

**PROCEEDINGS
OF THE 14TH INTERNATIONAL CONFERENCE
“LINGUISTIC RESOURCES AND TOOLS FOR NATURAL
LANGUAGE PROCESSING”
CLUJ-NAPOCA, 18-20 NOVEMBER 2019**

Editors

Mihaela Onofrei

Anca-Diana Bibiri

Constantin Dragoş Nicolae

Dan Tufiş

Dan Cristea

Organisers

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

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

Institute for Computer Science
Romanian Academy, Iaşi Branch

Faculty of Mathematics and Computer Science
“Babes-Bolyai” University of Cluj-Napoca

Romanian Association of Computational Linguistics

Under the auspices of the Academy of Technical Sciences

The publication of this volume was supported by
the Faculty of Computer Science,
“Alexandru Ioan Cuza” University of Iași

ISSN 1843-911X

PROGRAM COMMITTEE

Tiberiu Boroș, ADOBE Bucharest

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

Corneliu Burileanu, Faculty of Electronics, Telecommunications and Information Technology, University Politehnica of Bucharest and Research Institute for Artificial Intelligence “Mihai Drăgănescu”, Romanian Academy, Bucharest

Camelia Chira, Faculty of Mathematics and Computer Science, “Babes-Bolyai” University of Cluj-Napoca

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

Svetlana Cojocaru, Institute of Mathematics and Computer Science, Academy of Sciences of Moldova, Chișinău

Horia Cucu, Faculty of Electronics, Telecommunications and Information Technology, University Politehnica of Bucharest

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

Nils Diewald, Leibniz-Institut für Deutsche Sprache, Mannheim, Germany

Liviu Dinu, Faculty of Mathematics and Computer Science, University of Bucharest

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

Mircea Giurgiu, Faculty of Mathematics and Computer Science, “Babes-Bolyai” University of Cluj-Napoca

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

Gabriela Haja, “A. Philippide” Institute of Romanian Philology, Romanian Academy, Iași Branch

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

Diana Zaiu Inkpen, School of Electrical Engineering and Computer Science, University of Ottawa, Canada

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

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

Dana Lupșa, Faculty of Mathematics and Computer Science, “Babes-Bolyai” University of Cluj-Napoca

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

Vivi Năstase, Institut für Computerlinguistik, Universität Heidelberg, Germany

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

Mihaela Onofrei, Institute for Computer Science, Romanian Academy, Iași Branch and Faculty of Computer Science, “Alexandru Ioan Cuza” University of Iași

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

Adriana Stan, Faculty of Mathematics and Computer Science, “Babes-Bolyai” University of Cluj-Napoca

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

Horia-Nicolai Teodorescu, Institute for Computer Science, Romanian Academy, Iași Branch and “Gheorghe Asachi” Technical University of Iași

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

Ștefan Trăușan-Matu, Faculty of Automation, Control and Computer Engineering, University Politehnica of Bucharest and Research Institute for Artificial Intelligence “Mihai Drăgănescu”, Romanian Academy, Bucharest

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

Marius Zbancioc, “Gheorghe Asachi” Technical University of Iași and Institute for Computer Science, Romanian Academy, Iași Branch

ORGANIZING COMMITTEE

Anca Andreica, Faculty of Mathematics and Computer Science, “Babes-Bolyai” University of Cluj-Napoca

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

Camelia Chira, Faculty of Mathematics and Computer Science, “Babes-Bolyai” University of Cluj-Napoca

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

Corina Forăscu, 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 of Iași and Institute for Computer Science, Romanian Academy, Iași Branch

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

Simona Motogna, Faculty of Mathematics and Computer Science, “Babes-Bolyai” University of Cluj-Napoca

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

Mihaela Onofrei, Institute for Computer Science, Romanian Academy, Iași Branch and Faculty of Computer Science, “Alexandru Ioan Cuza” University of Iași

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

Horia Pop, Faculty of Mathematics and Computer Science, “Babes-Bolyai” University of Cluj-Napoca

Cristian Pădurariu, Faculty of Computer Science, “Alexandru Ioan Cuza” University of Iași and Institute for Computer Science, Romanian Academy, Iași Branch

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

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	ix
CHAPTER 1 CORPUS APPROACHES	1
POLISHING MONERO, THE MORPHOLOGICALLY AND MEDICAL NAMED ENTITIES ANNOTATED CORPUS OF ROMANIAN.....	3
<i>Ioana Marinescu, Verginica Barbu Mititelu, Maria Mitrofan</i>	
THE COBILIRO PROJECT: BUILDING AND DISTRIBUTING A BIMODAL CORPUS FOR ROMANIAN LANGUAGE	13
<i>Dan Cristea, Cristian Pădurariu, Șerban Boghiu, Daniela Gîfu, Mihaela Onofrei, Diana Trandabăț, Ionuț Cristian Pistol, Anca-Diana Bibiri, Andrei Scutelnicu</i>	
SPANISH LEARNER CORPUS CAES: A RESEARCH ON HOW RUSSIAN LEARNERS USE IMPERFECT TENSE IN SPANISH.....	25
<i>Maria Adelaida Gil Martinez</i>	
AUTOMATIC IDENTIFICATION AND CLASSIFICATION OF LEGAL TERMS IN ROMANIAN LAW TEXT	39
<i>Andrei Coman, Maria Mitrofan, Dan Tufiș</i>	
LOOKING ALONG ZIPF'S LAW FOR THE DISTRIBUTION OF WORDS BEGINNING AND ENDING SENTENCES IN LITERARY PRINTED ROMANIAN CORPORA.....	51
<i>Bogdan Hanu, Adriana Vlad, Alexandru Dinu, Adrian Mitrea</i>	
CHAPTER 2 NEURAL APPROACHES	63
ROMANIAN AUTOMATIC DIACRITICS RESTAURATION CHALLENGE	65
<i>Florin Iordache, Lucian Georgescu, Dan Oneață, Horia Cucu</i>	
TRIPOD: LEARNING LATENT REPRESENTATIONS FOR SEQUENCES	75
<i>Tiberiu Boroș, Andrei Cotaie, Alexandru Meterez, Paul Iliaica</i>	
CHAPTER 3 SYMBOLICAL APPROACHES VS. STATISTICS	85
SOME LOGICAL AND COMPUTATIONAL METHODS FOR THE ANALYSIS OF THE INEFFABLE	87
<i>Cătălina Mărânduc, Anca-Diana Bibiri</i>	
REVISITING THE STATISTICAL INDEPENDENCE FOR THE PRINTED ROMANIAN LANGUAGE	99
<i>Alexandru Dinu, Adriana Vlad, Bogdan Hanu, Adrian Mitrea</i>	
CHAPTER 4 DIACHRONIC STUDIES.....	113
SEMI-AUTOMATIC WORD LEVEL ALIGNMENT FOR RHOTACISING PSALTERS IN ROMANIAN.....	115
<i>Ana-Maria Gînsac, Maria Moruz, Mihai Alex Moruz, Mădălina Ungureanu</i>	
SOLUTIONS FOR SCANED DOCUMENTS SEGMENTATION AND LETTER RECOGNITION	127
<i>Constantin Cristian Pădurariu, Dan Cristea</i>	
MORHO-SYNATCTIC REGULARITIES IN UD_ROMANIAN NONSTANDARD PARSING	139
<i>Cătălina Mărânduc, Victoria Bobicev, Roman Untilov</i>	

CHAPTER 5 MULTIWORD EXPRESSIONS AND FRAME SEMANTICS	151
.....	
WRAPPING OUR HEADS AROUND VMWES AND THEIR DERIVATIVES ..	153
<i>Ivelina Stoyanova, Svetlozara Leseva, Verginica Barbu-Mititelu, Maria Todorova, Mihaela Cristescu</i>	
FRAME SPECIALISATION MOTIVATED BY INTER-FRAME RELATIONS IN FRAMENET	167
<i>Svetlozara Leseva, Ivelina Stoyanova, Maria Todorova, Hristina Kukova</i>	
CHAPTER 6 NLP TOOLS AND LANGUAGE-BASED APPLICATIONS	179
.....	
INTEGRATION OF ROMANIAN NLP TOOLS INTO THE RELATE PLATFORM	181
<i>Vasile Păiș, Dan Tuș, Radu Ion</i>	
READ-ME – MORPHO-SYNTACTIC ANALYSES FOR ROMANIAN LANGUAGE	193
<i>Anda-Mădălina Florea, Irina Toma, Robert Botarleanu, Mihai Dascălu, Ștefan Trăușan-Matu</i>	
QUERY EXPANSION – AUTOMATIC GENERATION OF SEMANTIC SIMILAR PHRASES USING WORDNET	203
<i>Diana Lucaci, Adrian Iftene</i>	
INDEX OF AUTHORS	217

FOREWORD

Today's communication is more and more mediated by the information technology and, by analogy with the already established term international communication language, one could speak about e-communication languages with direct reference to the languages used in the cyberspace. This concept, beyond its direct significance, has profound cultural, social, economical and ethical implications, stating the right of any citizen to have access in her/his own language to the knowledge and services of the cyberspace. Thus, it is not surprising the huge volume of research in the area of human language engineering, consistently supported through the international or national research programs which produced, in the last 20 years or so, practical results that paved the way to the multilinguality industry. The unprecedented increase of computers' performances and of data storage capacities, the success of neural networks and deep-learning approaches generated in the last decade not only a research paradigm change but also more consistent and convincing results than ever.

Romanian language is one of the emerging languages in the virtual space and, as such, it is present in the international research programs, by united forces of Romanian experts and their colleagues from important institutions around the world. With a visionary anticipation of this trend, the Section for Information Science and Technology of the Romanian Academy initiated in 2001 the Consortium for Informatization of the Romanian Language, which, afterwards, organised in Iași a first workshop in what will become a series of events named ConsILR (by the acronym for *Consortiul pentru Informatizarea Limbii Române*). They were meant to bring together linguists and computational linguists, researchers of the humanities, PhD students and master students in Computational Linguistics, all with a major interest in the study of the Romanian language from a computational perspective. The events have run, with few exceptions, every year, and since 2010 – the series was transformed into an annual conference with an international scope. To reflect this opening towards general methods, which, although initially addressing one language interest could easily be generalised to cope with other languages and multilingual issues, the ConsILR conference is now called Linguistic Resources and Technologies for Natural Language Processing.

To the three traditional organisers of the conference, the Faculty of Computer Science, “Alexandru Ioan Cuza” University of Iași, the Research Institute for Artificial Intelligence “Mihai Drăgănescu”, Romanian Academy, and the Institute for Computer Science of the Iași Branch of the Romanian Academy, this year have joined also the “Alexandru Philippide” Institute of Romanian Philology from the Iași Branch of the Romanian Academy, the Faculty of Mathematics and Computer Science, “Babeș-Bolyai” University of Cluj-Napoca, as the local organiser, and the Romanian Association of Computational Linguistics. We, the organisers, thank again the Academy of Technical Sciences, which accepted to place our conference under their high auspices.

This year the Conference was organised in Cluj-Napoca, a beautiful city located in the centre of Transylvania, with a very strong scientific life gravitating around two big universities and including one of the most vivid IT community in Romania. The 17 contributions, quite heterogeneous, were organized in 6 chapters: Corpus Approaches, Neural Approaches, Symbolic Approaches vs. Statistics, Diachronic Studies, Multiword Expressions and Frame Semantics, and NLP Tools and Language-Based Applications. The current selection of papers is inherently incomplete but the multitude of topics approached reflects the current wide coverage of the domain.

To conclude, this volume offers a valuable collection of contributions presenting research and development results. Several issues discussed throughout the book are present-day concerns of the scientific community irrespective of the investigated language. Therefore, we believe that this volume will be a very useful instrument for NLP researchers, for students preparing their MSc dissertations or PhD theses in the area of computational studies of natural languages.

ACKNOWLEDGMENTS

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.

November 2019

The editors

CHAPTER 1

CORPUS APPROACHES

POLISHING MONERO, THE MORPHOLOGICALLY AND MEDICAL NAMED ENTITIES ANNOTATED CORPUS OF ROMANIAN

IOANA MARINESCU¹, VERGINICA BARBU MITITELU², MARIA MITROFAN²

¹ *Mircea cel Bătrân National College, Râmnicu Vâlcea*

² *Research Institute for Artificial Intelligence “Mihai Drăgănescu”, Romanian Academy*

ioanam2509@gmail.com

{vergi, maria}@racai.ro

Abstract

MoNERo is a Romanian specialized corpus, reflecting the medical language from three fields (cardiology, endocrinology and diabetes). It contains 4,987 sentences and 154,825 tokens with an unbalanced distribution among the fields. The corpus is one of the few gold standard ones for Romanian. It was automatically processed (tokenized, part of speech-tagged and lemmatized) and then manually checked. It is also annotated with medical terms, using the IOB2 format and contains labels belonging to four entity semantic groups: anatomy (ANAT), chemicals and drugs (CHEM), disorders (DISO) and medical procedures (PROC). The annotation with medical entities was manually done, but only a small part of the corpus was doubly annotated. Given the inherent rate of errors characterizing manual work, even that of specialists, we present here: (i) the steps taken for an automatic identification of the errors left in the corpus, (ii) the types of errors found in the corpus; (iii) the way in which they were solved in the corpus. MoNERo has been publicly released for the community and can be used as a reliable resource for training NER systems and NLP systems.

Key words — medical corpus, part of speech tagging, medical terms, automatic identification of errors, gold standard.

1. Introduction

Language resources in the form of written corpora are a great asset, useful for language specialists and language engineers alike: they mirror the linguistic (lexical, morphological, syntactic, discourse) phenomena characterizing a language at the time when the texts they contain were written. When the corpus is specialized, *i.e.* it contains texts from a certain domain of activity, it is also interesting for terminologists (who are concerned with the correctness of the terms form according to the language characteristics and with standardizing the terms specific to the domain) and for the specialists of that domain, who need to mine the text for specific information.

In this paper we present the characteristics of the first Romanian medical corpus, MoNERo, linguistically annotated with lexical and morphological information and with medical information in the form of several types of terms; the main interest here lies in applying an automatic procedure for the identification of the errors left after manual correction of the annotations. After a brief general presentation of the corpus (section 2), we present related work in the field of automatic identification of annotation errors

in corpora (section 3), then we describe our methodology for spotting such errors in MoNERo (section 4). The types of errors found are discussed in section 5 and conclusions are drawn in section 6.

2. MONERO – general presentation of the corpus

Romanian benefits of the existence of a reference contemporary corpus, CoRoLa (Barbu Mititelu *et al.*, 2018), whose design was meant to ensure a large coverage of the language styles, of as many as possible domains and subdomains of activity. As a consequence, the medical scientific domain is represented among the many others. Mitrofan and Tufiş (2018) reported on BioRo, made up of 9,864,707 tokens from texts belonging to the medical domain. They are unequally distributed into 9 subdomains: oncology, cardiology, diabetes, surgery, endocrinology, neurology, nephrology, psychiatry, and alternative medicine. The texts come from several types of sources: scientific books, medical journals, medical blogs and medical school lecture notes. The corpus is automatically split into sentences, tokenized, part-of-speech tagged and lemmatized with TTL (Ion, 2007).

A part of BioRo that raised no copyright issues has been turned into a freestanding corpus, MoNERo (Mitrofan *et al.*, 2019). The corpus was extracted from sources that guarantee the users unlimited access to the site and gives them the right to modify or to reproduce the texts in whole or in part. Since it is difficult to obtain texts with copyright-free status, MoNERo contains texts from only three out of the nine medical subdomains in BioRo: cardiology, diabetes, and endocrinology, distributed as in Table 1 below, where one can notice their unbalanced representation.

Table 1: Subdomain distribution of the texts in MoNERo

Domain	Number of tokens
Cardiology	63,043
Diabetes	69,085
Endocrinology	22,697
TOTAL	154,825

Although not large as a corpus, MoNERo is large enough to be used as a resource for training systems dealing with named entity recognition (NER) in the medical domain (Mitrofan *et al.*, 2019). Some quantitative data about it are provided in Table 2:

Table 2: Quantitative data about MoNERo

Number of:	#
Sentences	4,987
Tokens	154,825
Words	134,084
Sentence length (words/sentence)	26.9

MoNERo is comparable in size with other annotated medical corpora available within the scientific community: CLEF corpus (Roberts *et al.*, 2009), i2b2 corpus (Uzuner *et*

al., 2010), NCBI corpus (Doğan *et al.*, 2014), Quaero corpus (Névél *et al.*, 2014) CHEMDNER corpus (Krallinger *et al.*, 2015), IxaMedGS corpus (Oronoz *et al.*, 2015), DrugSemantics corpus (Moreno *et al.*, 2017), etc.

Being part of BioRo, MoNERo is sentence split, tokenized, part-of-speech (POS) tagged and lemmatized. What distinguishes it from BioRo is the fact that all these levels of processing and annotation have been manually checked: sentence limits, tokens, lemmas and the morphosyntactic descriptions have been corrected when necessary. Another distinguishing aspect of MoNERo is its manual annotation with medical named entities. These are medical terms of varying length, i.e. made up of a variable number of words, either continuous or discontinuous in their occurrences in the corpus. For example, in the sentence in Ex. 1 the entity *peretelui aortic* is continuous, of length 2, the entities *vasului* and *lumenul* are of length 1, while *anevrismele fusiforme*, *anevrismele sacciforme*, *pseudoanevrismele sacciforme* are discontinuous ones, of length 2.

Example 1:

Din punct de vedere morfologic, aneurismele adevărate pot fi fusiforme (dilatatie simetrică, aspect cilindric al vasului) sau sacciforme (afectează doar o porțiune a peretelui aortic și comunică printr-un colet cu lumenul), iar pseudoanevrismele, doar sacciforme.

‘From point of view morphologic, aneurysms-the real can be fusiform (dilatation symmetric, aspect cylindrical of vessel-the) or saccular (affects only a portion of wall aortic and communicates through-a colet with lumen-the), and pseudoaneurysms-the, only saccular.’

“From a morphological point of view, real aneurysms can be fusiform (symmetric dilatation, cylindrical aspect of the vessel) or saccular (affecting only a portion of the aortic wall and communicating with the lumen through a colet), while pseudoaneurysms can be only saccular.”

The manual annotation with named entities (NEs) observed some guidelines drafted beforehand and adjusted whenever new cases were encountered and was made by two annotators, who discussed together the problematic cases during the annotation process (Mitrofan *et al.*, 2019).

In order to ensure a high-quality annotation of this gold corpus, besides the manual correction of all annotations, we also used some automatic strategies of spotting inconsistencies left in the annotation after the manual correction, as described in section 4 below.

3. Related work

The focus of automatic POS error detection is identifying errors in POS tags assigned by human annotators. Variation in POS tags in the corpus can be caused either by ambiguous word forms which, depending on the context, can belong to different word classes, or by incorrect judgments made by the annotators (Eskin, 2000; van Halteren, 2000; Kveton and Oliva, 2002; Dickinson and Meurers, 2003; Loftsson, 2009). They

are the results of several factors, among which we mention: complex annotation scheme, rare occurrences of words, even hapax legomena, which are difficult to analyze, especially if the words characterize specialized vocabulary, factors affecting the humans' work, e.g. tiredness, tight deadlines, etc.

The variation n-gram algorithm (Dickinson and Meurers, 2003) allows users to identify potentially incorrect tagger predictions by looking at the variation in the assignment of POS tags to a particular word n-gram. The algorithm produces a list of varying tagger decisions which have to be processed by a human annotator.

Results obtained from work with another focus could also be combined with our consistency checking approach. Deriving and searching for bigrams of tags which should never be allowed (Kveton and Oliva, 2002) shows that inconsistencies are mostly bigrams. Sparse Markov transducers used to detect anomalies, i.e., rare local tag patterns (Eskin, 2000) show that inconsistencies are mostly recurrent, not rare. Parsing failures are used to detect ill-formed annotation serving as parser input (Hirakawa *et al.*, 2000; Muller and Ule, 2002). Searching and correcting with hand-written rules is also proposed (Oliva, 2001; Blaheta, 2002).

4. Work description

The solution for detecting errors in the part-of-speech tags annotation is programed based on the variation n-gram method. The implementation was made using C++ libraries vector, string, and map. The program was developed in three parts, by coding the functions for reading (section 4.1), generating the n-grams (section 4.2), and identifying the errors (section 4.3).

4.1. Reading

The reading function stores each word from all lines in a vector. Then for every word, it stores a list of all occurrences of the word in the corpus using the number of the line (e.g. *vascular*: 28567, 60245, 123160).

4.2. Generating N-grams

For each word, each occurrence is analyzed separately. For each occurrence, there are N possibilities for the position of the word in a context of N words, from first to last. Thus, each group of N words acts like a unit (gram). For each gram, a list of its occurrences in the corpus is stored. The list contains the indexes of the lines where the word analyzed in the context of the current gram occurs. Having stored information about the n-grams, the POS tags are then analyzed as in Section 4.3.

While any value of N relatively small ($N \leq 100$) would have preserved the linear complexity of the algorithm, the results presented were obtained using $N = 3$ (e.g. analyzing the word *perioadă* in the context of *pe o perioadă* ("over a period") appears eight times and the context of *perioadă de urmărire* ("period of observation") occurs three times).

4.3. Identifying errors of POS-tags

Considering each N-gram separately, for every occurrence of it, the annotation of the initial word is checked and for each different annotation, the tag and its position are stored. Supposing that the most frequent tag is the right one, all the rest are errors and must be replaced. For examples, the adjective *cronice* (“chronic”) in the context *boli pulmonare cronice* (“chronic lung disease”) occurs on the lines 40851, 50127, 62431, with *Afpfprn* as the most frequent tag assigned to it, while *Afpfp-n* is the tag for the occurrence on line 62431 (in the file this is represented as: *cronice: boli pulmonare cronice – 40851, 50127, 62431 Afpfprn Afpfp-n 62431*). The tag *Afpfprn* is to be read as adjective (A) of the type qualificative (f), with the degree positive (p), gender feminine (f), number plural (p), case direct (r), non-definite (n), while the tag *Afpfp-n* expresses the same information with the exception of the case position which is left underspecified (-) because the same form can be used for all cases of the plural number, feminine gender and non-definite use.

5. Results

5.1. Improved morphological annotation

The TTL tool is reported good annotation results on news texts: 98.23% accuracy. When run on medical data, the accuracy decreased, as expected, however, not to a high extent: it was 97.83% (Mitrofan and Ion, 2017). This implies a low overload of manual annotation corrections. As no pointers to the wrong tags or lemmas are provided, the human annotator had to go through all tokens, one by one, and make sure they are correctly tokenized, POS-tagged and lemmatized. This is similar to finding a needle in the haystack. The complex tagset used for Romanian adds to the difficulty of the task: it contains 714 complex tags.

The types of errors that were corrected are:

- Errors of tokenization:
 - most of the times they are connected to the different types of uses of the hyphens: between numbers (years: 1960-1990, periods: 15.03.1997-15.09.1997, etc.) or between words (especially clitics: *s-a* ‘Refl.Cl.3.sg.Acc.-has’): the automatic tokenization of such cases was not consistent: consider the first example: 1960-1990: sometimes this is considered a single token, some other times it is split into three tokens;
 - sometimes, there are spaces inside a word, thus it is interpreted as two tokens (e.g., *fi cat* instead of *ficat* “liver”); they are typos, but require the deletion of the blanc and the recovery of the word;
 - inconsistent treatment of several symbols (% , +, etc.): sometimes they are treated as a separate token, while some other times, together with the previous token, they make up one token: for example, 50% is sometimes analyzed as a single token, and some other times it is split into 50 and %;

- Errors of lemmatization: their causes are multiple:
 - words unknown to the annotation tool: given the fact that MoNERo is a specialized corpus, the number of words specific to the medical domain and even to the three subdomains represented in the corpus is quite high: *stenoză* (“stenosis”), *aterom* (“atheroma”), etc. Moreover, some of the words are foreign (mostly English): *biomarker*, *strain*, etc. Another linguistic phenomenon encountered is the calque: *cost-eficientă* (“cost-effective”) is a strange compound in Romanian, copying the inner structure of its English equivalent;
 - morphologically ambiguous words: one of the frequent wrong lemmatization is of the word *copii*, ambiguous between the indefinite plural of the masculine noun *copil* (“child”) and the indefinite plural of the feminine noun *copie* (“copy”); its most frequent occurrence is with the first value, but the latter also occurs and the TTL tool is not able to correctly distinguish between them;
- Tagging errors: they affect either the part of speech or the morphological classes:
 - POS errors: they occur in the case of homonyms of different parts of speech: nouns - adjectives (*prezent* “present”), nouns - adverbs (*seara* “the evening”); adjectives - adverbs (e.g.: in the string *Un studiu recent a arătat că...* “A recent study has shown that...” the word *recent* is annotated as an adverb, although it is an adjective), verbs in the participles - participial adjectives (e.g.: in the context ... *vor trebui upgrdate ulterior sau chiar înlocuite* “... will have to be eventually updated or even replaced” the words *upgrdate* and *înlocuite* are annotated as adjectives, but they are verbs);
 - morphological classes are misinterpreted as a result of the homonymy in the inflectional paradigm of the respective word: *folositoare* (“useful”) is an adjective either in the oblique (dative or genitive) case of the singular number or in any of the cases of the plural number; many morphological categories are affected: number, person, case, gender, verbal mood, tense, etc.

Running the script presented above for checking the consistency of the morphologic annotation 565 tokens were found for which inconsistencies are suspected to happen in their annotation. An automatic replacement of the least frequent with the most frequent of the annotation of a token in a certain context cannot be made: consider the tri-gram *care este apoi* “which is then”: the competing POS-tags are Vmip3s (Verb main indicative present 3rd.person singular) and Vaip3s (Verb auxiliary indicative present 3rd.person singular), which means that the token is ambiguous between a main and an auxiliary verb, but the context is not large enough for deciding upon the correct value. Thus, a human annotator must check each and every occurrence and decide for each case separately. Such ambiguous examples are not rare in the corpus.

After the manual intervention following this automatic identification of possibly inconsistent annotations, we counted 5279 lines that were modified: 923 lemmas and 4530 POS tags were corrected. There were 174 cases in which both the lemma and the

POS tag were corrected. The interventions on the POS tags meant changing the part of speech in 243 cases and/or changing the values of their attributes in 4287 cases.

The most frequent change of the part of speech is the result of the homonymy between participle and the participial adjectives:

- 88 adjectives (A) were changed into participles (V): example: when the trigram *asociată sau nu* (“associated or not”) occurs in a syntactic position in which *asociată* is a modifier of a noun, it should be annotated as a participle (V): there is an ellipsis of the auxiliary verb, so the underlying structure would be *care este asociată sau nu este asociată* (“that is associated or is not associated”);
- 28 participles (V), into adjectives (A): example: in the context *în stadii avansate* (“in advanced stages”) the underlined word is definitely an adjective, as it modifies the noun.

The homonymy between adjectives and nouns also affects 31 tokens, which had to have their part of speech changed from noun (N) into adjective (A): in the string *limita superioară a normalului* (“the upper limit of normal”) the word *normalului* (“normal”) is a noun, not an adjective (which would be the most frequent case of this word in the language).

The word *a* was found 17 times wrongly annotated as a(n auxiliary) verb (V) instead of an (infinitive) particle (Q).

Many abbreviations were annotated as nouns by TTL, although there is a specialized tag for them, namely Y. The initial manual correction of the corpus involved a large number of corrections of such cases. However, after checking the consistency we found 13 abbreviations containing also numbers that we decided to treat as residual (thus, annotating them with the tag X): examples: Ca2, T1DM (type 1 diabetes mellitus), etc.

The other types of modifications involving the part of speech are not numerous, each type having at most 5 occurrences. A few examples are: particles (Q) that should have been annotated as prepositions (S) (4 cases), nouns (N) that should have been annotated as adverbs (R) (3 cases), etc.

5.2. Improved NE annotation

The manual annotation with NEs was done by two annotators, one a physician, the other one a computer scientist, using the IOB2 format (Sang and Veenstra, 1999). The labels attached to the NEs are: ANAT (for anatomy-related entities), CHEM (for chemicals and drugs), DISO (for different types of disorders), and PROC (for all medical procedures).

The annotators’ different backgrounds, the linguistic characteristics of the medical texts (with a high frequency of abbreviations), the lack of terminological resources in Romanian (which could have helped the annotators understand the meaning of various terms) are the main explanations for the errors on the NEs annotation in MoNERo. The task being so complex, discussions were allowed between the annotators during their work. This explains the high agreement between them on a small part of the corpus

(1,628 tokens) that was doubly annotated by them, at the end of the task, that is when both had gathered experience in this work.

The types of errors found are:

- different lengths of the same NE - this happens especially when coordinating conjunctions occur between two NEs: *ocluzia/B-DISO arterelor/I-DISO mici/I-DISO și/I-DISO mijlocii/I-DISO* (“occlusion of small and medium sized arteries”) is interpreted by the physician as only one NE, while the computer scientist analyzed it as two NEs connected by the conjunction *și*;
- different labels for the same NE: this is mainly the result of the wrong interpretation of abbreviations: e.g. *ACE* can be either *CHEM* or *PROC*, depending on the context of occurrence and telling them apart is not always trivial for the computer scientist.

The errors found through the extraction of all NEs from the corpus and by comparing their annotations are only a handful, thus showing again the high quality of this manual annotation. Table 3 shows the number of errors found in each NE category. There was a total of 131 annotation errors. Most of them, almost 40%, were confusions between disorders and others categories (50) followed by chemicals tagged wrongly (30).

Table 3: Table of corrections of NEs

	<i>Corrected Annotations</i>						
		<i>ANAT</i>	<i>CHEM</i>	<i>DISO</i>	<i>PROC</i>	<i>O</i>	<i>TOTAL</i>
<i>Initial Annotations</i>	<i>ANAT</i>	0	7	11	2	1	21
	<i>CHEM</i>	15	0	12	0	3	30
	<i>DISO</i>	7	5	0	36	2	50
	<i>PROC</i>	0	4	11	0	2	17
	<i>O</i>	0	4	5	4	0	13

6. Conclusions

MoNERo is a valuable lexical resource for Romanian, reflecting the medical domain, with three subdomains: cardiology, diabetes, and endocrinology. Its size, the quality of the annotation with morphological information and NEs types are important assets. It is also made available for free to the community¹, to those interested to use it for different types of analysis of such texts (linguistic or terminological characteristics of such texts, etc.), or for training, improving and testing systems for automatic medical NEs recognition and classification.

A new version of MoNERo, with a new level of annotation, namely syntactic, will be made available soon, as part of the Romanian contribution to the Universal Dependencies initiative².

¹ <http://www.racai.ro/tools/text/>

² universaldependencies.org

Acknowledgements

This work was supported in part by a grant of the Romanian Ministry of Research and Innovation, PCCDI - UEFISCDI, project number PN-III-P1-1.2-PCCDI-2017-0818/72, within PNCDI III.

References

- Barbu Mititelu, V., Tufiş, D., Irimia, E. (2018). The Reference Corpus of the Contemporary Romanian Language (CoRoLa). In *Proceedings of LREC 2018*, May, Japan, 1178-1185.
- Blaheta, D. (2002). Handling noisy training and testing data. In *Proceedings of the 7th conference on Empirical Methods in Natural Language Processing*, 111–116.
- Dickinson, M. and Meurers, D. W. (2003). Detecting errors in part-of-speech annotation. In *10th Conference of the European Chapter of the Association for Computational Linguistics (EACL-03)*, April, Budapest, Hungary, 107-114.
- Dogan, R.I., Leaman, R., Lu, Z. (2014). NCBI disease corpus: a resource for disease name recognition and concept normalization. *Journal of biomedical informatics*, 47, 1-10.
- Eskin, E. (2000). Automatic corpus correction with anomaly detection. In *1st Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Seattle, Washington, 148-153.
- Hirakawa, H., Ono, K., Yoshimura, Y. (2000). Automatic Refinement of a POS Tagger Using a Reliable Parser and Plain Text Corpora. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING)*, Saarbrücken, Germany: ICCL.
- Ion, R. (2007.) *Word Sense Disambiguation Methods Applied to English and Romanian* (in Romanian). Ph.D. thesis, Romanian Academy.
- Krallinger, M., Rabal, O., Leitner, F., Vazquez, M., Salgado, D., Lu, Z., Leaman, R., Lu, Y., Ji, D., Lowe, D.M. *et al.* (2015). The CHEMDNER corpus of chemicals and drugs and its annotation principles. *Journal of Cheminformatics*, 7(1):S2.
- Kveton, P. and Oliva, K. (2002). (Semi-)Automatic detection of errors in PoS-tagged corpora. In *Proceedings of 19th International Conference on Computational Linguistics (COLING-02)*.
- Loftsson, H. (2009). Correcting a POS-tagged corpus using three complementary methods. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, March, Athens, Greece, 523-531.
- Meurers, D. (2010). Detecting Errors in Corpus Annotation. *LingLunch*, UFR de Linguistique, Université Paris Diderot, Paris 7, March 10, 2010.
- Mitrofan, M., Barbu Mititelu, V., Mitrofan, G. (2019). MoNERo: a Biomedical Gold Standard Corpus for the Romanian Language. In *Proceedings of the BioNLP workshop*, 1 August, Florence, Italy, 71-79.
- Mitrofan, M. and Ion, R. (2017). Adapting the TTL Romanian POS tagger to the biomedical domain. In *Proceeding of BiomedicalNLP@ RANLP*, 8-14.

- Moreno, I., Boldrini, E., Moreda, P., Roma-Ferri, M. T. (2017). DrugSemantics: a corpus for named entity recognition in Spanish summaries of product characteristics. *Journal of biomedical informatics*, 72:8–22.
- Muller, F. H. and Ule, T. (2002). Annotating topological fields and chunks – and revising POS tags at the same time. In *Proceedings of COLING*, 1-7.
- Neveol, A., Grouin, C., Leixa, J., Rosset, S., Zweigenbaum, P. (2014). The Quaero French medical corpus: A resource for medical entity recognition and normalization. In *Proc BioTextM*, Reykjavik, 24-30.
- Oliva, K. (2001). The Possibilities of Automatic Detection/Correction of Errors in Tagged Corpora: A Pilot Study on a German Corpus. In V. Matousek, P. Mautner, R. Moucek and K. Tauser (eds.), *Text, Speech and Dialogue. 4th International Conference, TSD 2001*, September 11-13, Zelezná Ruda, Czech Republic, 39-46.
- Oronoz, M., Gojenola, K., Perez, A., de Ilarraza, A.D., Casillas, A. (2015). On the creation of a clinical gold standard corpus in Spanish: Mining adverse drug reactions. *Journal of biomedical informatics*, 56: 318–332.
- Rehbein, I. (2014). POS error detection in automatically annotated corpora Published. In Levin, Lori/Stede, Manfred (eds.), *Proceedings of the 8th Linguistic Annotation Workshop in conjunction with COLING 2014 (LAW-VIII)*, August 23-24, Dublin, Ireland, 20-28.
- Roberts, A., Gaizauskas, R., Hepple, M., Demetriou, G., Guo, Y., Roberts, I., Setzer, A. (2009). Building a semantically annotated corpus of clinical texts. *Journal of biomedical informatics*, 42:5, 950-966.
- Sang, E. F. and Veenstra, J. (1999). Representing text chunks. In *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*, 173-179.
- Uzuner, O., Solti, I., Xia, F., Cadag, E. (2010). Community annotation experiment for ground truth generation for the i2b2 medication challenge. *Journal of the American Medical Informatics Association*, 17:5, 519-523.
- Van Halteren, H. (2000). The detection of inconsistency in manually tagged text. In *Proceedings of the COLING2000 Workshop on Linguistically Interpreted Corpora*, August, Centre Universitaire, Luxembourg, 48-55.

THE COBILIRO PROJECT: BUILDING AND DISTRIBUTING A BIMODAL CORPUS FOR ROMANIAN LANGUAGE

DAN CRISTEA^{1,2}, CRISTIAN PĂDURARIU^{1,2}, ȘERBAN BOGHIU¹, DANIELA GÎFU^{1,2}, MIHAELA ONOFREI^{1,2}, DIANA TRANDABĂȚ¹, IONUȚ CRISTIAN PISTOL¹, ANCA-DIANA BIBIRI³, ANDREI SCUTELNICU^{1,2}

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

² Institute of Computer Science, Romanian Academy, Iași Branch,

³ “Alexandru Ioan Cuza” University of Iași, Institute for Interdisciplinary Research, Social Sciences and Humanities Research Department

{dcristea, cristian.padurariu, serban.boghiu, daniela.gifu, mihaela.onofrei, dtrandabat, ionut.pistol, andrei.scutelnicu}@info.uaic.ro
anca.bibiri@gmail.com

Abstract

CoBiLiRo (*Corpus Bimodal pentru Limba Română - Bimodal Corpus for Romanian Language*) is an on-going research project aimed to collect, standardise and make available a collection of Romanian language files containing both text and audio recordings, aligned at boundaries of sentences, words, phones, and/or other linguistic levels. This paper describes the current efforts carried out as part of this project. We present the design of the format aimed to serve as an annotation standard for bimodal resources, the main operations of the web Platform which hosts the corpus, and the automatic conversion flow that brings the submitted file at the format accepted by the Platform.

Key words — bimodal corpus, annotation standard, web platform, speech and text processing, metadata of linguistic resources.

1. Introduction

CoBiLiRo is a component part of ReTeRom¹, a project aiming to push forward the state of the art in Romanian language technology, grouping researchers from four natural language processing laboratories² that work on speech understanding, speech synthesis, text processing, alignment of speech - text resources and organisation of big repositories of language data for research and public use. CoBiLiRo aims to create a thesaurus with audio and textual resources annotated on different levels of acoustic and linguistic

¹ www.racai.ro.

² RACAI – Research Institute for Artificial Intelligence “Mihai Drăgănescu”, Romanian Academy; the Speech Processing Laboratory at the Faculty of Electronics, Telecommunications and Information Technology of Politehnica University of Bucharest; the Speech Processing Lab at the Department of Communications, Faculty of Electronics, Telecommunications and Information Technology of the Technical University of Cluj-Napoca, and the Natural Language Processing Group at the Faculty of Computer Science of the “Alexandru Ioan Cuza” University of Iași.

achievement, which will stand as the most important reference of this type for the Romanian language, addressing future developments of human-machine interfacing technologies. As such, the project makes a careful inventory of existing bimodal resources at partners, finds ways to harmonize the representation, the annotation and the metadata formats, designs and implements an infrastructure that will finally house the resources, and does a wide dissemination of the bimodal corpus, for research valorisation and use in applications.

In ReTeRom, but also sharing other views (Mihăilă and Mekhaldi, 2009), by a bimodal corpus we understand a collection of oral records accompanied by their transcripts and their corresponding metadata. A bimodal corpus is hosted on a specialized platform, together with its web access, maintenance services, processing technology and possible applications.

Stages in the evolution of linguistic corpora included: the first generation, which contained only texts (e.g. British National Corpus³), the second generation, which contained both oral and written texts (e.g. Michigan Corpus of Academic Spoken English⁴), and the third generation, which included also the alignment between written and oral components. Recent stages in the evolution of corpora are characterized by the inclusion of the video dimension (Rasso and Mello, 2014: 29). The object of our interest is the third generation of corpora.

In order to advance Romanian technologies integrated with natural language processing, our goal is to construct bimodal corpora to be used in learning. Part of our bimodal corpora is annotated on different linguistic levels of the bimodal corpus generated within CoBiLiRo based on some conventions, which will then be taken as a model for an automatic process, the proposal of an inventory of unstructured data and specifications for the realization of user interfaces.

The next sections will briefly describe existing approaches of bimodal resources, the CoBiLiRo format, the design and implementation of converters, and the CoBiLiRO web platform.

2. Bimodal resources

Research on automatic speech recognition have grown dramatically since the 1960s (Halle and Stevens, 1962; Denes and Mathews, 1960; Denes, 1960), although the use of oral corpora as a language storage device that should be interpreted or generated by the machine is much more recent. In order to make available these corpora, specialized interfaces have begun to be created. A bimodal corpus should be hosted on a specialized platform, together with its web access, development and maintenance services and applications, where the researcher can find methods and algorithms for corpus use and,

³ <http://www.natcorp.ox.ac.uk/>

⁴ <https://www.lib.umich.edu/database/link/11887>

from where, in some cases, examples of applications of the corpus in training and evaluation of the technology can be downloaded.

Even if the terms *speech corpus* and *oral corpus* are sometimes differentiated (Llisterri, 1996), they are often used interchangeable, without a clear distinction between them. For simplicity reasons, we will not differentiate between these two categories of corpora, mixing what some researchers consider to be specific characteristics, and using the generic term of *speech corpus*.

A speech corpus, in general, means a database of audio files and their textual transcripts, in a format that can be used to create acoustic models that support both speech recognition and speech synthesis research. An example is the Switchboard transcripts reviewed at the Institute for Signal and Information Processing (Godfrey and Hollman, 1997). Audio files and their transcripts can be aligned at phoneme, syllable, word and sentence levels, sometimes marking also prosody elements. Within speech recognition systems, prosody models are mainly used to predict prose events (syntactic and semantic accents) associated with a text. Research in which the emphasis is on the audible signal, on its acoustic properties, and the articulatory properties of the vocal tract, is making heavy use of speech corpora. The symbolic representation in this case is the phonetic alphabet. The transcript may include marks, in an enriched spelling system, of various auditory phenomena that accompany the pronunciation: murmurs, pauses, coughs, laughter, etc.

On the other hand, a speech corpus can be of particular interest for researchers dedicated to the use of a language and the characteristics of the various linguistic levels: lexical, morphological, syntactic, semantic, discourse, conversation, for pragmatic studies of communication, in sociolinguistics, dialectology, etc. Ideally, both language researchers and natural language technology researchers should use the same data set, regarding data collection, transcription, coding and annotation.

At the moment of writing this paper, the ReTeRom collection of bimodal corpora contains the following items: the audio files and their transcripts from the CoRoLa⁵ corpus - contributed by ARFI-IIT and RACAI, the Read Speech Corpus (RSC) - contributed by UPB, the Spontaneous Speech Corpus (SSC-train), Spontaneous Speech Corpus (SSC-eval) and Spontaneous Speech Corpus 2 (SSC-eval2) – all contributed by UPB, the SWARA⁶ Speech Corpus – contributed by UTCN, the text corpus Adevărul.ro, Mara⁷ – an Audiobook, Ro-GRID contributed by UTCN, IIT⁸ corpus – contributed by ARFI-IIT and the SoRoEs⁹ corpus – contributed by UAIC. Most of the 11 linguistic resources are bimodal, therefore including both audio files and transcripts, together with their alignments. In total, there have been reported more than 450 hours of recordings.

⁵ <http://corola.racai.ro>

⁶ <https://speech.utcluj.ro/swarasc/>

⁷ <https://speech.utcluj.ro/corpora/mara.html>

⁸ <89.38.230.231a/home/corola/corpusIasi/>

⁹ <http://soro.es.ro/>

As will be presented in Section 4, the inventoried corpora expose a large diversity of formats, containing audio files and their text transcriptions, but in some cases also simplified text transcriptions (without punctuation and segmentation), and TextGrid files, which put in evidence alignment of boundary clues between segments of speech and text. To unify this diversity, a standard format has been proposed by the consortium members, a bunch of conversion modules operating on the CoBiLiRo platform to bring any of the input files to the standard one.

3. Design criteria in building the hosting Platform and the adoption of the corpus format

To fulfill CoBiLiRo's purposes, we designed a platform capable to host linguistic resources of the Romanian language, intended to be used for the development of automatic speech recognition and synthesis systems. Many of these resources, used to train acoustic models, are speech corpora containing recordings of different speakers, paired with their corresponding textual transcripts.

In order to ensure naturalness in automatic speech systems, the recordings to be placed at the base of learning experiments should generally be acquired in spontaneous interactions between speakers, therefore readings in lab conditions and eBooks containing actors' voices are less recommended. On the other hand, production of linguistic resources of this type is expensive, in terms of time, cost and necessary involvement of experts, which are difficult to find. For these reasons, the interest of the ReTeRom consortium was focused towards sources of real life speech, audio corpora available online and in the media: radio and television shows, recordings of public meetings for certain institutions, and ad-hoc interviews addressed to people on the street intended to evidenciate local accents and dialectal voices. As not all of these resources have intrinsic textual transcriptions, we had to prepare the ground for transcribing parts of the corpus, which is by no means a trivial task. Adding transcriptions can be done manually, by listening to the recordings and simultaneously writing down the related text, or by using already existing automatic speech recognition systems. The apparent vicious circle (of using automatic systems to transcribe naturally produced speech, followed by training recognition systems out of the parallel corpus produced this way) is broken down by involving the use of several architecturally different speech recognition systems, which are supposed to make non-symmetrical errors, keeping as correct identical transcribed spans and manually correcting the regions which display discrepancies.

As such, recently, the CoBiLiRo Platform frontend has been upgraded to accommodate the process of uploading bimodal corpora files (speech plus text) even at different moments of time, as each textual transcript can be decoupled from its speech component. Properly annotated, these two components can be paired later, on the Platform, when both are uploaded there. The alignment is assured through segmentation clues, which can be placed at sentence, word or even letter/phoneme boundaries, as will be explained in the following section. Moreover, the frontend includes functionalities

that allow online editing of the two components in view of creating the speech-text alignment, more precisely the inclusion and synchronisation of boundary markers.

The CoBiLiRo annotation format is inspired by the TEI-P5¹⁰ standard (Sperberg-McQueen and Burnard, 2018), the well known scheme for representing a diversity of document types, but also includes elements from other proposals to best fulfil our goals (Li and Yin, 2007). This standard has been simplified in some aspects and augmented in others to best accommodate the requirements of our bimodal corpora of speech and text data.

The CoBiLiRo format includes a header, which encapsulates metadata related to the resource. This section holds information about: the source of the object stored, the identity of speakers (in conditions of respecting confidentiality terms), the type of voice (spontaneous or voice-in-reading), aspects regarding technical conditions of the recording, its duration, the type of file stored (*mp3* or *wav*), the segmentation level, etc. The most common level of segmentation is the sentence, but voice can also be segmented in morphological units (words), phonological (phonemes), prosodical (pitch, raise and decrease of the fundamental frequency), or syntactic (nominal group, clause, etc.). These pieces of information are stored in appropriate *xml* tags and attributes, within the *teiHeader* tag. Figure 1 shows an example of such a header.

```
<teiHeader Description="..." Collection="..." Keywords="bimodal, text,
speech" Language="ro" Contributor="cristian.padurariu"
Distribution="...">
  <speechSection SpeechCreator="..." AcousticMedia="wav"
Duration="10:01" SamplingFrequency="10" Resolution="4"
RecordDate="2010-02-01" RecordTime="11:24" Equipment="microphone"
SpeechFile="speech.wav" SpeechSegmentation="Start-stop">
    <speaker SpeakerName="..." SpeakerAccent="moldavian" Gender="Male"
Age="20-30" />
  </speechSection>
  <textSection TextCreator="..." TextFormat="UTF8" />
  <dataSection MetadataCreator="a" AnnotationCreator="a"
AnnotationLevel="SentenceAlign" />
  <title Ident="Resource" LongTitle="Bimodal Resource" />
</teiHeader>
```

Figure 1: Extract from a *teiHeader* tag

The segmentation and alignment of the resource is available in *unit* tags, which can be of three types, depending on the manner the resource is represented. The first type (“*file*”) is for resources that are stored in multiple files. So, for this case, each *unit* tag will have a child node called *speech*, which indicates the name of the file containing the speech component and a child node called *text* containing the textual transcription associated with the audio file specified.

¹⁰ <http://www.tei-c.org/>

The second type of segmentation is called “*start-stop*” and is used for those resources that present just one speech file, segmented and aligned at temporal boundaries, the text being reproduced between each two such consecutive markers. So, the *unit* tag will contain a *speech* subelement with two attributes *start* and *stop* (in seconds). Along with the *speech* tag, the *text* tag contains a reproduction of the text being spoken. Other components can be added in each *unit* group under specific tags (see an example in the next section).

The third type of segmentation is “*file-start-stop*”, which is a combination of the two types presented above. It is meant to accommodate those resources that contain multiple audio files and a “*start-stop*” segmentation for each of them. So, for each *unit* tag pointing to a speech file, a series of child nodes called *subunits* are also created. Each *subunit* will hold the “*start-stop*” segmentation, similar to the one described above. An example of such an annotation is shown in figure 2.

```
<teiHeader>...</teiHeader>
<unit>
  <speech speechFile="file1.wav"/>
  <subunit>
    <speech start="0.0000" stop = "8.5673"/>
    <text>...</text>
  </subunit>
  ...
  <subunit>
    <speech start="8.5673" stop="15.6652"/>
    <text>...</text>
  </subunit>
</unit>
...
<unit>
  <speech speechFile="file2.wav"/>
  <subunit>
    <speech start="0.0000" stop = "4.1231"/>
    <text>...</text>
  </subunit>
  ...
  <subunit>
    <speech start="3.7234" stop="10.423"/>
    <text>...</text>
  </subunit>
</unit>
```

Figure 2: Extract of a *file-start-stop* type of file

4. Designing and implementing convertors

In the bunch of resources contributed by different partners of the ReTeRom project we have identified three specific types of formats, according to which we have designed the first set of convertors, supposed to “understand” the corresponding files and on which they act accordingly to transform them to the CoBiLiRo standard.

THE COBILIRO PROJECT: BUILDING AND DISTRIBUTING A BIMODAL CORPUS FOR
ROMANIAN LANGUAGE

The first format is composed of groups of four files: a *wav* file - containing the audio recording, a *txt* file - containing the text associated with the recording speech, a *lab* file - containing the same text as the *txt*, but from which the punctuation has been eliminated and all letters are reduced to lowercase, and a *phs* file - containing a list of all letters present in the recording along with their start-stop moments. The conversion of this format to the CoBiLiRo standard starts with the creation of the header containing the metadata. Part of the information that fills in the header should be provided by the contributor through the form imposed by the interface. This type of resource is converted to the *file-start-stop* standard representation described above. All files belonging to the same group will have a unique name. After grouping the files, four subunits are created: *speech*, *text*, *lab* and *phs*. Their contents are extracted from, respectively: a *wav* file - containing the recorded segment of voice; a *txt* file - containing the textual transcription on the segment; a *lab* file - containing the same text but in only lower case letters and without punctuation; a *phs* file - containing the sequence of letters in the segment, each paired with time marks showing its start and end as it is pronounced in the *wav* file. In figure 3 you can see an example of such a file group.

PHS

```
0 3700000 pau pau
3700000 4400000 p purt\304\203torul
4400000 4900000 u
4900000 5200000 r
5200000 5600000 t
5600000 6200000 @
6200000 6700000 t
6700000 7200000 o
7200000 7500000 r
7500000 8100000 u
8100000 8400000 l
8400000 8800000 d de
8800000 9200000 e
```

LAB

purătorul de cuvânt al biroului electoral central marian muhuleț a adăugat că au mai rămas de centralizat procese verbale cu voturile exprimate în județele bacău bihor constanța olt vrancea și tulcea

TXT

Purtătorul de cuvânt al Biroului Electoral Central, Marian Muhuleț, a adăugat că au mai rămas de centralizat procese verbale cu voturile exprimate în județele Bacău, Bihor, Constanța, Olt, Vrancea și Tulcea.

Figure 3: First type of aligned corpora

The second format is called MULTTEXT/TEI and is composed of some audio files and an *xml* file containing metadata (not relevant to the scope of our platform) and a series of *div* tags mapping text to the audio files. The first step of the conversion, as in the previous case, is the creation of the CoBiLiRo header and it is done in the same manner as for the first format. Considering that there are multiple audio files of this type, the “*file*” representation is used. The next step is to identify the *div* tags that contain the mapping of the text to the audio files from the original *xml* file. Then the texts in-

between consecutive *div* tags are extracted and inserted into *text* tags belonging to different *units*. A *div* tag also contains an *url* tag, where the name of the audio file associated with the corresponding text can be found. This information is inserted into the *speech* tag of the output format belonging to the appropriate *unit* element. As such, the expected pairs of *xml* elements *<speech/>* - *<text/>* are formed. A fragment from such a file, including the corresponding speech resource, is given in figure 4.

```
<div id="sro.2" n="01" type="block">
<head>*BLOCK: 01</head>
<p id="sro.2.2"><s id="sro.2.2.1">Am o problem&#x103; cu aparatul de
dedurizare a apei.</s>
...</p>
<ab>[<xref url="../../spc/spch01-ro.wav">speech file</xref>]</ab></div>
```

Figure 4: A fragment of a MULTTEXT type alignment

The third format discussed here is called TEXTGRID and it contains groups of three files. The first type of file is an audio recording that contains the speech part of the resource. The *textgrid* file contains tuples of values (*letter*, *xmin*, *xmax*) referring to letters extracted from the text and the time interval between which each letter was spoken. The third file (*txt*) contains information about the energy of the enunciation of each letter, expressed in decibels and the speech frequency. After copying the header information, *unit* tags for each of the audio files are created, with the attribute *speechFile* containing the name of the audio file. Next, a series of child nodes, each containing a sub-element called *speech* and receiving the attributes *start* and *stop* are created. These attributes' values will represent the *xmin* and *xmax* values from the tuples present in the *textgrid* file. The *letter* values from the tuples will be placed in each *subunit* under the tag *text*.

5. The CoBiLiRO web platform

In order to provide a unified working space where all users can upload, store and find resources, we have created a web platform which facilitates collaboration. The platform is available for all CoBiLiRo users that have an account and a password¹¹. It integrates the roles of *Admin*, *Contributor* and *Trustee*.

The *Admin* controls the list of users and their credentials and can get information, through logs, on the flow of data on the Platform. This person also manages the creation of accounts from requests addressed by unregistered users.

A *Contributor* may upload its own resources. A user can gain this quality when she/he makes the first request to upload a resource to the Platform¹². After a resource has been uploaded, the platform processes the content and, provided its format is compatible with one it knows already (as explained in the previous section), it creates one or more *xml*

¹¹ <http://85.122.23.18:81/>

¹²For security reasons, this status can presently be trusted only to members of the ReTeRom consortium.

files that concentrate the content of the source files and one header file that includes all metadata that could be automatically extracted. Other information that complements the metadata need to be manually filled in.

Finally, a *Trustee* is a user that has the right to access resources, downloading them or only browsing their metadata.

All the roles described above can access statistics about the types, formats and number of records stored on the Platform.

In order to create an intuitive and friendly User Interface we used Razor, an ASP.NET programming syntax used to create dynamic web pages, and jQuery, a JavaScript library designed to simplify HTML DOM tree traversal and manipulation.

Metadata is client-side validated by checking that all required input fields are submitted. Server-side validation is done by assigning a format type and checking if specific conditions are satisfied for each type of input file. For data persistence, we have used a relational database named MariaDB. As an ORM¹³, in order to query our data easier, we used the Pomelo Entity Framework. Our project repositories are hosted on GitHub using Git for source control. In order to provide a high performance and cross-platform application we used the ASP.NET Core framework written in the C# programming language. Another reason we used this technology was that it provides us with data security (protection against SQL Injection and Cross-site request forgery) and a way to secure REST APIs using JSON Web Token. The website is hosted on premises on a CentOS Linux machine.

Figure 5: Information about an available resource

Our main concern in design was related to efficiency. As such, we tried to reduce at maximum the response time of each upload request, including format conversion runs. The server running the conversions has a E5-2650 v3 de 2.30GHz, with 20 cores reserved for conversion tasks.

6. Conclusions and future work

The Platform described hosts at present more resources contributed by the ReTeRom project partners from Bucharest, Cluj and Iași. Their different research interests (speech analysis, speech synthesis, recordings containing prosodic clues and linguistic enquiries

¹³A programming technique for converting data between incompatible type systems using object-oriented programming languages.

from different Romanian provinces) made up the initial bunch of formats for which we have built converters. The designed standard tried to find an equilibrium between these diverse needs.

At the moment, the Platform is ready to host resources observing the types of formats described, but we are open to find other formats and write converters accommodating them as well. What is still to come is the other facet of the transformation technology, convertors receiving in input the CoBiLiRo standard and exporting any of the ones we considered now in input. When this will be done, CoBiLiRo will become a real hosting-island platform, as an intermediate hub for text2speech and speech2text research technologies.

Through the Platform, a user can also trigger text processing operations, by calling the web NLP technologies installed on the RELATE Platform (Păiș *et al.*, in this volume). Other advanced speech processing operations will be integrated in the processing chain as soon as they will reach technological maturity in the ReTeRom project. This way the CoBiLiRo-RELATE tandem will become a research hub for advanced speech and text processing addressing the Romanian language.

Acknowledgements

This work was supported by a grant of the Ministry of Research and Innovation, Program PN-III-P1-1.2.-PCCDI, nr. 73/2018, as part of the ReTeRom project.

References

- Cho, K., van Merriënboer, B., Gulehre, F., Bougares, F., Schwenk, H., and Bengio Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*.
- Denes, P. and Mathews, M (1960). Spoken Digit Recognition Using Time-Frequency Pattern Matching. In *The Journal of the Acoustical Society of America*, 30, 1450-1455.
- Denes, P. (1959). The Design and Operation of the Mechanical Speech Recognizer at University College, London. In *Journal of the British Institute of Radio Engineer*, 19, 219-229.
- Donahue, J., Hendricks, L. A., Guadarrama, S., Rohrbach, M. (2015). Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, 677-691.
- Gîfu, D., Petic, M. (2014). Transliteration and alignment of parallel texts from Cyrillic to Latin. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*.

- Girshick, R. (2015). Fast R-CNN. In *Proceedings International Conference on Computer Vision (ICCV)*, 1440-1448.
- Godfrey, John J. and Edward Hollman (1997). Switchboard I, Release 2. In *Linguistic Data Consortium*, Philadelphia.
- Halle, M., and Stevens, K. (1962). Speech recognition: A model and a program for research. In *IRE Transactions on Information Theory*, 8(2), 155-159.
- Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. In *Proceedings Neural computation*, vol. 9, no. 8, 735–1780.
- Karpathy, A., Joulin, A., and Fei-Fei, L. (2014). Deep fragment embeddings for bidirectional image sentence mapping. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, 1889-1897.
- Khemakhem, Mohamed, Foppiano, Luca and Romary, Laurent (2017). Automatic Extraction of TEI Structures in Digitized Lexical Resources using Conditional Random Fields. In *Electronic Lexicography, eLex 2017*.
- Li, Ai-jun and Zhi-gang, Yin (2007). Standardization Of Speech Corpus. In *Data Science Journal*, Volume 6, Supplement, 18 November, 806-812.
- Llisterri, J. (Ed.). (1996). Preliminary recommendations on spoken texts. In *EAGLES Document EAG-TCWG-CTYP/P*, May 1996.
- Luo, R., Price, B. L., Cohen, S., Shakhnarovich, G. (2018). Discriminability objective for training descriptive captions. In *Proceedings IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6964–6974.
- Mihăilă, C. and Mekhaldi, D. (2009). Bimodal Corpora Terminology Extraction: Another Brick in the Wall. In *International Conference RANLP 2009 – Borovets, Bulgaria*, 236-240 (<https://www.aclweb.org/anthology/R09-1044/>).
- Păiș, V., Tufiș, D. and Ion, R. (2019). Integration of Romanian NLP Tools into the RELATE Platform. In *Proceedings of the 14th International Conference “Linguistic Resources and Tools for Natural Language Processing”*, Cluj-Napoca, 18-20 November 2019, 181-192.
- Rasso, T. and H. Mello (eds.) (2014). *Spoken Corpora and Linguistic Studies*, John Benjamins.
- Redmon, J., Divvala, S., Girshick, R., Farhad, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779-788.
- ReTeRom project webpage (2019), consulted as of September 2019 (http://www.racai.ro/p/reterom/index_en.html).
- Sperberg-McQueen, C.M. and Burnard, L. (2018). Original editors, revised and expanded under the supervision of the Technical Council of the TEI Consortium. In *EI P5: Guidelines for Electronic Text Encoding and Interchange*, version 3.3.0, last update: 31st January 2018, revision: f4d8439.

SPANISH LEARNER CORPUS CAES: A RESEARCH ON HOW RUSSIAN LEARNERS USE IMPERFECT TENSE IN SPANISH

MARÍA ADELAIDA GIL MARTÍNEZ

Instituto Cervantes de Moscú,

adelaida.gil@cervantes.es

Abstract

This article examines the use of imperfect tense in the interlanguage (IL) of second language (L2) Spanish learners whose first language (L1) is Russian. The study is based on a qualitative and quantitative analysis of written texts extracted from CAES (Corpus de Aprendizajes de Español) and conducted at the Instituto Cervantes of Moscow. The aim of this analysis is to highlight the influence of learner L1 on the selection of Spanish past tense verbal morphology, imperfect versus simple past, concerning the difference between tense and aspect in Russian and Spanish. The purpose of this article is, then, twofold. The first part will be a description of the CAES, a Spanish learner corpus consisting of written texts issued by Spanish learners of six different L1. The second part will deal with the results of a research on the selection of imperfect or simple past tenses by Russian students of Spanish when they refer to actions in the past with a durative aspect. The article will end with some pedagogical observations about designing language tasks for Spanish classes which may help to improve the learning teaching process.

Key words — learner corpora, CAES, interlanguage, aspect, tense, Russian learners.

1. Introduction

Learner corpora (LC) are becoming one of the most relevant and interesting linguistic resources for foreign language teachers and researchers. They not only describe second language learner's interlanguage but they can be used to identify typical difficulties in learners of a certain level (e.g. intermediate or upper-intermediate learners) or learners of a certain native language (Russian learners, for instance). Learner corpora, thus, will provide the basis for identifying the frequently occurring mistakes in the process of learning a language¹.

Although the last decade has witnessed the creation of large corpora of first language (L1) English and second language (L2) Spanish, such as CEDELE 2 (Lozano, 2009a; Lozano and Mendikoetxea, 2013), the oral SPLLOC (Mitchell *et al.*, 2009), it has not been until five or six years ago that Spanish corpora including learners with different (L1) backgrounds has become an important tool for researching, e.g. the oral CORELE (Campillos Llanos, 2014a) and the written CAES (Instituto Cervantes, 2015). Indeed, the CAES (*Corpus de Aprendizajes del Español*) has become one of the most promising electronic corpora in Spanish as a second and foreign language, since it includes written

texts issued by Spanish learners from L1 languages such as Russian or Chinese, in which not much research has been done.

This paper is divided as follows. Section 2 will be devoted to make known how CAES was designed and the functionalities it offers for future research on L2 Spanish learner corpora. Section 3 will examine the use of imperfect tense in the interlanguage of second language L2 Spanish learners whose L1 is Russian.

2. The CAES corpus: general features

The CAES corpus is a computerized corpus of Spanish as a foreign language, designed by the University of Santiago de Compostela and funded by the Instituto Cervantes. The project team is led by two established professors at the University of Santiago, Prof. Guillermo Rojo and Prof. Ignacio Palacios. It has been compiled between 2012 and 2014 and it contains at this moment 575.000 words taken from written texts issued by Spanish learners from six different native languages: Arabian, Mandarin Chinese, French, English, Portuguese and Russian.

Table 1: Main features of the CAES project

Compilers	Participant's native language		Participants' gender		Participants' level		Participants' main countries represented	
(G. Rojo, I. Palacios).	Arabian	497	male	521	A1	526	Brasil	31
	Portuguese	361	female	902	A2	421	Marruecos	9
	English	227			B1	252	EE.UU.	31
	French	143			B2	162	China	2
	Mandarin	128			C1	62	Francia	13
	Chinese	67					Siria	9
	Russian						Rusia	12
							Afganistán	7
							Irlanda	92
							Argelia	70
							Portugal	62
							Líbano	52
							Jordania	38
							Túnez	32
								31
								26
								21
								16

Table 1 shows 1.423 participants from over twenty different countries, 502 males and 921 females, whose level ranges from A1 to C1 based on CEFRL. The greatest number of participants come from Brazil and Morocco, followed by United States and China due to the number of Instituto Cervantes (IC) in those countries: there are five ICs in Brazil and Morocco. There are four ICs in United States and one in each of the participant countries. The number of participants in China can be explained for the growing interest of the Chinese in learning Spanish. As it has been said before, it is easily seen here the main value of this corpus, i.e., the online availability of a great

SPANISH LEARNER CORPUS CAES: A RESEARCH ON HOW RUSSIAN LEARNERS USE IMPERFECT TENSE IN SPANISH

amount of written texts coming from students of different native languages and ranging from different proficiency levels.

2.1. The CAES Corpus.

Table 2: Participants' distribution according to their L1 and proficiency

	Arabian	Chinese	French	English	Portuguese	Russian
A1	599	189	132	77	494	66
A2	364	100	88	344	257	58
B1	232	69	85	127	123	41
B2	99	15	48	41	99	11
C1	48	0	18	26	28	0

Table 2 reveals some important facts to consider in future research: most of the participants are found in the lowest level of language proficiency (A1, A2 and B1). There are no C1 participants in Chinese and Russian and there are few in the other four languages. Samples in these levels cannot be representative enough to carry out fine-grain corpora-based studies.

Table 3: Participant's distribution according to their L1

L1	Words	Sample units (texts)
Arabic	168.231	497
Mandarin Chinese	53.163	128
French	58.412	143
English	106.968	227
Portuguese	165.231	361
Russian	20.713	67

Table 3 shows participant's distribution according to L1 and proficiency. It might seem that the size of the corpus is very humble compared to other important corpora. For instance, *Aprescrilov* (*Aprender a escribir*), a corpus of Spanish as a foreign language, with Dutch as L1, includes one million words. It contains 2700 written texts in English with levels of proficiency ranging from A1 to C1. Besides, there are some research carried out with data from *Aprescrilov*: difference between SER/ a research, prepositions (Buyse and Melón, 2013; Buyse *et al.*, 2015; 2016). Furthermore, CEDEL2 contains 750.000 words from 2400 English participants. ESPALEX, a corpus designed for studying the acquisition of Spanish as a foreign language and built on written texts coming from DELE B2 exams, is also worth mentioning. The corpus contains 12.576 written texts organized according to text-types criteria (Gisbert and Iglesias, 2011). (2) So, at this point, it would be advisable to ask what is special about CAES.

2.2. CAES' particularities

According to Lozano and Mendikoetxea (2013), it is important to use learner corpora that have well-established corpus design principles. The three strengths of CAES are the

following: the variety of L1, the search tool and the CAES reference guide, all of them characteristics that “permiten construir una vision de conjunto de la procedencia de los textos y de las fortalezas y limitaciones que ellos encierran (Parodi, 2015:195).

i. Six different L1. All the texts compiled in CAES come from six different L1 languages, some of them hardly represented within the realm of second language acquisition, such as Russian and Arabian. This means that the new data provided by CAES will allow to carry out research on interlanguages different from English as L1, as in most already mentioned corpora. That is to say: English will no longer be the only L1 in learner corpora, but it will be another language to carry out linguistic research. In multilingual societies such as the European one, it is good to see that research might be done in a wide variety of languages as L1. Furthermore, it is important to keep in mind that many students in Europe speak three or more languages, and many of them come from bilingual and trilingual contexts. The results obtained will vary if the start point is a bilingual, monolingual or trilingual context. Plurilingual competence seems to appear as an important characteristic to be taken into account as far as teacher’s decisions are concerned.

ii. The search tool. The search tool retrieves statistical information and textual examples of elements, lemmas, word classes and grammatical categories with filters (learner’s L1 and level of proficiency, age, sex, country of origin, etc.).

Figure 1: www.galvan.usc.es/caes/search

Moreover, lower and higher case words may be distinguished as well as accented or non-accented, a relevant feature in Spanish written texts. Searches based on co-occurrence of several elements can also be conducted.

iii. The CAES reference guide¹. It is, without any shadow of doubt, one of the most revealing documents concerning CAES. It is much more than a consulting guide for the application, since it becomes a kind of solid road map of how to move through CAES application. In the first place, it explains, step by step, how to use and exploit all the search options by offering a wide variety of examples. Each search category is illustrated by the corresponding screen in CAES, something that it is really helpful and it is missed in another language corpora.

2.3. Stages in CAES compilation

There is another important characteristic to highlight concerning CAES: how the written texts are compiled, encoded and annotated. In order to do that, we have seen so far the kind of contextual information CAES may provide according to the characteristics of researching. I would like to focus now on the four stages for the compilation and collection of the data: before the data collection, data collection, text and encoding annotation and the search tool.

2.3.1. Before the data collection

A computer application was designed for the data collection so that participants themselves could enter the data directly in the computer. Then, a protocol was prepared and distributed among all the institution involved in the project, such as the Instituto Cervantes centers and a large number of universities worldwide. Before that, the computer application was piloted with several groups of students to prove all the protocols and refine all the process of collecting data so that it became as easier and effective as possible. Participants, on the other hand, are asked to sign a consent form for the use the data obtained.

2.3.2. Data collection

All participants (3 on average). These tasks were designed according to the CEFR descriptors and DELE test as well as in accordance with the CI's General Curricular Document. Here we have some examples of activities: writing emails to friends and relatives, applying for a job, booking a hotel room, making a complaint, writing a funny story, etc.

2.3.3. Text encoding and annotation

Texts integrated into CAES adopt the forma of XML documents and all of them were tagged both automatically and manually. A total of 702 different tags were used. *FreeLing*, an open source language analysis tool suite, was used to make the necessary adjustments of the equivalences between the *FreeLing* tagging system and the one our team intended to use. Finally, the texts were manually disambiguated.

¹ <http://galvan.usc.es/caes/guia>

2.3.4 The Search tool

As it has been said before, the search tool allows to retrieve all kind of statistical, textual and grammatical information.

Resultados 1 a 7 de 7

1	B1/Ruso	Pues este día justo antes de salir de fiesta	estaba aprendiendo las palabras que tienen que ver con la cocina+la coc
2	B1/R	Estudiante: 1311 Tarea: 3710 País: Rusia Sexo: Mujer Edad: 23 Estudios que realiza / ha realizado: Universidad Edad de inicio en el estudio del español: 22 Número de meses en países de habla española: 5 Número de meses estudiando español: 5	todos , un chico estaba contando a mi amiga sobre la systema de limpieza en la ciudad .
3	B1/R	Contactos personales: Amigos	Mi maleta estaba de color amarillo , muy grande+ ,+112_1 grande+ ,+112_1 grande+ ,+1
4	B1/Ruso	Esta situasion	estaba en la calle_de_Venezia.
5	B1/Ruso	Y cuando	estaba en un pueblo cerca de Leon , en casa de mis amigos que tienen los
6	B1/Ruso	Quería a escribir te antes , pero es que	estaba tan ocupada todos los días , que no tenía tiempo para nada .
7	B1/Ruso	La semana pasada	estaba viajando desde Madrid a Moscú a las 22.30 en este avion .

Figure 2: <http://galvan.usc.es/caes/search>

3. Study on how the contrast imperfect /simple past tenses in Spanish: a brief study on aspect

Verbal aspect in Russian language is a complex phenomenon that has been at the forefront of attention at the realm of language acquisition in both L1 and L2. I have been teaching Spanish language in Moscow for six years and one of the main challenges for the students is to understand how to convey the contrast between the imperfect tense (*Ella bailaba salsa al salir de clase* “She used to dance salsa after school”) and the simple past (*Ella bailó salsa al salir de clase* “She danced salsa after school”). The two past forms in Spanish refer to an action that took place in the past, but morphologically encode different aspects, perfective and imperfective, respectively. By using the imperfect tense, it is meant that “bailaba” used to be a habit or routine for this girl in the past. By using the simple past, it is meant that the girl *bailó salsa* only once, in a punctual moment in the past. Russian students understand in both sentences that the girl *bailó salsa* for an unlimited period of time. She just *danced*, and “danced or was dancing” implies *durative aspect*. It is difficult for them to note that the form “bailaba” in Spanish also encodes the meaning of “habitual action in the past” and not only the feature of “*durative aspect*”. Russian students think in terms of aspect or, as Comrie defined it, “internal temporal constituency of the one situation” (Comrie, 1976:5) while in Spanish language, aspect is conveyed, among others, through periphrastic structures such as verb *estar* + *gerund*, “*Estuvo bailando toda la noche*”.

3.1. Aspect in Russian language: some basics.

In Russian, speakers distinguish between a binary opposition of perfective and imperfective aspect with morphologically distinct forms. According to Timberlake (1985: 286-287), imperfective forms acquire semantic features of “cancellation of an action, conation, taking the emphasis away from the result and a type of state that comes from the result”. The perfective forms, by contrast, “denotes completion and result”. In the example *ты смотрел фильм на английском?*, an imperfective form is used, since “the speaker is merely interested in the fact that the type of action named did occur or

SPANISH LEARNER CORPUS CAES: A RESEARCH ON HOW RUSSIAN LEARNERS USE
IMPERFECT TENSE IN SPANISH

din not occur” (Forsyth, 1970:82), and not whether the speaker saw the entire film or not. In that case they would say: ты посмотрел фильм на английском?

Tatiana Drosdov (2002: 36) points out at four sub-categorical features to understand the semantic-aspectual opposition between Russian and Spanish.

Table 5: Sub categorical features in the semantic-aspectual opposition Russian-Spanish. Cf. Drosdov, 2002.

RUSSIAN		SPANISH	
Aspectual meanings	ASPECT		Verbal forms to express aspectual meanings in Russian.
	Imperfective	Perfective	
1. Complete action (limit of time)	-	+	Present perfect; simple past; past perfect Periphrasis: <i>romper a/echar a/acabar de + infinitive</i>
2. Durative aspect (certain period of time); <i>Action as a process.</i>	+	-	Imperfect Periphrasis: <i>estar/andar/venir + gerund</i>
3. Iterative aspect. <i>The repetition of an event observable in one single occasion; Repeated, habitual actions.</i>	+	-	Imperfect and present tense.
4. Result of an action.	-	+	Present perfect; simple past; simple future Periphrasis: <i>tener/estar/llevar /dejar + participle</i>

Table 5 provides four semantic criteria to understand the relationship between the aspectual meanings in Russian and how they are encoded in the Spanish verbal system. Drosdov points out to the fact that the imperfective aspect occurs in situations “que designan una circunstancia reiterativa, habitual o en progreso y evolución” (Drosdov, 2002: 36), i.e. situations that do not indicate limit of time. Perfective aspect, by contrast, occurs in situations “que abarcan acontecimientos puntuales o acabados”, i.e. they denote completion and result.

By transferring these aspectual meanings to the learning of the Spanish language, it is clearly seen that Russian learners choose the imperfect tense to convey durative and iterative aspect, while it is conveyed through simple past in Spanish. These will be the aspectual meanings to work with by designing materials and teaching.

3.2. Imperfect tense used as a simple past: findings in CAES

It is convenient to remember that this brief study on the use of the Spanish imperfect tense rather than the simple past with a “durative” aspect is based on an initial hypothesis. This is:

“Russian students of L2 Spanish will choose the imperfect tense and not the simple past in circumstances in which they have to talk about an action in the past that they feel lasts a long time despite a finished process being referred to”.

The CAES was chosen to verify this hypothesis. The purpose was to find out in what contexts and to what degree Russian students use imperfect tense instead of simple past when encoding durative aspect and. Once analyzed these contexts, next step would be to understand why and how to transform these findings into class activities.

Before analyzing the CAES findings, some data should be highlighted. The number of Russian participants was 176. According to their proficiency level, the distribution was the following:

Table 6: Participants' distribution according to their level proficiency

Proficiency level	Number of participants
A1	66
A2	58
B1	41
B2	11
C1	0

Most participants were in level A1 and level A2, as for all languages in CAES. In order to carry out this study, only participants from A2 to B2 were taken into account, because past tenses are taught for the first time in A2 level. That means 110 participants in total. The type of search was based on grammatical tags: *principal verb, indicative mood, imperfect tense (copreterite)*. The others options chosen were: *level, country and L1*.

The screenshot shows the CAES search interface. The top navigation bar includes 'CAES', 'Datos', 'Frecuencias', 'Búsquedas', 'Ayuda', and 'Contacto'. The 'Búsquedas' section is active, showing search criteria and filters. The search type is 'Elem. gramaticales', and the result type is 'Ejemplos'. The order is 'Coincidencia' and the page size is '50'. The filters section includes 'Sensibilidad' (Acentos: No, Mayúsculas: No), 'Filtros' (Nivel de español: B1, País: Rusia, L1: Cualquiera, Sexo: Cualquiera, Edad entre:), and 'Elementos gramaticales' (estaba, VMil*, Lema). The search results show 7 results, with the first two displayed. The first result is 'B1/Ruso' and the second is 'B1/Ruso'.

Figure 3: www.galvan.usc.es/caes/search

3.3. Results

Table 7: Statistical inferences

	A1	A2	B1	B2	TOTAL
IMPERFECT TENSE	0	5/92 7,6%	4/53 7,5%	2/22 9,09%	11/167 6,5%

Table 7 shows that there are 167 examples in which Russian students use the imperfect tense. However, the imperfect tense is used instead of simple past only in 11 examples (6,5%). It is seen that this pattern occurs at all levels of proficiency except at A1. This is expected, since imperfect and simple past tenses are taught at A2 level. The highest percentage of occurrence takes place at level B2 (2 examples out of 22, which means 9,09%), but the total percentage (6,5% 11 tokens out of 167) would not indicate a serious problem. It is felt the other way around when teaching Spanish or correcting exams, however, as it is seen in the sample of A2 results.

Table 8: Sample of A2 results taken from CAES

1. Visitamos las casas de Gaudi, Sagrada_Familia, museo de Picasso ... Ademàs de est	paseabamos por el parque_Gúel, comimos paella , bebimos el vino nativo y hicimos
2. El se	preparaba bien y empezó a estudiar en este Univer sidad en Moscu.
3. Y, sin duda, como tu sabes,	probabamos toda la comida local y tomabamos m ucho cerveca,
4.	teníamos bastante tiempo para ver toda Bavaria, por eso quiero
5. Y, sin duda, como tu sabes, probabamos toda la comida local y	tomabamos mucho cerveca, tu sabes mi marido ...

As it is seen in these examples, the use of the imperfect tense in Spanish instead of simple past is due to the fact that the action lasts for a long time (*durative aspect*). Since they use the imperfective form in Russia (they think in terms of aspect and not in terms of linguistic time), the transfer into Spanish is made automatically: *paseábamos* (*go for a walk*) instead of *paseamos*. *Pasear* indicates durative aspect, they feel that the action is not finished. *Preparaba* (*he was preparing*) instead of *prepare*. They do not consider the fact that the period of time for he was preparing himself is ended. They think about

the fact of being prepared. *Teníamos* (we had) instead of *tuvimos*. Although they are talking about an amount of time (*bastante tiempo*), they consider the internal time of the action “*tener*” and not whether the action is over or not.

4. Conclusions

The main objective of CAES is to become a computational tool designed to carry out contrastive analysis among different languages in order to research how students learn a foreign language (in this case, Spanish) and which aspects are of greatest difficulty for them. This study examines how Russian learners of Spanish use the imperfect tense instead of simple past in order to convey the concept of durative and iterative aspect. The hypothesis of the study is that Russian students transfer from their L1 the feature of “durative aspect” into imperfect tense in Spanish. However, durative aspect in Spanish is encoded within simple. Hence, aspect/tense mistakes occur and produce a kind of miscommunication, since the realization of aspect is completely different in both languages. This is the reason why Russian students of Spanish cannot understand, perceive and transfer the meaning of “aspect” from Russian into Spanish. As a matter of fact, they do transfer the meaning of aspect, but the realization in both languages does not match.

The main idea to improve the process of learning Spanish is to teach periphrasis from the very beginning, so that the students might understand, perceive and sense how to transfer the concept of durative aspect into Spanish, and how to use the right form in the different contexts they occur. Moreover, there are some recommendations by teaching this grammatical theme in class:

1. It is important to analyze the meaning of “tenses” in Spanish, since they do not match with Russian tenses.
2. Students should be familiarized with periphrasis from the A2 level, so that they can feel how to convey actions with a limit of time.
3. Contrastive analysis should be implemented in class by means of activities that makes them aware of this difference between tense and aspect.
4. CAES should be a heuristic tool to be implemented in class, since it will be help them to do some linguistic research.

References

- Alonso Ramos, M. (Ed.) (en prensa). *Spanish learner corpus research: current trends and future perspectives*. Amsterdam: John Benjamins.
- Arche, M. J. and Domínguez, L. (2011). Morphology and syntax dissociation in SLA. A Study on clitic acquisition in Spanish. En Galani, A., Hicks, G. and Tsoulas, G.

- (eds.), *Morphology and its Interfaces*, Amsterdam & Philadelphia: John Benjamins, 291-320.
- Bailini, S. (2013). SCIL: A Spanish Corpus of Italian Learners. *Procedia — Social and Behavioral Sciences, Corpus Resources for Descriptive and Applied Studies. Current Challenges and Future Directions: Selected Papers from the 5th International Conference on Corpus Linguistics (CILC2013)* 95, 542-549.
- Baralo Ottonello, M. (2004). La interlengua del hablante no nativo. En Sánchez Lobato, J. and Santos Gargallo, I. (Dirs.), *Vademécum para la formación de profesores. Enseñar Español como segunda lengua (L2)/Lengua extranjera (LE)* Madrid: SGEL, 369-389.
- Bullock, B. and Toribio, A.J. (2011). Creating a linguistic landscape of Spanish speakers in the US, starting in Texas. From an August 2011 interview with COERLL [Centre for Open Educational Resources and Language Learning]. Accesible en <http://sites.la.utexas.edu/voices/barbara-bullock-and-almeida-jacqueline-toribio/> [visita 30.10.2016].
- Bustos Gisbert, J. M. and Sánchez Iglesias, J. J. (2011). Espalex: un corpus para el estudio de la adquisición del español como lengua extranjera. En Hernández González, C., Carrasco Santana, A. and Álvarez Ramos, E. (Coord.), *La Red y sus aplicaciones en la enseñanza-aprendizaje del español como lengua extranjera*, Valladolid: ASELE, 149-159.
- Buyse, K., Fernández Pereda, L. and Verveckken, K. (2015). The reference to L1 and L2 in SFL: proposals based on the Aprescrilov learner corpus. *Procedia: Social and Behavioral Sciences*, 173, 274-278.
- Buyse, K., Fernández Pereda, L. and Verveckken, K. (en prensa). The Aprescrilov corpus, or broadening the horizon of Spanish language learning in Flanders. En Alonso Ramos, M. (Ed.), *Spanish learner corpus research: current trends and future perspectives*. Amsterdam: John Benjamins.
- Buyse, K. and González Melón, E. (2013). El corpus de aprendices Aprescrilov y su utilidad para la didáctica de ELE en la Bélgica multilingüe. En Blecua, B., Borell, S., Crous, B. and Sierra, F. (eds.), *Plurilingüismo y enseñanza de lenguas*, Girona: ASELE, 247-252.
- Campillos Llanos, L. (2012). *La expresión oral en español lengua extranjera: interlengua y análisis de errores basado en corpus*. Tesis doctoral UAM.
- Campillos Llanos, L. (2014). A Spanish Oral Learner Corpus for Computer-Aided Error Analysis. *Corpora* 9 (2), 207-238.
- Cestero Mancera, A. M., Penadés Martínez, I., Blanco Canales, A., Camargo Fernández, L. and Simón Granda, J. (2001). Corpus para el análisis de errores de aprendices de E/LE (CORANE). En Gimeno Sanz, A. M. (Ed.), *Tecnologías de la información y de las comunicaciones en la enseñanza de ELE: Actas del XII Congreso Internacional de ASELE*, Valencia: Universidad Politécnica de Valencia, 527-534.
- Cestero Mancera, A. M. and Penadés, I. (2009). *Corpus de textos escritos para el análisis de errores de aprendices de E/LE (CORANE)*. CD-ROM. Universidad de Alcalá.
- Conrad, S. (2005). Corpus Linguistics and L2 Teaching. En Hinkel, E. (Ed.), *Handbook of research in Second Language Teaching and Learning*, Mahwah, NJ: L. Erlbaum Associates, 247-256.

- Díaz, L. (2007). *Interlengua Española: Estudio de Casos*. Barcelona: Printulibro Intergrup.
- Drosdov, T. (2002). Gramática del tiempo y aspecto en ruso y en español: similitudes y divergencias. *Eslavista Escolastica*, Complutense: Madrid.
- García Salido, M. (2014). O uso de construcións con verbo soporte en aprendices de español como lingua estranxeira e en falantes nativos. *Cadernos de Fraseoloxía galega*, 16, 181-198.
- Granger, S. (2002). A Bird's-eye view of learner corpus research. En Granger, S., Hung, J. and Petch-Tyson, S. (eds.), *Computer Learner Corpora, Second Language Acquisition and Foreign Language Teaching*, Amsterdam: John Benjamins, 3-33.
- Granger, S. (2009). The contribution of learner corpora to second language acquisition and foreign language teaching. En Aijmer, K. (Ed.), *Corpora and Language Teaching*, Amsterdam & Philadelphia: Benjamins, 13-32.
- Granger, S., Gilquin, G. and Meunier, F. (2015). Introduction: Learner corpus research —past, present and future. En Granger, S., Gilquin, G. and Meunier, F. (eds.), *The Cambridge Handbook of Learner Corpus Research*, Cambridge: Cambridge University Press, 1-5.
- Instituto Cervantes (2014). *Corpus de aprendices de español como lengua extranjera (CAES)*. Aplicación y guía de consulta en <http://galvan.usc.es/caes> [visita 30.10.2016].
- Koike, D. (2007). *Spanish Learner Corpus and Exercises*. Austin: Universidad de Texas.
- Lozano, C. (2009). CEDEL2: Corpus Escrito del Español L2. En Bretones Callejas, C. M. et al. (eds.), *Applied Linguistics Now: Understanding Language and Mind / La Lingüística Aplicada Hoy: Comprendiendo el Lenguaje y la Mente*, Almería: Universidad de Almería, 197-212.
- Lozano, C. (2014). Word Order in Second Language Spanish. En Geeslin, K. L. (Ed.), *The Handbook of Spanish Second Language Acquisition*, Oxford: Wiley-Blackwell, 287-310.
- Lu, H. (2012). An Annotated Taiwanese Learners' Corpus of Spanish, CATE. *Corpus Linguistics and Linguistic Theory*, 6 (2), 297-300.
- Mark, K. L. (1998). The Significance of Learner Corpus Data in Relation to the problems of Language Teaching. *Bulletin of General Education*, 312, 77-90.
- Mas Álvarez, I. and A. Gil Martínez (2018). Los corpus de aprendices: un terreno en expansión para la enseñanza del español. En Ellison, M. et al. (Orgs.), *As línguas estrangeiras no ensino superior: propostas didáticas e casos em estudo*, Oporto: FLUP e-DITA, 35-55.
- Mendikoetxea, A. (2014). Corpus-Based research in Second Language Spanish. En Geeslin, K. L. (Ed.), *The Handbook of Spanish Second Language Acquisition*, Oxford: Wiley-Blackwell, 11-29.
- Mitchell, R., Domínguez, L., Arche, M., Myles, F. and Marsden, E. (2008). SPLLOC: A New Corpus for Spanish Second Language Acquisition Research. En Roberts, L., Myles, F. and David, A. (eds.), *EUROSLA Yearbook 8*, Amsterdam: John Benjamins, 287-304.
- Myles, F. (2005). Interlanguage Corpora and Second Language Acquisition Research. *Second Language Research* 21 (4), 373-391.

SPANISH LEARNER CORPUS CAES: A RESEARCH ON HOW RUSSIAN LEARNERS USE
IMPERFECT TENSE IN SPANISH

- Orol González, A. and Alonso Ramos, M. (2013). A Comparative Study of Collocations in a Native Corpus and a Learner Corpus of Spanish. *Procedia-Social and Behavioral Sciences*, 95, 563-570.
- Palacios Martínez, I. M. (2016). Applications of learner corpora to Language Teaching. The CAES 'Corpus de aprendices de español' project as a case in point. Comunicación presentada en el 34 Congreso Internacional AESLA, Alicante (España) 14-16 de abril de 2016.
- Parodi, G. (2015). Reseña del Corpus de aprendices de español (CAES). *Journal of Spanish Language Teaching*, 2 (2), 194-200.
- Rojo, G. (2016). Corpus textuales del español. En Gutiérrez-Rexach, J. (Ed.), *Enciclopedia Lingüística Hispánica*, Oxon: Routledge, 285-296.
- Rojo, G. and Palacios Martínez, I. M. (en prensa). Learner Spanish on computer: The CAES 'Corpus de Aprendices de Español' project. En Alonso Ramos, M. (Ed.). *Spanish learner corpus research: current trends and future perspectives*. Amsterdam: John Benjamins.
- Seidlhofer, B. (2002). Pedagogy and local learner corpora. Working with learning-driven data. En Granger, S., Hung, J. and Petch-Tyson, S. (eds.), *Computer Learner Corpora, Second Language Acquisition and Foreign Language Teaching*, Amsterdam: John Benjamins, 213-234.
- Sinclair, J. (1996). *EAGLES: Preliminary recommendations on Corpus Typology*. <http://www.ilc.cnr.it/EAGLES/corpus/corpus.html> / [visita 30.10.2016].
- Tracy-Ventura, N., Mitchell, R. and McManus, K. (en prensa). The LANGSNAP Longitudinal Learner Corpus: Design and Use. En Alonso Ramos, M. (Ed.). *Spanish learner corpus research: current trends and future perspectives*. Amsterdam: John Benjamin.

AUTOMATIC IDENTIFICATION AND CLASSIFICATION OF LEGAL TERMS IN ROMANIAN LAW TEXTS

ANDREI COMAN¹, MARIA MITROFAN², DAN TUFIȘ²

¹ *Colegiul Național de Informatică “Tudor Vianu”*

² *Research Institute for Artificial Intelligence “Mihai Drăgănescu”, Romanian Academy*

andreicomman299@gmail.com

{maria, tufis}@racai.ro

Abstract

This paper presents the process of organizing a corpus consisting of approximately 150,000 legal documents into domains and subdomains. The high number of documents creates the need of a highly efficient term identification and classification algorithm. We combine several string-matching techniques and language specific properties in order to develop a linear time approximate-string-matching algorithm with a reduced number of false positive matches. Thus, we tackle the issue of terms identification and also integrate the IATE and EuroVoc descriptors within the Romanian legal corpus. Another important result presented within this paper is the ability to classify the legal documents according to the categories defined by EuroVoc.

Key words — document classification, legal language, natural language processing, terminology extraction.

1. Introduction

In this paper, we describe the results of the “Multilingual Resources for CEF.AT in the legal domain” (MARCELL)¹ action, whose final goal is to deliver clean, validated domain-specific language resources based on the national legislation (laws, decrees, regulations) of the seven involved countries, including Romania. The presented work will help create a large-scale pre-processed monolingual corpus and a comparable corpus of seven languages aligned at topic level domains. In addition, through this project the automatic translation in CEF. AT on the body of national legislation in seven EU official languages (Bulgarian, Croatian, Hungarian, Polish, Romanian, Slovak, Slovene) will be enhanced. All the documents are classified using the top-level domains of the EuroVoc thesaurus² and are also annotated with legal terms extracted from the Interactive Terminology for Europe (IATE)³ terminology database. All the texts exist in electronic form.

This paper is structured as an iterative process, showcasing the work behind the term annotation step as an approximate-string-matching issue and the steps of designing a

¹ <https://marcell-project.eu/>

² <https://eur-lex.europa.eu/browse/eurovoc.html>

³ <https://iate.europa.eu/home>

suitable conceptual model, based on several algorithms from the literature. Thus, we present the solutions used together with the downsides of the already existent methods and also the problems which we encountered while developing the algorithm (i.e. processing unlabeled documents, dealing with large data volumes and limited time and space complexity, defining a model to map the IATE terms onto more compact structures etc.).

The first part of the paper presents the statistics of the collected corpus and the results of the preparation step for the subsequent term identification – tokenization, lemmatization and annotation with POS tags. For this purpose, we have used the tools developed in the Research Institute for Artificial Intelligence “Mihai Drăgănescu”, Romanian Academy (RACAI). These tools have been designed for general Romanian language, but they can also be exploited for the law sublanguage. In the second part we present the methodology used to identify the legal terms in each document and the obtained results.

2. Related Work

The approximate-string-matching literature has lately been heavily oriented towards two main directions: deterministic algorithms based on string distance functions – such as the Levenshtein or the Hamming distance – and probabilistic Natural Language Processing techniques.

The algorithms falling in the former category usually become impractical due to their space and time complexity, especially when processing large databases of texts. In this paper, we develop our algorithm starting from the concepts of the k-mismatch Levenshtein Automaton, built around the concept of Levenshtein distance, (Schulz and Mihov, 2001; Touzet, 2016), and the Aho-Corasick data structure (Aho and Corasick, 1975) and merge them in order to obtain an algorithm with linear time and satisfactory results. Other algorithms address similar problems with the Hamming distance or periodicity properties, but often require the corpus to be processed for each entry of the dictionary (Amir *et al.*, 2004; Cole and Hariharan, 2002).

Currently, the most suitable algorithms now fall in the latter category, where issues of terms annotation are dealt with more frequently (Pala *et al.*, 2010). As those usually require prior knowledge to some extent or involve a long training period, we are aiming to develop an efficient, linear-time deterministic algorithm to solve the approximate-string-matching problem in the context of the Romanian legal corpus.

3. Collecting and Processing the Data

Text corpora have become an indisputable and mandatory source of research data in linguistics during the last few decades. Often, (huge) web corpora are used for either research or as training data, but in general most of the texts gathered in a corpus have licensing and copyright restrictions. However, there is a type of text data that is exempt from copyright protection (unlike web-based texts) – the law body, which is often (as it happens in Romania, too) specifically excluded from copyright laws.

In the context of MARCELL project we collected via crawling the law documents available on⁴ and a corpus of legal documents was created.

This resource contains 334,035,306 words distributed in 148,272 files ranging from 1881 to 2018 (see Table 1). It can be seen that the majority of the documents have been issued after the year 2000.

Table 1: Number of documents per year

Period	No. of documents
1881-1949	308
1950-1989	3,821
1990-1999	21,005
2000-2018	123,138

This corpus includes mainly: governmental decisions (25%), ministerial orders (18%), decisions (16%), decrees (16%), laws (6%) and others (19%).

Initially all the text collected were collected in .txt format, but in order to be used in further complex tasks such as automatic identification and classification of terms, all the legal documents were subjected to a preprocessing step. Therefore, using the Romanian Portal of Language Technologies (RELATE)⁵, the corpus was tokenized, lemmatized and annotated with part-of-speech tags.

4. Proposed Methodology

Having to find the approximate matches of a *partially unlemmatized dictionary* (as IATE is) within a *lemmatized corpus* (as all our corpora are) yields the need for a prior definition of a match. Thus, as a starting point we define a match as a pair of terms at a Levenshtein distance not exceeding a certain threshold (a predefined fraction of the length of the shortest term). This definition leads to an intuitive solution involving Levenshtein automata (Schulz and Mihov, 2001). As their proposed method involves prior knowledge of the terminology database (the automaton accepting the entire database) and passing each word of the corpus through an automaton, the size of the legal database makes this method unviable. Moreover, after matching the individual words, we would then need to compute the matches formed by associating these words. Thus, this approach was excluded from the conceptual stage.

Ideally, we would want to adapt linear-time perfect-string-matching algorithms to mimic the results yielded by the Levenshtein automata, according to our definition of a match. As the issue we are dealing with is identifying the approximate matches of dictionary terms in a database, the two most suitable techniques in the literature may be either the Knuth-Morris-Pratt algorithm (Knuth *et al.*, 1977) or the Aho-Corasick algorithm (Aho and Corasick, 1975).

⁴ <http://legislatie.just.ro/>

⁵ https://relate.racai.ro/index.php?path=teprolin/doc_dev

Instead of viewing the Levenshtein automaton as an automaton which takes a term as an input and returns its Levenshtein distance from a standard term, we can redefine it as an automaton which only accepts a term if it coincides with another term, which is previously defined as having an acceptable distance from the standard. Thus, we obtain a dictionary of acceptable terms (by their Levenshtein distance), which we have to match against a database. Obviously, this verification can be done in linear-time by inserting all the acceptable terms in an Aho-Corasick trie-like structure (with additional links between different internal nodes).

Through this approach, we are confronted with another issue. As the length of the dictionary entries increases linearly, the number of accepted terms increases exponentially – for a certain entry of length l , in a dictionary of cardinality σ and a maximum acceptable Levenshtein distance of d , the total number of acceptable terms would have the order of $(\sigma l)^d$ making it impossible to process all the possibilities. Thus, this approach was also excluded before its actual implementation, due to the unviable space complexity.

Therefore, instead of inserting all of these valid entries, we want to compress their essential information as much as possible. One possible and intuitive solution is to not insert the terms themselves, but instead a compressed mask which is a representative of all the acceptable forms of a term (if we had prior information, this mask would have been the lemma of the dictionary entry).

Based on this idea, we want to identify a function f defined over W (where W is the set of all Romanian terms in the IATE database) which compresses any given term to an associated mask. The function should be chosen such that different forms of the same term yield the same mask and different terms yield different masks (otherwise we could not distinguish different terms between each other).

For any such function, in order to better analyse its behaviour, we define two parameters which are the most relevant:

- i) The compression rate, which is equal to the average over W of the ratio between the length of the mask and the length of the initial term. As a more formal definition, we write:

$$C(f) = \frac{\sum_{w \in W} \frac{\text{len}(f(w))}{\text{len}(w)}}{|W|} \quad (1)$$

- ii) The matching rate, which is calculated based on both the correct and incorrect matches. As these definitions are only used for an intuitive analysis, we could use the following formal definition for the matching rate:

$$M(f) = p(1 - f) \quad (2)$$

where p is the fraction of positive matches which are detected and f is the fraction of all matches which are false-positives.

As we do not have any prior knowledge of the dictionary (i.e. the lemmatized forms), the functions subject to analysis will be purely based on the terms' structure. Thus, if the compression rate is close to unity (there is little to no compression), the function will not be able to discard the differences between several forms of the same input term. Similarly, if the compression rate is close to zero, too much information will be lost and the number of false positives will increase considerably. Therefore, we expect the graph of matching rate in relation to the compression rate to have a concave shape and to register a maximum value, which we would ideally want to achieve.

Regarding the compression itself, in order to maximize the identified positive matches, we want to find a general way to obtain an "invariant form" of any term. In other words, we want to manually identify general redundancies of the Romanian language, whose deletion would make forms of the same term to coincide.

By manually analyzing a medium-sized sample of Romanian nouns, together with their various forms, it was observed that the consonants making up the term were not affected. It is worth mentioning that almost the entire database consists of nouns. Most of the times, the terms had exactly the same string of consonants, while, in other cases, the final string had the lemma's string as a prefix. Therefore, the first conclusion to be drawn was that the consonants of a term were a relatively strong invariant.

Example 1:

excepție de neconstituționalitate → *xcpt d ncnsttñlnt*

agenție → *gnț*

In order to further test the function, we considered a testing sample of medium-large sized terms and created a standard annotation to compare our results with. The testing sample consisted of 100 randomly chosen terms from the Romanian terms in the IATE database which we manually lemmatized. These terms were introduced in an Aho-Corasick structure which the corpus was also passed through.

Considering this testing sample and the function which, for a given term, only keeps the consonants for the mask, yielded nearly perfect results in terms of positive matches, missing at most 2% of matches. But, although the parameter p (the fraction of positive matches) was close to unity, the parameter f (the fraction of all matches which are false-positives) had a slightly worse value. Indeed, the consonant string itself created several false positives, at a rate of about 10% of matches. Thus, this function had a matching rate of 88%.

In order to reduce the number of false positives, as described earlier, we wanted to slightly refine the compression algorithm. To decide which additional letters should be kept when creating the mask, we used former observations stating that, as the human mind processes words, the positions of the first and last letters are essential, while the others may be randomized to some extent without affecting our capacity of comprehension. Based on these observations, we decided that we should stress the importance of those letters in our masks as well. Thus, our second function constructed the mask as follows: it kept the first letter of each word, then added the string of consonants.

Example 2:

excepție de neconstituționalitate → excpț d ncnsttñlitt
agenție → agnț

This function yielded the same number of correct matches over the testing sample, keeping the p (the fraction of positive matches) parameter at the value of 98%, but the number of false positives decreased to zero. Thus, adding the first letter to the masks increased the matching rate to 98% over the testing sample. As the miss-matches were mainly due to alterations involving diacritical letters (when changing form, some diacritical letters became normal letters and the other way around, i.e. “suport – suporti”), ignoring diacritics (i.e. processing “ț” as “t”) finally increased the matching rate to 100% over the testing sample. It is worth mentioning that, although we obtained a perfect identification over the testing sample, the actual rate over the entire corpus is expected to be slightly lower.

While calibrating the compression function, spacing and punctuation proved to be crucial factors. We decided to ignore any form of punctuation and keep all spacing in order to avoid mismatches as the ones below:

Example 3:

“ACR” was identified as different from “A.C.R.”
“alga nori” was identified as matching “legea nr.” (the mask of both of them was “lgnr”)

After keeping spacing and deleting punctuation, both of these issues were solved.

Another aspect which helped remove false positives was identifying abbreviations by their uppercase-structure. Thus, if a term consisted of only uppercase letters, we included it as an abbreviation and kept its capitalization, otherwise we converted it to lowercase letters.

This method only implies linear-time transformations of the IATE dictionary (through the compression function) and a single pass through the Aho-Corasick structure, the overall complexity of the proposed algorithm is linear.

When tested against shorter terms, the algorithm behaved considerably worse, considering the fact that the compressed masks started to coincide, thus violating the rule of injectivity over the IATE database. Analysing the definition of a match in the context of shorter terms, it is easy to notice that defining the match according to the Levenshtein distance is impractical. But, as the short entries of the dictionary are mostly single-word entries and are already lemmatized, we can directly apply a perfect match algorithm.

It is obvious that the two algorithms proposed earlier are very similar, but cannot be merged because they use different versions of the corpus (the first algorithm passes the

text through the compression function, while the second doesn't). The second algorithm yielded slightly worse results than the perfect ones from our first algorithm, but, after a short manual analysis of the mismatches, we concluded that better results are almost impossible to obtain with a deterministic algorithm (i.e. mistaking "Au" (the chemical element) for the commonly used verb form "au"). The matching rate of the second algorithm was of approximately 96% (the mentioned mismatches created many false positives and few false negatives; thus, when ignored, the number of omitted terms had a small influence over the matching rate).

Having both of these algorithms, the final proposed method represents a combination of these two (we will call it the *hybrid algorithm*). As each of them behaves almost perfectly on large, respectively short terms, the question which arises is where do we impose the border between them two. After several attempts, we defined two categories of terms:

- i) Short terms: terms having at most 4 consonants,
- ii) Long terms: terms having at least 5 consonants.

The separation was done by consonants because of the structure of the first proposed algorithm, yielding a matching rate of approximately 98%. Thus, the short and the long terms were processed in parallel, using two completely separate Aho-Corasick structures. At the start of the processing, the long terms were inserted in an Aho-Corasick structure corresponding to the first algorithm (the long terms are passed through the compression function, then inserted in the Aho-Corasick structure), and the short terms were inserted in a different Aho-Corasick structure corresponding to the second algorithm (the short terms are directly inserted in the Aho-Corasick structure). Then, the corpus is passed separately through both structures in order to identify the matches (for the first structure, the corpus is passed through the compression function, then through the Aho-Corasick containing long terms, while for the second structure, the corpus is passed directly through the Aho-Corasick containing short terms). Following the processing, the matches resulted through these two separate algorithms are combined in the same document, producing the final annotation.

It is worth mentioning that, when multiprocessor computing is available, the two algorithms can be run in parallel, using half the computation time required for monoprocessor computing.

Besides the compression function, the Aho-Corasick structures were not language-specific and were created during runtime (for the Romanian IATE terms, consisting of about 55,000 terms, the computation time was of approximately 10 seconds). Thus, the algorithm would be available for any language, given a specific compression function which is relevant to that respective language. When dealing with larger databases, the Aho-Corasick structures could also be stored offline, such that they should not be re-built at each processing.

The manual analysis of several documents yielded no false positive matches. The final complexity of the hybrid algorithm is linear – $O(dict + corpus)$, where *dict* is the size of the IATE dictionary and *corpus* is the size of all documents.

5. Results

By running the proposed algorithm over the Romanian legal corpus, we obtained several remarkable results, which we are going to discuss.

Firstly, the processing allowed the annotation of the corpus through the introduction of EuroVoc and IATE labels. Thus, every occurrence of the IATE term inside the corpus is now annotated by its respective position and is accompanied by the corresponding EuroVoc categories (also mentioned in IATE, but added here for further processing purposes). In the analysed testing sample, we were able to detect no false positive matches and a nearly perfect precision for detecting true matches.

The overall matching rate over the used testing samples was of approximately 98%-99%, but, as mentioned earlier in the paper, we expect the real matching rate to have a slightly lower value due to unexpected collisions which may have occurred during the term identification process. Moreover, the matching rate, as defined in the paper, has a rather simple definition and only presents our work's accuracy to some extent.

Secondly, by using the earlier mentioned annotation with EuroVoc categories, we were able to create a statistical database resembling the occurrence frequency of all the categories in each legal document. For each file, we determined the number of terms falling in each of the 21 EuroVoc categories. After multiplying each frequency with a predefined weight, the mentioned file was placed in the category corresponding to the maximum number of terms.

The predefined weights were roughly determined over a testing sample of medium size (approx. 100 documents), due to the lack of pre-processed legal data. As most of the EuroVoc categories present in the description of the IATE terms yielded correct classifications, we only needed to slightly modify some of them (such as Geography) to obtain a better classification over the testing sample.

In comparison with other formerly used approximate-string-matching algorithms, as mentioned in the section 2, the proposed algorithm has a better computational time than most of the deterministic methods (the time complexity is purely linear) and fulfills the goal of achieving a linear-time complexity while still maintaining an almost-perfect matching rate. We have presented an algorithm which processes the entire term dictionary once, unlike (Amir *et al.*, 2004; Cole and Hariharan, 2002) – thus improving the time complexity by a factor of its size – and which does not require the preprocessing of any automaton, whatever its nature, unlike (Schulz and Mihov, 2001) – as the dimension of the terms increases, the number of allowed mismatches proportionally increases, creating size issues when building the automata. Moreover, our algorithm does not depend on a universal constant concerning the allowed number of matches, but instead on a Compression Function, which limits the matches structurally.

The computation time for simply identifying the matches with the hybrid algorithm was of approximately 4,250 seconds, which yields a rate of almost 35 documents per second. Thus, as the XML-formatted corpus had a size of approximately 31.2 Gb, the processing rate was of 7.5 Mb of text per second. Because of using the Aho-Corasick structure, the

memory usage was also linear, which made the computation possible on almost any machine.

Below, there is a fragment of an annotated document. The structure corresponds with the CoNLL-U format, and the last column is formatted in the format: “match_number: IATE_id-EuroVoc_categories”.

```
# sent_id = 2
# text = MONITORUL OFICIAL nr. 101 din 8 februarie 2012
1  MONITORUL monitor  Ncmsry  _  _  _  _  _  _  5:1394636-3206;6:3522817-3221,7206,7231
2  OFICIAL oficial Afpms-n _  _  _  _  _  _  6-3221,7206,7231
3  nr. nr. Yn _  _  _  _  _  _
4  101 101 Mc-p-d _  _  _  _  _  _
5  din din Spsa _  _  _  _  _  _
6  8 8 Mc-p-d _  _  _  _  _  _
7  februarie februarie Ncms-n _  _  _  _  _  _
8  2012 2012 Mc-p-d _  _  _  _  _  _ spaceafter=no _
```

Figure 1: Example of an annotated sentence⁶

The number of matches was significantly increased by working with the lemmatised corpus instead of the unlemmatised one. Working with the unlemmatised corpus yielded a matching rate of about 50%, while the lemmatised form produced almost perfect results.

Through the usage of the hybrid algorithm, we have identified a total of 51,517,877 matches (IATE terms), out of which 29,162,667 were short terms (at most 4 consonants, identified through perfect matching) and 22,355,210 were long terms (at least 5 consonants, identified through matching with the compressed function). This corresponds to an average of 347 matches per document, which is an acceptable value for a classification to be made. The verification of these results would require preprocessed data in the used language (which, in our case, is currently unavailable), because the annotation accuracy widely depends on the Compression Function (how well it fits the actual language which it is applied on). Thus, we cannot yet provide a precise value of precision, but only approximate it based on our testing samples, namely approx. 98% precision over the testing sample. We could not compute the recall since there is no available reference annotation. Yet, we hypothesize that it would be similarly high.

Below, there is the classification of the documents into the 21 EuroVoc categories, based on the maximal number of matches in each category. We intend to further improve these results in several ways described under the section 7.

⁶Actually, the final representation of a text in the MARCELL corpus is much richer in information: it contains the UD tags, the MSD expansion, the dependency parsing, the noun phrases mark-up and the name entities annotation. We did not show them here for the sake of readability.

Table 2: Number of documents in each EuroVoc category

EuroVoc domain	No. of documents	EuroVoc domain	No. of documents
POLITICS	17,290	EMPLOYMENT	78
INTERNATIONAL RELATIONS	1,819	TRANSPORT	4,324
EUROPEAN UNION	27,988	ENVIRONMENT	274
LAW	46,037	AGRICULTURE, FORESTRY AND FISHERIES	1,937
ECONOMICS	75	AGRI-FOODSTUFFS	95
TRADE	798	PRODUCTION, TECHNOLOGY AND RESEARCH	56
FINANCE	7,271	ENERGY	745
SOCIAL QUESTIONS	4,315	INDUSTRY	4,522
EDUCATION AND COMMUNICATION	11,858	GEOGRAPHY	81
SCIENCE	187	INTERNATIONAL	4
BUSINESS AND COMPETITION	18,518		

6. Conclusions

We have presented the preliminary results of the process of collecting and processing the corpus of Romanian law documents together with the methodology and results of the terms identification step. The term identification step is based on the encoded list of IATE and EUROVOC terms, which brings our approach closer to a gazetteer-based processing. This is why the estimated precision is so high; the reason for not reaching a 100% precision is that sometimes the term compressing algorithm, currently, generates the same code for different terms. Yet, we plan to also experiment with ML algorithms, trained with the current annotated corpus and compare the results with what we obtained through the present more traditional approach.

The classification of documents using standard EUROVOC thesaurus and the cross-lingual linkage that EUROVOC ensures, makes this resource useful for training and testing machine translation systems, especially for under-resourced languages. Moreover, the knowledge that can be extracted from this resource is necessary for a deeper understanding of how law terminology works and how it can be made more consistent.

7. *Further work*

Following our classification of the Romanian legal corpus, which, as stated in the Results section, was done by choosing the (weighted) most frequent EuroVoc category in each document, we are planning to complete several more classifications of the corpus and realise a comparison between them. We are currently working on a classification based on Semantic Vectors (term embeddings) and another one using the JRC Eurovoc Indexer JEX⁷, which is pre-trained for all EU official languages.

Furthermore, we are working towards releasing another version of the annotated corpus, which completely ignores matches entirely included inside other matches.

Acknowledgement

This research was supported by the EC grant no. INEA/CEF/ICT/A2017/1565710 for the Action no. 2017-EU-IA-0136 entitled “Multilingual Resources for CEF.AT in the legal domain” (MARCELL).

References

- Aho, A. and Corasick, M. (1975). Efficient string matching: An aid to bibliographic search. *Commun. ACM*. 18:6, 333-340.
- Amir, A., Lewenstein, M. and Porat, E. (2004). Faster algorithms for string matching with k mismatches. *Journal of Algorithms*. 50:2, 257-275.
- Cole, R. and Hariharan, R. (2002). Approximate String Matching: A Simpler Faster Algorithm. *SIAM J. Comput.*, 31:6, 1761–1782.
- Knuth, D. E., Morris, J. H. and Pratt, V. R. (1977). Fast Pattern Matching in Strings. *SIAM J. Comput.*, 6:2, 323–350.
- Pala, K., Rychlý P. and Šmerk, P. (2010). Automatic Identification of Legal Terms in Czech Law Texts. *Semantic Processing of Legal Texts. Lecture Notes in Computer Science*, 6036.
- Schulz, K. and Mihov, S. (2001). Fast String Correction with Levenshtein-Automata. *International Journal on Document Analysis and Recognition*. 5:1.
- Touzet, H. (2016). On the Levenshtein Automaton and the Size of the Neighbourhood of a Word. *Language and Automata Theory and Applications. LATA 2016. Lecture Notes in Computer Science*. 9618, 207-218.

⁷ <https://ec.europa.eu/jrc/en/language-technologies/jrc-eurovoc-indexer>

LOOKING ALONG ZIPF'S LAW FOR THE DISTRIBUTION OF WORDS BEGINNING AND ENDING SENTENCES IN LITERARY PRINTED ROMANIAN CORPORA

BOGDAN HANU¹, ADRIANA VLAD^{1,2}, ALEXANDRU DINU¹, ADRIAN MITREA¹

¹*Faculty of Electronics, Telecommunications and Information Technology,
Politehnica University of Bucharest,*

²*Research Institute for Artificial Intelligence "Mihai Drăgănescu", Romanian Academy*

*hanu.bogdan@gmail.com, avlad@racai.ro
{adrianmitrea, alexandrudinu89}@yahoo.com*

Abstract

A detailed experimental study was conducted following the statistical structure of the words beginning and ending a sentence or complex sentence in printed Romanian (literary corpora with orthography and punctuation marks, with 47 signs extended alphabet). The analysis focused on the rank-frequency distribution of the beginning words of the (complex) sentence, by reference to Zipf's law. The study was conducted on an overall corpus of over 6 million words in length (consisting of 49 books written by 9 authors), with details on 5 author sub-corpora. Since there are many distinct words (the number of ranks corresponding to Zipf's law in the whole corpus is over 260000, and the number of distinct words at the beginning of the sentence is over 28000), the analysis required and benefited from the existence of 3 areas of priority interest established in our previous studies. The priority areas of interest offer the researcher a much smaller number of words to carry out his analysis, maintaining a high coefficient of word use in the language. The distinct words used to begin a sentence can appear in the text in various situations (not only as a beginning of sentence), an aspect that complicates the study and further supports the importance of the existence of priority areas delimited on the axis of Zipf's law ranks. A first plan of the analysis led to mapping the overall situation regarding the statistical structure of words (along Zipf's law) on the overall corpus and author corpora. Another plan of the analysis focused directly on the parallel tracking of the rank-frequency distribution of the beginning words of the sentence and of the distribution of the same words along Zipf's law in the corpora.

Key words — natural language statistical description, orthography and punctuation in language modeling, Romanian literary corpora, priority areas for the NL user under Zipf's Law

1. Introduction

The paper focused on two types of problems: a) the impact of orthography and punctuation in the language model and b) the importance of the beginning and ending words of the sentence in the statistical description of the language (seen as a chain of words). The association between these two issues had as starting point the choice of the

delimiting signs that mark the end / beginning of the sentence, namely, in the present analysis four punctuation marks were considered as a delimiter: a) FULL STOP; b) QUESTION MARK; c) EXCLAMATION MARK; d) ELLIPSIS.

Word tracking was done along the rank-frequency distribution of words (Zipf's law) across various text collections, (Kanter and Kessler, 1995; Dinu, 1996; Vlad *et al.*, 2003).

The study was conducted on an overall corpus of 6377720 words in length, consisting of 49 books written by 9 authors: Isaac Asimov – 9 books; Constantin Chiriță – 5 books; Alexandre Dumas – 12 books; Colin Falconer – 1 book; Frank Herbert – 8 books; Niven Larry – 1 book; Orson Scott Card – 3 books; Michel Zevaco – 7 books; J. R. Tolkien – 3 books. The corpus name is #NLCO (New Literary Corpus with Orthography and Punctuation), (Vlad *et al.*, 2013). The writing with extended alphabet to 47 signs was considered (15 being orthography and punctuation marks).

Since there are many distinct words (the number of ranks corresponding to Zipf's law in the whole corpus is over 260000, and the number of distinct words at the beginning of the sentence is over 28000 (see Table 1)), the analysis required and benefited from the existence of 3 areas of priority interest established in our previous studies, denoted as Area 1, Area 2 and Area 3, (Hanu *et al.*, 2018). The present paper contains a detailed analysis on 5 author sub-corpora, namely #ASIMOV, #CHIRITA, #DUMAS, #HERBERT and #TOLKIEN, which allowed a comparative point of view (the 5 author corpora were used in previous studies for other statistical comparisons, (Ciucă *et al.*, 2012)). The priority areas of interest (rank ranges along Zipf's law) offer the researcher interested in natural language processing a much smaller number of words to carry out his analysis, maintaining a high coefficient of word use in the natural language.

It should be noted that writing with orthography and punctuation marks has not been sufficiently studied yet, at least not enough to be included in the mathematical modeling of natural language, (Say and Akman, 1996; Vlad *et al.*, 2007; Rodriguez, 2011; Vlad *et al.*, 2013).

We resume some aspects regarding the writing considered with the alphabet extended to 47 signs. The 31 characters are the upper case letters composing the Romanian alphabet, A Ă Â B C D E F G H I Î J K L M N O P Q R S Ș T Ț U V W X Y Z, no distinction being made between the upper case and the lower case letters. The space character (blank) and 15 orthography and punctuation marks (Vlad *et al.*, 2007; Mitrea *et al.*, 2012; Vlad *et al.*, 2010; Vlad *et al.*, 2013) are added to these. These 15 marks are: hyphen, full stop, comma, colon, semicolon, question mark, exclamation mark, quotation dash, em dash, abbreviation point, ellipsis, quotation marks, parentheses and apostrophe. The parentheses contribute with two characters to the alphabet corpus (the open parenthesis and the close parenthesis are distinctly marked). For the quotation marks, there is no open / close difference, so they contribute by a single character to the extended alphabet.

1.1. Some details related to the analysis carried out:

- The word is the sequence of characters between two spaces, so the punctuation marks are attached to the word. Thus, in writing with 47 characters, punctuation is part of the word, for example the following items:

SPUS SPUS, SPUS. SPUS? SPUS: SPUSE SPUSE: SPUSESE

are eight distinct words under Zipf's law representation.

- The symbol representing the dialogue, i.e. quotation dash, appears as a distinct word and the symbol representing the em dash also appears as a distinct word. The em dash contributes by less than 0.5% to corpus coverage, and the quotation dash contributes by less than 2.5% (the ratio of the number of the respective word occurrences to the corpus length expressed in words).

- In our study, we considered as the beginning word of the sentence the word that follows one of the four characters in the extended alphabet: FULL STOP, QUESTION MARK, EXCLAMATION MARK and SUSPENSION. The ending words of the sentence are the words that contain one of the four characters on the last position.

Chapter 2 contains the first part of the study, respectively a view of the overall situation regarding the statistical structure of words along the graph of Zipf's rank-frequency law, on #NLCO and on the author corpora. There were mainly aspects regarding the impact of orthography and punctuation in the language and the importance of the beginning / ending words of a sentence. This overview also contains a brief presentation of the areas (rank regions) of priority interest on which this analysis is carried out. This chapter highlights a series of quite stable numerical values (with small variations) in the comparison between corpora.

Chapter 3 represents another plan of the analysis that focused directly on the parallel tracking of the rank-frequency distribution of the beginning words of the sentence and of the distribution of the same words along Zipf's law in the corpora. A new type of rank-frequency distribution is highlighted, conditioned by the beginning of the sentence.

Chapter 4 concludes the paper.

2. About words beginning and ending sentences

An overview of the statistical structure of words is presented along the graph of Zipf's rank-frequency law, on literary corpora. This analysis was initially performed on the overall corpus (#NLCO) and continued on the five author corpora (#ASIMOV, #CHIRITA, #DUMAS, #HERBERT, #TOLKIEN). The areas that we consider to be of priority interest for the natural language processing (NLP) user are delimited on the rank axis of the Zipf's law. The analysis based on these areas reveals the impact of orthography and punctuation in the language and the importance of the beginning and ending words of a sentence. The 3 areas of priority interest for the researcher in the NLP field (i.e. Area 1, Area 2 and Area 3, (Hanu *et al.*, 2018)), were established along the rank axis of the Zipf's law, as next indicated. We mention that the areas of priority interest contain a large part of the rank interval in Zipf's law where the experimental graph best fits with the theory.

For the #NLCO corpus, the areas were initially delimited by taking into account the number of word occurrences. These boundaries associated with Areas 1, 2 and 3 are also highlighted by the relative frequencies corresponding to the boundary ranks (which

is useful to delimit areas on smaller corpora). We detail the three priority areas of interest for #NLCO.

Area 1 contains the ranks from 1 to 2762; the delimiting rank is experimentally obtained as the last word with 201 occurrences in the corpus; equivalently, the relative frequency corresponding to the border rank, 2762, is 0.00315% (number of occurrences divided by the corpus length in words, $201/6377720$).

Area 2 contains the ranks from 1 to 5334; the delimiting rank is experimentally obtained as the last word with 101 occurrences in the corpus; equivalently, the relative frequency corresponding to the border rank, 5334, is 0.00158% (number of occurrences divided by the corpus length in words, $101/6377720$).

Area 3 contains the ranks from 1 to 9993; the delimiting rank is experimentally obtained as the last word with 51 occurrences in the corpus; equivalently, the relative frequency corresponding to the border rank, 9993, is 0.00079% (number of occurrences divided by the corpus length in words, $51/6377720$).

Table 1 presents details of the analysis carried out on the overall corpus and on the 5 author sub-corpora. Columns 1 to 8 present general information regarding the analyzed corpora. The rest of the columns from 9.x to 17.x, where $x = 1, 2$ or 3 , contain the analysis on the areas of priority interest. We mention that some of the experimental results in Table 1 are results previously obtained, (Hanu *et al.*, 2018).

The first column contains the name of the analyzed corpus, specifying the number of books for each corpus. To read Table 1, we follow line 1 representing information on the #NLCO overall corpus.

The #NLCO length is 6377720 words (column 2), out of which 260103 are distinct words (column 3) corresponding to the number of ranks in the Zipf's law representation (the word rank-frequency distribution in the corpus). Out of the 260103 distinct words, 170601 are words that contain orthography and punctuation marks (column 4). The words that contain orthography and punctuation marks cover the corpus in a proportion of about 23% (column 5), i.e. the occurrence number of the 170601 words in the corpus divided by the corpus length.

Column 6 highlights the large proportion of distinct words (about 66%) that contain orthography and punctuation marks in total distinct words (column 4 / column 3).

For these first 6 columns see also (Hanu *et al.*, 2018). Columns 7 and 8 contain information on the word found at the beginning of a sentence. In #NLCO there are (as occurrence number) more than 525000 words standing for the beginning of a sentence, which corresponds to the total number of sentences in the corpus. These words at the beginning of sentences cover the corpus in a proportion of about 8.3% (column 7). The distinct words of the beginning a sentence represent about 11% of total distinct words in #NLCO (column 8).

We next follow the analysis on the priority interest areas for #NLCO. We exemplify the information for Area 1; this contains the distinct words with over 200 occurrences in the corpus (column 9.1).

LOOKING ALONG ZIPF'S LAW FOR THE DISTRIBUTION OF WORDS BEGINNING AND
ENDING SENTENCES IN LITERARY PRINTED ROMANIAN CORPORA

The number of ranks experimentally obtained for Area 1 is 2762 (column 10.1). The 2762 distinct words cover the overall corpus length in a proportion of about 71.8% (column 11.1). There are 743 ranks in Area 1 that contain words with orthography and punctuation marks (column 12.1). The coverage of the #NLCO corpus by the 743 words is about 10.4% (column 13.1). Out of the 2762 distinct words, 2229 words can be used at the beginning of a sentence (column 14.1), which represents a proportion of 81% of the distinct words in Area 1 (column 15.1). These distinct words that can form the beginning of sentences cover the corpus in a proportion of 68% (column 16.1). Strictly used at the beginning of sentences, the 2229 distinct words cover the corpus in a proportion of 7.1% (column 17.1); we note that this is a large proportion, considering that all the words found at the beginning of sentences cover the corpus in a proportion of about 8.3% (column 7).

Similarly, we can follow the analysis in Table 1 regarding Areas 2 and 3 of priority interest. We mention that Areas 2 and 3 do not bring a significant increase in the use of the distinct words in the language, compared to Area 1. However, the detailed information regarding the distinct words is much richer (including words that contain orthography and punctuation and those that form the beginning of a sentence). For example, in #NLCO, the number of distinct words in Area 2 is about twice the number of distinct words in Area 1 ($5334/2762 = 1.94$ columns 10.2 and 10.1), but their use in the natural language has slightly increased, from 71.8% to 77.5% (columns 11.2 and 11.1). Also, the number of distinct words with orthography and punctuation in Area 2 is about twice the number in Area 1 ($1579/743 = 2.13$ columns 12.2 and 12.1), but with a slight increase of their use in the language, from 10.4% to 12.2% (columns 13.2 and 13.1).

The number of distinct words that can form the beginning of sentences has significantly increased in Area 2 compared to Area 1 ($3891/2229 = 1.75$ columns 14.2 and 14.1), but there is a small increase in their use in the language in the strict situation at the beginning of a sentence, from 7.1% to 7.4% (columns 17.2 and 17.1).

As a first conclusion, Area 1 can be an important starting point for research in the NLP field.

A similar analysis was conducted for the 5 authors, with a general view on the corpus on columns 1-8 and continuing with information on the priority areas (columns 9.x - 17.x where $x = 1, 2$ or 3). It can be seen to what extent we can talk about a stability (small variations) of the quantitative results and how large the difference is, depending on the author and on the corpus length.

The coverage of the analyzed corpus by words containing orthography and punctuation represents a proportion from 21% to 25% for author corpora. By comparison, in #NLCO we have 23% (column 5).

The distinct words that contain orthography and punctuation represent a large proportion from the total distinct words in the corpus, namely from 54% to 60% for author corpora. By comparison, in #NLCO we have 66% (column 6).

The coverage of the analyzed corpus by the beginning words of the sentences (the same for the ending words of the sentences) represents a proportion ranging from 7% to

10.4% for the author corpora, which leads to an average sentence length of about 14-15 words and 9-10 words, respectively. By comparison, in #NLCO we have about 8.3% coverage (column 7), which leads to an average length of about 11-12 words.

Out of total distinct words from the analyzed corpora, those that can be used to form the beginning of a sentence represent a proportion from 7% to 12%. By comparison, for #NLCO we have 11% (column 8).

We shall next describe the information regarding the author corpora for the 3 areas of priority interest, respectively columns 9.x to 17.x, where $x = 1, 2$ or 3 (Table 1).

We remind that the landmark in the delimitation of Areas 1, 2 and 3 for all author corpora was the relative frequency corresponding to the border rank of the respective Area in #NLCO, namely: 0.003% for Area 1; 0.0016% for Area 2; 0.0008% for Area 3.

For each author corpus, the number of occurrences of the word on the border rank of the analyzed priority area was experimentally determined (columns 9.x where $x = 1, 2, 3$).

It can be noticed that, for all the analyzed corpora, the total number of distinct words in each corpus has significantly decreased (columns 10.x where $x = 1, 2, 3$ compared to column 3). For example, the ranks in Area 1 range from 2857 to 3322; by comparison, for #NLCO, Area 1 comprises 2762 ranks (column 10.1).

The coverage by the distinct words existing in Area 1 is between 72% and 79.2% for the author corpora (column 11.1). By comparison, for #NLCO, this coverage is 71.8%.

The number of distinct words with orthography and punctuation in Area 1 ranges from 790 to 998; in #NLCO, there are 743 such words. The percentage of corpus coverage by these words with orthography and punctuation is from 9.8% to 13.8%. By comparison, in #NLCO, the coverage is 10.4% (column 13.1).

The distinct words that can form the beginning of a sentence relative to the total number of distinct words existing in Area 1 represent a proportion ranging from 46% to 65% for author corpora. By comparison, in #NLCO, this is 81% (column 15.1).

The distinct words from Area 1 that can form the beginning of a sentence cover the corpus in a proportion from 64% to 67% for author corpora. For #NLCO, this proportion is 68% (column 16.1).

The corpus coverage by the distinct words that are strictly used at the beginning of a sentence varies from 6.4% to 9.2%. By comparison, in #NLCO, this coverage is 7.1% (column 17.1).

2.1. Regarding the words ending sentences versus words beginning sentences

The total number of occurrences of the words ending sentences in the corpus is equal to the total number of occurrences of the words beginning sentences (it results the same corpus coverage, see column 7).

The set of distinct words ending sentences differs from the set of distinct words beginning sentences, the overlapping of the two sets meaning one-word sentences.

The number of distinct words beginning a sentence is smaller than the number of distinct words ending a sentence. For example, in #NLCO, the distinct words beginning a sentence represent 11% of the total distinct words in the corpus (column 8), while the

LOOKING ALONG ZIPF'S LAW FOR THE DISTRIBUTION OF WORDS BEGINNING AND
ENDING SENTENCES IN LITERARY PRINTED ROMANIAN CORPORA

distinct words ending a sentence (83642 words) represent 32% of the total distinct words in the corpus (Hanu *et al.*, 2018). Regarding the author corpora, the proportion of distinct words ending a sentence (out of total distinct words in the analyzed corpus) varies from 22% to 33%.

Table 1: 1. Corpus Name; 2. Corpus length in words; 3. Number of distinct words in corpus; 4. Number of distinct words with orthography and punctuation; 5. Corpus coverage by words with orthography and punctuation (%); 6. The share of distinct words containing orthography and punctuation in total distinct words (column 4/column 3, %); 7. Corpus coverage by words beginning a sentence (%); 8. The share of distinct words beginning a sentence in total distinct words in corpus (%); 9.x. Limit of x priority interest area expressed in number of occurrences; 10.x. Limit rank of the x area; 11.x. Corpus coverage by the words from the x area; 12.x. Number of distinct words with orthography and punctuation marks from the x area; 13.x. Coverage of the corpus by words with orthography and punctuation, from the x area; 14.x. The number of words from x area that can be used as beginning a sentence; 15.x. The share of distinct words in x area that can begin a sentence from the total ranks in x area; 16.x. Corpus coverage by words from x area that can form beginning of sentence; 17.x. Corpus coverage by words from x area when strictly used as beginning a sentence. x = 1, 2 or 3.

								Area 3								
1	2	3	4	5	6	7	8	9.3	10.3	11.3	12.3	13.3	14.3	15.3	16.3	17.3
#NLCO	6377720	260103	170601	23	66	8.3	11	50	9993	82.7	3310	14.1	6276	63	74	7.6
#ASIMOV	909490	74295	41390	23	56	8.49	9	7	10741	87.7	3777	15.8	3454	32	68	8.1
#CHIRITA	619071	64124	38282	23	60	10.36	10	5	11355	87	4515	15.6	3508	31	68	9.8
#DUMAS	2022193	114623	68715	25	60	6.96	7	16	9822	86.5	3562	17.7	3780	39	70	6.7
#HERBERT	1149056	99560	56084	22	56	9.66	12	9	12047	84.7	3923	14.1	5017	42	68	8.9
#TOLKIEN	486134	45287	24672	21	54	7.07	9	4	11071	90.2	4194	15.8	2641	24	66	6.8

(continued)

Area 2									Area 1								
9.2	10.2	11.2	12.2	13.2	14.2	15.2	16.2	17.2	9.1	10.1	11.1	12.1	13.1	14.1	15.1	16.1	17.1
100	5334	77.5	1579	12.2	3891	73	72	7.4	200	2762	71.8	743	10.4	2229	81	68	7.1
14	5677	82.5	1789	13.8	2480	44	67	7.9	29	2857	76.5	817	11.8	1626	57	65	7.6
10	5530	80.9	1902	12.9	2447	44	67	9.6	19	3001	75.5	937	10.9	1682	56	66	9.2
31	5531	81.9	1851	15.9	2754	50	68	6.4	63	2937	76.4	904	13.8	1805	61	67	6.4
18	6350	78.8	1799	11.9	3388	53	67	8.6	36	3134	72	790	9.8	2039	65	64	8.2
8	5976	84.9	1964	13.5	2048	34	66	6.6	15	3322	79.2	998	11.4	1520	46	64	6.4

As an observation, out of total 83642 distinct words ending a sentence (see Table 2), more than half are distinct words that contain the character FULL STOP on the last position.

The distinct words beginning a sentence that have a high use in the language are found on ranks included in the priority interest areas (Areas 1, 2, 3). The distinct words ending a sentence belong to the set of words with orthography and punctuation and are spread throughout the entire graph of Zipf's law, the framing in the priority interest areas is not significant from the point of view of their use in the natural language. For this reason, Table 1 selected only the information regarding the words beginning a sentence, information largely found in the priority interest areas.

3. The rank-frequency distribution of the words beginning a sentence in various corpora

The analysis focused directly on the parallel tracking of the rank-frequency distribution of the beginning words of the sentence and of the distribution of the same words along Zipf's law in the corpora. Table 2 presents results obtained on the #NLCO corpus and on the #ASIMOV and #CHIRITA author corpora.

NOTE: Zipf's law concerns the rank-frequency distribution for all the distinct words in the corpus, while the topic of this chapter is a rank-frequency distribution conditioned by the beginning of a sentence. Consequently, the total number of ranks in the new distribution law (conditioned by the beginning of a sentence) will correspond to the total number of distinct words beginning a sentence in the analysed corpus.

For the author corpora, we kept as reference the words from #NLCO ordered in the decreasing hierarchy of frequencies (according to Zipf's law in #NLCO), aiming to rearrange the ranks according to Zipf's law for the author corpus, as well as to change the rank-frequency distribution when conditioned by the beginning situation of sentence.

Table 2 presents only information regarding the words in Area 1 of priority interest in #NLCO. The columns 1, 2.1 and 3.1 of Table 2 present information related to the Zipf's law on #NLCO, respectively the word, rank and word frequency (in the decreasing hierarchy of the word relative frequencies). The columns 4.1 and 5.1 give information about the new rank-frequency distribution of the words, conditioned by the beginning of a sentence. For example, the word DE (column 1), the most frequent word in #NLCO (rank 1, see column 2.1), has the relative occurrence frequency of about 3.46% (column 3.1), obtained as a ratio of the number of occurrences of the word DE in #NLCO to the corpus length in words (220679/6377720). Under the new rank-frequency word distribution conditioned by the beginning a sentence, the word DE is on rank 7 (column 4.1) and has the relative frequency of about 1.23% (column 5.1), obtained as a ratio of the number of occurrences of the word DE at the beginning of sentences to the number of sentences in the corpus (6527/528976).

NOTE: Let us consider k a line in Table 2, where k (column 2.1) is the word rank according to Zipf's law. The notation $f(k)$ represents the word relative frequency in #NLCO and is obtained by dividing the number of occurrences of the word (on rank k , column 1) by the corpus length ($L_c = 6377720$). The notation $f(k\text{-begin})$ is the value on column 5.1 and represents the relative frequency of the word on line k , strictly when it is used at the beginning of sentences; $f(k\text{-begin})$ is obtained by dividing the number of word occurrences as word beginning a sentence by the number of sentences in the corpus, the denominator being the value 528976 (Hanu *et al.*, 2018). The $k\text{-begin}$ notation (column 4) means the rank of the word in the new hierarchy as word beginning a sentence. The information given in Table 2 for #ASIMOV and #CHIRITA can be read in the same manner.

3.1 Some considerations on Table 2

The first 20 words belong to the *common word set*, meaning words found in identical form in all 49 books of the #NLCO corpus. We mention that *the set of common words* has been the subject of previous studies, (Mitrea *et al.*, 2014), contributing to defining

LOOKING ALONG ZIPF'S LAW FOR THE DISTRIBUTION OF WORDS BEGINNING AND
ENDING SENTENCES IN LITERARY PRINTED ROMANIAN CORPORA

the representativeness of a language resource and of a corpus for the literary domain of the language and also revealing some statistical mathematical aspects, (stationarity and ergodicity of the language as a chain of words, (Vlad *et al.*, 2003; Shannon, 1951)). These *common words* by default will be found in all the analyzed author corpora.

NOTE: There are 905 *common words* in all 49 books in the #NLCO corpus. This set of words is in Area 1 of priority interest and covers about 60% of the overall corpus #NLCO, (Hanu *et al.*, 2018).

There are words frequently found in the corpus, but not used as words beginning sentences; as for example FI which is on rank 19 in the descending hierarchy of word occurrences in #NLCO (with a relative frequency of about 0.54%). Otherwise, all the other 19 illustrated words can be found at the beginning a sentence.

Following Table 2 for #ASIMOV, the 20 *common words* are found on ranks close to those from #NLCO, namely in the first 25 ranks of the #ASIMOV corpus. Similarly, the 20 *common words* are found up to rank 21 in #CHIRITA.

Table 2: 1. Word; 2.x The k rank corresponding to Zipf's law; 3.x Word frequency in the corpus, f(k) (%); 4.x The word rank (k-begin) in the hierarchy conditioned by the beginning a sentence; 5.x Word frequency, f(k-begin), in the rank-frequency distribution conditioned by the beginning a sentence (%).
x= 1 for #NLCO; x= 2 for #ASIMOV; x= 3 for #CHIRITA

	#NLCO				#ASIMOV				#CHIRITA			
1	2.1	3.1	4.1	5.1	2.2	3.2	4.2	5.2	2.3	3.3	4.3	5.3
DE	1	3.46	7	1.23	1	3.29	8	1.38	1	3.25	6	1.6
ȘI	2	2.47	2	2.51	3	2.52	3	2.14	2	2.89	2	4.13
{	3	2.28	1	24.71	2	2.59	1	24.93	6	1.80	1	14.6
ÎN	4	2.07	5	1.85	5	1.90	4	1.85	3	2.18	8	1.39
SĂ	5	1.97	21	0.5	4	2.05	29	0.44	4	1.89	16	0.8
SE	6	1.58	6	1.69	9	1.30	7	1.44	5	1.84	4	2.74
PE	7	1.44	16	0.58	10	1.29	16	0.63	11	1.27	31	0.56
O	8	1.36	9	0.97	8	1.41	14	0.75	10	1.40	10	1
NU	9	1.35	3	2.14	6	1.76	2	3.04	7	1.62	5	2.56
CU	10	1.32	15	0.59	11	1.24	15	0.66	8	1.51	29	0.57
LA	11	1.19	13	0.68	15	0.92	22	0.51	9	1.43	25	0.63
CĂ	12	1.10	147	0.07	7	1.42	159	0.07	13	1.20	72	0.2
MAI	13	0.93	26	0.39	14	0.92	33	0.42	12	1.25	15	0.84
UN	14	0.93	17	0.55	13	0.95	25	0.46	14	0.97	34	0.49
CARE	15	0.91	159	0.06	16	0.85	152	0.08	15	0.76	98	0.13
A	16	0.79	30	0.35	12	1.03	20	0.51	17	0.66	41	0.38
DIN	17	0.76	27	0.37	17	0.74	42	0.34	16	0.71	36	0.45
CE	18	0.72	14	0.61	18	0.73	17	0.58	18	0.64	19	0.72
FI	19	0.54			19	0.71			20	0.50		
CA	20	0.51	46	0.24	24	0.46	57	0.24	21	0.49	39	0.43
DINTRE	101	0.09	575	0.015	92	0.11	528	0.0156	135	0.073	1173	0.0062
VORBĂ	801	0.01			1458	0.0065			862	0.011		
VĂZÂND	901	0.01	300	0.031	5617	0.0015			1778	0.0052	849	0.0094
VEDERE	1001	0.0088			446	0.022			5476	0.0016		
NOȚII.	2501	0.0035			5249	0.0016			2352	0.0039		

The word DE is the most frequent word in #NLCO, but also in #ASIMOV and #CHIRITA. As word beginning a sentence, the word DE is on rank 7 in #NLCO, 8 in #ASIMOV and 6 in #CHIRITA.

There are situations when the words on a higher rank on the graph of the Zipf's law in the analysed corpus appear on a lower rank in the new hierarchy conditioned by the beginning of the sentences. For example, according to Zipf's law in the corpus, the word DAR is on rank 21 in #NLCO, on rank 20 in #ASIMOV and on rank 19 in #CHIRITA. When used as a word beginning a sentence, the word DAR is found on rank 4 in #NLCO, on rank 5 in #ASIMOV and on rank 3 in #CHIRITA.

There are situations when the words on a lower rank on the graph of the Zipf's law in the analysed corpus appear on a higher rank in the new hierarchy conditioned by the beginning of the sentences. For example, according to Zipf's law in the corpus, the word CĂ is on rank 12 in #NLCO, on rank 7 in #ASIMOV and on rank 13 in #CHIRITA. When used as a word beginning a sentence, the word CĂ is found on rank 147 in #NLCO, on rank 159 in #ASIMOV and on rank 72 in #CHIRITA.

The words CARE (rank 15 in #NLCO) and LUI (rank 24 in #NLCO) are other examples of frequent words in the corpus (even in the *common words* set), but rarely used at the beginning of a sentence.

Table 2 continues with information on 5 words extracted from the Zipf law in #NLCO. All such examples belong to Area 1 of priority interest for #NLCO, where the maximum rank (delimiter of the area) is 2762.

We mention a main result referring to the entire Area 1 of #NLCO, maintaining the organization in Table 2. There are 2229 ranks (distinct words) in Area 1, meaning over 80% of the total of 2762 ranks, see Table 1, which can be used at the beginning of a sentence. Out of these distinct words that can be used to begin a sentence, 901 words have the following property: the k-begin rank (column 4.1) is strictly lower than the k rank (column 2.1). This experimental result is obtained by making a comparison between columns 2.1 and 4.1 in an extended table of Table 2 type, which contains the whole Area 1. These 901 words have a number of 330775 occurrences strictly at the beginning of the sentences and cover the #NLCO in a proportion of 5.19%, which is significant considering that the corpus coverage by all the distinct words from Area 1 that are strictly used at the beginning of a sentence is about 7%, see Table 1.

4. Conclusions

An overview of the statistical structure of words along the Zipf's Law graph was achieved for the Romanian language, on literary corpora, highlighting the importance and position occupied by the beginning and ending words of sentence under this representation, as well as the impact of orthography and punctuation in the natural language model.

The main results obtained are organized in three rank areas under Zipf's law graph, considered in the paper as areas of priority interest for the NLP researcher. These areas

comprise that part of the rank axis that corresponds to a better match between the theory and the experimental study regarding Zipf's law.

A first area of priority interest for the NLP researcher is Area 1, which facilitates its analysis, both in terms of words in general and in the analysis of the impact of orthography and punctuation and of the possibility of forming the beginning of the sentence. Thus, the 260103 distinct words existing in the overall corpus (which has a length of 6377720 words) are reduced to 2762 words (reduction which means a proportion of about 1% of the total 260103). These distinct words from Area 1 keep a high degree of use in the language (cover about 72% of the corpus length).

Also, the words with orthography and punctuation are reduced from 170601 distinct words to 743 distinct words (reduction which means a proportion of about 0.4% of the total 170601), with a corpus coverage by more than 10% (of the total of about 23% coverage brought by all words with orthography and punctuation in the corpus). Similarly, for the words beginning a sentence, the over 28000 distinct words (found in the corpus at the beginning of the sentences) were reduced to 2229 words found in Area 1. In the strict situation of words used at the beginning of the sentences, these 2229 words cover the corpus by about 7.1% (of the total of about 8.3% brought by all the words beginning a sentence).

Another plan of the analysis focused on the parallel tracking of the rank-frequency distribution of the beginning words of the sentence and of the distribution of the same words along Zipf's law in the corpora. Referring again to Area 1, we mention an important result regarding the words of the beginning a sentence. 901 distinct words of beginning of sentence (existing in Area 1) were found that have an important contribution in the formation of the beginning a sentence and a significant use coefficient in the language.

As far as we know, this type of information brought by the current study represents a novelty, at least for the Romanian language (possibly for other languages as well). Such an analysis of words by framing them into areas of priority interest might be a good starting point in the studies on discourse, in developing speech training applications, in creating linguistic resources for research in life sciences in general.

Acknowledgments

The authors acknowledge the continuous scientific support from Prof. Dan TUFÎȘ, member of the Romanian Academy. The work is part of an ongoing research theme at the Research Institute for Artificial Intelligence, Romanian Academy.

References

- Ciucă, S., Vlad, A., Mitrea, A., (2012). Mathematical Comparison Between Several Single Author Corpora. *U.P.B. Sci. Bull., Series A*, Vol. 74, Iss. 1, 2012, 69-82, ISSN 1223-7027.
- Dinu, M., (1996). *Romanian language personality*, Ed. Cartea Românească, Bucharest.

- Hanu, B., Vlad, A., Mitrea, A., (2018). Aspects Revealing the Orthography and Punctuation Impact in Printed Romanian: A Literary Corpus Based Study. In *The 12th International Conference on Communications (COMM), Bucharest*, 95-100. IEEE, DOI 10.1109/ICComm.2018.8484819.
- Kanter, I., and Kessler, D. A., (1995). Markov processes: linguistics and Zipf's law. *Physical review letters* 74, no. 22: 4.
- Mitrea, A., Vlad, A., Luca, A., (2012). Statistical study on a literary Romanian corpus for the beginning and ending of the Words. In *The 9th International Conference on Communications (COMM), Bucharest*, 81-84. IEEE, DOI 10.1109/ICCOMM.2012.6262546.
- Mitrea, A., Vlad, A., Hodea, O., Dragomir, R., (2014). A study on the common words found in different literary Romanian corpora. In *The 10th International Conference on Communications (COMM), Bucharest*, 123-127. IEEE, DOI 10.1109/ICComm.2014.6866729.
- Rodríguez-Castro, M., (2011). Translationese and punctuation: An empirical study of translated and non-translated international newspaper articles (English and Spanish). *Translation and Interpreting Studies*, 6: 1, 40-61, (John Benjamins Publishing Company).
- Say, B. and Akman, V., (1996). Current approaches to punctuation in computational linguistics. *Computer and the Humanities*, Vol. 30, 457-469.
- Shannon, C., E., (1951). Prediction and entropy of printed English. *Bell System Technical Journal*, vol. 30, 50-64.
- Vlad, A., Mitrea, A., Mitrea, M., (2003). *Limba română scrisă ca sursă de informație*, Editura Paideia, București, (286 pag.), ISBN 973-596-185-7.
- Vlad, A., Mitrea, A., Mitrea, M., (2007). Printed Romanian Modelling: A Corpus Linguistics Based Study with Orthography and Punctuation Marks Included. In *Lecture Notes in Computer Science*, vol. 4705 (ICCSA 2007), Springer Verlag, Berlin Heidelberg, 409-423.
- Vlad, A., Mitrea, A., Mitrea, M., Ciucă, S., (2010). Enriching Printed Romanian Statistical Description: an Approach by Mathematically Comparing Two Independent Literary Corpora. In *Multilinguality and Interoperability in Language Processing with Emphasis on Romanian*, Dan Tufiş, Corina Forăscu (eds.), The Publishing House of the Romanian Academy, 245-271.
- Vlad, A., Mitrea, A., Luca, A., Hodea, O., (2013). Considerations Regarding the Statistical Compatibility of Two Romanian Literary Corpora with Orthography and Punctuations Marks Included. In *Towards Multilingual Europe 2020: A Romanian Perspective*, Dan Tufis, Vasile Rus, Corina Forascu (eds.), The Publishing House of the Romanian Academy, 99-12.

CHAPTER 2

NEURAL APPROACHES

ROMANIAN AUTOMATIC DIACRITICS RESTORATION CHALLENGE

FLORIN IORDACHE, LUCIAN GEORGESCU, DAN ONEAȚĂ, HORIA CUCU

*Speech and Dialogue Research Laboratory,
University Politehnica of Bucharest*

*{florin.nicolae.iordache, dan.oneata}@gmail.com,
lucian.georgescu@speed.pub.ro,
horia.cucu@upb.ro*

Abstract

Recent work from various research groups addresses the task of diacritics restoration for the Romanian language. However, since each method is trained and evaluated on a different dataset and using different metrics, these approaches remain somewhat isolated and incomparable. In this paper, we propose a solution by introducing the diacritics restoration challenge. Our contributions are three-fold. First, we release a unique dataset of talk-shows transcripts, consisting of more than 40M words and 2.5M sentences. Second, we provide an online framework through which teams can submit results, have them automatically evaluated and displayed in a competition leaderboard. Third, we provide baseline results using methods that range from unigram models to character-level recurrent neural networks. Our best model is a two-layer BiLSTM, which achieves a word-level accuracy of 99.46%¹.

Key words — challenge, diacritics restoration, neural networks, Romanian language

1. Introduction

Many languages use scripts in which the usual letters can receive marks, named *diacritics*. Albeit small in size, diacritics are important as they alter the pronunciation and meaning of a word. The lack of diacritics can lead to ambiguous sentences with several meanings: *un roman la Roma* (*a Romanian in Rome* or *a Roman in Rome*). Electronic texts are usually written using English keyboards, which lack these characters, making it inconvenient to type correctly; this process results in large amounts of electronic texts without diacritics. In this paper we tackle the task of *automatic diacritics restoration* (ADR) – inserting the correct marks (comma, cedilla, umlaut, breve, acute, slash, etc.) and in the correct position (above, below, over, etc.) for the usual letters in an orthographic text.

ADR is an active field of study with on-going efforts in a wide variety of languages, including Czech, Polish, Romanian and Hungarian (Novák and Siklósi, 2015; Tufiş and Ceauşu, 2008; Mihalcea and Năstase, 2002), Turkish (Arslan, 2016), Arabic (Khorsheed, 2012; Belinkov and Glass, 2015; Nelken and Shieber, 2005; Schlippe *et*

¹ All information regarding the challenge and the dataset is available at <https://diacritics-challenge.speed.pub.ro>.

al., 2008), Maori (Cocks and Keegan, 2011), Uyghur (Tursun and Cakici, 2017), Urdu (Raza and Hussain, 2010), Vietnamese (Pham *et al.*, 2017; Luu and Yamamoto, 2012; Do *et al.*, 2013), as well as West African languages like Igbo (Ezeani *et al.*, 2016) and Yoruba (Adegbola and Odilinye, 2012; Scannell, 2011; De Pauw *et al.*, 2007).

We study ADR for the Romanian language. Romanian uses three diacritic marks (breve, circumflex, comma below) that can be used for four letters (*a*, *i*, *s*, *t*), as follows: *ă* (*a* breve), *â* (*a* circumflex), *î* (*i* circumflex), *ș* (*s* comma) and *ț* (*t* comma). Although there are only a few characters that can support diacritics, 30-40% of the words in a Romanian running text contain at least one character with diacritics (Tufiș and Chițu, 1999). Moreover, many of these words can take multiple valid forms depending on the choice of diacritics; for example, *pana*, *pană*, *până* are all valid words. A baseline that restores diacritics according to the most common occurrence (a unigram model) gives an error rate of 29.71% – making Romanian the most challenging European language in this regard (Náplava *et al.*, 2018).

Missing diacritics lead to poor intelligibility: the text needs to be read several times to eliminate ambiguities or, worse, the sentence context might not be enough to disambiguate the text meaning. More importantly, texts without diacritics cannot be correctly processed by natural language processing modules such as part-of-speech (POS) taggers, named entity recognizers and cannot be used as input for text-to-speech systems. Hence, an ADR system is of critical importance in all the above situations.

The task of ADR for the Romanian language was approached by several research groups in the past two decades. A knowledge-based method using POS tagging to disambiguate the different diacritical words hypotheses, was introduced in (Tufiș and Chițu, 1999) and later refined in (Tufiș and Ceașu, 2008). A statistical approach using a character n-gram model, a memory-based learning system (TiMBL) and a decision tree classifier (C4.5) is described in (Mihalcea and Năstase, 2002). Word-level n-grams with a history of up to five words were employed in (Cucu *et al.*, 2014; Petrică *et al.*, 2014), while in (Ungurean *et al.*, 2008) the diacritics restoration task is regarded as a sequential filtering process based on unigrams and bigrams of diacritical words and trigrams of diacritical word-suffixes. Finally, starting from 2018 several deep learning architectures (Cristescu, 2018; Rușeți *et al.*, 2018; Nuțu *et al.*, 2019) were proposed for this task.

The various works on ADR for the Romanian language were performed on different training and evaluation datasets and used different evaluation metrics. Table 1 summarizes the sizes of the datasets and the performance metrics used. Moreover, even if the evaluation metrics seem similar, there are still subtle differences in how they are computed. For example, (Cucu *et al.*, 2014) computes the word error rate as the number of incorrectly restored words relative to *the total number of words* in the evaluation dataset. This metric is, indeed, equivalent to the complement of word-level accuracy, used in (Nuțu *et al.*, 2019), but it differs from the word-level accuracy of (Rușeți *et al.*, 2018), as they consider only words that might contain diacritics (that is, words that contain at least one of the characters *a*, *i*, *s*, *t*).

Our work aims to facilitate the comparison across different methods and metrics by providing a common testbed for the task of automatic diacritics restoration. We frame

our proposal as a challenge to the community by providing datasets (both for training and evaluation) and a web interface where participants can log-in to submit their results and have the predictions automatically evaluated.

Table 1: Romanian diacritics restoration initiatives, datasets and evaluation metrics. The accuracies marked with * are computed relative to the words or characters that might contain diacritics, as explained above.

Work	Dataset	Evaluation metric
(Tufiş and Chiţu, 1999)	Fiction (118k tokens), Philosophy (135k tokens), Journalism (92 k tokens)	Accuracy (word-level)
(Mihalcea and Năstase, 2002)	3M words	Char F-score
(Tufiş and Ceaşu, 2008)	Journalism (6.6M words), Juridical (3.5M words)	Accuracy* (word/ char - level)
(Ungurean <i>et al.</i> , 2008)	Online Romanian websites (10M words)	Accuracy (word-level), Char F-score
(Cucu <i>et al.</i> , 2014)	Europarl (5M words), misc (11M words)	WER, ChER, Char F-score
(Petrică <i>et al.</i> , 2014)	Europarl (5M words), misc (11M words), talkshows (44M words), Antena3 (27M words), Realitatea (68M words), Libertatea (72M words)	WER, ChER
(Cristescu, 2018)	unknown	Accuracy (word-level)
(Ruşeti <i>et al.</i> , 2018)	PAR (50M words)	Accuracy* (word/ char - level), Char F-score
(Nuţu <i>et al.</i> , 2019)	Subset of CoRoLa (1M words)	Accuracy (word/ char/ 3-gram -level)

The rest of the paper is organized as follows. Section 2 describes the diacritics restoration task and it presents the evaluation metrics. Section 3 provides a detailed description of the dataset, such as some statistics regarding it. Section 4 introduces the online leaderboard used to track all the approaches and their accuracy results. Our baseline systems are presented in Section 5. The last section is reserved for the conclusions.

2. Task description and performance metrics

The task of automatic diacritics restoration implies identifying the characters which require diacritics and replacing them with their correct diacritical form. This decision is made according to each word and the context in which it is found.

In the Romanian language, there are 4 characters that can have diacritics, each of them being able to have one or more diacritical forms, as follows: (i) $a - \tilde{a}, \hat{a}$, (ii) $i - \hat{i}$, (iii) $s - \text{ş}$, (iv) $t - \text{ţ}$.

There are multiple ways of evaluating an automatic diacritics restoration system. The most natural metric is accuracy. Accuracy can be computed at both word or character level, as the number of correct characters (or words) in the output text, relative to the number of characters (or words) in the groundtruth text. These metrics are presented in equations (2) and (4) and further called CA2 and WA2.

Provided that only a few characters accept diacritics (*a, i, s, t*), a more appropriate manner to compute the character accuracy would be to take into consideration only these characters, as in equation (1). When it comes to words, the accuracy can be computed as the number of correct words accepting diacritics in the output text relative to the number of words which could accept diacritics in the groundtruth text. The term “word accepting diacritics” refers to any word that contains at least one character that might accept diacritics: *a, i, s, t*.

$$CA1 = \frac{\# \text{ correct characters accepting diacritics in output text}}{\# \text{ characters accepting diacritics in groundtruth text}} \quad (1)$$

$$CA2 = \frac{\# \text{ correct characters in output text}}{\# \text{ characters in groundtruth text}} \quad (2)$$

$$WA1 = \frac{\# \text{ correct words accepting diacritics in output text}}{\# \text{ words accepting diacritics in groundtruth text}} \quad (3)$$

$$WA2 = \frac{\# \text{ correct words output text}}{\# \text{ words in groundtruth text}} \quad (4)$$

The primary metric that we decided to use for the proposed challenge is WA1. For a more detailed analysis of the results, the application also offers information about the following metrics: WA2, CA1, CA2, F-score per character and total F-score. F-score represents the harmonic mean of two other well-known metrics: precision and recall. The closer these values are to 1, the higher the performance. On the other hand, a low value for the precision means that the correction of the diacritics has been done, but in a wrong way, while a low value for the recall means that some corrections of the diacritics have been omitted.

3. Dataset description

The Diacritics restoration text corpus was created by the Speech and Dialogue Research Laboratory (Speed). With about 42 million words, this is one of the largest Romanian text corpus containing diacritics freely available for research. The texts represent transcriptions of several Romanian talk-shows and online news. The raw text was preprocessed and normalized using *TextCorpusCleaner*, a tool created by Speed to uniformize diacritics and hyphens, replace URLs, emails and abbreviations with their spoken form, replace numbers (different formats) with text, handle special characters, lowercase text, and remove text in other languages. The corpus contains a training set of 39.7 million words with correct diacritics and an evaluation set of 2 million words without diacritics. Table 2 describes the training corpus in terms of the number of words, letters and sentences, considering the presence of diacritics.

A word without diacritics there may take one or more forms with diacritics, each having a different meaning. Here are some such examples from the dataset:

- suta → șuta, suta, şuță, șută, sută

- pana → pană, pana, până
- fata → fâța, fătă față, fătă, fata, fată, fața
- pasa → pașa, pasă, pașă, păsă

Table 2: Training corpus description. A “word with diacritics” is a word which contains at least one letter with diacritics (*ă, â, î, ș or ț*), while a “word accepting diacritics” is a word which contains at least one letter that might accept diacritics (*a, i, s, t*)

Statistics	Number of occurrences
Words with diacritics	13M
Words accepting diacritics	29M
Total number of words	39M
Letters with diacritics	14M
Letters accepting diacritics	56M
Total number of letters	224M
Number of sentences	2.6M
Number of unique words	0.16M

Figure 1 and Figure 2 present some statistics regarding the training set. The first figure is a heatmap that illustrates the relationship between the words length and their number of diacritical forms: the darker the colour, the greater the number of words (the colour is on logarithmic scale). We noticed that most words have lengths of 6-11 characters and it is obvious that most words have a single diacritical form. Another interesting fact we observed is that shorter words have more diacritical forms. The second diagram is a histogram that shows the distribution of sentences according to their length as the number of words. Most sentences contain between 4-7 words.

4. Evaluation details

Our laboratory introduces a diacritics restoration challenge for the Romanian language². The purpose is to keep track of performance using the current state-of-the-art technology, while systems training and testing takes place on the standard dataset, proposed in Section 0. More information about the dataset can be found on the main page of the challenge, under the *Data* section. The usage of other datasets than those specified is not permitted.

First of all, the users must register using a username and password in the Register section of the menu, as well as providing some personal info: first name, last name, email, affiliation, personal webpage URL. After registration, the user will receive an email to confirm his identity. The Submit section allows users to upload their output file obtained after restoring diacritics on the evaluation subset. The submission file should have the same number of words as the evaluation file, but with the characters a, i, s, t being replaced with their correct version. The leaderboard menu helps users to track the performance of the submitted results. There is a public leaderboard, which reports the best performances of each user, along with information about the total number of

² Available at <http://diacritics-challenge.speed.pub.ro/>

submissions, as well as the date of the last submission. The menu also includes a private leaderboard, showing all the user’s submissions and their performance.

This challenge intends to be of interest to all researchers working on diacritics restoration task on Romanian language.

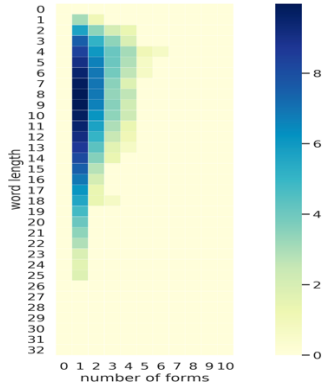


Figure 1: Word length vs. number of forms

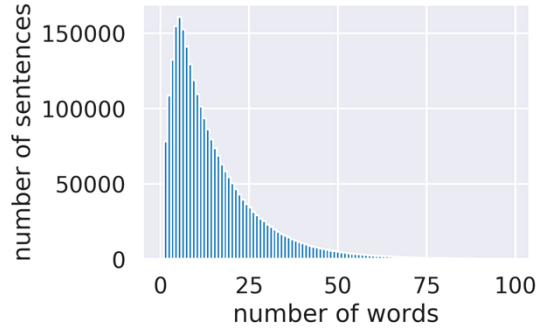


Figure 2: Sentences length histogram

5. Challenge baselines

5.1. Simple baselines

Two simple baselines were created in order to set a reference for the range of the possible performance values. The first and most straightforward approach has involved comparing the evaluation set without any diacritics against the groundtruth. We obtained a WA2 of 68.11%, which means that this is the proportion of words that have no diacritics. The second approach involved the use of a 1-gram probabilistic model, to replace each word that might have diacritics with its most commonly encountered form containing diacritics. In this case, we obtained a WA2 of 97.57%. The conclusion that emerges from these two baselines is that any simple algorithm should not obtain accuracies under 68%, while a more elaborate algorithm should definitely be more accurate than 97% (WA2 obtained with the 1-gram approach).

5.2. Recurrent neural network approach

Because the task is of the type sequence-to-sequence, a natural option is the use of a recurrent neural network (RNN). Its role is to extract useful information about the sequence of characters making up the words and then the sentences. We work at the character level to prevent the situations when some words are not found in the vocabulary. All the diacritics were removed from the training set in order to be used at the network input. At each iteration, the loss function, categorical cross-entropy, is computed as the difference between the text obtained at the network output, which is supposed to have the correct diacritics, and the training set containing diacritics. The network was trained using character level examples. A total of 44 symbols were used: non-diacritical letters, hyphens, space or end of line symbols. Each one was assigned to an integer number. The next step involves moving to a vector representation, where

each trainable example is associated with a vector, which corresponds to a recurrent cell in the network. On top of the RNN is a fully connected layer, characterized by the softmax activation function. Two approaches regarding the network output were used:

- each symbol is represented at the output as its index from the list of possible output symbols. Because a total of 44 symbols are used, the network could perform the classification task into 44 classes. But this method is not optimal, because the large number of output classes leads to a large number of parameters, implicitly to a longer duration of training.
- to prevent the previous issue, the network output consists into 3 possible classes as follows: 0 - no diacritics required, 1 - one of the $\hat{i}$, $\hat{s}$, $\hat{t}$, $\hat{a}$ (the final decision depends on the input character) and 2 - $\hat{a}$.

The architecture depicted in Figure 3 represents the general concept of the network. In our experiments, it has undergone changes regarding the type of the recurrent cell (LSTM or GRU), the type of the recurrent layer (unidirectional or bidirectional) or the number of recurrent layers.

Data Preprocessing. The sentences in the training set are of different length in terms of the number of characters. Although RNNs allows variable length sequences to be trained, this results in a longer training time, since operations cannot be parallelized. Thus it is desired to train on sequences of equal lengths. Therefore, a couple of data preprocessing approaches can be taken into account. The simplest assumes that all sequences are brought to the length of the largest sequence, by filling them with the space character. However, this method creates a lot of redundancy and leads to a very long training time, so we did not use it. Instead, we used the following two methods:

1. inside cutting: all training examples were joined together into one single large string, followed by dividing it into equal pieces of characters. Empirically it was determined that the optimal value of this length is 256 characters. The method works, but there are situations when some words are cut inside or words from different sentences are joined.
2. smart grouping: an optimized method was applied that adaptively processes the sentences. The aim was to obtain sequences of 256 characters, but without cutting inside the words. Words shorter than a threshold are selected, and then grouped so that the resulting strings have a length of 256 characters or less. Those that have less than 256 characters are filled with the space character.

Our first two experiments consisted of using a recurrent network with LSTM cells, in which we applied the above listed data preprocessing methods. We obtained a value equal to 98.67% WA2 when we applied the inside cutting method, respectively 99.03% WA2 when we applied the smart grouping method. Therefore, there is a considerable relative improvement brought by the optimized data preprocessing method, being 27% better than the other one. We use this method in future experiments.

The experiments from Table 3 aimed at trying several modifications in terms of network architecture. Recurrent unidirectional and bidirectional networks were tested, both with LSTM cells and GRU cells, and the number of layers ranged from one to two layers.

The performances were higher for the two-layer networks, compared to the single-layer ones. GRU cells obtained weaker results, but close to the results obtained with LSTM.

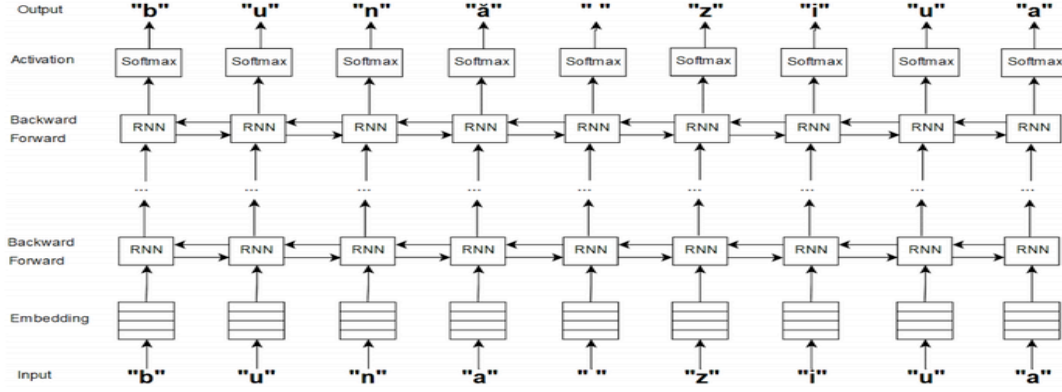


Figure 3: Character-level RNN: it takes as input a character, it predicts its diacritical form

Table 3: Results varying the cell type and the number of layers

Network type	Parameters [millions]	Epochs	Training time [h]	Evaluation (WA2 [%])
BiLSTM-1	0.8	62	4.1	98.79
BiLSTM-2	2.4	55	9.2	99.42
LSTM-1	0.4	71	2.9	88.47
LSTM-2	0.9	88	7.0	88.88
BiGRU-1	0.6	65	4.0	98.59
BiGRU-2	1.8	47	6.1	99.29

Finally, the bidirectional two-layer LSTM architecture obtained the best results, but it is also the most expensive, both in terms of the number of parameters, implicitly of the memory, but also of the time required for training. The accuracy has been further enhanced by running hyperparameter tuning experiments, but also by applying optimization techniques, such as: early stopping, dropout on the recurrent layers or on the output layer, finding a suitable value for the learning rate or dimensioning the size of the batches used for training.

6. Conclusions

In this paper we introduce a diacritics restoration challenge for the Romanian language. We provide a dataset containing both training and evaluation subsets, summing 42 million words, as well as an online leaderboard to keep track of the various submissions and their performances. We also offer baseline results based on a unigram model and recurrent neural networks. By carefully tuning the hyperparameters (number of layers, cell type, learning rate, dropout) and applying optimization methods such as early stopping, we obtained the best results using a two-layer BiLSTM network, gaining an

accuracy value of 99.46% (WA2). We invite the community to further improve upon these results.

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, within PNCDI III.

References

- Adegbola, T., Odilinye, L. U. (2012). Quantifying the effect of corpus size on the quality of automatic diacritization of Yoruba texts. In *Spoken Language Technologies for Under-Resourced Languages*, 48–53.
- Arslan, A. (2016). Deasciification approach to handle diacritics in Turkish information retrieval. In *Information Processing & Management*, vol. 52, 326–339.
- Belinkov, Y., Glass, J. R. (2015). Arabic diacritization with recurrent neural networks. In *EMNLP*, 2281–2285.
- Cocks, J., Keegan, T. T. (2011). A word-based approach for diacritic restoration in Maori. In *Proceedings of the Australasian Language Technology Association Workshop*, 126–130.
- Cristescu, H. (2018). Romanian-diacritics-restoration. Available: <https://github.com/horiacristescu/romanian-diacritic-restoration>. [Accessed at 26 June 2019].
- Cucu, H., Buzo, A., Besacier, L., Burileanu, C. (2014). SMT-based ASR domain adaptation methods for under-resourced languages: Application to Romanian. In *Speech Communication Journal*, Vol. 56 – Special Issue on Processing Under-Resourced Languages, 195-212.
- Do, T. N. D., Nguyen, D. B., Mac, D. K., Tran, D. D. (2013). Machine translation approach for Vietnamese diacritic restoration. In *Asian Language Processing (IALP), 2013 International Conference on. IEEE*, 103–106.
- Ezeani, I., Hepple, M., Onyenwe, I. (2016). Automatic restoration of diacritics for Igbo language. In *International Conference on Text, Speech, and Dialogue*. Springer, 198–205.
- Khorsheed, M. (2012). A HMM-based system to diacritize Arabic text. In *Journal of Software Engineering and Applications*, vol. 5, 124-127.
- T. A. Luu, T. A., Yamamoto, K. (2012). A pointwise approach for Vietnamese diacritics restoration. In *Asian Language Processing (IALP), 2012 International Conference on. IEEE*, 189–192.
- Mihalcea, R., Năstase, V. (2002). Letter level learning for language independent diacritics restoration. In *Proceedings of the 6th conference on Natural language learning-Volume 20*, 1-7.
- Náplava, J., Straka, M., Straňák, P., and Hajic, J. (2018). Diacritics restoration using neural networks. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.

- Nelken, R., Shieber, S.M. (2005). Arabic diacritization using weighted finite-state transducers. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*. Assoc. for Computational Linguistics, 79–86.
- Novák, A., Siklósi, B. (2015). Automatic diacritics restoration for Hungarian. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2286–2291.
- Nutu, M., Lorincz, B., Stan, A. (2019). Deep Learning for Automatic Diacritics Restoration in Romanian. In *Proceedings of the IEEE 15th International Conference on Intelligent Computer Communication and Processing*, Cluj-Napoca, Romania.
- Orife, I. (2018). Attentive Sequence-to-Sequence Learning for Diacritic Restoration of Yoruba Language Text. In *Proceedings of Interspeech*, 2848-2852.
- De Pauw, G., Wagacha, P. W., De Schryver, G.-M. (2007). Automatic diacritic restoration for resource-scarce languages. In *International Conference on Text, Speech and Dialogue*. Springer, 170–179.
- Petrică, L., Cucu, H., Buzo, A., Burileanu, C. (2014). A robust diacritics restoration system using unreliable raw text data. In *Proceedings of the 4th International Workshop on Spoken Language Technologies for Under-resourced Languages (SLTU)*, St. Petersburg, Russia, 215-221.
- Pham, T.-H., Pham, X.-K., Le-Hong, P. (2017). On the use of machine translation-based approaches for Vietnamese diacritic restoration. In *arXiv preprint arXiv:1709.07104*.
- Raza, A. Hussain, S. (2010). Automatic diacritization for Urdu. In *Proceedings of the Conference on Language and Technology*, 105–111.
- Rușeți, Ș., Cotet, T.-M., Dascălu, M. (2018). Romanian Diacritics Restoration Using Recurrent Neural Networks, In *Proceedings of the International Conference on Linguistic Resources and Tools for Natural Language Processing – ConsILR, Iași, 22-23 November 2018*.
- Scannell, K. P. (2011). Statistical unification of African languages. In *Language resources and evaluation*, vol. 45, no. 3, 375-386.
- Schlippe, T., Nguyen, T., Vogel, S. (2008). Diacritization as a machine translation problem and as a sequence labeling problem. In *AMTA-2008. MT at work: In Proceedings of the Eighth Conference of the Association for Machine Translation in the Americas*, 270–278.
- Tufiș, D., Chițu, A. (1999). Automatic diacritics insertion in Romanian texts. In *Proceedings of the International Conference on Computational Lexicography COMPLEX*, Vol. 99, 185-194.
- Tufiș, D., Ceaușu, A. (2008). DIAC+: A professional diacritics recovering system. In *Proceedings of LREC 2008*, 167-174.
- Tursun, O., Cakici, R. (2017). Noisy Uyghur text normalization. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, 85–93.
- Ungurean, C., Burileanu, D., Popescu, V., Negrescu, C., Derviş, A. (2008). Automatic diacritic restoration for a TTS-based e-mail reader application. In *University Politehnica of Bucharest Scientific Bulletin, Series C*, vol. 70, no. 4, 3-12.

TRIPOD: LEARNING LATENT REPRESENTATIONS FOR SEQUENCES

TIBERIU BOROS¹, ANDREI COTAIE¹, ALEXANDRU METEREZ², PAUL ILIOAICA²

¹ *Adobe Inc., Security Coordination Center*

² *Intern/External Contributor*

{boros, cotaie}@adobe.com

{alexandrumeterez, ipaulbeniamin}@gmail.com

Abstract

We propose a new model for learning and extracting latent representations from sequences, which generates a tripartite representation: global style, memory-based and summary-based partitions. We show the relevance of these representations on a couple of mainstream tasks such as text similarity and natural language inference. We argue that the generic nature of this approach makes it applicable to many other tasks that involve modelling of discrete-valued sequences (time-ordered) and, with some modifications even to image and speech processing. We encourage everyone to try our opensource code and our Python3 API.

Key words — machine learning, latent representations, embeddings, Natural Language Processing, sequence, sentences, paragraphs.

1. Introduction

The emergence of word2vec (Mikolov *et al.*, 2013) has enabled research in the direction of learning unsupervised models that generate “semantic”-oriented representations of words, tokens, images, speech etc. Such approaches include (but are not limited to) Bert (Devlin *et al.*, 2018), ELMo (Peters *et al.*, 2018), Glove (Pennington *et al.*, 2014) as well as Paragraph2vec (Le and Mikolov, 2014) and the speech-oriented Global Style Tokens (GST) (Wanget *et al.*, 2018). Previous approaches limit their scope to language processing, whereas Code2vec (Alon *et al.*, 2019) focuses on learning latent representations for computer code.

Our methodology combines lessons learned from several models and is designed to compute the latent representations of long sequences. Our approach is generic, in the sense that we extract latent representations for both text and code (described elsewhere). Thus, the same model architecture can be used for natural language sentences, code segments and, with minor modifications, for speech and images. However, our focus for the present work will be limited to input sequences composed of discrete valued tokens, such as text or code. We stress out that our goal is to generate latent representations for long sequences and not for individual tokens. To our knowledge,

methods that achieve the same goal are Paragraph2vec (Le and Mikolov, 2014), Doc2vec (Gupta *et al.*, 2016) and Code2vec (Alon *et al.*, 2019).

In Section 2 we focus on describing our model, training procedure and the intuition behind our choice of architecture. Later on, in Section 2 we show the results of experiments conducted on text similarity and natural language inference. Finally, we bring conclusions and discuss further development plans in Section 4.

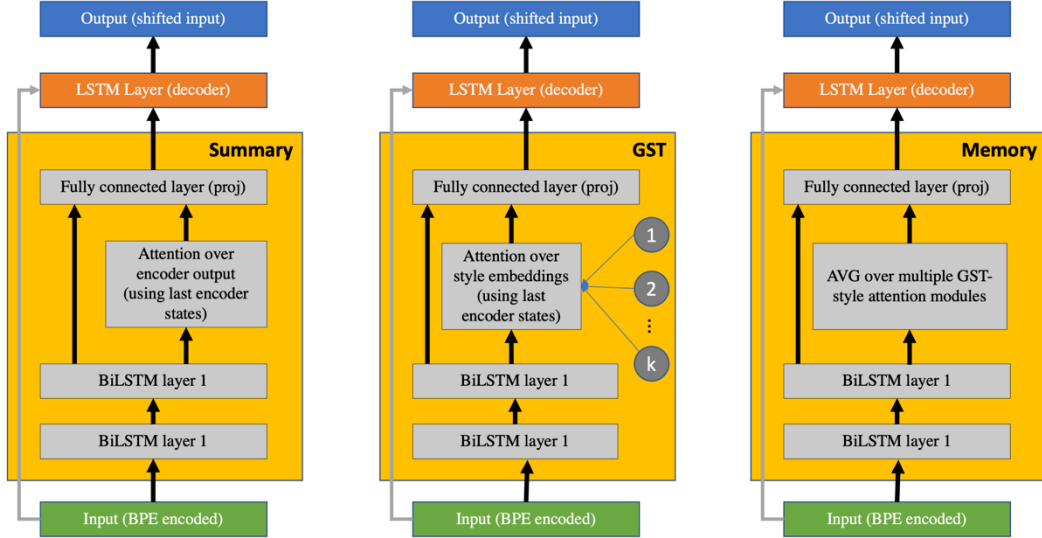


Figure 1: Overview of Tripod's architecture

2. Tripod model

Several network architectures and training strategies have been previously proposed for learning how to compute latent representations in an unsupervised manner. We draw the attention that most proposed methodologies use the reconstruction of the input as the target objective, based on a type of “condensed” numeric representation, which is often the same as the latent one.

Our approach has the same training objective. However, the **difference is that we use and encoder-decoder seq2seq model** that is conditioned to reproduce the input using **three different strategies**. These strategies dictate the three latent-representation partitions: summary-based, GST-based and memory-based.

In order to generate a latent numeric representation for a sequence, we independently train three models that share a similar architecture (Figure 1) but have no parameters in common. All three methods are based on the intuition that the decoder will act as a language model (LM) that extends the standard LM estimation to incorporate “context-conditioning” (i.e. computes $P(w_k | w_1, w_2 \dots w_{k-1}, c)$, where c is a context-vector a.k.a. latent representation).

The summary-based representation uses a standard attention mechanism over the encoder states. Just like in self-attention we use the final encoder (not decoder) states in

the affine transformation. Also, the context vector (Eq. 1, 2 and 3) is computed only once and the same latent representation is used to condition each decoder state in the generation phase. Because the same representation is used for every time-step, our intention is that the encoder-attention model will learn to summarize the input and retain information that is not captured by the Language Model (LM), also known as the decoder.

$$c = \sum_{i=1}^n h_i a_i \quad (1)$$

$$a_i = \frac{\exp e_i}{\sum_{k=1}^n \exp e_k} \quad (2)$$

$$e_k = \text{MLP } h_k \oplus s_n^f \oplus s_n^b \quad (3)$$

In the above notations c is the context vector, n is the sequence length, h_k is the k -th output of the top-level concatenated encoder states, s_n^f , s_n^b are the k -th hidden states of the top level forward(f) and backward(b) encoder and MLP is a three-layer perceptron with one hidden layer ($\tanh$ activation) and scalar output.

The global style-tokens (GST) representation is inspired by (Wang *et al.*, 2018) which is used for prosody embeddings in end-to-end text-to-speech (TTS) synthesis. This approach uses a LSTM encoder that “sees” mel-cepstral coefficients computed from human speech samples. The last state of the encoder is used to compute an attention over a fixed number of “embeddings”¹ and, based on this attention, a conditioning vector that guides a decoder to reconstruct the input speech. Their experiments show that the global style tokens are able to capture speech-specific characteristics, such as emotion, sloppiness and speaking speed, which are transferable across utterances. In our approach the encoder is bi-directional, and we use trainable embeddings instead of cepstral coefficients.

The memory-based representation is inspired after the GST strategy, but instead of using one attention over one set of tokens we combine multiple trainable attentions with multiple groups of token groups. While the small number of tokens in GSTs forced the model to learn generic styles, this larger model is able to “memorize” fine-grained details about the input examples. Finally, by concatenating the context vectors from each token group and applying an affine transformation we obtained the context vector c that is used to condition the decoder. This is required in order to reduce the dimensionality of the input and computational costs.

The final latent representation is obtained by concatenating the context vectors that are independently computed by the three models.

¹ Equations 1, 2 and 3 can be adapted by replacing n with the number of style-tokens and using the trainable embeddings e_i instead of the encoder outputs h_k .

In Section 3 we validate Tripod vectors on mainstream NLP tasks. All the experiments were conducted using a Tripod model which was pre-trained on the Wiki-103 corpus (Merity *et al.*, 2016). Before that, we invite readers to take a look at Figure 2 which shows a graphic representation² of Tripod vectors³ for the following paragraphs:

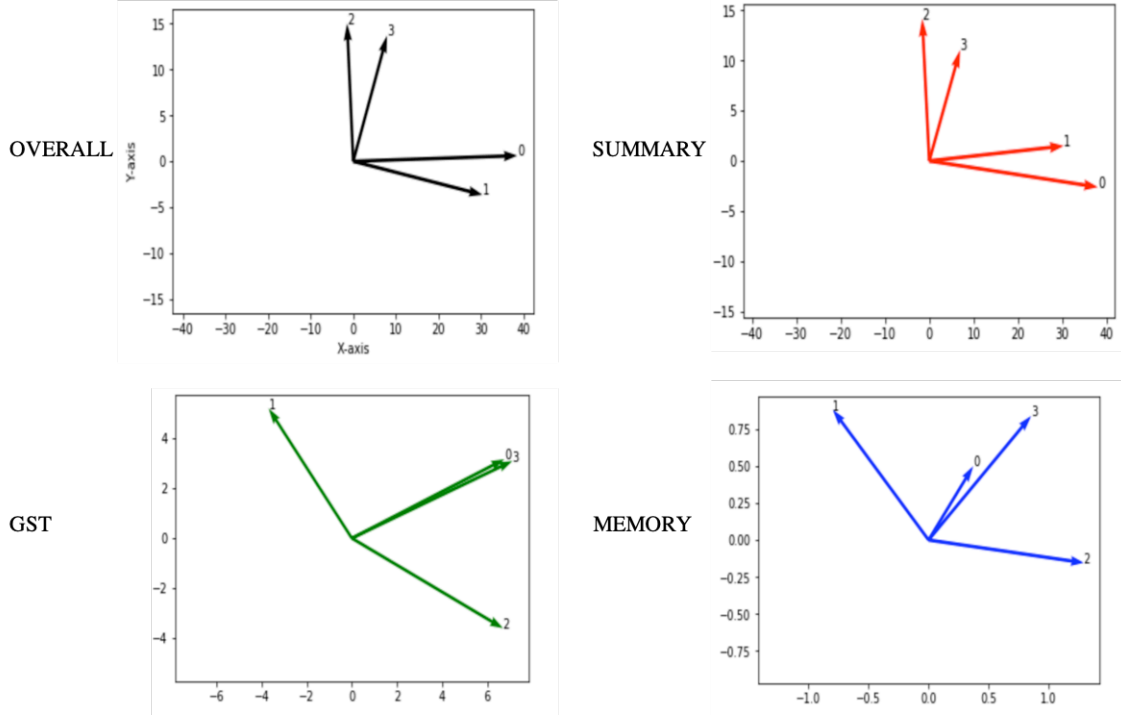


Figure 2: Vector representation for the four sentences

“Science (from the Latin word scientia, meaning “knowledge”) [1] is a systematic enterprise that builds and organizes knowledge in the form of testable explanations and predictions about the universe.”

“Their contributions to mathematics, astronomy, and medicine entered and shaped Greek natural philosophy of classical antiquity, whereby formal attempts were made to provide explanations of events in the physical world based on natural causes”

“Bucharest is the capital and largest city of Romania, as well as its cultural, industrial, and financial centre. It is located in the southeast of the country”

“The city proper is administratively known as the “City”, and has the same administrative level as that of a national county, being further subdivided into six sectors, each governed by a local mayor.”

² We used SVD to get 2D coordinates.

³ “overall” – is the concatenated representation and the other three vectors represent the individual partitions for summary, GST and MEM.

Note 1: In order to speed-up convergence, in the data preparation phase we used TF-IDF and SVD (20 dimensions) followed by K-Means clustering to k clusters, with k equal to the number of GST tokens. The assigned a cluster index to each data entry and was used the cluster index to guide the attention on the GST tokens.

Note 2: During the experiments in Section 3 the Tripod model was not fully converged.

Note 3: Initially, the computed representations were close to the centroids of the clusters. As the model converged, they drifted away from these centroids and the accuracy on the test datasets (SNLI and SICK – see Section 3) got better.

Note 4: Without the measures describe in (1), some of our models did not converge at all.

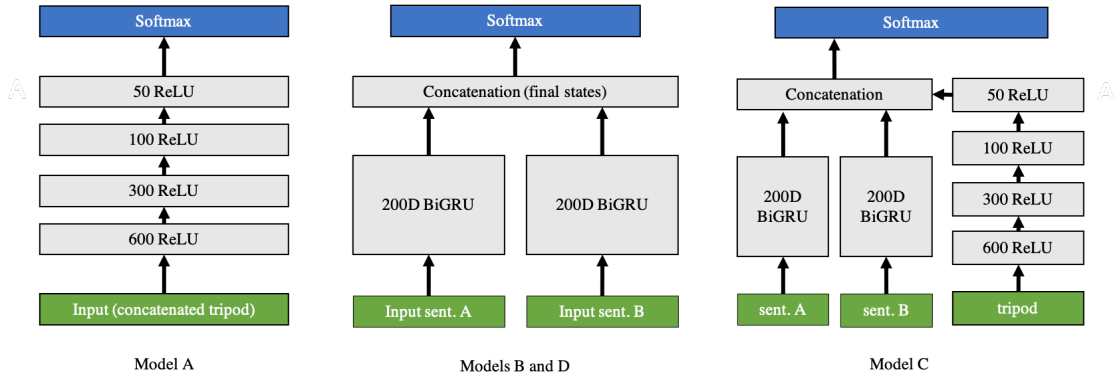


Figure 3. The three model architectures used in our experiments

3. Experimental validation

We validate our approach on two datasets that require Natural Language Inference: the SICK dataset (see Section 3.1) and the SNLI dataset (see Section 3.2).

3.1. The SICK 2014 dataset

Sentences Involving Compositional Knowledge (SICK) dataset (Marelli *et al.*, 204) is composed of 10K English sentence pairs. Each pair is annotated with relatedness score and entailment information, which is strictly directional (A in ENTAILMENT with B does not imply B in ENTAILMENT with A).

Table 1: Examples from the SICK dataset

#	Example	Rel.	Ent.
1	A group of kids is playing in a yard and an old man is standing in the background	4.5	Neutral
	A group of boys in a yard is playing and a man is standing in the background		
2	A man with a jersey is dunking the ball at a basketball game	4.9	Entailment

	The ball is being dunked by a man with a jersey at a basketball game		
3	There is no biker jumping in the air	4.2	Contradiction
	A lone biker is jumping in the air		

Table 1 includes a couple of examples extracted from the SICK 2014 dataset. Column “*Rel.*” stands for relatedness and is a numeric value from 0 (totally unrelated) to 5 (extremely related), while column “*Ent.*” stands for entailment and takes discrete values (N) - neutral, (C) - contradiction and (E) - entailment. It is obvious that detecting contradictions or entailment between two sentences is a fine-grained task, which requires the model focusing on small details (such as simple negations) that would otherwise change the outcome of the prediction.

Given that the training objective of tripod is purely unsupervised, we designed two experiments to assess the usefulness of tripod vectors:

(A) Simple Multi-layer Perceptron (MLP): In this experiment we train an MLP to predict the entailment state between two sentences by feeding it their concatenated tripod vectors. Doing so allows us to see how well-suited the tripod captured context is in this granular task.

(B) LSTM encoders with auxiliary tripod vectors: In this experiment we measure the value of tripod vectors combined with fine-tuned task-specific models. Particularly, we rely on Gated Recurrent Units (GRUs) to walk over tokens in both sentences and we combine their final states through concatenation. We show results both with and without using tripod vectors for conditioning the final Softmax output. We also experiment with the GRU unit’s input (see below).

For clarity (also see Figure 3), the list of our models is:

- **Model A:** MLP with tripod vectors;
- **Model B:** Parallel GRUs over locally computed embeddings for input tokens;
- **Model C:** Parallel GRUs over locally computed embeddings for input tokens and additional MLP with tripod input;
- **Model D:** Parallel GRUs over locally computed embeddings for input tokens and tripod vectors conditioning (by concatenating the embeddings with repeated tripod sentence vectors).

Other complex architectures achieve better results on this task - see (Radford *et al.*, 2018; Kim *et al.*, 2019; Zhang *et al.*, 2018; Liu *et al.*, 2019; Tan *et al.*, 2018) and many others for details. However, our goal is to set our own baseline when doing the evaluation. Arguably, we could fine-tune our task specific models to yield better performance. However, this would be out-of-scope, since we are measuring the contribution of tripod vectors to this task, instead of going against the SOTA. The fact that we were able to directly use tripod vectors with the MLP for textual entailment, given the unsupervised training goal used in pre-training, is itself promising.

3.2. The Stanford Natural Language Inference (SNLI) corpus

SNLI (Bowman *et al.*, 2015) is a well-known corpus for Natural Language Inference (NLI), which was extensively used to assess the performance of SOTA systems. Needless to say, it was also a good candidate for us, and we decided to further evaluate tripod and show the results obtained on this particular task and dataset.

The SNLI corpus contains 570K sentence pairs, which makes it 57 times larger than the SICK dataset. It does not include a relatedness score, but it includes fine-grained annotations at chunk level.

For simplicity, we used the exact same models described in the previous section (3.1) and we did not alter the train-validation-test split.

Table 2 shows the results obtained by our four models on the SICK and SNLI datasets. For completeness we include relevant SOTA results from comparable models.

Table 2: Experimental results⁴ for SICK and SNLI test datasets (accuracy on entailment prediction).
 “Sum of words” is from Bowman *et al.*, 2015

	Model	SICK	SNLI
Ours	Model A	73.68	77.83
	Model B	71.16	78.41
	Model C	72.20	72.51
	Model D	71.58	68.94
Other	100d sum of words	N/A	75.3
	Arora <i>et al.</i> (2016)	84.60	78.20

4. Conclusions and future work

We introduced a new model for extracting latent representation from sequences, that relies on a tripartite representation: summary, style and memory.

To evaluate the relevance of our model's learned representations we ran a series of experiments on several main-stream NLP tasks. We showed that the unsupervised training objective is sufficient to make the model usable on tasks that rely on fine-grained observations.

We argue that our model is smaller than several alternative models, it is computationally fast (we achieve 1K sequences per second on an RTX 2070) and the code is fully available as Opensource⁵ under the Apache v2 License.

We also provide a python API and quick installation method via pip/conda packages, as well as collaborative notebooks (check the documentation uploaded on GitHub).

⁴ These tests are reported on an older Tripod model, not the one available on Github.

⁵ <https://www.github.com/adobe/tripod>.

Work was done to evaluate Tripod vectors on JavaScript (JS) (latest model already available) and experiments are being conducted on detecting malicious JavaScript code based on just latent representations.

Furthermore, we plan to extend our support to other languages and provide Wikipedia-based pre-trained models.

You can leave any suggestions and requests for support on the official GitHub repository.

References

- Alon, U., Zilberstein, M., Levy, O. and Yahav, E. (2019). code2vec: Learning distributed representations of code. In *Proceedings of the ACM on Programming Languages*, 3(POPL):40.
- Arora, S., Liang, Y. and Ma, T. (2016). A simple but tough-to-beat baseline for sentence embeddings. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 1-14.
- Bowman, S. R., Angeli, G., Potts, C. and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.
- Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional trans-formers for language understanding. arXiv preprint arXiv:1810.04805.
- Gupta, M., Varma, V., et al. (2016). Doc2sent2vec: A novel two-phase approach for learning document representation. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 809–812. ACM.
- Kim, S., Kang, I. and Kwak, N. (2019). Semantic sentence matching with densely-connected recurrent and co-attentive information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 6586–6593.
- Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *International Conference on Machine Learning*, 1188–1196.
- Liu, X., He, P., Chen, W. and Gao, J. (2019). Multi-task deep neural networks for natural language understanding. arXiv preprint arXiv:1901.11504.
- Marelli, M., Bentivogli, L., Baroni, M., Bernardi, R., Menini, S. and Zamparelli, R. (2014). Semeval-2014task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, 1–8.
- Merity, S., Xiong, C., Bradbury, J. and Socher, R. (2016). Pointer sentinel mixture models. arXiv preprint arXiv:1609.07843.

- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119
- Pennington, J., Socher, R. and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.
- Peters, M. E., Neumann, M., Zettlemoyer, L. and Yih, W. (2018). Dissecting contextual word embeddings: Architecture and representation. arXiv preprint arXiv:1808.08949.
- Radford, A., Narasimhan, K., Salimans, T. and Sutskever, I. (2018). Improving language understanding by generative pre-training, (https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf).
- Tan, C., Wei, F., Wang, W., Lv, W. and Zhou, M. (2018). Multiway attention networks for modeling sentence pairs. In *IJCAI*, 4411–4417.
- Wang, Y., Stanton, D., Zhang, Y., Skerry-Ryan, R., Battenberg, E., Shor, J., Xiao, Y., Ren, F., Jia, Y. and Saurous, R. A. (2018). Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. arXiv preprint arXiv:1803.09017.
- Zhang, Z., Wu, Y., Li, Z., He, S., Zhao, H., Zhou, X. and Zhou, X. (2018). I know what you want: Semantic learning for text comprehension. arXiv preprint arXiv:1809.02794.

CHAPTER 3

SYMBOLICAL APPROACHES VS.

STATISTICS

SOME LOGICAL AND COMPUTATIONAL METHODS FOR THE ANALYSIS OF THE INEFFABLE

CĂTĂLINA MĂRĂNDUC^{1,2}, ANCA-DIANA BIBIRI³

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

²*Iorgu Iordan – Al. Rosetti Institute of Linguistics, Romanian Academy, Bucharest*

³*Institute for Interdisciplinary Research, Social Sciences and Humanities Research Department, “Alexandru Ioan Cuza” University of Iași*

catalinamaranduc@gmail.com

anca.bibiri@gmail.com

Abstract

In this article we aim to show that measuring the probability that a combination of terms in a text will appear in the language model is a method of classifying the text as figurative and finding the points where the meaning is symbolically unlimited. The classification of texts according to the discourse emitter's intention defines the type of (lyric) text. The text is classified from the optimization of the communicative intention's fulfillment point of view, a fact that leads the emitter to select combinations of words which have 0% frequency in a language model. The “semantic induction” shows that three or more such strengths of the text inter-influence and orient the interpretation towards a unique textual meaning. An experiment is described, by which receivers are determined to interpret and produce figurative meaning. The answers must fit some criteria, correctly interpreted and reproduced. The results were classified by dividing them into distinctive features.

Key words — contiguity, distinctive features, language figures, language model, semantic induction, substitution.

1. Introduction

The figurative meaning is studied in the NLP especially as being achieved by metaphors, that is, transfers of semantic attributes from a source concept to a target concept (Ovchinnikova *et al.*, 2014). In linguistics those semantic attributes are called the medium term, and the figures are classified after the formal realization as comparisons, if the three terms are all present, metaphors if the target is substituted by the source, and contiguities if the terms are indirectly present in context (Coteanu, 1973). This way of presenting the problem is meant to detect and then interpret lexicalized style figures, which are about to turn into MWE-s, but how will we detect and interpret an original figure?

The computer can find the combinations of terms which express feelings by calculating their frequency in texts written in a given language. The method of making word indices and calculating their frequency provides information about the dominant themes and the insistence of their being pursued.

The communicative intention led to the optimization of its implementation. To express a feeling in a poetic text, a banal selection of terms cannot be used, because they would not attract the attention of the receiver and would not convey an unrepeatable experience, so the communicative intention would not be achieved. It can be considered that any experience, being new and irretrievable, is transmitted by an expression that does not be worn-out by long-term use.

The paper is structured in 6 sections as follows: after a brief introduction about the figurative meaning, section 2 is dedicated to some approaches to express feelings in lyrical text; section 3 presents an attempt of identifying figurative meaning through a language model; section 4 deals with an experiment called „Semantic induction” made in the Poetic Group of Bucharest led by Ion Coteanu, to show that analogies occur both in the meaning and form of the figurative expression, while section 5 presents the textuality standards, i.e. expectability, and, finally, the last section draws some conclusion upon the subject discussed.

2. Related Work

Numerous analyses of the feelings expressed by the lyrical text were made by stylists and theorists of literature. At the same time, attempting to analyze the poetic text with mathematical and computer resources is not a novelty. The more representative researcher in the field of Mathematical Poetics, after the mathematician Solomon Marcus (1970), is the philologist Ecaterina Mihăilă.

She debuted with the study of Mathematical Methods in the Analysis of the Literary Text, in the collective volume *Analysis and Interpretation* (1972), written by the members of the Poetic Group of Bucharest, led by Ion Coteanu. She did research on poetic language from the perspective of mathematics and information theory (Mihăilă 1980; 1995; 2001; 2003).

The analysis of the poetic text with mathematical and computer resources is less in the attention of the computer scientists, because they studied the sentiments expressed in non-literary texts. In Tufiş (2011) more development of WordNet for the affects expression are cited. The developments of WordNet Affect (Valitutti *et al.*, 2004) and SentiWordNet (Esuli and Sebastiani, 2006), try to remedy the lack of information regarding the sentiment load of the words recorded in a semantic dictionary such as WordNet (Fellbaum, 1998).

Opinion mining (also known as “sentiment analysis”) is a subdiscipline at the crossroads of information retrieval and computational linguistics. Opinion-driven content management has several important applications, such as determining critics’ opinions about a given product by classifying online reviews (Pang and Lee, 2008), or tracking the attitudes of the general public towards a political candidate in forums or blogs (Wiebe *et al.*, 2004).

This type of research is not new; more than 50 years ago the paper (Osgood *et al.*, 1957) on the theory of semantic differentiation shows that connotative meanings could be outlined and measured by a semantic differential technique.

WordNet is a dictionary network that encompasses almost all of today's natural languages. Acad. Dan Tufiş led a project that created RoWN, i.e. the Romanian language was aligned to PWN (Princeton Word Net). This is a dictionary organized on meronyms and hyperonyms, i.e. hierarchic concepts less or more generals.

The network is different for nouns, verbs, adjectives. Dan Tufiş (2011) describes a possible logical organization of the adjectives network, starting to Osgood's (1957) theory. According to this theory, the connotative meaning of some adjectives can be rated on a scale, finishing by antonymic adjectives. Such a pair of antonymic adjectives is called a factor. A synset is a node of the network, containing all the words that express a notion in the languages comprised in the PWN (all synonyms). The synsets are annotated with the positive or negative sentiment evaluation for each notion. Examples of polarised factors rendered as types, i.e followed by their type:

({unfairness:2...}<-> {fairness:1...}) NormativeAttribute;

({discomfort:1...} <-> {comfort:1...}) StateOfMind

({decrease:2...} <-> {increase:3...}) QuantityChange;

({bad:1...}<->{good:1...}) SubjectiveAssessmentAttribute;

({inactive:2...}<->{active:1}) BiologicalAttribute

We can see that these polarizations do not refer to feelings expressed in poetry, which are very personalized and require more refined assessments; but we hold on to the antonymic polarization method.

What about figurative meanings? The tasks which NLP try to solve are the detection and the interpretation of the conceptual metaphors. The third task, i.e. the generation of metaphors is also neglected, however, it can have more applications (Miyazawa and Miyao, 2017). In this paper, the metaphors are evaluated by comparing the contextual meaning with the lexical meaning and their comprehensibility and novelty are statistically calculated. Shutova (2010) makes an inventory of metaphoric and metonymic generation of figures showing that the meaning can be decoded only in the given context.

3. The Romanian Language Model

The figurative meaning can be a non-lexicalized one, not used in the conversation or non-lyric texts. Examples:

(1) Noapte bună! (*Good night!*)

(2) Fie-ți îngerii aproape, somnul dulce! (*May the angels be close to you and your sleep be sweet!*)

(Mihai Eminescu, *Somnoroase păsărele* (*Sleepy Birds*))

(1) is a polite and banal wish; it does not convey a special feeling, but marks the end of a discussion in a situation of communication (in the end of a day). No figurative expression will be found in the first example; their frequency in the language is high. Classical poetry (2) has the aim of finding an expression that is original and beautiful. As far as modern poetry is concerned, the problem must be not only a more beautiful expression than what has been said before, but also something new, similar with nothing that has been said before, while almost everything has already been said. No one wants to read in a new book what they have already read in five other books.

We can make the computer detect the figurative meaning of the words, by statistical methods (Maning and Schütze, 1999). We can build bigram or trigram models of the language that list all combinations of two or three word-occurrences in a language. The texts of no matter how big book written in Romanian will be made to continue one after the other. Then, the words will be separated with a tokenizer so that we have one word per line and the punctuation will be deleted. Then, we place the same string on two or three columns, so that the second column starts one row higher and the third one starts two rows higher. The string of words must contain millions of word-occurrences. The computer can calculate the frequency or the probability of the occurrence for a given combination of terms (in the rows of the model). If its frequency tends to 0%, it is very probably an expression used in a figurative sense. One warning is in order here: Do not use uncorrected texts obtained by an Optical Character Recognizer (OCR) program! The mistaken words have also a 0% frequency and are not figurative expressions.

Table 1. The beginning of a trigram language model.

	somniază		sleepy
	somniază	păsărele	birds
somniază	păsărele	pe	at
păsărele	pe	la	their
pe	la	cuiburi	nests
la	cuiburi	se	they
cuiburi	se	adună	are gathering
se	adună	se	they
adună	se	ascund	hide
...	...	...	...

To express the feeling is enough to choose terms whose lexical meaning is consistent with what the emitter/addressee feels. But it should be noted that for what he/she expressed to be received by someone and not ignored, he/she needs to choose an expression that attracts the attention of the receivers.

The method used by literary creators is to seek out new combinations of terms. Consequently, such a combination is even more capable to focus the receiver's attention as the frequency in the language model tends to 0%. The frequency 0% marks a combination of terms with a figurative meaning that is not sharpened by use. We will compare the text above with a poem written online by an unknown writer, Vasile Gheorghe.

Păsări de lemn Se înălțau, Sau, prin soarele căzut Crengile treceau înalte? Înlemnea lumina în raze.

SOME LOGICAL AND COMPUTATIONAL METHODS FOR THE ANALYSIS OF THE
INEFFABLE

(*Wooden birds Flying up. Or, through the fallen sun, did the high branches pass? Light encased in wooden rays.*)

To find the figurative expressions (with a computer program), we must recast the text in the trigram format:

Table 2: The trigram model of the Vasile Gheorghe's poetry.

			păsări	Frequency
	păsări	de		
păsări	de	lemn	5%	
de	lemn	se		
lemn	se	înălțau	5%	
se	înălțau	sau		
înălțau	sau	prin		
sau	prin	soarele		
prin	soarele	căzut	0%	
soarele	căzut	crengile	0%	
căzut	crengile	treceau	0%	
crengile	treceau	înalte	5%	
treceau	înalte	înlemnea	5%	
înalte	înlemnea	lumina	0%	
înlemnea	lumina	în	0%	

Even if we had a big string of Romanian written texts on a single row of words a table like Table 1, we would still not encounter combinations of terms with frequencies ranging from 5% to 0 as in Table 2. We rarely find such 0% possible expressions in the language in a classic text. Example of an exception:

(3) cuibar rotind de ape (*turning nest in the water*).

The truth value of 0% frequency expressions is also 0 = false in a real-world:

- Wooden birds flies =0;
- Light encased in wooden rays =0;
- The sun fall =0;
- The branches passed =0.
- Turning nest in the water=0

Although for one whole century now, since the paper authored by John Searle (1981) has shown that fictional texts describe possible worlds in which real-world truth is suspended, this fact has not been taken into consideration.

To find the meaning of such a figure that in Logical semantics is equal to the truth value (here suspended) of a construction/of an expression, no dictionary of symbols can help us. The more distant the terms used in the combination, the more possibilities of interpretation that find their place in the space between them. But the key to the meaning can be at the end of the text, and with it the reader can be sent to the text

beginning by feedback to read the text in light of the meaning suggested by this key.
Example:

(4) Pe când tot mai adânc dormeam / În pământ, ca un sămbure. (As more and more I was asleep /In the ground, like a kernel).

In the poem “Sleepy Bird”, the key found is “ți” (your), the person to whom the poet directly addresses himself tenderly and full of care. It seems to be a kind of love poem. In the second poem, the end key is “I”, the poet, who is sleeping in the ground to give birth to a seed, like a kernel; the sunset awakens the premonition of death.

4. The Semantic Induction

An experiment called “Semantic induction” was made in the Poetic Group of Bucharest led by Ion Coteanu, to show that analogies occur both in the meaning and form of the figurative expression. The experiment is based on bigrams like the 0% frequent in the previous chapter. The person being tested received two sets of terms between which the same relationship is possible to find, and a free term, for which the subject had to find a correspondent. The distance between the related terms was very big, like in the modern poetry; like that those we have shown above have the frequency 0% in the language model. Example:

senin – pelin	(<i>serene - absinth;</i>
cald – smarald	<i>warm - emerald;</i>
sus – ?	<i>top - ?)</i>

The opposition of terms in the two inductive groups takes place at a high level of abstraction, the subject carrying out the test is required to decode and generate figurative meaning (in this example, to decode and generate metaphorical opposition).

The results of these experiments were extremely interesting, but their interpretation was very difficult. We will propose a computational solution, the terms being searched in WordNet (Fellbaum, 1998), where we find the related notions. Each related notion will be considered as distinctive feature and evaluated with – or + values.

The first pair is towards a semantic field of character traits and another in the field of taste – bitter. But the character feature “serene” is a +pleasant one and the absinth taste is a –pleasant one, so we have +opposition between the terms of the first pair. The second pair is a synesthetic equivalence of the color sphere with the temperature sphere. For example, we can extract the distinctive features:

– bitter +; – color +; – opposite +; – pleasant +; – warm +; – cold +.

For the painters, the concept of cold refers to the colors that are in the spectrum closer to the blue and the notion of warm refers to the colors that are in the spectrum closer to the orange. This opposition has more affairs with the expression of feelings than with the physical sensation of cold-hot. So, for the painters, emerald green is a cold color that expresses feelings of sadness, and orange is a warm color that expresses the love and the joy. Emerald opposes to warm.

The induction group was tested on children aged between 11 and 14 years, of a creative camp, so having artistic inclinations. The following responses were received to this test: apus (sunset), expus (exposed), pus (put), curs (flowed), presupus (supposed), supus (subjected), mus (apprentice), jos (down).

In order not to keep away from the field of stylistics, I. Coteanu (1973) classifies the figures of language in contiguous figures (metonymy and synecdoxa) and substitution figures (metaphor).

In order to function, i.e. to optimally polarize the receiver's attention, formal constraints are also required. One is the rhyme; the other can be the determinant-determiner syntactic structure (which may assume formal agreement).

So, we must add distinctive features as:

– contiguous +, – substitution +, – determiner +, – rhyme +.

We can then make the components of sense analysis and automatically intersect the sets of terms obtained for each of the first 2 pairs of terms, and then for the last two. (The sixth term being the response received from the user.) The meaning of the combination of terms will be at the intersection of these sets. The answer is appropriate if the intersection of the third group of two terms is similar to the intersection of the first two groups. An adequate answer of this induction group is “supus” (*subjected*). It has the distinctive features: + opposite, + rhyme, + determiner, – contiguous, +substitution.

This answer is not a perfect adequate one, because there are not a fort figurative meaning, the frequency in the language model is not 0%. The answer “curs” (flowed) has the same distinctive features and also a unattended character for the receiver, it is more adequate. The answer “apus” (*sunset*) is opposed to the inductor term, the sunset is not “sun at the top” but “sun down”. It is also adequate.

The answers “expus, pus, presupus, mus” have only a formal relation with the inductor term (+ rhyme) and are non-adequate. The answer “jos” has only an adequate distinctive feature, (+opposition).

We will propose another induction group, more complicated:

luminiș de-ntunecime – văpaie de neguri	(<i>clearing of darkness – flame of blacks</i>)
cer prăbușit – cădere înaltă	(<i>collapsed sky – high fall</i>)
parfum împietrit – ?	(<i>petrified perfume – ?</i>)

This group of induction has been experienced on advised lecturers of poetry, students of the Letters Faculty. It is formed by metaphoric nominal determiners with opposite meaning. The first line is formed of terms often encountered in poetry, poetic common expressions, but their antonymic connection is unusual. For some receivers, their decoding is superficial, they are perceived as common terms in poetry and the antonymic relation is not observed. It follows that feelings cannot be expressed by common expressions bunt by use, which results in the fact that lecturers do not perceive their meaning. The fact that poetry must avoid common expressions, valid for both the classical and the modern ones, is known in the Romanian critique from the nineteenth century (Maiorescu, 1878).

In the usual language, in a fixed phrase, the relationship between the terms is not decoded. El cere iertare (*He asks for forgiveness*) (and nothings is given to him), el își aduce aminte (*“he brings remember”*) and he brings nothings to himself.

Answers received:

diamant vaporos (vaporous diamond) (1), apă de colonie înghețată (frozen cologne water) (2), tăcere fluidă (fluid silence) (3), plăcere eternă (eternal pleasure), (4), parfum nemirositor (odourless perfume) (5);

piatră mirositoare (smell stone) (6), miros de piatră (smell of stone) (7), piatră fluidă (fluid stone) (8), floare vie (living flower) (9), împrejur intangibil (intangible around) (10), surâs de fotografie (*smile of photography*) (11). 20% of tested persons have no answer of this very difficult induction group. Applying a test with a more difficult induction group should be preceded by a care familiarization with the tool by applying simple tests.

For this test, the distinctive features must be:

- complexity +, – opposition +, – metaphor +, – inedit +, – synesthetic +,
- determiner +.

The answers (1), (2), (5), (6), (7), (8), (10) are adequate from the point of view of these features. (5) and (7) are inductions of the relation in the first line of induction. The other are inductions of the relation on the second line. The answer (10) is an original copy of oxymoronic metaphorical mechanism without relation of the first or second line.

Due to the long distance between the terms, the meaning decoded by the lecturer can be symbolically omitted, although the set of appropriate interpretations will always be less than the inadequate interpretations, both tending to infinite. This is the paradox of the “smaller” infinite.

If a larger number of advised lecturers tested give a greater number of appropriate responses, it would be possible to find other distinctive features that we did not initially notice. Example:

- – confusion gas /liquid /solid + (this is more appropriate than:)
- – synesthetic + (that means confusion visual /auditory/ odor).

The experiment can provide data on the comparative study of the ability of subjects to decode and reproduce more correctly the contiguity or substitution relationships.

Decoding is a process that depends on abstraction ability and logical thinking that is not specific to the literature. Good results were obtained in these tests for students who were better trained in mathematics than literature. The ability to reproduce poetic relationships depends both on the decoding capacity and on the wealth and property of vocabulary. The breadth of the cultural reference system, sensitivity, talent are also required. The ability to reproduce also depends on the ability of customization, the realization of the abstract and the artistic sense.

5. Expectability

A text can be studied from the point of view of the textuality standards: (Beaugrande and Dressler, 1984). The textual property of an ensemble of signs is circumscribed here by eight standards: cohesion (1), or how text surface components are interconnected in sequences, coherence (2), or how the configurations of concepts and relationships are mutually accessible, relevance (3), intentionality (4), acceptability (5), informality (6),

situationality (7), and intertextuality (8). We can add the expectability (9) (Vasiliu, 1990). For Plett (1983), all these qualities of the text are internal premises of the emitter and of the receiver. The (3), (5), (9) standards concern the receiver, and (4), (6) concern the emitter.

In his book, E. Vasiliu (1990) replaced all of the eight standards of the textuality by a single one: the expectability. He demonstrates that no syntactic or semantic rule can be made for establishing if a linguistic string is a text or it is not. If text-making rules would be written, they would be of pragmatic nature.

As the receivers have a great deal of decision-making and have a native textual sense, we can rely on their diagnosis, called expectability (=E) (in Măranduc, 2005 expectancy) to know whether a string of phrases is a text or it is not.

The string is formed by units, we can consider as its units the words, or the letters, or the clauses. It may be added at the right by a PAD (pas adițional la dreapta = *aditive step to the right*) but just as well to the left side by a PAS (pas adițional la stânga = *aditive step to the left*). Any such step can be accepted or rejected by the receptor's expectability. For each step, the E can have a set of values, let's say, $E = (0,1,2,3,4,5)$ like in a modal logical system, that has no only 2 truth values, 0 and 1.

Example (if the units are plain-meaning words):

(1) $PAD_1 = \text{Birds} + \text{of wood}$ $E = 1$

(2) $PAD_2 = \text{Birds of wood} + \text{flies}$ $E = 0$

The string evaluated as having 0% frequency in the chapter 3 is no evaluated as $E=0$, i.e. rejected by the receiver. But the expectability is a kind of truth value; it is suspended for the literary text.

Or, in other words:

The implicit textual rules for the evaluation of the Expectability are in relation to the type of the text. The advised lecturer can reject a string of words without such as strange combination, if the text is marked as modern poetry.

In Măranduc (2005) 15 experiments are described. The orthographic, phonetic, morphologic, syntactic, semantic rules affect the textual quality of a string only if the global meaning is affected by an intolerable ambiguity. The semantic and pragmatic rules must be conceived in accordance with the type of text, well defined by specific marks. The negative publicity is rejected. Test number 14 is estimated to be incorrect by the majority of the 50 tested subjects; students aged 12 to 18 years.

Evaluating the text from a pragmatic point of view means assessing how it fulfills the issuer's intentionality.

The Intentionality depends on the classification of the text in a certain type, which is made by its marks, sometimes internal, sometimes meta-textual (title, subtitle, cover, situationality).

The text below does not meet the intention of the issuer that is supposed to be to induce the purchase of the object, because the text is marked by exclamations and the initial imperative as an advertisement.

But if we reinterpret these marks as showing an ironic text, with the intention of ridiculing the advertisements, then it is correctly formed.

- Cumpărați ciocolata Dănuț!

Cine o va cumpăra va vedea că nu este bună pentru copii, căci e amăruie. Deoarece este adusă din Sighetul Marmației, este puțin mai veche. Ciocolata Dănuț te face prostuț !

- *(Buy the Chocolate “Dănuț”! Whoever will buy, will see that it is not good for the children, because it is bitter. It is brought from Sighetu Marmatiei, so it is a little older. The Chocolate “Dănuț” makes you goofy!)*

The considerations in this chapter are applicable of all types of texts. For the poetic text, an advised lecturer will reject a string without unexpected expressions, with 0% frequency in the language model. The intentionality of the poetic text is to convey a whole new experience, intense and fresh feelings, which the lector's sensitivity has never met.

Test number 7 is conceived for the investigation of rules for the lyrical transmission of a feeling. The following string is presented to the tested persons and the ennonce ask them to appreciate if it is a text and (if there are) to correct its mistakes:

- “Sâmburele îi curge din mână și ea nu vede acest zgomot. Apoi vede că sâmburele e amar și nu-l ridică. Aude că s-a făcut întuneric. Îți tace ceva și tu îi răspunzi. Fiind absent lângă tine, te-ar putea ajuta. Ușa îți citește gândul că acest copil îți e tată”.
- *(The kernel flows out of her hand and she does not see this noise. Then she sees that the kernel is bitter and does not pick it up. She hears that it has gotten dark. She is silent something to you and you answer her. Being absent next to you, he could help you. The door reads your thought that this kid is your father.)*

The way of creating a modern text is parodied. The expression: “the kernel flows” has the frequency 0% in a poetic language model, and the following expressions have the truth value 0 in the real world:

- to see a noise=0,
- to see that it is bitter=0,
- to hear that it was dark=0,
- she is silent something=0,
- absent next you=0,
- the door reads=0,
- the kid is your father=0.

But as many as the tested persons tried, they were unable to construct the text so as to have a global meaning, although it results in a kind of meditative state of the character “She”, which is absent in relation to the contingent real world.

The poetic text is bad formed because it has no final key for decoding a global meaning, the closing mark is absent. The tested persons, high school students, preparing for the exams studying modern lyrical texts, have attempted to eliminate illogical constructions with the value of truth 0 (which is not an adequate answer, the lyric text suspending the value of truth, as we have shown above) and declared the text to be incorrectly formed, because “it has no sense”, that is an adequate answer. But if the intention of the emitter was to create an absurd literary text, it is correctly formed!

6. Conclusions

The experiment of semantic induction is a model of the poetic reception, because the poetic text proposes to lecturers several such pairs of terms. Each PAD is interpreted in accordance with one of more inductive combinations of words found in the text above. As reads, the earlier ones direct the decoding of each new combination encountered. A new pair of terms that seems to be contradictory with the above ones may require the resumption of reception and the re-interpretation of the previous one(s), to which the lecturer is sent via feed-back.

If the reader encounters a combination of terms that violate his textual sense, he may abandon the reception and qualify the string of expressions as being a non-text. It happens frequently with the non-advised readers of modern poetry. The Intertextuality governs the Expectability.

The computer can detect the expressions with possible figurative meaning of a combination of words by calculating the frequency by rapport of a model of the language. But a human person can validate the result of this detection; a mistake (without emitter intention) can be evaluated in the same way. A lot of logical, mathematical, computational methods can be found to qualify a text as poetic. The automatic decoding of the global sense can also be approximated using the intersection of distinctive features.

The fillings of the emitter, in a poetical or non-poetical text can be approximate using the Senti(ment)WordNet, the network of dictionaries annotated with positive or negative fillings and their gradual intensity. However, for now, we cannot imagine any way of automatically generating metaphors, lyrical text, expressing feelings that the computer does not have, and requiring human talent and creativity.

References

- Beaugrande, R. A. de, and Dressler, W. U. (1984). *Text production. Toward a Science of composition*, Advances in Discourse Processes, New-Jersey, ALEX Publishing Corporation.
- Coteanu, I. (1973). *Functional Stylistics of the Romanian Language I, Style, Stylistics, Language*. Bucharest, Academy Publishing House.
- Esuli, A., Sebastiani, F. (2006). SENTIWORDNET: A publicly available lexical resource for opinion mining. In *Proceedings of the 5th Conference on Language Resources and Evaluation LREC*, Genoa, 2006, 417–422.
- Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*, Academic Press, Cambridge, MA.
- Maiorescu, T. (1878) Critical research on Romanian poetry from 1867. In *Works*, First volume, Bucharest, Minerva Publishing House, 5-69.
- Maning, C., and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*, The MIT Press, Cambridge, Massachusetts, London.
- Marcus, S. (1970) *Mathematical Poetics*, Bucharest, Romanian Academy Publishing House.

- Mărănduc, C. (2005). *Rules of Correct Text Formation – from a Pragmatic Perspective*, Bucharest, National Foundation for Science and Art Press.
- Mihăilă, E. (1972) Mathematical Methods in the Analysis of the Literary Text. In Rață-Dumitriu, N., Bratu, S., Sincu, Al., Mihăilă, E., Crețulescu, I., Șerban, I. V., Tudorică, Al., *Analysis and interpretation. Orientations in Contemporary Literary Critique* Bucharest, Scientific Publishing House.
- Mihăilă, E. (2001). *The Language of the Romanian Neomodern Poetry*, Bucharest, Olimp Publishing.
- Mihăilă, E. (2003). *The Semiotics of the Romanian Neomodern Poetry*, Bucharest, Cartea Românească Publishing House.
- Miyazawa, A., Miyao, Y., Evaluation Metrics for Automatically Generated Metaphorical Expressions. In *Proceedings of the 12th International Conference WCS 2017* - aclweb.org.
- Ovchinnikova, E., Zaytsev, V., Wertheim S., Ross Israel, R. (2014). Generating Conceptual Metaphors from Proposition Stores. In *ArXiv*, volume abs/1409.7619.
- Pang, B., Lee, L. (2008). *Opinion Mining and Sentiment Analysis*, Foundations and Trends in Information Retrieval, 2, 1–2, 1–135.
- Plett, H.F. (1983) *Text Science and Text Analysis. Semiotics, linguistics, rhetoric*, Bucharest, Universe Publishing House.
- Rață-Dumitriu, N., Bratu, S., Sincu, Al., Mihăilă, E., Crețulescu, I., Șerban, I. V., Tudorică, Al. (1972). *Analysis and interpretation. Orientations in Contemporary Literary Critique* Bucharest, Scientific Publishing House.
- Searle, J. R. (1981). The Logical Status of Fictional Discourse. In Borcilă, M. and McLain, R. (eds.), *American Poetics, Current Orientations*, Cluj-Napoca, Ed. Dacia.
- Shutova, E., (2010), Models of Metaphor in NLP. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 688–697, Uppsala, Sweden.
- Tuفیș, D., and Ștefănescu, D. (2011). Connotative Meaning Mark-up for WordNet. In *Proceedings of the Romanian Academy, Series A, Volume 12, number 2/2011*, 143-150.
- Valitutti, Al., Strapparava, C., and Stock, O. (2004). Developing Affective Lexical Resources. In *Psychology Journal*, 2, 1, 61–83.
- Vasiliu, E. (1990) *Introduction to Text Theory*, Bucharest, Scientific Publishing House.
- Wiebe, J., Willson, T., Bruce, R., Bell, M., Martin, M. (2004). Learning Subjective Language. In *Computational Linguistics*, 30, 3, 277–308.

REVISITING THE STATISTICAL INDEPENDENCE FOR THE PRINTED ROMANIAN LANGUAGE

ALEXANDRU DINU¹, ADRIANA VLAD^{1,2}, BOGDAN HANU¹, ADRIAN MITREA¹

¹*Faculty of Electronics, Telecommunications and Information Technology,
POLITEHNICA University of Bucharest,*

²*Research Institute for Artificial Intelligence “Mihai Drăgănescu”, Romanian
Academy*

*{alexandrudinu89, adrianmitrea}@yahoo.com,
avlad@racai.ro, hanu.bogdan@gmail.com*

Abstract

This paper revisits the notion of statistical independence for printed Romanian language, analysis on a literary corpus. The main goal is an improved perception of the concept of statistical independence for natural texts and of using this concept to evaluate the numerical properties of the printed language. The proposed analysis combines information theory and statistical instruments (mutual information, noise matrix, Chi-square test in contingency tables). The Chi-square test in contingency tables offers a global answer to the statistical independence problem and it requires a careful collection of experimental data to ensure the condition for the number of occurrences and for data complying with the statistical i.i.d. model (data coming out from identically and independently distributed random variables). The other tools used in the analysis are based on the mutual information concept (information channel, conditional probabilities and noise matrix, implicitly). The ergodicity feature of the printed language allows a better use of the corpus when using the mutual information compared to the case when the Chi-square test is used. Another topic that was investigated was the application of the two strategies described above when the corpus contains orthography and punctuation marks (alphabet of 47 symbols compared to 32 symbols as used in the first part of the analysis).

Key words — Chi-square test in contingency tables, mutual information, natural language statistical description, statistical independence sampling distance

1. Introduction

This paper is the follow-up of a larger study dedicated by the authors to the description of printed Romanian language as an information source (Vlad *et al.*, 2003). Here, the statistical investigation aims at obtaining an answer regarding the minimum statistical independence distance for printed Romanian language. To come out with an accurate result, the authors used concepts from the information theory field like mutual information, information/noise channel and conditional probabilities. Furthermore, the

Chi-square test in contingency tables was used to provide an alternative perspective on the topic of statistical independence and to speed up the analysis. The new experimental results, regarding the search for a minimum statistical independence sampling distance for printed Romanian, refer to *m-gram* entities (illustrations for letter, digram and trigram statistical structures) in a linguistics corpus (Vlad *et al.*, 2013) of 49 books written by 9 authors (Isaac Asimov – 9 books; Constantin Chiriță – 5 books; Alexandre Dumas – 12 books; Colin Falconer – 1 book; Frank Herbert – 8 books; Niven Larry – 1 book; Orson Scott Card – 3 books; Michel Zevaco – 7 books; J. R. Tolkien – 3 books). The length of the corpus including orthography and punctuation marks is 36 898 320 characters (over 6 million words). The books were randomly concatenated to form the corpus.

The study of the statistical independence distance was conducted first on the corpus version without orthography and punctuation marks (35 191 963 characters). In this version, the corpus alphabet consists of 32 symbols: 31 symbols are the upper case letters composing the Romanian alphabet, A Ă Â B C D E F G H I Î J K L M N O P Q R S Ș T Ț U V W X Y Z, and the space (blank). For the corpus with orthography and punctuation marks (section 3.3), the extended alphabet contains 15 additional symbols (hyphen, full stop, comma, colon, semicolon, question mark, exclamation mark, quotation dash, em dash, abbreviation point, ellipsis, quotation marks, parentheses and apostrophe). In what follows, we refer to the elements of the alphabet as being letters, symbols or characters, either proper Romanian language letters or orthography and punctuation marks (Vlad *et al.*, 2013).

In the previous research, it was assumed that a distance of around 200 characters ensures statistical independence. This hypothesis has not been fully tested until now but the indirect results from previous papers supported its validity. The main goal of this paper is to validate the above-mentioned statement and to provide a more accurate answer concerning what the minimum statistical independence distance means for a natural language. Furthermore, a second aim of the paper is to provide a more intuitive understanding of the concept of statistical independence for a natural text, by adding visual representations of the conditional probabilities and the error matrix.

2. Data collection and theoretical aspects

2.1. Data collection

In this paper, we shall illustrate the procedure carried out to evaluate the sampling distance that ensures statistical distance for *m-grams*.

An *m-gram* is a group of *m* successive letters. For example, in the sentence “**ACUM** **MERGEM** **ACOLO**”, *ACU* is an example of a trigram and *LO* is a digram.

When one talks about statistical independence, the main question is: what is the distance/number of characters between an *m-gram* and the next one, so that these two text items are not dependent/related to each other? In other words, knowing the first *m-gram*₁, does this convey any information on what the *m-gram* at *d* characters away is going to be (*m-gram*₂)? If the answer is No, the independence is achieved. Note: the

m -gram at d characters away means that the difference between the first letter of m -gram₁ and the first letter of m -gram₂ is exactly d .

This can also be written in terms of conditional or joint probabilities as in (1) and (2):

$$P(m\text{-gram}_2 | m\text{-gram}_1) = P(m\text{-gram}_2) \quad (1)$$

$$P(m\text{-gram}_1, m\text{-gram}_2) = P(m\text{-gram}_1) \cdot P(m\text{-gram}_2) \quad (2)$$

Table 1: Example of digrams and trigrams in the natural text

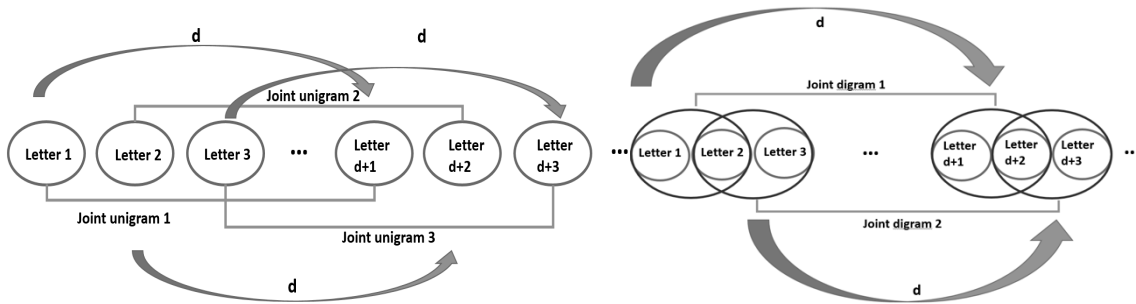
Text	A	C	U	M		M	E	R	G	E	M		A	C	O	L	O
Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

Comment on Table 1: If statistical independence is reached at a distance of 10 characters, the probability that letter M comes 10 characters after letter A (in fact, after any letter) is the same with the probability of letter M occurrence: $P(M|A) = P(M)$ – letter independence; similarly, $P(\text{Blank A}|CU) = P(\text{Blank A})$ – digram independence and $P(M \text{ Blank A}|ACU) = P(M \text{ Blank A})$ – trigram independence. This is valid for m -grams of any size (m stands for the size) but due to computation and text limitations, only the letter, digram and trigram behaviour has been investigated ($m = 1, 2, 3$ respectively).

2.2. The information theory track

The first approach used to assess the statistical independence distance is derived from the information theory field. We shall next describe the general procedure followed. This was applied to letters, digrams and trigrams.

The m -grams (letters, digrams and trigrams) are selected from the natural text, as it is shown in Fig. 1 and Table 1. This use of the text (overlapping and meshing m -grams) is possible based on previous ergodicity results (Vlad *et al.*, 2003; Vlad *et al.*, 2013). The distance d between the m -grams is a parameter in the applied statistical procedure.



(a) (b)

Figure 1: Letter and digram collection from the Romanian corpus, without jump

The next step is to create a table that contains the number of occurrences for every investigated pair $(m\text{-gram}_1, m\text{-gram}_2)$ – the two $m\text{-grams}$ being separated by distance d , the quantity we are trying to estimate. Fig. 2 shows the number of occurrences/ contingency table when letters are analysed and the order of the letters is from the highest to the lowest probability (top-down and left-right).

For example, n_{21} is the number of occurrences in the corpus for the pair (E, Blank) with d letters between them. That corresponds to the information/noise channel assigned to the d investigated distance, with the $m\text{-grams}$ on the rows being input and the ones on the columns being the output of the channel (Shannon, 1951). At the same time, the number of occurrences for the individual $m\text{-grams}$ can be computed based on the entire corpus (Vlad *et al.*, 2003; Vlad *et al.*, 2013). By dividing the two quantities, one can obtain an estimate for the conditional probability of a certain combination of $m\text{-grams}$ separated by distance d (see (3)).

Letter 2 Letter 1	Blank	E	...	W	Q
Blank	n_{11}	n_{12}	$\dots$	$n_{1,k-1}$	n_{1k}
E	n_{21}	n_{22}	$\dots$	$n_{2,k-1}$	n_{2k}
.	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$
.	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$
.	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$
W	$n_{k-1,1}$	$n_{k-1,2}$	$\dots$	$n_{k-1,k-1}$	$n_{k-1,k}$
Q	n_{k1}	n_{k2}	$\dots$	$n_{k,k-1}$	n_{kk}

Figure 2: Occurrence number for letter pairs separated by d distance

$$P(m\text{-gram}_2 | m\text{-gram}_1) \cong \frac{\text{Occurrence number pair } (m\text{-gram}_1, m\text{-gram}_2)}{\text{Occurrence number } m\text{-gram}_1} \quad (3)$$

When (1) and (2) are met, (they are equivalent), independence is achieved and the mutual information value becomes zero. When the statistical independence is achieved, on each column in Fig. 2 we shall find the same value that is equal to the occurrence probability of the header character occurrence. For instance, the entire column 1 will have the same value, namely $P(\text{Blank})$; the entire column 2 will have the same value as the occurrence probability of letter E, namely $P(E)$.

Only the $m\text{-grams}$ that fulfil the condition for the number of occurrences (which is expressed by (4) and related to the De Moivre – Laplace condition) will be considered in the probability evaluations (Vlad *et al.*, 2003; Frieden, 1983).

$$\text{Occurrences for the pair } (m\text{-gram}_1, m\text{-gram}_2) > 5 \quad (4)$$

One quick step to assess whether the independence is achieved is just by visually checking the conditional probability table: when the same value is on each column (same colour, in a colour bar representation), the mutual information is 0 and the independence is achieved.

A more mathematical evaluation involves the concept of error matrix associated to the information channel. The deviation of each conditional probability from the expected value is denoted in (5). When independence is reached, we expect to have the entire error matrix close to 0.

$$\text{Error}_{ij} = |P(m\text{-gram}_j|m\text{-gram}_i) - P(m\text{-gram}_j)| / P(m\text{-gram}_j) \quad (5)$$

To summarize, this approach was followed for the study of statistical independence for letters, digrams and trigrams. The matrix of occurrences was computed for different distances. Next, the conditional probabilities were calculated based on (3). The final step is represented by the computation of the error matrix, which is evaluated based on (5). The experimental results are presented in section 3.

2.3. The Chi-square test in contingency tables

The investigation based on mutual information described in the previous subsection has the advantage that alongside with the mathematical aspect, we can also provide a visual representation for the perception of statistical independence concept (see Fig. 7).

Chi-square test in contingency tables is a useful tool also when the numerical evaluations concerning statistical independence are made based on the information channel and mutual information. This test allows a global view of the situation illustrated in Fig. 2 (implicitly of the associated noise/error matrix) and, together with the mutual information approach, it leads to a faster answer concerning the statistical independence distance.

One prerequisite of the Chi-square test is its known sensitivity and need for i.i.d. data - data coming out from identically and independently distributed random variables (Craiu, 1972). In order to ensure this and based on previous results of the authors (Vlad *et al.*, 2001; Vlad *et al.*, 2003; Vlad *et al.*, 2013), a jump needs to be performed in the corpus before selecting the next pair of *m*-grams for analysis. In other words, the corpus will not be fully used because after a pair of *m*-grams is selected as shown in Fig. 1, the next pair is selected after 200 characters (jump value) from where the first *m*-gram is located. Fig. 3 depicts the procedure followed in order to collect i.i.d. data for the Chi-square test.

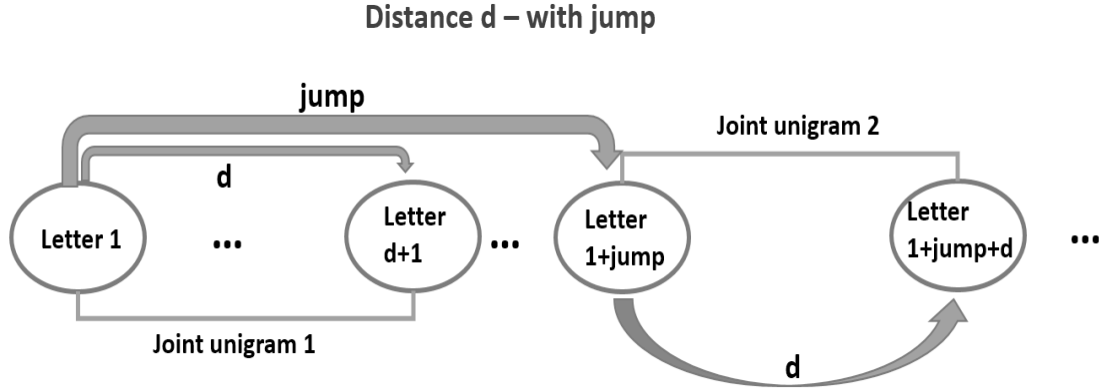


Figure 3: Performing a jump to collect the i.i.d. data set for the contingency table

The Chi-square test mainly uses, as the name implies, the contingency table, i.e., the matrix with the number of occurrences depicted in Fig. 2. Note: The n_{ij} values obtained now will be lower than the corresponding ones in Fig. 2 (due to the reduced size of the data set volume after the jump, *the total corpus length/jump value*). The condition for the number of occurrences applies here, too – see (4). When the condition for the number of occurrences is not met, the corresponding rows and columns in the table are concatenated into an artificial symbol, thus reducing the size of the contingency table. For example, in the case of letters, where the corpus alphabet has 32 symbols, the reduced matrix has a lower number of symbols, namely 25. The algorithm to reduce the contingency table and prepare it for the test is depicted in Fig. 4.

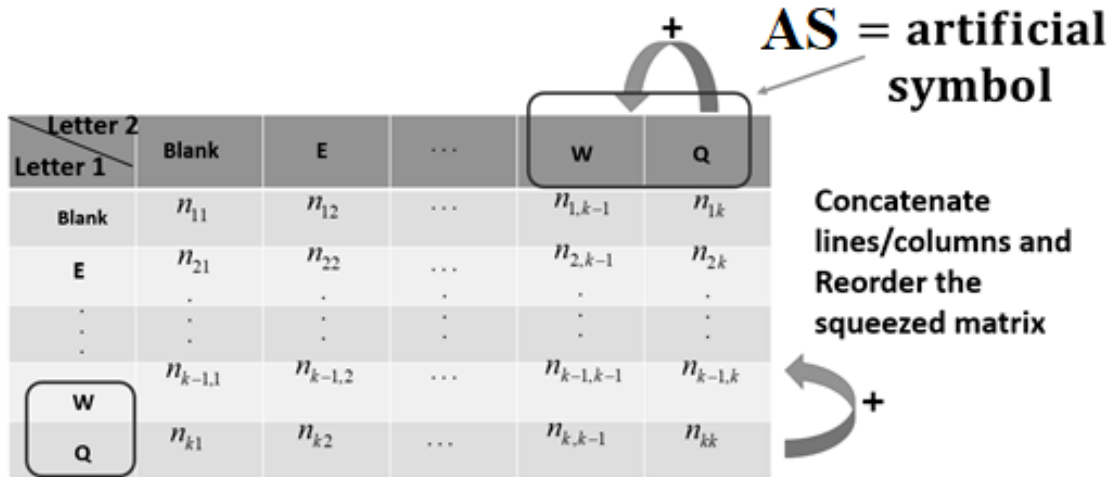


Figure 4: Reducing the size of the contingency table for the Chi-square test

Once the data is properly collected with a sufficiently large jump (see Fig. 3) and the procedure from Fig. 4 is followed, the test can be applied on the reduced contingency table. The algorithm is straightforward: for every distance d , a test value is computed

with (6) and it is compared to an accepted/admissible value (the α -quantile of a χ^2 distribution with $(k-1) \cdot (k-1)$ degrees of freedom, denoted by z_α).

$$z = \sum_{i=1}^k \sum_{j=1}^k (n_{ij} - \frac{n_i \cdot n_j}{n})^2 / (\frac{n_i \cdot n_j}{n}) \quad (6)$$

If $z < z_\alpha$, the Chi-square test passes and the independence is achieved; otherwise, the test fails and the next distance is checked (Craiu, 1972; Walpole *et al.*, 2016; Devore, 2008).

3. Experimental results

For the experimental section, we shall switch the order and start with the Chi-square approach, which gives faster results to be used as starting point for the statistical independence distance evaluation by means of mutual information.

As we shall see, the Chi-square test is less accurate than the information theory approach. Thus, it allows us to skip over analysing the d distances based on mutual information, when the Chi-square test in contingency tables fails (independence is not met for the respective d value).

3.1. The Chi-square test in contingency tables

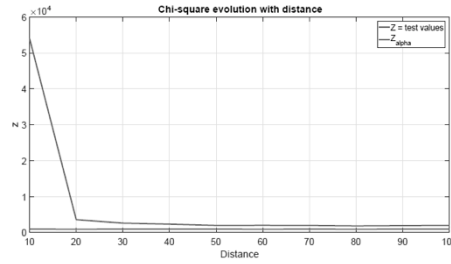


Figure 5: Chi-square test results without a jump (d distance – horizontal axis; z test value – vertical axis, see (6))

Fig. 5 shows the behaviour of the Chi-square test when the data is wrongly collected, without performing a jump. In this case, the experimental data are not i.i.d. and the test does not pass, no matter how large the d distance value is. Working with i.i.d. data is a prerequisite for the Chi-square test and that is why it becomes of paramount importance to collect the data correctly.

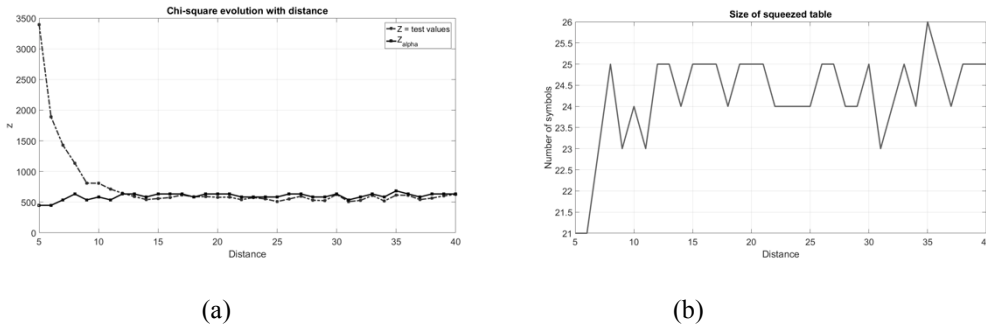


Figure 6: Chi-square test results for letters

Fig. 6 shows the results obtained for letters with a jump of 200 letters. Fig 6.a depicts the evolution of the test value z against z_α for different distances. Fig. 6.b shows the size of the reduced occurrence matrix (which complies with the De Moivre – Laplace condition). Note that z_α varies with distance because it depends on the size of the reduced matrix. The conclusion is that independence is achieved for letters at distance $d = 13$. The size of the reduced matrix is 25 (for $d = 13$) and these 25 most frequent characters in the corpus, for which the De Moivre – Laplace condition is met, cover 98.68% of the text. The results obtained for digrams and trigrams in Table 2 can be interpreted similarly.

Table 2: Chi-square test results for m-grams

<i>m-gram</i>	Minimum distance when the test passes	Reduced matrix size	Corpus coverage
Letters	13	25	98.68%
Digrams	18	67	61.02%
Trigrams	15	26	13.06%

An important conclusion is that the Chi-square test offers a quick answer on the minimum distance from which the detailed analysis based on mutual information starts. For example, in the case of digrams, we will analyse the results at distances higher than 18 because, based on the results from Table 2, independence will not be reached earlier. The conclusion is that by applying the Chi-square test in contingency tables, the statistical independence distance is obtained quite fast. It supports the results from previous studies of the authors (Vlad *et al.*, 2001; Vlad *et al.*, 2003; Vlad *et al.*, 2013), where a distance of 200 characters was considered large enough to guarantee statistical independence. Secondly, the number of *m-grams* for which the condition for the number of occurrences is met decreases (compared to the total number of *m-grams*) when the size of the *m-gram* increases. This is a direct consequence of the limited size of the corpus. Nevertheless, the coverage is quite decent for letters, even for digrams (example: 98.68% of the corpus length consisted of the first most frequent 25 letters).

3.2. The information theory approach

For the information theory approach (Marcus, 1995), the conditional probabilities matrix and the associated error matrix (calculated based on (3) and (5)) are of paramount importance. Fig. 7a and Fig. 7b show that as the d distance between the *m-grams* (letters, in this case) increases, the conditional probabilities (top pictures) begin to look very much like the unconditional probabilities (bottom pictures) – constant values for every column, and according to (1) this is a sign that independence is being achieved.

REVISITING THE STATISTICAL INDEPENDENCE FOR THE PRINTED ROMANIAN LANGUAGE

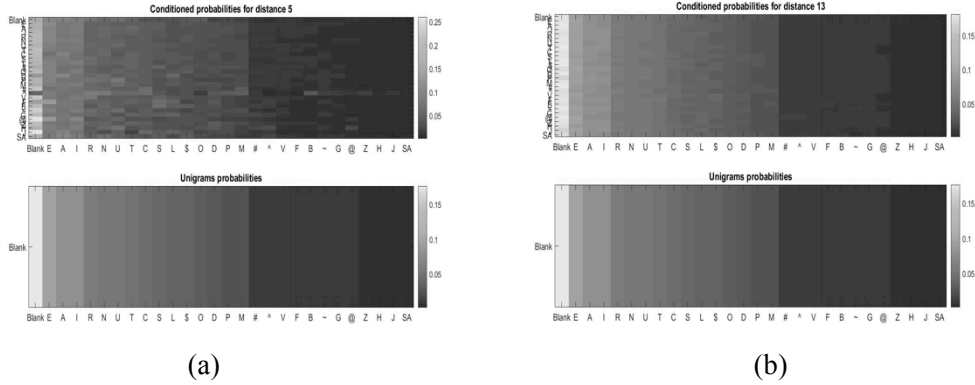


Figure 7: Conditional probabilities and error matrix for letters at distances 5 (a) and 13 (b)_de verificat

Table 3: Conditional probabilities for letters(%) at distance $d = 13$

Letters/ Probabilities	Blank	E	A	...	AS	...	$\hat{A}$
Blank	17.56	9.40	8.13	...	2.61	...	0.80
E	17.75	9.70	8.02	...	2.54	...	0.73
A	17.79	9.54	8.08	...	2.52	...	0.72
...	...	...	...	...	...	...	...
AS (Artificial symbol)	17.53	9.48	8.30	...	2.63	...	0.79
...	...	...	...	...	...	...	...
$\hat{A}$	18.03	9.73	7.92	...	2.38	...	0.86
Letter probabilities	17.63	9.54	8.09	...	2.57	...	0.75

Table 3 depicts the conditional probabilities for comparison with the individual letters probabilities shown in Fig. 7.b for distance $d = 13$. The two quantities should be equal when independence is reached. One can notice that the equality between the conditional and individual probabilities is much more visible for the most frequent symbols (top left) and the highest errors are for the less frequent symbols (bottom right).

Comment on Table 3: The conditional probability that letter A comes 13 characters after letter E is 8.02% (we started the analysis with $d=13$ based on the conclusions from section 3.1). AS is the artificial symbol obtained by concatenating the least 8 frequent characters of the alphabet for which the occurrence number does not meet (4).

The data from Table 3 were further used to compute the error matrix by means of (5). The result is visually shown in Fig. 8. When the statistical independence is reached, the error should be 0 and this corresponds to a black image.

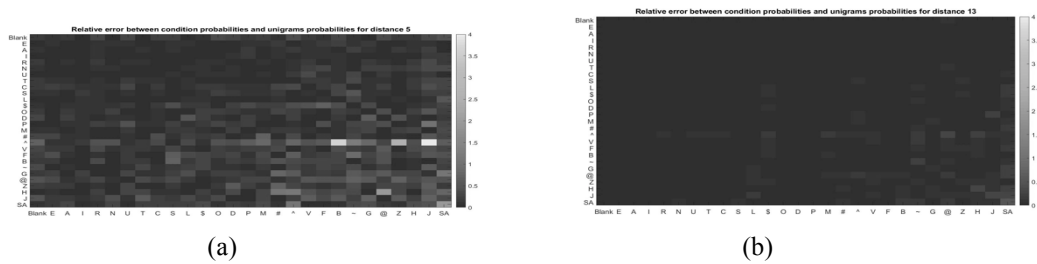


Figure 8: Error matrix for letters for distances 5 (a) and 13 (b)

Note that if at distance $d = 5$ (Fig. 8.a) there are still some high errors between the conditional and the individual probabilities (different shades of grey), once the distance reaches the value $d = 13$ (Fig. 8.b), the errors are much smaller. Fig. 8 becomes more understandable if one has a closer look at the error matrix.

Table 4: Relative error assigned to letter conditional probability by using (5), at $d = 13$ distance (%)

Letters/ Error	Blank	E	A	...	AS	...	$\hat{A}$
Blank	0.38	1.50	0.48	...	1.68	...	7.16
E	0.67	1.61	0.79	...	0.99	...	2.20
A	0.95	0.03	0.15	...	1.76	...	3.34
...	...	...	...	...	...	...	
AS (Artificial symbol)	0.57	0.62	2.63	...	4.94	...	5.29
...	...	...	...	...	...	...	
$\hat{A}$	2.28	1.93	2.10		7.34		15.39

Comment on Table 4: The relative error assigned to the conditional probability for letter A coming 13 characters after letter E is 0.79%.

The data shown in Fig. 8 can be better understood if one looks at the top 5 highest conditional probability errors for letters at distance $d = 13$ and their evolution as the distance increases until $d = 100$ (see Table 5).

Table 5: Top 5 conditional probability errors for letters (%) at distance $d = 13$ and their evolution with increasing distance

Pair	d=13	d=40	d=60	d=80	d=100
J J	28.54	26.26	15.61	16.90	12.03
H H	27.06	20.03	21.36	20.07	15.32
$\hat{I}$ $\hat{A}$	26.62	5.09	3.77	9.47	2.36
$\hat{I}$ H	24.77	11.33	9.00	2.48	6.36
T T	22.98	12.89	8.25	1.27	9.14

How to read Table 5: The relative error between $P(\hat{I} | H)$ (the conditional probability for letter $\hat{I}$ coming 13 characters after letter H) and the probability of letter $\hat{I}$ occurrence is 24.77%. This relative error probability becomes 9% for distance $d = 60$.

Similar results were obtained for digrams and trigrams. The difference is that the errors are slightly bigger than for letters. The errors are decreasing while the distance is increasing (see Table 6 and Table 7).

REVISITING THE STATISTICAL INDEPENDENCE FOR THE PRINTED ROMANIAN LANGUAGE

Table 6: Top 5 conditional probability errors for digrams (%) when increasing distance

Pair	d=20	d=40	d=60	d=80	d=100
Blank N NI	46.43	10.05	2.38	1.81	0.21
Blank N Blank N	44.07	31.06	16.33	13.79	13.84
TI TI	41.66	27.95	29.68	20.5	19.64
Blank V Blank V	36.44	42.35	33.14	29.43	28.53
Blank O Ă Blank	32.06	2.86	3.46	2.21	0.34

Table 7: Top 5 conditional probability errors for trigrams (%) when increasing distance

Pair	d=20	d=40	d=60	d=80	d=100
LE Blank LE Blank	61.8	37.58	31.34	22.09	16.37
Blank SE UL Blank	16.22	0.22	8.77	4.2	9.73
Blank SĂ UL Blank	44.51	5.68	4.19	2.59	0.71
Blank SĂ LE Blank	38.15	1.67	7.61	0.48	1.04
Blank SE IN Blank	36.95	9.07	11.51	10.29	8.67

3.3. Experimental results for the extended alphabet of the corpus

The Chi-square test and the information theory approaches have also been applied to an extended version of the literary corpus. The difference was that instead of 32 letters, punctuation and orthography marks have also been considered. We illustrate the results only for the letter case for the extended alphabet corpus.

Table 8 shows the Chi-square test results. The independence distance is just slightly bigger for the 47-symbol-corpus compared to the 32-symbol-corpus. For only 29 out of 47 symbols, the condition for the number of occurrences is fulfilled and these 29 letters cover more than 98% of the corpus.

Table 8: Chi-square test results for the 47-symbol corpus

m-gram	Independence distance	Reduced matrix size	Corpus coverage
Letters	15	29	98.55%

Table 9: Conditional probabilities for letters (%) at distance $d = 15$

Letters/ Probabilities	Blank	E	A	...	AS	,	...	.	...	Z
Blank	17.04	9.00	7.75	...	1.75	1.33	...	1.03	...	0.53
E	17.25	9.24	7.75	...	1.85	1.33	...	1.05	...	0.53
A	17.49	9.13	7.77	...	1.88	1.34	...	1.05	...	0.53
...	...	...	...	...	...	...	...	...	...	...
AS (Artificial symbol)	17.53	8.75	7.54	...	2.31	1.58	...	1.05	...	0.55
, (Comma)	17.14	8.80	7.82	...	2.33	1.52	...	0.98		
...	...	...	...	...	1.71	...	...	...	...	...
. (Full stop)	17.52	8.93	7.48	...	1.69	1.59	...	0.74		

...	...	...	...	...	...	...	...	...	...	...
Z	17.29	9.29	7.67	...	1.87	1.39	...	1.21	...	0.57
Letter probabilities	17.28	9.10	7.71	...	1.85	1.37	...	1.06	...	0.53

Comment on Table 9: The conditional probability that letter A comes 15 characters after letter E is 7.75%. The additional symbols, which were analysed for the extended alphabet of the corpus (with orthography and punctuation marks), are comma, full stop, em dash and letter Z.

4. Conclusions

This paper combines the mutual information concept and the Chi-square test in contingency tables under a new procedure, aiming to reveal the meaning of statistical independence in the natural text and a modality to evaluate the minimum statistical independence distance concerning *m-grams* extracted from a corpus/natural text. According to our knowledge, the information has not been analysed this way before. In addition, this research contributes with a more intuitive understanding to concepts that the authors previously studied and which could benefit other researchers' understanding via the visual tools and techniques introduced.

The results presented in this paper confirm the idea that a distance of 200 characters for the Romanian printed language is large enough to ensure statistical independence between *m-grams* extracted/sampled from the natural text. In addition, this sampling distance is further analyzed by means of information theory concepts and the Chi-square test in contingency tables, and information about the minimum distance ensuring independence is presented.

This investigation revealed that even though the Chi-square test is faster and it gives a first information that independence is obtained at a much smaller distance (of about 20 characters for letter case investigation), the approach by mutual information is more accurate. The results obtained via the mutual information and error matrix track increase the accuracy of estimating the statistical independence distance to approximately 80 characters (taking into account letter, digram and trigram investigation for a 32-symbol alphabet).

Furthermore, the 47-symbol extended version of the corpus alphabet is also analyzed and the independence is also reached in less than 100 characters. Further research might look into the analysis of statistical independence in the case of digrams and trigrams for the 47-symbol alphabet and similar results are expected. Secondly, another investigation track could be analyzing the statistical independence for the natural text by using words instead of *m-grams*.

References

- Craiu, V. (1972). *Verificarea Ipotezelor Statistice*. Editura Didactica și Pedagogica, Bucuresti, Romania.
- Devore, J.L. (2008). *Probability and Statistics 7th (Seventh) Edition*. Duxbury Press.
- Frieden, B.R. (1983). *Probability, Statistical Optics and Data Testing*. Springer Verlag, Berlin – Heidelberg – New York.
- Marcus, M. (1995). New trends in natural language processing: statistical natural language processing. In *Proceedings of the National Academy of Sciences*, 92(22), 10052-10059.
- Shannon, C.E. (1951). Prediction and Entropy of Printed English, *Bell Syst. Tech. J.*, vol. 30, 50-64.
- Vlad, A., Mitrea, A., Mitrea, M. (2001). The trigram statistical structure in printed Romanian, In *ROMJIST (Romanian Journal of Information Science and Technology)*, vol. 4, no. 3, 353-372.
- Vlad, A., Mitrea A., Mitrea, M. (2003). *Limba romana scrisa ca sursa de informatie*. Editura Paideia, Bucuresti, ISBN 973-596-185-7.
- Vlad, A., Mitrea, A., Luca, A., Hodea, O. (2013). Considerations Regarding the Statistical Compatibility of Two Romanian Literary Corpora with Orthography and Punctuations Marks Included. In Dan Tufis, Vasile Rus, Corina Forascu (eds.), *Towards Multilingual Europe 2020: A Romanian Perspective*, The Publishing House of the Romanian Academy, 99-122.
- Walpole, R., Myers, R., Myers, S., Ye, K. (2016). *Probability and Statistics for Engineers and Scientists, MyLab Statistics Update (9th Edition)*, Pearson.

CHAPTER 4

DIACHRONIC STUDIES

SEMI-AUTOMATIC WORD LEVEL ALIGNMENT FOR RHOTACISING PSALTERS IN ROMANIAN

ANA-MARIA GÎNSAC¹, MARIA MORUZ¹, MIHAI ALEX MORUZ²,
MĂDĂLINA UNGUREANU¹

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

² *Faculty of Computer Science, “Alexandru Ioan Cuza” University of Iași, Romania*

{*anamaria.gansac, mhusarciuc, mandronic*}@gmail.com
mmoruz@info.uaic.ro

Abstract

In the field of Romanian paleolinguistics, which, by necessity is based on translations of religious texts from the 16th and 17th centuries, it is becoming increasingly important to develop an electronic corpus that could be used as a basis for comparative linguistic studies. The Psalter has an important textual tradition in old Romanian, as there are at least 16 translations from that period, many of them still having unidentified sources; also, the relationship between these texts has not been exhaustively studied, because of the great difficulty of managing such large amounts of information without electronic tools. Therefore, there becomes apparent the need for a tool to aid in digital alignment that would allow comparative linguistic studies. To this end, we have chosen three Psalters, the oldest documented for Romanian, and for which previous studies have shown that they share the same base translation: the Voronețeană, Scheiană and Hurmuzaki rhotacising Psalters, so-called because they retain an old phonetic phenomenon, the passage of intervocalic *n* to *r*. From these we have extracted and transcribed into modern Latin alphabet tree psalms (113, 138, 142); these were subsequently automatically aligned at the word level, and then manually validated. The present paper describes the manner in which the alignment was obtained, verified and then utilized for determining lexical, syntactic and source similarity.

Key words — automatic word alignment, rhotacising Psalters, old Romanian approach.

1. Introduction

In the field of the study of the old Romanian language, there has been a growing need for parallel texts aligned at the word level. These are necessary for various philological historical-linguistic studies, such as finding the source of a translation, the provenance of distinct texts, identifying the changes in word order, lexicon and syntax, etc.

Lately, there have been endeavors concerned with the digitalization of old Romanian texts, consisting mainly of scanning and online publishing. The *Medievalia*¹ project

¹ <http://medievalia.com.ro/>

makes available facsimiles for 60 texts from the 16th – 18th centuries; *Dacoromanica*², provides access to facsimiles of books from the collection “*Carte veche tipărită pe teritoriul României (1508-1830)*”, without any processing or tools; *Electronic Corpus of The Old Romanian Texts: 1521-1640* (CETRV)³, contains a significant number of texts, together with a search engine for finding words.

The Psalter and Gospel Book are among the first texts translated into Romanian (the 16th century), which, together with a large number of versions, shows their importance in Christianity. Among the old Romanian Psalters there are relations that have not been fully determined and ramifications that continue to the modern age. The rhotacising Psalters are the first known translations of the Psalter into Romanian and have similar sources. Studies have led to the supposition that the three texts have as a common source one translation into Romanian, which is not preserved; from this common translation, and following successive revisions and comparisons with versions in Slavonic, and, possibly, Latin and Hungarian, the three rhotacising Psalters (see below) were created. Therefore, it is interesting to determine the degree to which they are similar and the types of differences. However, these issues are difficult to examine with classical philological tools, given the size of the corpus. This may be the reason for which investigations have not gone beyond hypotheses (see below) and small verifications. A tool that offers the possibility of aligning the text at the word level and allows for simultaneous viewing would be useful for comparative research, increasing efficiency.

We therefore propose to test the possibility of aligning the rhotacising Psalters, transcribed from Romanian Cyrillic into Latin alphabet. This aligning process builds upon research described in (Moruz *et al.*, 2012), which also concerns alignment of parallel texts, the 1688 Bible and two manuscripts (ms. 45 and ms. 4389) of the time (see the MLD series), which had been previously aligned at the verse level, lemmatized and morphologically analyzed. The novelty of the current approach comes from the nature of rhotacising texts, which makes alignment much more complicated, as the previously used metrics yield poor results, and the lexicon built at that time does not contain many of the words encountered at present. To this end, we have expanded on the previous system by using new similarity metrics, described in section 4, which yield good results even on non-lemmatized text.

For testing the alignment tool, we have chosen three psalms (113, 138, 142); this is because of the fragmented nature of one of the manuscripts (we have chosen psalms that are complete in all three texts and are large enough to have sufficient comparable material) and the different circulation of each psalm (Ps. 142 is better known, because of its frequency in liturgical texts and hymnography).

2. Corpus Description

Some of the oldest available translations of the Psalter into Romanian, the “rhotacising Psalters” (so-called because of the phenomenon of transforming intervocalic *n* into *r*, sometimes by the intermediate phase *nr*, in words coming from Latin) date from the 16th

² <http://digitool.dc.bmms.ro:8881/R>

³ www.textvechi.ro

SEMI-AUTOMATIC WORD LEVEL ALIGNMENT FOR RHOTACISING PSALTERS IN ROMANIAN

century and are the basis for other 16th century Romanian Psalters (“Coresi Psalters”). For a long time, these Psalters were studied by linguists with regards to their geographical location, date of creation, whether they were copies or originals translations, the identity of the translator/ copier or their source. There are three such Psalters, all in manuscript (Fig. 1):

a) H = Hurmuzaki Psalter (ms. rom. 3077 B.A.R.), named after the person that donated it to the Library of the Romanian Academy, has 134 pages, of which 125 contain the text of the Psalter and the final 9, added at a later date, contain the ordinary of the gospels in Slavonic; the text is not complete, and is missing some psalms and parts of psalms; the most probable date is before 1516, possibly even as early as late 15th century (Mareş, 2001:51, on the basis of filigree analysis); it is the only one of the rhotacising Psalters that has been edited (Gheţie and Teodorescu, 2005);

b) V = Psalter from Voroneţ (ms. rom. 639 B.A.R.), named after the monastery in which it was discovered, dated to between 1551 and 1558 (Mareş, 2001:51); the remaining manuscript starts at psalm 77, and the available text is not complete; it is a bilingual Slavonic – Romanian Psalter, with interleaved text;

c) S = Scheiană Psalter (ms. rom. 499 B.A.R.), named for the donor of the manuscript, dated to between 1573 and 1578 (Mareş, 2001:51); it has been edited twice, transcribed and published with facsimiles (Bianu, 1889) and with reconstructed text (Candrea, 1916).

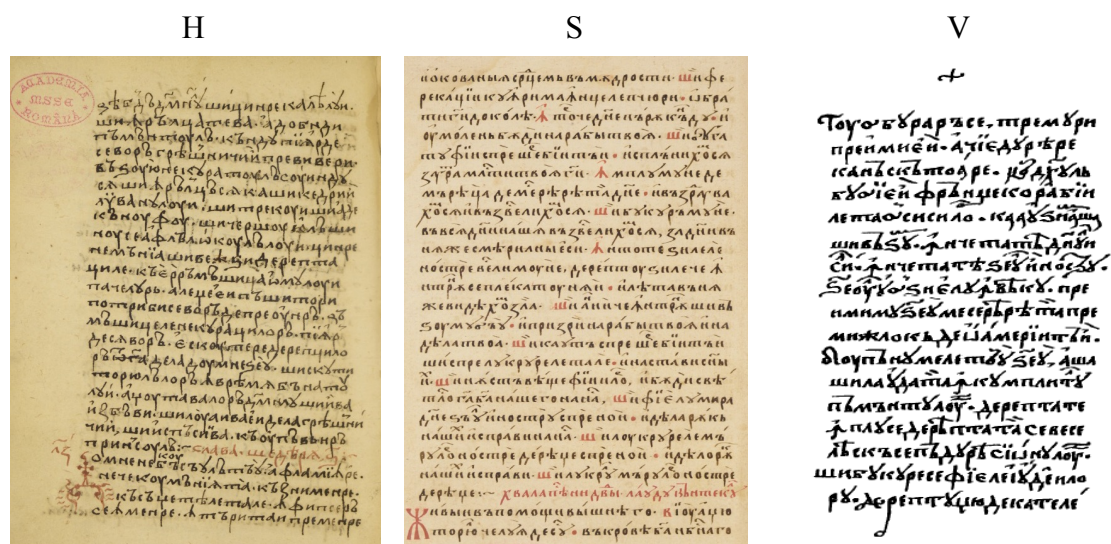


Figure 1: Facsimiles for the rhotacising Psalters

None of these Psalters contain any information regarding the source used for translation, and, as such, there have been a number of hypotheses in this regard. Mareş (1982, 2005) and Gheţie (1982) have proven, using lexical and syntactic arguments that the source of the translation is Slavonic; the exact source has not been identified (see the hypotheses of Mareş, 2005:276-277).

The complex relations between the early Romanian Psalters have recently generated new hypotheses with regards to the source of translation. On the basis of some Latin

origin words in S (e.g. *mesereare*, *a deșidera*, *a deștinde*; cf. Latin *miserere*, *desiderare*, *descendere*), which are thought to have been kept from the old translation from Latin, Chițimia (1981) supposes the existence of an old Latin source for the very early translations, before Slavonic being imposed as the language of culture (14th to 15th centuries); the resulting text would have been revised using a Slavonic version, and the words mentioned above replaced with Slavonic equivalents of the time. Munteanu (2008:128 *et sqq.*) agrees that a Latin source was used during the writing of the Romanian version “exclusively or at the same time as a Slavonic version”, influenced by Catholic and Calvinist Romanian communities in Banat-Hunedoara; he mentions sequences in S which correspond to Latin text (*Psalterium Romanum*), as well as some words that could only have been used under the influence of the Latin source (e.g. *a deșidera* for Latin *desiderare*; *a deșcinde* for *descendere*). Regarding the possible Latin source, Pavel (2013:27) shows that the more probable version is *Psalterium Gallicum*. Moldovanu (2009:58) thinks that a Hungarian source might have been used, as the Psalter was fundamental for the protestant community, and less so for the Orthodox, and so a translation would not be justified. The Slavonic aspect of the texts would be due to subsequent revisions, as was the case for other texts, such as Apostles and Gospels.

The possible relations between rhotacising Psalters are difficult to determine because of the numerous copies and intermediate revisions (Candrea, 1916:LXXX). Mareș (2005:279) does prove that the three Psalters come from the same early translation from Slavonic (which has been lost), supported by the proliferation in all three versions of the same translation errors, as can be seen in Ps. 104:32, where the Slavonic *gradŭ* was translated *cetate* (S), because of a confusion between homonyms *gradŭ* “city” and *gradŭ* “hail”, cf. lat. *grando* “hail”, *grindină* (H). Mareș (1982:260), however, places V and H on the one side, and S on the other, in different groups (Fig. 2).

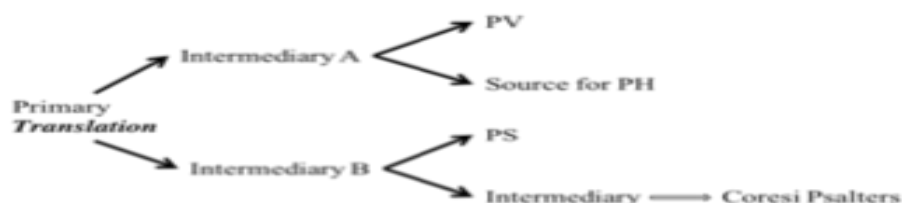


Figure 2: Relations between Psalters, Mareș 1982

The current interest for the source/sources of the early Psalters in Romanian proves the necessity of an exhaustive comparative study of the texts, both with regards to each other and with the supposed sources. Such a study, however, is difficult in the absence of philological editions for all the Romanian Psalters of the 16th century. Because of this, it is necessary to align the texts in an electronic corpus, together with their potential sources, which would greatly aid such comparisons.

3. Linguistic Processing of the Corpus

The first step in the linguistic processing of any old Romanian text written in Cyrillic alphabet is that of transposing into the Latin alphabet, for ease of access to the content. This can be achieved by transliteration (which replaces each letter in the Cyrillic alphabet with the equivalent letter in Latin alphabet, which is always the same) or by transcription (which supposes dropping graphic differences in the case of the same

phoneme, dropping letters with no phonetic value, or, on the contrary, marking the phonetic differences in the case of the same Cyrillic letter). Transliteration does not disambiguate the phonetic value of letters, and is simply a mechanical replacement; the advantage of this procedure is its precision, as it exactly reproduces the original. Transcription, on the other hand, allows for easier access to the text and offers a broader view with regards to phonetic peculiarities of the text; the disadvantage comes from the somewhat arbitrary interpretation of the letters. Nevertheless, transcription is preferred as it is more useful (Gheție and Teodorescu, 2005:76), even accounting for the risk of incorrectly interpreted symbols. The Cyrillic text can always be accessed for verification.

The fact that the rhotacising Psalters are the first religious texts in Romanian (also, the first large texts in Romanian; in the case of H, according to Al. Mareș (*apud* Gheție and Teodorescu, 2005:19), predating *Scrisoarea lui Neacșu* (*The Letter of Neacșu*) from 1521, currently believed to be the oldest written Romanian text available) implies that they are difficult to process. They belong to the first period of the existence of the Romanian Cyrillic alphabet (Boerescu, 2014:88), during which a local “tradition” of using it, specific to Romanian, is built. These texts are difficult to process because of the use of continuous script (*scriptio continua*), and the editor is the one that tokenizes the words, as well as oscillating writing, characteristic of the early period of Romanian writing; added to this are graphical errors because of the lack of experience of the copier.

All three Psalters have been edited, but using different editing methods. S was published by Bianu (1889), with Latin text accompanied, on the next page, by the facsimile. The reproduction of the text with Latin script, which Bianu (1889: XV) calls “transcription”, has many characteristics specific to transliteration, as the different versions of the same Cyrillic letter are not reproduced. Also, since the edition is more than a century old, the text is difficult to work with. The second edition for S, (Candrea, 1916), is in fact a rebuilt text; the exact text of each Psalter is available as a footnote, which makes the text difficult to use. V was published in Cyrillic alphabet by Giuglea (1910-1911). The only modern transcribed edition is available for H (Gheție and Teodorescu, 2005).

In the volume for the edition of H, Gheție and Teodorescu (2005:76) describe the difficulties in transcribing the text, which are due, on the one hand, to the graphic instability due to the lack of writing tradition for Romanian, and, as such, lack of hard rules, and on the other hand to the “graphical fancy of the copier”. These observations also apply to S. The most stable with regards to graphic representation is V, which has the least oscillations and less uncommon or traditionally Slavonic letter representations (at least in the three psalms analyzed for this paper). For the rest of the chapter, we will briefly describe the Slavonic letters that are difficult to transcribe, together with our solutions.

The letter ѣ is frequent in all the three Psalters, and is most often used to represent the vowels *ă* and *î*. Usually, its phonetic value is *ă*: *ръдикъсе* (*rădică-se* S 113:7), *грѣскоу* (*grăiescu* S 113:12), *къпръ* (*cătră*, all three Psalters), etc. There are, however, numerous cases where it is difficult to say whether it notes the vowel *ă* or *î*; Gheție and Teodorescu (2005:31) have not reached a conclusion with regards to the letter ѣ in those

contexts where there is an *i*, such as: **мѣнрѣ** (*mǎnrǎ?/mînǎrǎ?*, H 113:12), **се зѣкъ** (*se dzîcă?*, V 113:10), **пѣмѣнѣтѣ** (*pǎmîntul?/pǎmǎntul?*, H 113:25; S 113:7 etc.), **сѣмѣ** (V 113:22), etc. In such cases we have chosen the vowel *i* (the same as Gheție and Teodorescu, 2005), as more precise conclusions regarding this letter cannot be drawn without analyzing the entire text, and, even then, those conclusions might not be relevant given graphical alternations. In this respect, the word **пѣмѣтѣ**, found often in H, was transcribed as *pǎmîntul*, and **пѣнрѣ** (H 113:10) as *pǎgîni*. The Cyrillic letter **ѣ** also occurs at the end of some words: **оилорѣ** (*oilor*, H 113:6), **сѣмерѣ** (S 142:10), **сѣфакѣ** (S 142:11), cases in which we have decided it probably has no phonetic value. In cases such as **ѣнтурека** (*întureca*, H 138:11; the same as in Gheție 2005:32), **ѣнтрѣ** (*întru*, H 142:2), the letter **ѣ** seems to denote the vowel *u* and has been transcribed as such. Also, in H it is used to denote a group of two consonants: **пѣмѣнтѣлѣ** (H 138:14) transcribed *pǎmîntului*; cf. **чѣнѣстици** (H 138:16), transcribed *cînstiți*.

The Cyrillic letter ѣ is often present as the final symbol, after a consonant, with no phonetic value: лорѣ (S 113:10 *et passim*), and, sometimes, inside a word: хиклѣньшорѣ (*hicleanșug*, S 138:3); this is frequent in S, and also appears sporadically in H (аорѣ, 113:12, which can be transcribed as *auru* or *aur*; кѣндѣва 113:10), where ѣ has the role of marking a final consonant. Sometimes, a final ѣ with no phonetic value can be seen in V: лakovѣ (V 113:7). The letter is also used for representing the vowels *ă* and *i*: пѣгѣри (*păgîri*, H 113:1), коутремѣрѣсе (*cutremură-se*, H 113:7), фѣрѣ (*fără*, H 142:6), форѣ (*fură*, V 138:16), etc. For the sequence препѣмѣнтлорѣ (H 142:12), we have chosen the transcription *pre pămîntulu*, taking into account the commutability of ѣ/ѣ and the possibility that ѣ might sometimes notes the final vowel *u*. The letter was transcribed as *i* in форишь (S 113:5), by analogy with contexts in which the letter и appears at the end of this word.

The letter ҃ appears as the first symbol, and notes the vowel *î*, as in ҃нпѡрсе (S 113:8), ҃нема (S 138:22), ҃нпѡу (V 113:26), or the sequence *î + n/m* depending on the consonant following: ҃ѣіе (S 142:12), ҃пѡрсау (H 113:8), ҃вѡж (V 113:15), and sometimes both in the same context: ҃пѡуѣрекоуѡнѣсе҃нпѡуѣреѣ (înturearecul nu se înturecǎ, V 138:11). The Cyrillic letter also appears in less common contexts, such as маѡсѣмѣ҃нраю (*me asǎmǎnraiu*, H 142:8). Noting the uncommonness of the situation, Gheție and Teodorescu (2005:27) conclude that it is due to a copier who was not sufficiently knowledgeable in the value of each letter in Romanian writing. The same explanation holds for letter sequences such as де҃ (H 113:1 and *passim*; дѣ҃ in S), transcribed as *den* and *din* respectively; however, such sequences are recorded in texts of the second half of the XVII century, where this explanation is less likely.

The vowel *î* is sometimes denoted by *paieric* (apostrophe), exclusively in H: πᾶλιν ποῦλ (H 113:7), ἔλπις (*sîmt*, H 142:14, cf. V εἴλπις).

The letter *ж* appears, in the case of the three psalms we have analyzed, only in V (Gheție and Teodorescu, 2005:30) note that it might rarely appear in H), and most of the time notes the letter *ѣ*: *кѣтрѣ* (142:6), *вѣждичіиѣ* (138:16), but more rarely than *ѣ* or *ѣ*. In cases such as *жнши* (138:15) and *мжрѣлор* (142:6), we have transcribed it as *î*.

The letters Ѧ and ꙗ both note the *ea* diphthong: чѡѧ (H 142:10), with ꙗ being encountered more often used as such rather than Ѧ (which usually denotes the *ia* diphthong: поуѡаꙗкеаши, *tu me încheiași*, H 138:4). One peculiarity of H is the common use of the two Cyrillic letters as *e* (transcribed as such): єпурєає (*întoarse-se*, 113:3); тоаѡѧ (*toate*, 113:11), or a strange notation such as сѡпѧѡѧъ (*se tem*, 113:19). This is also found in S, but rarely: нєнѧвѧдѧскѧи (*nenăvidescu-i*, 138:20). Conversely, there are numerous cases where є is used for *ea*: єнцєлєпчїєнє та (*înțelepciunea ta*, S 138:5), дєрєптатєтє (*dereptatea ta*, S 142:1).

Traditional notations such as **ерѣпмоу** (H 113:1) and **ерѣпет** (S 113:1), cf. Slavonic **ѣрѣппа** (V, in which **и** is used) were transcribed as *Eghiptu*, and *Eghipet* respectively. Also, the sequences which keep the Slavonic groups **лъ**, **ръ** (transformed, by metathesis, **ѣл**, and **ѣр** respectively in Romanian) were transcribed according to the current form: **єрѣтошесе** (*învîrtoșe-se*, S 138:5), **єрѣтосъ** (*vîrtos*, S 138:17; H 138:13; **єрѣтосоу** *vîrtosu*, V 138:17), **єфрѣшити** (*sfîrșit*, H 138:21). The reason that these notations are not phonetically useful is provided, in part, by graphical alternations (cases in which, for the same contexts, **ѣл**, **ѣр** are used), or, even more telling, notations where this characteristic is extended to words coming from Latin, with no phonetic reason: **єрѣбаци** (V 138:18), **єрѣгѣ** (V 142:10).

The errors due to the copier have been corrected in the text and the procedure was marked in footnotes. Abbreviations were extended to the full form, and the added letters have been marked by square brackets; the letters written above the lines of text were lowered into line and marked in italics. We have capitalized the first letter of each psalm and of proper names, according to current standards. Punctuation is also according to modern standards and the logic of the text, and also takes into account separation into verses.

Apart from assigning values to the more difficult to interpret letters, the linguistic processing also performed word segmentation, as the three Psalters use continuous writing. Since the psalms are not segmented into verses, we have decided to split the text into verses, a procedure also employed by Gheție and Teodorescu (2005). We have taken as reference, conventionally, the psalms in the Bucharest Bible (Bianu, 1889; Gheție and Teodorescu, 2005) have used the *Septuagint* for their segmentation. We have kept the segmentation in V in only one instance (142:12-13), as this fit better with the logic of the text.

For the automatic alignment we have used as a pivot the psalms in H. This is due to the fact that, on the one hand, H is the oldest of the texts, and on the other, the hypotheses about the relations between Psalters in literature: S and V appear to constitute a group, while H is a different revision of the base translation (Gheție and Teodorescu, 2005:21-22; Gheție, 1978). Thus, the alignment to H gives us the opportunity to better determine the differences between the two ramifications of the base text. Finally, we have manually aligned the Romanian version in V to a Slavonic version; this serves to check a hypothesis in literature which states that, in bilingual Psalters (H is also a copy of a bilingual Psalter), the Romanian text is not the translation of the parallel Slavonic text.

4. Automatic Word Alignment

In order to assist in the creation of the corpus, it is necessary to automatically align the words in parallel verses. To this end, we have built on a tool which was previously developed for the automatic alignment of old Romanian texts in the *Monumenta linguae Dacoromanorum* (MLD) project (Moruz *et al.*, 2012). The previous system performed alignment on the basis of the Levenshtein distance between candidate word in parallel verses, and, in those cases where the differences were too great, on semi-automatically created lexicon.

However, although the alignment tool performed well on 17th century Romanian texts (Bucharest Bible 1688, ms. 45 and ms. 4389), it fared poorly on the texts of the rhotacising Psalters taken into account for our corpus. The reasons for this are many fold: the Psalter texts are significantly more varied than those of the MLD texts, and, because of this, the Levenshtein distance approach worked poorly; the lexical composition of the Psalters is also significantly different, and the previously created lexicon does not contain most of the needed word equivalence pairs; the current texts are not tokenized at the word level, as the previously mentioned texts are (also, in the case of the text of the Bucharest Bible, POS tagging and word sense disambiguation is also available, greatly aiding in the alignment).

Since the rhotacising process can be manifested in different ways, the same word can vary significantly between versions, e.g. *rruga* (H), *rugăciure* (S), *rrugăciurea* (V); because of this, these equivalent words have low Levenshtein similarity, and are not cannot be aligned. To solve this issue, we have tested various word similarity metrics, and have determined that a combination of string distance and k-shingle similarity, weighed by the relative position of the candidate words, yields the best results. The string distance method we have chosen is Longest Common Subsequence, which computes the distance between two candidates according to the formula below:

$$\text{distance} = |\text{word1}| + |\text{word2}| - 2 * |\text{longestCommonSubsequence}|$$

K-shingle similarity draws its name from the fact that the candidate words are split into shingles (n-grams) of length k; similarity is then computed on the basis of these k-grams. The k-shingle similarity measure we have chosen is Cosine similarity, which transforms the candidates in vectors of occurrences of k-shingles, and then computes similarity as the cosine of the angle between these two vectors (the value of k we have settled on is 3). Cosine distance between two strings is computed as 1-cosine similarity.

Given these two measures, we have aggregated them multiplicatively to obtain the compound difference between the candidate strings (it is important to note that this compound value is no longer a distance metric, but rather of score of dissimilarity). Since verses are of similar length, and equivalent words appear in the same general position, we have only taken into account words which are in the same vicinity as the candidate word (within an 8 word window, centered on the position of the pivot), and negatively weighed alignments the further away they occur from the center of the target widow. This accounts for syntactic variability sufficiently to cover the large majority of cases, and also does not break alignment in the case of differences in word numbers in parallel verses.

SEMI-AUTOMATIC WORD LEVEL ALIGNMENT FOR RHOTACISING PSALTERS IN ROMANIAN

If the value thus obtained is lower than the sum of the length of the words, multiplied by a factor of 0.75 (determined experimentally), we consider the words aligned. Below is given an example of a fully automatic alignment for 142:9.

S: Audzită fă mie demăneța meserearea ta, că în tine upuvăiiu.
H: A[u]dzită fă mie demînrața m/la tă, că pre tinre nedejđiiu.
V: Audzită fă mie demăreța mes[e]rearea ta, că spre tire nedejđuiiu.

Figure 3: Automatic alignment using similarity measures

As mentioned previously, we have used H as a pivot for alignment; it can be seen that, although most words are correctly aligned, there are two notable exceptions: *mila* vs. *meserearea* and *nedejđiiu* vs. *upuvăiiu*. In order to solve these kinds of semantic similarities, we have created a lexicon of equivalent word forms. Unaligned words were highlighted in the corpus, together with the most likely equivalent, determined on the basis of the alignment of neighboring words. These proposed equivalencies were manually validated from the automatically suggested versions, and this lexicon was then used for further alignments. The example given below shows, in dotted arrows, the alignment of such words.

In the case of subsequent alignments using both the similarity metrics described and the bootstrapped lexicon, we need to take into account the fact that we do not have access to a proper lemmatizer. Because of this, we proposed the solution of calculating a many to many distance between all the versions of our candidates, and choosing the lowest possible one. If this is below the threshold described above, we allow for the alignment of the initial candidates, and also propose, if needed, a new entry in the lexicon.

S: Audzită fă mie demăneța meserearea ta, că în tine upuvăiiu.
H: A[u]dzită fă mie demînrața m/la tă, că pre tinre nedejđiiu.
V: Audzită fă mie demăreța mes[e]rearea ta, că spre tire nedejđuiiu.

Figure 4: Automatic alignment using similarity measures and lexicon

Since one of the intended uses of the corpus is the study of translation sources for Romanian rhotacising Psalters, we have also manually aligned one of the Romanian versions, V, with the Slavonic version. We have chosen V as it is available as a bilingual manuscript, and, as stated above, we suppose that the Slavonic version was not the text used for the translation. Below we give an example of such an alignment (113:17).

S: Casa lui Is[rai]l upăvâi în D[om]nul, agiutoriu și scutitoru lor este.
H: Casa lui I[sra]il upovăi pre D[o]m[nu]lu, agiu[to]riu și sprejenitoriu lor e.
V: Casa lui Iizdrai/ nedejđui spre D[u]mn[e]dzău, agiutoriu și scutitoriu lor iaste.
ДОМ ІІЗДРАІ ОУПОВА НА ГА ПОМОЦНИКЪ И ЗАЩИТНИКЪ ИМЪ ЕС(ТЬ)

Figure 5: Alignment of the 3 Romanian Psalters and the Slavonic text

As can be seen from the examples given above, the automatic alignment, augmented with the lexicon, has good results in most cases. There are, however, some instances where the texts cannot be fully aligned as there are differences, such as tense or voice for the verb, conjunctions or prepositions, etc., which add words with no corresponding

lexical equivalent in the parallel texts. Such is the case of verse 113:20, given below, where the circled words cannot be lexically aligned.

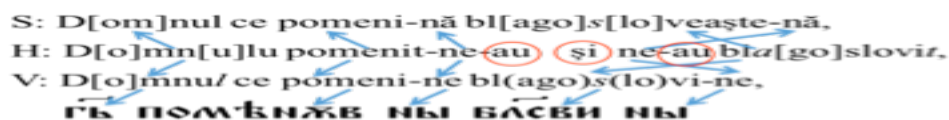


Figure 6: Word forms that cannot be aligned

Although the corpus we have created is small, the word level alignment of the three psalms allows for some pertinent remarks. The comparison between versions appears to support the findings of Gheție (1978): S and V are more alike than any of the two and H; these similarities are mostly on the lexical and grammatical level.

On the lexical level we note synonymic series such as: *a pogorî* (H) – *a deșinde* (S, V 113:26); *păgînrii* (H) – *limbile* (S, V 113:10); *năstăvi* (H) – *dere(a)ge* (S, V 138:9); *pîntecele* (H) – *zgăul* (S, V 138:12); *spuniu-me* (H) – *ispovedescu-me* (V, S 138:13); *vîrtos* (H) – *foarte* (S, V 138:16); *socoteaște* (H) – *ia aminte/amente* (S, V 142:1); *vrăjmașul* (H) – *dracul* (S, V 142:3), etc. This is also the case for *mila* (H) – *meserearea* (S, V 113:10), the second word coming from Latin (*miserere*), which might have been chosen because of a Latin source, even though the interposed Slavonic text for V contains *млѣсти* “milă” (see above for the hypothesis regarding a Latin language source).

This does not mean that the match, as described above, is true in all circumstances; for example, in 113:18, the H and V versions have, as an equivalent for the Slavonic *оупова́*, the verb *a nedejdui*, while in the case of S *upovăi* is used; in 113:19, the same Slavonic verb is given as *a upovăi* in H and S, and *a nedejdui* în V. In some other cases, all three versions use different translations: *încheiași* H, *feceși* S, *zidiși* V (138:4) for the Slavonic *сѣздá*; *căile* H, *cărăirile* S, *drumurele* V (138:3) for the Slavonic *пѣти*. Sometimes, H provides a version completely different from S and V, and also from the Slavonic version in V; the Slavonic *несѣкомъ* (113:8) is translated as *netăiată* (cf. *несѣкомъ* *durissimus, indomitus*, Miklosich, s.v.) in S and V, while H uses *lacurile*; *păgîri* H, but *varvari* S and V, cf. the Slavonic *варвар(и)*. Verse and word level alignment allows for clearer view of these kinds of issues, as well as the determining of synonymic sets in the three versions.

Word level alignment also allows for clarification in the case of obscure writing. For example, the writing *наоупоути* (S 113:14) can only be deciphered by making use of the parallel contexts of the other two versions, *nu aput* (H, V); since the copier did not understand the context, they switched two letters, and the writing needs to be corrected to *nu aput* or *n-apat*. Also in S, 138:12, we find the text “Că tu feceși zgăul meu și preimiși-me din mațele măriei mele”, which can only be understood by using the parallel texts, which use *mumînriei* (H), and *maiciei* (V); *măriei* is another incorrect interpretation of the copier, who did not understand the copied text.

Word level alignment also aids in the correct interpretation and segmentation of words. For example, the writing *фapтaи* (S 138:14) can be interpreted as *fapta-i* (noun + copulative) or *fapt-ai* (the verb *a face* in past perfect). Comparison to the parallel texts shows that the correct interpretation is the latter (H *ai faptu*, cf. V *ai făcutu*).

The alignment of the Romanian texts to the Slavonic version allows for some remarks regarding translations at the grammatical level: a) morphological differences, such as the prevalence of past perfect tense in H and past tense in S and V; the translation of the present tense in Slavonic to present or future tense in Romanian; different translations of participles as adjective clauses or nouns (e.g. *temuții* S și V, cf. participle *бѣжѣхъ*, but *carii se tem* H); different noun forms (e.g. *de mînă* H 113:12, dar *mînilor* S, *mîrului* V, cf. *ржкѣ*; dative plural *ceriurilor* H, but dative singular *cerului* S, *ceriului* V, cf. dative singular *нѣси*); b) copying of word sequences specific to Slavonic, such as *есѣи сѣдоваль* (auxiliary + active perfect participle) copied as *ești sledgeriu* in H and translated as past perfect in Romanian as *ai isle(a)dit(u)* in S and V. Therefore, the word level alignment to the Slavonic version allows for conclusions with regards to the translation strategies used for the texts.

5. Conclusions

The results described above, made over a small part of the Psalter (2%), serve as a feasibility study for the alignment and use of the aligned texts. We intend to extend verse and word-level alignment to all the psalms in the three versions, which will allow for the checking of relations between texts. Also, in order to draw a pertinent conclusion with regards to translation sources, it is necessary to also align more Slavonic, Latin and Hungarian texts of the age.

In terms of future development, it would be useful to create a lexicon for recognizing the synonym sets, which could prove the Latin lexicon used in S. This can also be used as a resource for determining alignment to the Electronic form of the DLR, which would also help increase the size of synonym sets. In order to aid in the validation process, we intend to provide a means by which text convergence can be highlighted, and also to build a more accessible interface for viewing the alignment and performing parameterized searches.

References

- Bianu, I. (1889). *Psaltirea Scheiană (1482). Ms. 449 B.A.R. Tomul I. Textul în facsimile și transcriere cu variantele din Coresi (1577)*, Edițiunea Academiei Române, București, Tipografia Caril Göbl.
- Boerescu, P. (2014). *Din istoria scrierii românești*, București, Editura Academiei Române.
- Candrea, I. A. (1916). *Psaltirea Scheiană comparată cu celelalte psaltiri din sec. XVI și XVII traduse din slavonește*, București, Atelierele Grafice Socec&Co.
- Chițimia, I. C. (1981). Urme probabile ale unei vechi traduceri din latină în Psaltirea Scheiană. *Revista de istorie și teorie literară*, 30:2, 151-156.
- Gheție, I. (1978). Psaltirea Hurmuzaki și filiația psaltrilor românești din secolul al XVI-lea și al XVII-lea, *Limba română*, 27:1, 51-57.

- Gheție, I. (1982). A existat un izvor latin al psaltirilor românești din secolul al XVI-lea?. *Limba română*, 31:2, 181-185.
- Gheție, I., Mareș, Al. (2001). *De când se scrie românește*, București, Univers Enciclopedic.
- Gheție, I., Teodorescu, M. (2005). *Psaltirea Hurmuzaki*, vol. I, Studiu filologic, studiu lingvistic și ediție de Ion Gheție și Mirela Teodorescu, București, Editura Academiei Române.
- Giuglea, G. (1910-1911). Psaltirea Voronețeană, editată de G. Giuglea. *Revista pentru istorie, arheologie și filologie*, XI (1910), 444-467, XII (1911), 194-209 și 475-487.
- H = *Psaltirea Hurmuzaki*, în ms. rom. 3077, Biblioteca Academiei Române, București.
- Mareș, A. (1982a). Originalele primelor traduceri românești ale Tetraevanghelului și Psaltirii. În Ion Gheție (coord.), *Cele mai vechi texte românești. Contribuții filologice și lingvistice*, București, Universitatea din București, Institutul de Lingvistică, 183-205.
- Mareș, A. (1982b). Filiația psaltirilor românești din secolul al XVI-lea. În Ion Gheție (coord.), *Cele mai vechi texte românești. Contribuții filologice și lingvistice*, București, 207-261.
- Mareș, A. (1984). Datarea Psaltirilor Scheiană și Voronețeană. *Limba română*, 33:3, 191-198.
- Mareș, A. (2005). *Scriere și cultură românească veche*, București, Editura Academiei Române.
- Miklosich = Fr. Miklosich. *Lexicon Palaeoslovenico-Graeco-Latinum*, Vindobonae, Guilelmus Braumueller, 1862-1865.
- MLD = *Monumenta linguae Dacoromanorum. Biblia 1688*, serie nouă, Eugen Munteanu (coord.), Iași, Editura Universității „Alexandru Ioan Cuza”, 2011-2015.
- Moldovanu, D. (2009). Finalitatea textelor slavo-române intercalate din Evul Mediu românesc. În *Text și discurs religios* (ed. Alexandru Gafton, Sorin Guia, Ioan Milică), Iași, Editura Universității „Alexandru Ioan Cuza”, 2009, 103-134.
- Moruz, M., Iftene, A., Moruz, A., Cristea, D. (2012). Semi-Automatic alignment of Old Romanian Words Using Lexicons. In *Proceedings of the 8th International Conference “Linguistic Resources and Tools for Processing of the Romanian Language”*, 8-9 December, 2011, 26-27 April 2012, 119-126.
- Munteanu, E. (2008). *Lexicologie biblică românească*, București, Humanitas.
- Pavel, E. (2013). Modelul slavon versus modelul latin în textele biblice românești. *Revista de istorie și teorie literară*, 7:1-4, 23-36.
- S = *Psaltirea Scheiană*, în ms. rom. 499, Biblioteca Academiei Române, București.
- V = *Psaltirea Voronețeană*, în ms. rom. 639, Biblioteca Academiei Române, București.

SOLUTIONS FOR SCANED DOCUMENTS SEGMENTATION AND LETTER RECOGNITION

CONSTANTIN CRISTIAN PĂDURARIU^{1,2}, DAN CRISTEA^{1,2}

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

² *Institute of Computer Science, Romanian Academy, Iași Branch*

{cristian.padurariu94, danu.cristea}@gmail.com

Abstract

The aim of this research is to build a robust workflow for processing old Romanian scanned texts written in the Cyrillic alphabet, in order to obtain the editable Latin alphabet transcript. If successful, the solution will become a pioneering approach to deciphering old printed texts and manuscripts, with special attention given to those using the Cyrillic alphabet. The paper examines existing deep learning and computer vision approaches with the intention to find the most adequate methods for solving the problem of converting scanned documents representing Cyrillic Romanian writings into editable forms.

Key words — Cyrillic alphabet, Latin alphabet, deep learning, interpreting old Romanian texts

1. Introduction

The Romanian Cyrillic alphabet has suffered changes during the last centuries and researchers distinguish 3 periods when different versions of the alphabet have been used. One version is the one used before the year 1830, when the transition to the Latin alphabet started. Between 1830 and 1862 a transition alphabet was already in use, when the Cyrillic alphabet was mixed with some Latin letters. The final version of the Cyrillic alphabet which supported Romanian language writings is the one used after 1930 in Bessarabia and Transnistria (Cojocaru and Colesnicov, 2016).

Many of the known approaches, some of which being described in section 3, focus on building a form of optical character recogniser (OCR) working on scanned documents, an enterprise which is usually known as not being very reliable. Standard OCR tools fail to process old scanned documents, as they are less readable, they contain colour stains, blots or other defects, it is usual that printings display irregular spacing between letters and words, and the old fonts vary substantially in both form and size even inside the same document (Cojocaru and Colesnicov, 2016; 2017).

The solution that we advance plans to leverage the latest computer vision and deep learning techniques with the purpose of obtaining a robust process for extracting text from Cyrillic Romanian scanned document, without any human intervention.

We think it is important to zoom in on the concept of Deep Learning, which is the backbone of most of the methods and approaches that we will see explained in this paper. Computers have always been good at solving problems that are hard for humans,

tasks that involved computational speed and complex calculations are easily done by a machine. The main challenge of Artificial Intelligence is to solve tasks that are easy for humans but resist at the trials of being formalised, like recognizing objects in an image or spoken words in a vocal stream. This is where Deep Learning is of help, with its high capacity to learn from former human experiences, by understanding in terms of a hierarchy of concepts. Complex concepts can be explained as a structure of simpler ones and so on, which sets about in the organisation of a deep structure graph with many layers (Goodfellow *et al.*, 2016).

2. Solution

One of the main challenges of this project is to formulate a unified solution that can accommodate the deciphering of printed Cyrillic Romanian texts and which could also be placed at the basis of deciphering manuscripts.

Although the need to decipher old Cyrillic writings is not new, only recently computer scientists started to trust that their algorithms could help researchers working in paleo-linguistics in a significant way. The approaches done so far were primarily focused on using classical approaches in OCR, which are very hard to customize. As these systems are not prepared to handle noisy writings of old Cyrillic texts, the OCR output needed always to be corrected by a human expert. Figure 1 shows some examples of noisy original scans.

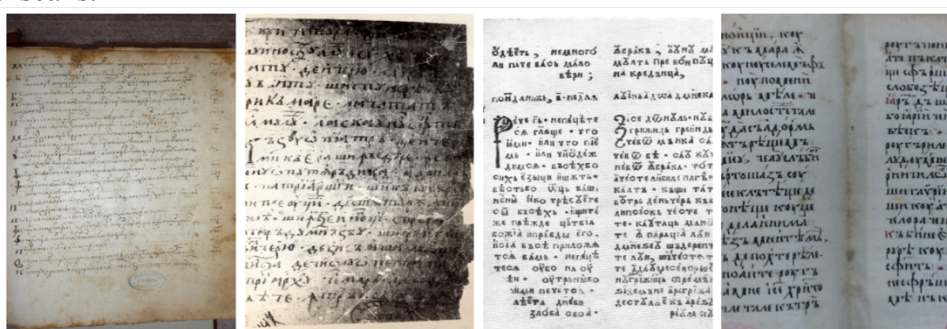


Figure 1: Samples of scanned old documents displaying different types of noise

It is very hard to come up with a robust pipeline of image processing steps expected to output a clean version of those documents, therefore containing nothing but the letters of the text. There are many types of noise that might appear and they are very hard to predict. Mainly because any error that might occur at this first step of the process will propagate throughout the entire workflow and because, sunk in noise or not, objects should still maintain their primary features, we decided to completely eliminate this type of operation and instead to study techniques of object detection and see to what extent they could be applicable directly to the original image.

As can be easily seen, our objects are nested: pages include columns, columns include lines, lines include words and words include letters. Part of these objects can be determined by techniques of document structure identification and segmentation. Identifying lines usually involves dynamic programming techniques on horizontal histograms after a grid is traced over the document. This attempts at generating a globally optimal segmentation solution based on histogram peaks (Figure 2). Since, the

final goal of deciphering scanned texts is to identify a sequence of words, recent approaches reconsider the idea of composing words from sequences of isolated characters and face directly the challenge of identifying words. So, the objects in intelligent word recognition (IWR) techniques are the words, which should be identified, irrespective of their huge variation in form, in the conglomerate of objects of the same kind populating a page. Taken from this perspective, deciphering of manuscripts differ from deciphering of prints not conceptually, but only by the much greater diversity of forms which characterises the first type of documents.

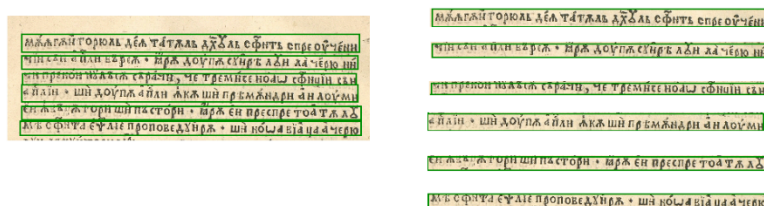


Figure 2: Line segmentation

As the recognition of words is basically a type of classification operation, it aims at finding a class ID to each object in a page. Usually, this goes in two steps: a cauterization operation, which groups similar words in clusters, followed by an operation of assigning labels to each found class. Seen as such, the assigned labels can be the predicted actual Latin versions of the original Cyrillic words (Figure 3). However, there is a drawback to such an attempt: the fact that an inventory of predicted Latin words is not known a-priori, which means that we do not know the set of labels to be assigned to original clusters. Moreover, training is problematic as well, since occurrences of many words could have very low frequencies in the training set.

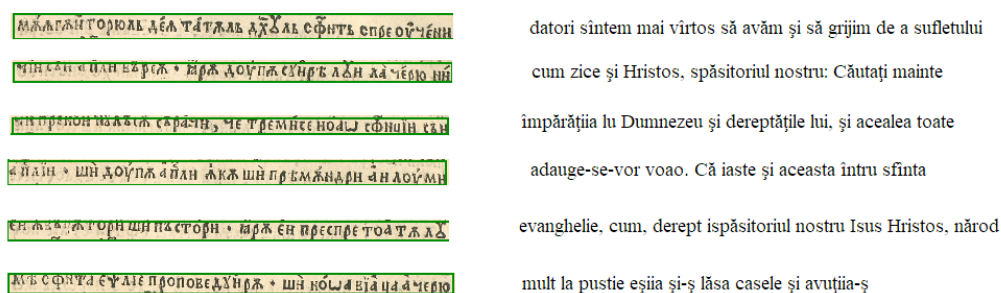


Figure 3: Transformation of an image containing original Cyrillic words into their Latin transcription

3. Related Work

3.1. Digitalization of Old Romanian Documents

In the area of processing old Cyrillic Romanian texts, one work that stands out (Cojocaru and Colesnicov, 2016) focuses on refining the output from the ABBYY FineReader OCR. Professional state-of-the-art OCRs work very well on modern texts. When dealing with historical documents, where the font may vary and the paper presents distortion, results are worse. OCRs don’t distinguish between old and modern

versions of the language, which usually use different orthography, so they tend to favour the modern one. Cyrillic Romanian script of the 18th century contains glyphs that are not supported by most OCR programs.

The article describes a pipeline of four processes: digitization and image pre-processing, OCR, text post-processing and quality evaluation. The first steps imply scanning and pre-processing the images, which mean applying actions like line straightening, image cleaning and conversion to grayscale. OCR is then applied, followed by the post-processing phase, where a manual correction of the OCR output is performed.

Another article (Cojocaru and Colesnicov, 2017) adds another step to this pipeline, which is the use of a lexicon to help with the correction of words. The vocabulary is iteratively enriched with every manual correction of the OCR output. In this paper the authors also distinguish another period in the use of Cyrillic scripts for Romanian. The pre-transition epoch is divided into early 19th century and 18th century scripts on one hand, while the 17th century scripts, due to the particular use of the language, is considered a different epoch.

Gîfu and Petic (2014) created a gold corpus for the Romanian language used in Transnistria, for both Latin and Cyrillic alphabet. They used the novel “Sania” (1955), written with the Cyrillic alphabet and applied OCR and transliteration. The original and the transliterated texts were aligned at sentence and word level. The Latin text was enriched with morpho-lexical information using the UAIC part-of-speech tagger (Simionescu, 2011). After the parallel alignment was done, a group of linguists have studied the output and came up with a series of observations regarding transliteration and how academic norms and editorial interventions affected the original text.

3.2. Object Detection

Previous work done on extraction of characters from historical documents, described in the previous section, use OCR to find and classify letters in images. This approach is prone to error especially when working with images of old documents. Also, these images might not always be scans, but also photos taken with a camera that might not contain just the text, as the portion of text may not be rigorously cropped out from the picture. It is very hard to estimate what types of distortions might occur. A more flexible solution to this problem is inspired from identifying text in scenes. Scene text detection is used to find text in any type of context. Scene text detection is related to object detection and uses similar detection algorithms. Some of the most common applications are licence plate detection, face detection, people detection, etc. One of the main problems when dealing with object recognition algorithms is that it is very hard to identify the region where to look for the object. Objects can be located anywhere within the image and can have different sizes. An ordinary CNN can't solve this problem. One naive approach would be to take many, randomly selected, regions of interest from the image and use a CNN to classify the cropped regions, but obviously the results would be poor. So, we will discuss some of the most common object detection approaches available now.

R-CNN (regions with CNN features) (Girshick *et al.*, 2014) is an algorithm that uses selective search (Uijlings *et al.*, 2013), which extracts just 2000 regions of interests. Now, instead of having to classify a large number of regions, the algorithms has to deal

with just 2000. These regions are wrapped into a rectangle and fed to a Convolutional Network that extracts a set of features. These features are then used as input for an SVM classifier (Cortes and Vapnik, 1995), which will output the region label. Given all the label scores, for all the proposed regions in an image, the algorithm will then eliminate all the proposed regions that have an intersection-over-union score lower than some threshold. But still, this approach does not efficiently solve the problem of finding candidate regions. Also, it is time consuming to classify 2000 regions for every image. Fast R-CNN (Girshick, 2015) is an improvement of the previously mentioned algorithm. In this approach, instead of selecting and inputting 2000 proposed regions, the algorithm takes the full image at once. A convolutional feature map is formed, similar to R-CNN, and the proposed regions are selected, using a selective search approach (Uijlings *et al.*, 2013). These are sent to a RoI pooling layer, which reshapes the regions into a fixed size, which then can be fed to the fully connected layers. The advantage of this algorithm is that, instead of doing the selective search in the beginning and having to call the CNN 2000 times, the feature map is extracted using CNN and the proposed regions are dynamically computed.

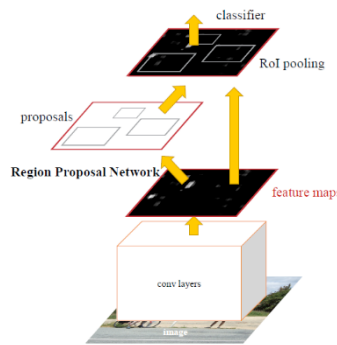


Figure 4: Faster R-CNN workflow (Ren *et al.*, 2015)

An even more efficient method to object detection is known as Faster R-CNN (Ren *et al.*, 2015) (Figure 4). The previous two approaches described use selective search for identifying the proposed regions, which is quite time consuming. Instead, Ren *et al.* propose a separate network for identifying the regions of interest. The selected regions are then reshaped using the RoI pooling layer, which are then used for classification and prediction of the bounding boxes. This approach is more efficient than the previous two, because instead of decoupling the region proposal and detection phases, they now do work together and learn one from another. For example, if selective search has false negatives, this will further generate errors in the network, but not in the case of Faster R-CNN. In this method the selective search is replaced with a Region Proposal Network (RPN) (Ren *et al.*, 2015). So, after the image goes through the Convolutional Network and the feature map is generated, this map is passed to the Region Proposal Network. Here a sliding window passes through the feature map. For each location in the feature map generate $k = 9$ anchors, which represent boxes at 3 scales (128, 256, 512) and with 3 aspect ratios (1:1, 1:2, 2:1) and this will be the proposed regions. For each box there is a classification layer, which outputs $2k$ scores and a regression layer, which calculates the $4k$ coordinates. If the feature map is of size $W \times H$, there are $W \times H \times k$ anchors in

total. The optimization described for the late version, still have to use some kind of region selection to localize the objects.

YOLO (you only look once) (Redmon *et al.*, 2016) is very different from the region based algorithms described above. It uses a single convolutional network to predict the bounding boxes and the labels. YOLO (Figure 5) takes an image and splits it into an $S \times S$ grid. For each grid cell centre, it takes B bounding boxes that have the same centre as the grid cell. The algorithm does this in order to detect cases where there are several objects that might occupy the same grid cell. So, the algorithm uses a typical CNN with a fully connected layer at the end. YOLO models detection as a regression problem and predicts an $S \times S \times (B \times 5 + C)$ tensor. That means that for each of the B bounding boxes associated with each grid cell, the network predicts 5 values ($bx1$, $by1$, $bx2$, $by2$ and c - confidence that an object exists in the box), and C class probabilities. In the end, boxes with low probability are removed and boxes with the highest shared areas are merged using a method called non-max suppression.

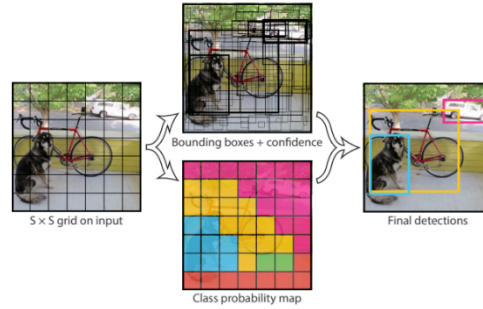


Figure 5: YOLO workflow (Redmon *et al.*, 2016)

3.3. Image Captioning and Sequence to Sequence Modelling

Another way of solving this problem is by applying a technique similar to image captioning. In image captioning, given an image, the system must output a sentence that describes the content of the picture (Figure 6). This might work just as well for our case where we try and predict the text (Latin version) from an image. Following the comprehensive survey of Image Captioning (Hossain *et al.*, 2018), we went through some of the most representative methods for generating text out of images. Some of the early approaches (before 2014) were not able to produce sentences that make sense and could not determine with enough precision the object-word association. One of the groundbreaking papers is (Vinyals *et al.*, 2015), which describes an approach inspired by the sequence-to-sequence model (Sutskever *et al.*, 2014). This method uses an encoder-decoder architecture, but it replaces the LSTM encoder with a CNN that generates a feature representation, which will be passed on to the decoder for generating the output sequence, similar to the “context” vector. A similar approach was done by (Donahue *et al.*, 2015), which also used a CNN instead of the LSTM encoder, but, this time, the CNN was a VGGNet (Simonyan and Zisserman, 2014).

The first “attention” based method used for image captioning was introduced in (Xu *et al.*, 2015). This method concentrates on the salient parts of the image and generates words accordingly. CNN based “attention” methods concentrate on features from the top level of the network, which will miss information from the image that might contribute to a more detailed described. So, the approach in (Xu *et al.*, 2015) will use “attention” on the lower level of the CNN to preserve information.

Other recent popular papers (Luo *et al.*, 2018; Rennie *et al.*, 2017) also use “attention” to better target the predictions. This time the attention is focused on the CNN feature map that is generated on the top layer of the network. The algorithm will assign weights to each feature in the map, which will contribute differently to the next output in the sequence. The difference between the two papers is that in (Luo *et al.*, 2018) the weighted feature map will be used as input for the input gate of the LSTM decoder, while in (Rennie *et al.*, 2017) the feature map is weighted and fed into every gate of the LSTM.

The methods described are based on a bi-modal model, which uses a common embedding space for images and sentences trained to learn relationships between two entities. A different approach is described in (Karpathy *et al.*, 2014), where the objects-words associations are considered at a finer level, predicting fragments of sentences rather than the whole sentence at once. To do so, an R-CNN (Girshick *et al.*, 2014) algorithm is used to detect specific regions in an image. Then, each region is described separately and the resulting descriptions of the individual fragments of the image are put together in a sentence with the help of Dependency Tree relations (Figure 6).

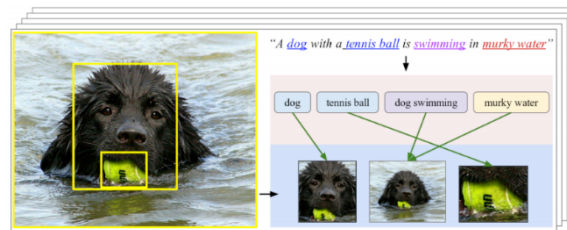


Figure 6: Image to text using object detection (Karpathy *et al.*, 2014)

Coming back to our goal, we see that these methods, although having rather similar goals with what we want to achieve, also differ in at least one important aspect: the structure of the generated sentence in Latin Romanian should be identical to that of the Cyrillic version, therefore here we do not face a proper constraints-free generation as in the last approach investigated. We retain however the possibility to understand zones of the original image, the denser they are the better, and try to guess the missing elements. A Dependency Tree could indeed be chosen to glue together the discovered segments of a sentence, by using constraints of lexical, syntactic and semantic nature, but guessing the missing words should also take in consideration lexical distances. One supplementary difficulty comes from the scarceness of morphological, syntactic and semantic resources for variants of the old Romanian languages.

Another way to solve image captioning is through sequence-to-sequence models (S2S), first described in two pioneering papers (Sutskever *et al.*, 2014) and (Cho *et al.*, 2014). Apart from image captioning, S2S models were also proved to achieve good results in: machine translation, text summarization, question answering, etc.

When referring to any S2S model, cf. to (Cho *et al.*, 2014), two key components should be retained: the encoder and the decoder (Figure 8). Both are basically Recurrent Neural Networks (Karpathy *et al.*, 2015). The Encoder takes each item in the input sequence and constructs a so called “context” vector, which has a fixed length and represents the output of the last hidden layer of this network. After the encoder finishes the process of the input sequence, it sends the “context” vector to the Decoder, which begins producing the output items one by one based on the “context” vector and the already outputted items (Figure 7).

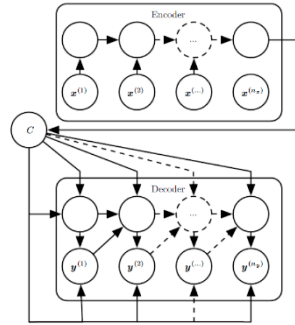


Figure 7: Encoder-Decoder architecture (Goodfellow *et al.*, 2016)

There are some problems though with this approach. One is that the “context” vector is of fixed size and might not be able to encapsulate all the information from the input sequence, especially when dealing with long sequences. Another issue is that RNNs are hard to parallelize, because they need to wait for the output of the previous recurrent step in the network. Also, there are still problems with long-range dependencies, for long texts or sentences, even for networks using LSTM (Hochreiter and Schmidhuber, 1997). In the case of image captioning, the encoder is replaced with a CNN that generates sets of image features to be used as context vectors.

3.4. The Dataset

The data that we have right now consists of a collection of 22 documents summing up 5520 scanned pages (Table 1) with their corresponding Latin transcriptions.

Table 1: Sources of our dataset

Source	# Scanned pages
<i>Bible</i>	98
<i>Pravila aleasa a lui Eustratievici (1632)</i>	728
<i>Codex Sturdzanus (1580 - 1619)</i>	253
<i>Coresi, Cazania II (1581)</i>	251
<i>Cuvint pentru curatie</i>	73
<i>Evangelhilarul de la Londra</i>	281
<i>Evangelhilarul de la Sibiu</i>	235
<i>Floarea darurilor (1491)</i>	341
<i>Leastvita (1672)</i>	620
<i>Prăvălia de ispravă</i>	53

<i>Prăvălia de la Govora</i>	348
<i>Prăvălia ritorului Lucaci</i>	144
<i>Psaltirea Hurmuzaki (1500-152)</i>	249
<i>Psaltirea Scheiană (secolul al XV-lea)</i>	530
<i>Psaltirea Voroneeană (secolul al XVI-lea)</i>	149
<i>Rujdenița (1620)</i>	10
<i>Manuscrisul de la Ieud (1610-1640)</i>	48
<i>Molitevnici Rumânesc</i>	185
<i>Octoih Rumânesc (1683 - 1686)</i>	322
<i>Palia de la Orăștie (1582)</i>	322
Total	5520

Starting from these images, we are currently building a new dataset consisting of pairs of letter/word/line bounding boxes and their corresponding decoded content in the Latin alphabet. A web tagging tool (Figure 8) was specially designed for this task, which enables a person to load a sequence of scanned images of pages into a web canvas and annotate any number of bounding boxes on their areas. In the right hand side we also have the original language text that will help us easily decode the image text. For each box the annotator will have to manually assign a value content, which the tool will pair with the calculated box coordinates. This kind of data are necessary to build an object recognition system. The tool will enable to create a reliable and complex dataset not just for our purpose but for many other annotation tasks.

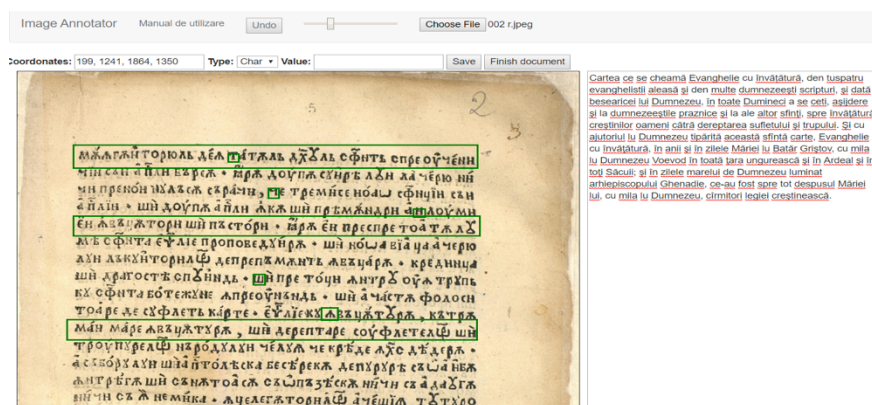


Figure 8: The image annotator frontend

For the application frontend we used HTML, CSS and JavaScript and the backend is written in PHP, which uses a MySQL database for persistence.

4. Conclusions

The task of deciphering images of old documents is not an easy one. Even though we have enough methods today to approach this problem more freely, we still have to make a lot of fine tuning and customization to make the existing tools suite our goal.

Tackling this particular subject gives a lot of room for experimentation, because of the multitude of parameters that have to be taken into consideration. What we have presented in this paper is merely a preparative work, intended as an inventory of what seems to us being promising models for deciphering scanned Cyrillic documents. We consider that the methods presented, together with the data now under preparation (and for which we have already developed an acquisition frontend), offer promising perspectives of doing original work and advance the state of the art in this domain.

AI technologies (a look at the chronology of the articles cited in this paper will reveal that few articles appeared before 2012) are applied in surprising new environments as compared to those they have been initially planned for. We are aware that the novelty of these approaches does not necessarily guarantee success. One big drawback of using Deep Learning is that it requires a high quantity of data from which meaning is inferred and generalizations reach the point where the learned concepts can safely be applied to new use cases. In fact, this represents one of the causes why much progress was not made in solving this problem and the formalizations have been rather stuck on classical OCR techniques. For the task described here we are determined to apply a bootstrapping approach to acquiring data, by iterating a sequence of correct-retrain-enlarge operations over the transcribed data obtained at each step, until the evaluation of the trained transcriber shows a flattening to a maximal value.

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, Romanian Academy, as well as under the PhD research theme of the first author.

References

- Cho, K., B. van Merriënboer, Gulehre, F. Bougares, H. Schwenk, and Y. Bengio (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*.
- Cojocaru, M. B., and Colesnicov (2016). *Optical character recognition applied to romanian printed texts of the 18th – 20th century*.
- Cojocaru, M., and Colesnicov (2017). *Digitization of old romanian texts printed in the cyrillic script*.
- Cortes, C., and V. Vapnik (1995). Support-vector networks. *Machine Learning*, vol. 20, no. 3 273-297.
- Donahue, J., L. A. Hendricks, S. Guadarrama, and M. Rohrbach (2015). Long-term recurrent convolutional networks for visual recognition and description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, 677-691.
- Gifu, D., and M. Petic (2014). Transliteration and alignment of parallel texts from cyrillic to latin.
- Girshick, R. (2015). Fast R-CNN. *International Conference on Computer Vision (ICCV)*.
- Girshick, R., J. Donahue, T. Darrell, and J. Malik (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *IEEE Conference on*

- Computer Vision and Pattern Recognition, ser. CVPR '14*. Washington, DC, USA. 580-587.
- Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep Learning*. MIT Press.
- Hochreiter, S., and J. Schmidhuber (1997). Long short-term memory. *Neural computation*, vol. 9, no. 8, 735–1780.
- Hossain, M. Z., F. A. Sohel, M. F. Shiratuddin, and H. Laga (2018). A comprehensive survey of deep learning for image captioning.
- Karpathy, A., A. Joulin, and L. Fei-Fei (2014). Deep fragment embeddings for bidirectional image sentence mapping. *NIPS*, 1889-1897.
- Karpathy, A., J. Johnson, and L. Fei-Fei (2015). Visualizing and understanding recurrent networks. *CoRR*.
- Luo, R., B. L. Price, S. Cohen, and G. Shakhnarovich (2018). Discriminability objective for training descriptive captions. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6964–6974.
- Ranzato, M., S. Chopra, M. Auli, and W. Zaremba (2016). Sequence level training with recurrent neural networks. *CoRR*.
- Redmon, J., S. Divvala, R. Girshick, and A. Farhad (2016). You only look once: Unified, real-time object detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 779-788.
- Zhao, Zhong-Qiu, Peng Zheng, Shou-tao Xu and Xindong Wu. Object Detection with Deep Learning: A Review. *IEEE transactions on neural networks and learning systems* (2018): n. pag.
- Ren, S., K. He, R. Girshick, and J. Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Proceedings of the 28th International Conference on Neural Information Processing Systems – Volume 1*, ser. NIPS. 91-99.
- Rennie, S. J., E. Marcheret, Y. Mroueh, J. Ross, and V. Goel. 2017. Self-critical sequence training for image captioning. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1179-1195.
- Simionescu, R. 2011. Hybrid POS Tagger. *Language Resources and Tools with Industrial Applications*. 21-28.
- Simonyan, K., and A. Zisserman (2014). Very deep convolutional networks for large-scale image recognition, arXiv:1409.1556 [cs.CV].
- Sutskever, I., O. Vinyals, and Q. V. Le (2014). Sequence to sequence learning with neural networks. *NIPS*.
- Sutton, R. S., and A. G. Barto (1998). *Introduction to Reinforcement Learning, 1st ed.* Cambridge, MA, USA: MIT Press.
- Uijlings, J. R., K. E. Sande, T. Gevers, and A. W. Smeulders (2013). Selective search for object recognition. *Int. J. Comput. Vision*, vol. 104, no. 2. 154-171.
- Vinyals, O., A. Toshev, S. Bengio, and D. Erhan (2015). Show and tell: A neural image caption generator. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3156-3164.
- Xu, K., J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio (2015). Show, attend and tell: Neural image caption generation with visual attention. *32nd International Conference on Machine Learning, ICML 2015*, 3, 2048-2057.

MORPHO-SYNTACTIC REGULARITIES IN UD_ROMANIAN NONSTANDARD PARSING

CĂTĂLINA MĂRĂNDUC^{1,2}, VICTORIA BOBICEV³, ROMAN UNTILOV³

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

² Iorgu Iordan – “Al. Rosetti” Institute of Linguistics, Bucharest

³ Department of Applied Informatics, Technical University of Moldova

catalinamaranduc@gmail.com,

victoria.bobicev@ia.utm.md,

kikmor97@gmail.com

Abstract

The paper explores statistics of morphological and syntactic regularities in UD_Romanian-Nonstandard corpus. The corpus is growing rapidly and has now 15843 sentences and 318869 tokens containing old texts (1592-1818), and folklore from Romania and Republic of Moldova. The statistics is made on the Malt parser accuracy evaluated on the whole corpus and on its different parts. The study is focused on the dependence between the parser accuracy and the link which it makes (or not) to morphological tags of the tokens (heads and dependents).

Key words — parsing, dependency relations, folklore, morphological classification, old and regional language, treebank

1. Introduction

The UD_Romanian-Nonstandard Treebank is part of the UAIC RoDia Dependency Treebank (Măranduc *et al.*, 2018) having now 30,000 sentences. UD_Romanian-Nonstandard was introduced to this family of treebanks in November 2017 (Măranduc and Bobicev, 2017) and it is ahead of the 5th release.

It is intended for the old and popular Romanian language. Table 1 contains detailed statistics about the corpus parts. Old Romanian language is presented in the “Flower of Gifts”, a collection of philosophical histories and maxims from 1592, the “New Testament from Alba Iulia”, the first printed in Romanian in 1648, and the “Caragea’s Law”, a codex printed in 1818.

Table 1: The Content of UD_Romanian-Nonstandard Treebank

Nr.Crt.	Type	Chron.	Style	Title	Sentences
1	OLD	1592	Narrative	Flower of Gifts	1,088
2	OLD	1648	Church	New Testament Gospel and Prefaces	5,172
3	OLD	1648	Church	New Testament Acts of Apostles	5,894
4	OLD	1818	Legal	Caragea’s Law	1,188
5	FOLK	-	Poetry	Folk from R. Moldova	1,806
6	FOLK	-	Poetry	Folk from Romania	695
Total	OLD=13,342 FOLK=2,501				15,843

Thus we have samples from three centuries of evolution and three styles of the language, narrative, church and legal. The weight of the church style is high, because “The New Testament” has over 11,000 sentences and the other two books only over 1,000. Due to its homogeneity and volume, this sub-corpus has better accuracy.

We will continue the diversification and balancing of the corpus by adding fragments from the Ion Neculce’s chronicle, 1745, and numerous quotations from legal documents.

The part of popular literature includes ballads and lyrical poems from Romania and the Republic of Moldova. The corpus in the Republic of Moldova has 1,800 sentences; the one in Romania now has about 700 and is still in work.

The intention is to demonstrate the identity of the language spoken in the two states and to make comparisons between the two sections, of which the one from the Republic of Moldova is most recently collected. Also, part of this corpus was manually annotated with rhymes, in order to create an automatic rhyme annotator.

To monitor the application of the same annotation conventions in all treebanks for 88 languages, numerous data validation programs have been created. After the release of 15 May 2019, a new validation program of the data entered has been implemented. It still creates problems for UD members.

At this moment, the statistics of valid treebanks are as follows: Total 175, valid 48 (32 languages), error 105, empty 22. The UD_Romanian-Nonstandard treebank is among the 48 valid corpora.

The new validation program aims to match the morphological analysis and syntactic dependencies. Each of these, according to the rules, can be expressed by certain parts of speech and if not, the program lists errors. Also, it checks if the annotation convention is respected so that some deprels (dependency relations) cannot have descendants, and if a clause has a single word in the syntactic position of the subject. Examples of errors revealed by the “/validte.sh” program:

```
[Line 293481 Sent train-14425 Node 18]: 'advmod' should be 'ADV'
but it is 'ADP'
```

```
[Line 293546 Sent train-14427 Node 18]: 'mark' should not be
'PRON'
```

```
[Line 293570 Sent train-14427 Node 42]: 'aux' should be 'AUX'
but it is 'PRON'
```

```
[Line 293613 Sent train-14428 Node 17]: 'mark' should not be
'VERB'
```

```
[Line 293611 Sent train-14428]: 'mark' not expected to have
children (17:chema:mark --> 15:ce-:obj)
```

For the analysis of the errors made by the parser trained on our treebank, we considered that it is really important the parser's ability to detect the rule of concordance between the morphological value given and the syntactic value that it must predict.

2. Related work

Several types of parsers are used in the natural language processing. The oldest are deterministic, based on grammars; however, rules cannot provide all the situations found in the creative natural language. The next generation of parsers is the statistical one, including those based on neural networks (Chen and Manning, 2014). There are also mixed models, which solve a part of the tree in a deterministic way, and in a statistic way the relations for which there are no constraints.

M. A. Covington (1990) describes an approach based on dependency grammars, with a deterministic parsing strategy. This grammar must define a Boolean function that returns *True* for two words w_1 and w_2 if w_1 can be the head of w_2 , according to the grammar, and *False*, otherwise.

It is an incremental parsing from left to right, starting from the first word of the sentence, the algorithm attempts to bind each word as a regent or dependent of the previous word. The model is intended for languages that have a free word order. The projectivity of the structure is obtained by introducing certain constraints (rules) to the relational operation.

Another type of parser is the statistical one, trained on corpora annotated by experts. It extracts from them rules and the frequency of each structure. They avoid situations in which natural language expressions appear outside the deterministic grammar. In (Nivre, 2003) is described a non-deterministic method having a classifier trained on annotated texts with syntactic structures. This parser has been developed and is now trained on 88 languages that have built dependency treebanks annotated in the same conventions (Nivre *et al.*, 2016), available on the Universal Dependencies (UD) site¹.

Another parser, based on annotated dependencies in the form of a syntactic structure, is (Wang and Harper, 2004) stochastic parser. The parsing is done in two steps:

- A list of constraints related to possible dependents and possible regents is created for each word;
- Dependency relationships are determined.

The parser for the Romanian language was created and developed with the treebanks. The first, (Hristea and Popescu, 2003) was manually annotated and contains about 4,000 short sentences, in standard language.

A similar corpus is described in (Călăcean and Nivre, 2008), and results from the processing of some texts using the Malt parser created by Nivre.

The UAIC corpus, created by Perez (2014) as part of its doctoral research, started to be manually annotated in 2007. In 2014; it had 4,600 phrases with 105,000 tokens (words and punctuation). The number of sentences is approximately the same but the number of

¹ <https://universaldependencies.org/>.

tokens is double in comparison with the previous treebanks for Romanian, which shows that sentences are long and complex (Perez *et al.*, 2016 a, b).

The first FDG parser was created in 2010, having a modest accuracy (Popa, 2010). LAS (Labeled attachment score) = 57.89, UAS (Unlabeled attachment score) = 66.30, LA (Label accuracy) = 64.01. The explanation is that by then the corpus, manually annotated in XML without an interface, had less than 2000 sentences.

In (Colhon and Simionescu, 2012), a new attempt is made, to build a parser on the same corpus hence a model that creates trees without labeling the arcs. The parser is trained on only 1,800 sentences, and annotation without the name of relationships is not very useful for creating a training corpus.

After this date, the training corpus grows faster, using the bootstrapping method, having 3,500, then 4,000 sentences. A variant of the Malt parser (Nivre *et al.*, 2007), was trained on it and a better accuracy was obtained:

Table 2: Evaluation of the UAIC-Romanian FDG Parser

Metrics	LA	UAS	LAS
Score 2010	62.02%	68.88%	58,56%
Score 2014	80%	85%	77%

A Romanian parser created by (Florea *et al.*, 2014), based on the Stanford model (Klein and Manning, 2003) trained on French, and on the Joakim Nivre Malt Parser (Nivre *et al.*, 2007) trained on Spanish (McDonald *et al.*, 2005), two romance languages of the same family as Romanian, were created because the authors did not have a Romanian training corpus. The results were compared with the UAIC parser web variant. The scores were: LAS 67%, UAS 73% with the Stanford Parser, and LAS 57%, UAS 62% with the Malt Parser, compared to the scores LAS 77% and UAS 85% obtained by the UAIC web parser, since it was trained on a corpus of 4,600 Romanian sentences (105,000 tokens) while the two proposed models were trained on sentences in other languages.

Another parser for Romanian was recently constructed at Mihai Drăgănescu's Academic Institute of Artificial Intelligence (Barbu-Mititelu *et al.*, 2016). The new parser, trained on the UD-Romanian-RRT treebank, uses semantic information from RoWordNet (Tufiş *et al.*, 2013), to obtain a modest increase of the accuracy. They get a maximum score UAS, from 85.3% with a standard Malt parser model to UAS 85.7% adding the semantic information.

All these tests regarding the parsers for the Romanian language demonstrate that the creation of a syntactic parser is based on the corpus and that the only way to increase its accuracy is to increase the size and consistency of the training corpus.

3. Conventions of Annotation

There are more differences between our conventions of annotation and the UD ones. We will discuss here only one of them. As it is known, dependency grammars do not support the subordination of a node of two hedges. And yet, this is the specificity of

annotated if it is descended from the verb or adverb, which have no case, but with the deprel “mark”.

All these observations and many more could be formulated as constraints if we had a mixed parser. However, for statistical parsers, the process of automatic deduction of the rules from the examples noted in the training corpus is important.

4. Parsing the UD_Romanian-Nonstandard with the UD parsers

We experimented with the Malt Parser (Nivre *et al.*, 2007) provided as one of the main parsers for Universal Dependencies project. It has implemented several parsing algorithms and we worked with all of them. The corpus we worked with has more sections, and its statistics is presented in Table 3.

The whole corpus is divided in three parts as it is demanded by UD organizers: training part (82% of the corpus), the biggest one and two smaller parts, evaluation and testing (9% of the corpus each). Thus, we trained on training part and tested on testing part.

Table 3: Accuracy of the different MaltParser parsing algorithms on UD_Romanian-Non-standard corpus

Nr. crt.	Parsing method	LAS	UAS
1	covproj	76.23	83.85
2	2planar	75.46	83.13
3	nivreeager	75.35	83.18
4	stackproj	75.34	83.27
5	nivrestandard	75.23	83.11
6	stacklazy	75.15	83.08
7	planar	74.99	82.76
8	stackeager	74.59	82.49

We obtained the best results using *covproj* algorithm. However, since the maximum accuracy of 83.85 % is an average between several types of sub-corpora that have different characteristics, we evaluate each one separately. Table 4 shows the accuracy calculated for each type of text. We note the accuracy correlation with the number of sentences and the average number of words per sentence.

Unexpected results are obtained for the oldest text, “Flower of Gifts”, which has the best score, although the morphological problems abound, and the dimensions are not large. It is probably a great similarity between the style of the philosophical maxims and the narrative of the 16th century and the church style of the *New Testament*, with which the parser was trained.

Another surprise is the lack of homogeneity of the results obtained for the text of the *New Testament 1648*. Although they have comparable dimensions, the Acts of the Apostles has a very good score, and the Gospel, a weak one. As we have shown in (Măranduc, 2018; 2019), the two parts are stylistically different, the Acts of the Apostles are narrative and in the epistolary style, written, and the Gospel is

characterized by direct speech, dialogues abound. Orality seems to give rise to another type of syntactic structures, which the parser analyzes with difficulty, because in the case of folklore we also have this characteristic and low score. See Table 4.

Table 4: The accuracy of each type of text in correlation with the sub-corpora dimension

Nr.crt.	Sub-corpus	Nr. of sentences	vords per sentence	method	LAS	UAS
1	apostles	5,894	20.86	nivreeager	82.99	87.42
2	Caragea.Law	1,188	23.95	2planar	79.09	83.62
3	Flower.gifts	1,088	19.46	covproj	83.69	87.56
4	Folk.Mol	1,806	18.69	2planar	82.22	86.75
5	Folk.Rom	695	23.61	2planar	78.29	86.08
6	gospel	5,172	17.86	covproj	76.47	84.06

The low score of Romanian folklore and Caragea's Law seems to be caused by the sentence length. It should be mentioned that the Gospel sub-corpus also includes book prefaces, where 99 sentences have an average of 40.06 words per sentence.

However, we have to increase the weight of the legal style sub-corpus, which has different syntactic peculiarities. Also, the two sections of folklore should be balanced and the dimensions of the sub-corpus increased.

5. The link among the morphological classification and the choices of the syntactic parser.

To see to what extent the results are explained by the lower consistency of the corpora with low accuracy, we studied the relationship that the parser establishes or not between the morphological category and the syntactic relation of words.

Table 5: The accuracy and the correct link deprel – morphological classification of the descendent

PoS	correct	total	accuracy
ADP	2011	2096	0.959447
DET	913	975	0.93641
AUX	909	1020	0.891176
PART	344	393	0.875318
SCONJ	351	408	0.860294
CCONJ	1218	1421	0.857143
ADJ	344	434	0.792627
NUM	168	212	0.792453
PUNCT	2284	3071	0.743732
PROPN	533	720	0.740278
PRON	1772	2413	0.734356
ADV	862	1180	0.730508
INTJ	55	76	0.723684
NOUN	2300	3553	0.64734
VERB	1785	3008	0.593418

The study is realized separately for each sub-corpus, to shows whether or not the parser linked correctly the syntactic deprel which it proposed with the morphological classification of the dependent word. See table 5.

For each category of words on the first column, there are a total number of occurrences in the sub-corpus tested, on the third column, and a number of those correctly annotated with correct link, on the second column. The fourth column is the rapport between the second and the third column, i.e. the accuracy.

This table has also unexpected results. The verb and the noun, some of the most frequent, should have good accuracy, but on the contrary it is the lowest. Certain verbal forms are homonymous with nouns and each time the post-tagger makes the wrong choice according to statistics made on another type of corpus. If the mistake remains uncorrected, it leads to a mismatch between the deprel and the category or a wrong choice of the type of the deprel and the type of descendants.

Table 6 evaluates the correct link of the deprel with the morphological classification of the head and of the descendent. The morphological category of the descendent is the first, the morphological class of the head is the last, and the deprel is placed between them, on the first column. The three next columns are the same as the preceding table. This table investigated all the possible combinations and it is very long. We transcribe here a small excerpt. See Table 6.

Table 6: The accuracy and the correct link deprel – morphological class of the head and the descendent (small excerpt)

DESCEND. deprel HEAD	CORR.	TOT	ACCUR	DESCEND.deprel HEAD	CORR	TOT	ACCUR
ADP case PROP	171	172	0.990	AUX cop ADV	16	24	0.67
ADP case NOUN	1280	1294	0.990	CCONJ cc PRON	16	24	0.67
PART mark VERB	326	330	0.987	CCONJ cc ADV	6	9	0.67
AUX aux VERB	616	624	0.987	PART mark ADV	4	6	0.67
ADP mark ADV	47	48	0.979	ADV advmod NUM	2	3	0.67
DET det NOUN	639	655	0.975	NOUN vocative ADJ	2	3	0.67
DET det NUM	36	37	0.973	NUM iobj VERB	2	3	0.67
NUM nummod NOUN	128	133	0.962	NUM nummod PRON	2	3	0.67
ADJ amod NOUN	273	285	0.958	PRON nsubj ADV	2	3	0.67
ADP fixed ADP	64	67	0.955	PRON nsubj PRON	2	3	0.67

This table is very important for the amelioration of the parsers. Linguists can improve parser accuracy by studying combinations with the least accuracy to see if there are some controversial analyzes that could otherwise be noted, and if not, to add a larger number of similar examples to the training corpus, such as the parser can learn the rule by which they are annotated. At the same time, computer scientists observing the shortcomings in the table can improve the program.

Concerning the problem of the *cop* annotation convention, discussed above, we can observe some unexplained linguistic combinations in the list, which of course have a small accuracy. Examples:

A descendent having the classification AUX is subordinated of a NOUN, or ADV, but these class of words do not need auxiliaries. The subject of the nominal clause is also subordinated to the predicative name, and we have the situations: NOUN with *deprel nsubj* is subordinated to a NOUN or a PRON.

6. Conclusions

Creating a parser for the Romanian language with accuracy over 90% that does not require results supervision is a difficult task. It is mainly based on increasing the consistency and size of the training corpus. As we cannot just be content with the contemporary standard language parsing, we will not abandon the digital access of the cultural heritage, so we will continue the parser training on texts of the Old and Nonstandard Romanian language.

Malt parser results are conditioned by the link between the morphological category of words and the dependency relation, so the POS-tagger for old Romanian needs to be improved by adding new data to its lexicon and making statistics on our types of texts.

References

- Barbu-Mititelu, V., Ion, R., Simionescu, R., Scutelniciu, A., Irimia, E. (2016). Improving parsing using morpho-syntactic and semantic information. Cluj-Napoca University Publishing House. The Romanian Human-Computer Interaction Journal 9, 4 (2016), 285–304.
- Călăcean, M., Nivre, J. (2008). Data-driven Dependency Parsing for Romanian. In *LOT Occasional*, Vol.12, LOT, Uppsala, Netherlands Graduate School of Linguistics, 65–76.
- Chen, D., Manning, C. D. (2014). A Fast and Accurate Dependency Parser using Neural Networks. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*. Doha, Qatar, ACL Publishing, 740–750.
- Colhon, M., Simionescu, R. (2012). Deriving a Statistical Parsing from a Treebank. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics (WIMS)*. Craiova, Romania, ACM Publishing, 34:1–34:8.
- Covington, M. (1990). *A Dependency Parser for Variable Word Order Languages*. Artificial Intelligence Programs AI-1990-01. Athens, Georgia, The University of Georgia.
- Florea, I. M., Rebedea, T., Chiru, C. G. (2014). Parser of dependencies for Romanian language based on parsers for other Romance languages. *The Romanian Human-Computer Interaction Journal* 7, 1, 2014, 1–20.

- Hristea, F., Popescu, M. (2003). A Dependency Grammar Approach to Syntactic Analysis with Special Reference to Romanian. Bucharest, Romania. University of Bucharest Press, 9–34.
- Klein Dan and Christopher D. Manning. (2003). Accurate Unlexicalized Parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*. ACL Publishing, Sapporo Japan, 423–430.
- Măranduc, C., Bobicev, V. (2017). Nonstandard Treebank Romania – Republic of Moldova in the Universal Dependencies. In *Proceedings of Conference on Mathematical Foundations of Informatics (MFOI-2017)*. Chişinău, Republic of Moldova, Moldovan Academy Publishing House, 111–116.
- Măranduc, C., Mititelu, C., Bobicev, V. (2018). Syntactic Semantic Correspondence in Dependency Grammar. In *Proceedings of the 16th International Workshop on Treebanks and Linguistic Theories*, Prague, Czech Republic, Charles University Press, 167–180.
- Măranduc, C., (2018), The structure of dialogue in the Gospel of the New Testament of Alba Iulia (1648), In *Proceedings of the 17th International Colloquium of the Department of Linguistics*, Bucharest University Publishing House, 237–337.
- Măranduc, C., (2019) Marks of the epistolary style in the seventeenth century, in the Acts of Apostles, New Testament of Alba Iulia (1648). In *Proceedings of the 18th International Colloquium of the Department of Linguistics*, Bucharest University Publishing House.
- McDonald, R., Pereira, F., Ribarov, K., Hajik, J. (2005). Non-projective Dependency Parsing using Spanning Tree Algorithms. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. Vancouver, Canada, ACL Publishing, 523–530.
- Nivre, J. (2003). An efficient algorithm for projective dependency parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*. Nancy, France, ACL Publishing, 149–160.
- Nivre, J., J. Hall, J. Nilsson, A. Chanev, G. Eryigit, S. Kübler, S. Marinov and E. Marsi (2007). MaltParser: A language-independent system for data-driven dependency parsing. In *Natural Language Engineering*, 13(2), 95–135.
- Nivre, J., Marneffe, M. C. de, Ginter, F., Goldberg, Y., Hajic, J., Manning, C. D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., Zeman, D. (2016). Universal Dependencies v1: A Multilingual Treebank Collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC-2016)*. Portoroz, Slovenia, (ELRA), 1659–1666.
- Perez, C. A. (2014). *Linguistic Resources for Natural Language Processing*. Ph.D. Dissertation. Iaşi, Romania, Al. I. Cuza University.
- Perez, C. A., Măranduc, C., and Simionescu, R. (2016). Including Social Media, a Very Dynamic Style, in the Corpora for Processing Romanian Language. In *Proceedings at EUROLAN 2015*. Switzerland, Springer Publishing, 139–153.
- Perez, C. A., Măranduc, C., and Simionescu, R. (2016). Social Media Processing Romanian Chat sand Discourse Analysis. *Computacion y Sistemas*, 20, 3, 2016, 404–414.

- Popa, C. (2010). FDG Parser for Romanian Language. Master's dissertation. Iași, Romania, Al. I. Cuza University.
- Tufiș, Dan Verginica Barbu Mititelu, Dan Ștefănescu, and Radu Ion. (2013). The Romanian Wordnet in a Nutshell. In *Language Resources and Evaluation* 47, 4, 2013, 1305–1314.
- Wang, W., Harper, M. (2004). A statistical constraint dependency grammar CDG parser. In *Proceedings of Proceedings of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together*, Barcelona, Spin, ACL publishing, 42–49.

CHAPTER 5

MULTIWORD EXPRESSIONS AND FRAME SEMANTICS

WRAPPING OUR HEADS AROUND VMWES AND THEIR DERIVATIVES

IVELINA STOYANOVA¹, SVETLOZARA LESEVA¹, VERGINICA BARBU
MITITELU², MARIA TODOROVA², MIHAELA CRISTESCU³

¹ *Department of Computational Linguistics, Institute for Bulgarian Language,
Bulgarian Academy of Sciences*

² *Research Institute for Artificial Intelligence “Mihai Drăgănescu”, Romanian Academy*

³ *University of Bucharest, Faculty of Letters*

{iva, zarka, maria}@dcl.bas.bg,
vergi@racai.ro, mihaella.ionescu@yahoo.com

Abstract

We propose a framework for the unified description of verbal multiword expressions (VMWEs) and their derivatives in wordnet. The framework is meant as linguistically independent and applicable to the encoding of VMWEs in a variety of linguistic resources. It will facilitate cross-linguistic studies on verbal MWEs, and will have application in NLP tasks such as syntactic parsing, semantic disambiguation, machine translation. Our current efforts are focused on the linguistic features needed for the morphological, syntactic, semantic, and derivational description of VMWEs in a cross-linguistic perspective with a special focus on Bulgarian and Romanian, and with a possible extension towards English and other languages.

Keywords — verbal multiword expressions, wordnet, Universal Dependencies, derivation.

1. Introduction

The rising interest of language specialists and language engineers in the phenomenon of multiword expressions (MWEs) in recent years has led to the use of various definitions, descriptions and classifications from different perspectives (Baldwin and Kim, 2010; Constant *et al.*, 2017; Savary *et al.*, 2018). MWEs are described as combinations of two or more tokens which display lexical, morphosyntactic, semantic, pragmatic and/or statistical idiosyncrasies (Baldwin and Kim, 2010). The most cited definition is “idiosyncratic interpretations that cross word boundaries (or spaces)” (Sag *et al.*, 2002). The term MWEs is used synonymously with *multiword units*, *multiword lexical items*, *phraseological units*, *fixed expressions*, etc. (Baldwin and Kim, 2010). The main features emphasized in MWEs’ description vary between compositionality and decomposability (Villavicencio *et al.*, 2004). Based on them, different classifications, types and subtypes are offered as the NLP researchers (Sag *et al.*, 2002; Savary *et al.*, 2019, among others) underline the importance of identification of different types of MWEs in each language and of the formalized description of their characteristics.

This paper deals with verbal MWEs (VMWEs) as a subcategory of MWEs according to the description of the PARSEME shared task on automatic identification of VMWEs, edition 1.1. (Ramisch *et al.*, 2018). Considering the multilingual characteristics of the project, in a previous research we applied the PARSEME classification of VMWE types on the wordnet data as discussed in Section 2. We elaborate on these Romanian – Bulgarian – English parallel data and go further by proposing a framework for a unified description of VMWEs and their derivatives combining the rich semantic information in wordnets with information of MWEs' grammatical and syntactic idiosyncrasies.

Section 2 contains an overview of previous work on the description of MWEs in dictionaries, while Section 3 deals with the treatment of MWEs particularly in wordnet and Universal Dependencies. Section 4 focuses on the elements of the description of each MWE entry, which include five categories: technical, morphological, syntactic, semantic and derivational features, and touch upon the necessity for defining language specific features at all levels of description. Section 5 presents the conclusions.

2. Current state of the research into the description of MWEs

The significance of MWE lexicons in NLP has been specifically underlined as their word-for-word interpretations are usually misleading or wrong (Savary *et al.*, 2019). This is due to the idiosyncrasy of MWEs at the morphological, syntactic and semantic level. The description of MWEs focused on the dictionary description of their components and structure has been the object of extensive research with a view to different languages and methods for MWE identification (Baldwin and Kim, 2010; Calzolari *et al.*, 2002; Gralinski *et al.*, 2010; Gregoire, 2010; Savary *et al.*, 2018; Villavicencio *et al.*, 2004, among others). Dyvik *et al.* (2019) and Markantonatou *et al.* (2019) underline the complexity and versatility of the regular and idiosyncratic phenomena exhibited by the MWEs. Timm Lichte *et al.* (2019) define a lexical resource as an electronic representation of lexical encodings usable in NLP applications. They conclude that the general-purpose encoding format should cover: (i) the scaled regularity of the properties of MWEs, (ii) co-occurring properties of different degrees of regularity in each MWE, (iii) rare idiosyncratic properties, (iv) the unpredictability of shared properties. The authors propose a set of 10 defective and restrictive properties of MWEs regular for their linguistic realization based on multilingual observations.

In another multilingual approach, Calzolari *et al.* (2002) underline the need for lexicons containing MWEs accompanied by syntactic and semantic information, with correspondences from one language to another. Their recommendations for the description of MWEs in lexicons with a focus on light verb constructions and on noun compounds include: morphological and syntactic constraints on the components, the syntactic pattern instantiated by each MWE (when existing), the semantic description of each component and identification of the semantic relations between components, listing of the particular instantiations of a certain semantic paradigm, and specification of the degree of lexicalization of each MWE. The authors also stress the need for a structural approach to such a lexicon, exhibiting the relational nature of each MWE.

Particular attention should be paid to the PARSEME initiative, which stands out from the rest due to the multilingual perspective adopted by it, *i.e.* the equal importance attributed to the various phenomena across languages. It focuses only on verbal MWEs, proposing and later refining (in a second version of the guidelines) types and subtypes for them, grouped as universal and quasi-universal categories (Savary *et al.*, 2018).

3. VMWEs in the framework of linguistic resources

3.1. Treatment of VMWEs in Wordnet

The set of MWEs in Princeton WordNet (PWN) has been discussed for different purposes in various languages, such as Arabic (Attia *et al.*, 2010) and Slovene (Vintar and Fiser, 2011), among others. A similar approach to the one adopted here is presented by Declerck *et al.* (2019) who propose porting two German resources – OdeNet (Open German WordNet) and the German version of the MMORPH morphological analyzer.

However, to the best of our knowledge there is no other cross-lingual research on MWEs of any kind based on wordnet data. The nature and the structure of the lexical semantic network afford broad opportunities for mono- and multilingual analysis of MWEs. The wordnet literals are lexemes expressed by a single word, by multi-token words or by compositional phrases. The second type represents the characteristics of MWEs – separate lexemes composed of 2 or more graphical words, with complex and unique concept meaning. They differ from the literals expressed by phrases with compositional meaning, which often serve as a paraphrastic solution to lexical gaps.

Several advantages of using wordnets as a database of MWEs may be pointed out: (a) the synset to which a MWE belongs has a semantic description expressed by the synset gloss; (b) the semantic relations between the literals in a synset: a MWE may be synonymous with a simple word or with (an)other MWE(s) in the same synset; (c) the semantic relations between the respective synset and other synsets in the wordnet, that is the wordnet structural relations (hyponymy, meronymy, troponymy, etc.), which further refine the semantics of the MWEs; (d) the existence of aligned wordnets provides an opportunity for multilingual comparisons; (e) those comparisons may include research on different lexical groups such as different parts of speech or MWEs, etc.; (f) wordnet provides parallel multilingual data which can be used for research on the lexicalization of the same meaning across languages with MWEs or for comparing different types of MWEs with a view to their meaning, structure and morphology.

The development of the Bulgarian WordNet (BulNet) and the Romanian WordNet (RoWN), both aligned to PWN (Fellbaum, 1998), started within the BalkaNet project (Tufiş *et al.*, 2004); after that BulNet and RoWN have been developed in parallel, following similar tasks and common projects which have provided crosspoints for the work of the teams. Both wordnets have been enriched with morphosemantic and derivational relations (Barbu Mititelu, 2013; Koeva *et al.*, 2016). Currently, BulNet¹ contains 92,910 manually verified synsets comprising a total of 164,418 literals, out of

¹ BulNet's user interface <http://dcl.bas.bg/bulnet/> provides access to both BulNet and RoWN, among other languages, as well as parallel visualization of corresponding synsets in two wordnets (Rizov *et al.*, 2015).

which 28.3% (63,930 literals) are multiword expressions. RoWN² contains 59,348 synsets in which 85,277 literals (representing 50,480 unique ones) occur, out of which 23.5% (20,031 literals) are multiword expressions.

The crosspoint for the work of the two teams on establishing a lexicon representation of VMWEs in Bulgarian (as a Slavic language) and Romanian (as a Romance language) started with the work on the PARSEME project and the annotation, encoding and classification of VMWEs. As a result, a comparative analysis of the VMWEs in the two wordnets was offered (Barbu Mititelu *et al.*, 2019). After the identification of VMWEs in the two resources, we found that 3,237 literals are VMWEs in BulNet (where aspectual verb pairs are grouped and counted as a single VMWE) and 1,752 literals are VMWEs in RoWN. The data was analysed cross-linguistically and the occurrences were classified and labeled into the subtypes defined in the PARSEME shared task edition 1.1. The set intersection of the synsets containing VMWEs in BulNet and in RoWN comprises 944 synsets, which represents 40% of the verbal synsets containing MWEs in BulNet and 45% of those in RoWN. This parallel data is big enough to serve as grounds for further grammatical and semantic comparisons between the two languages.

Moreover, we observed the distribution of VMWE types within the WN structure, presented in Table 1 below. For BulNet, we present data where aspectual verb pairs are grouped and counted as a single VMWE (Barbu Mititelu *et al.*, 2019).

Table 1: Types of MWEs with their distribution in RoWN and BulNet.

Type of VMWE	RO	RO #	BG	BG #	EN
verbal idioms (VID)	<i>citi printre rânduri</i>	614	<i>cheta mezhdu redovete</i>	775	<i>read between the lines</i>
light-verb constructions (LVC.full)	<i>lua parte</i>	102	<i>vzemam uchastie</i>	465	<i>take part</i>
light-verb constructions (LVC.cause)	<i>lăsa loc</i>	42	<i>hvărlyam vāv vāztorg</i>	63	= RO: <i>allow for, provide</i> = BG: <i>bring down (cause to be enthusiastic)</i>
reflexive verbs (IRV)	<i>se înfură</i>	989	<i>yadosvam se</i>	1,822	<i>anger, see red</i>
inherently adpositional verbs (IAV)	<i>not annotated</i>		<i>razbiram ot</i>	39	<i>be good at</i>

² RoWN can also be queried at <http://relate.racai.ro/>.

These Bg-Ro correspondences enable the analysis of the concepts which the two languages tend to lexicalize as VMWEs and the study of the degree to which such concepts are lexicalized by the same type of VMWEs. It is also a starting point for exploring the morphosyntactic idiosyncrasies of the VMWE types in the two languages.

3.2. VMWEs is the framework of Universal Dependencies

Universal Dependencies (UD³) is a framework for consistent annotation of grammar (parts of speech, morphological features, and syntactic dependencies) across different natural languages. The UD taxonomy contains three special relations aimed at describing MWEs and their properties: (a) ‘fixed’ for fixed grammaticized MWEs; (b) ‘flat’ for exocentric semi-fixed MWEs such as names and phrases based on coordination; and (c) ‘compound’ for expressions involving a head and dependent components. Most VMWEs fall in the third category. However, for our purposes the simple annotation of VMWE components is not sufficient since we are interested in both their automatic identification and grammatical annotation for which we need a complex description of their internal structure, compatibility between components and their syntactic realization. For that purpose, we adopt the UD taxonomy in general and apply it to describe the internal structure of VMWEs.

A similar approach is presented by Kahane *et al.* (2018), who describe the impact of idiomaticity on the syntactic annotation using the UD formalism based on the distinction of syntactically irregular structures from semantically non-compositional units. The authors analyze MWEs with an internal syntactic structure as a unit with its own POS according to their level of syntactic regularity using the CoNLL-U format.

4. Cross-linguistic description of MWEs in a wordnet – a provisional model

In this section we make an inventory of elements that we consider necessary in the description of VMWEs in a lexicon, making use of as much relevant information from wordnets as necessary, and we explain their importance. We conceive this MWE lexicon as a standoff file and therefore it must provide a link to the respective synsets to which the MWEs belong. The lexicon entries are the MWEs themselves. For the moment, we focus only on VMWEs, but we expect to extend it to other parts of speech as well. There are several types of information registered for each VMWE: technical details, lemmatized form, morphological, syntactic, semantic, derivational characteristics. Each level of description is extendable and may be further refined with the analysis of data. The elements discussed below are based on the consideration of the necessary elements in Bulgarian and Romanian with a recourse to English as well.

4.1. Technical description

The technical details comprise of information about the unique identifiers of the entities within the resource as well as their linking to other resources.

(Tech1) Synset ID (ILI – interlingual index) of the wordnet synset containing the VMWE (or unique BulNet or RoWN ID in case the synset is language specific);

³ <https://universaldependencies.org>

(Tech2) VMWE ID: the ID of a given VMWE contains its synset ID and a unique index to distinguish between multiple MWEs in a synset, as well as ambiguous VMWEs found in more than one synset. The ID serves as an identifier when we make references to the VMWE (e.g., in the description of derivatives, see Deriv1 below).

In such a way VMWEs are: (a) linked to other language resources such as BulNet or RoWN with information about synonymous single words and other MWEs and associated with all the synset-level linguistic information provided in the synset; (b) identified in terms of its cross-linguistic correspondences, thus allowing for the multilingual perspective of our work. For example, for the ID below we find equivalent VMWEs in all three languages we work with:

(1) Synset ID: eng-30-00621504-v (shared by the three examples)

(1a) BG: {*cheta mezhdu redovete:1*} VMWE ID: eng-30-00621504-v_bg01

(1b) RO: {*citi printre rânduri:1*} VMWE ID: eng-30-00621504-v_ro01

(1c) EN: {*read between the lines:1*} VMWE ID: eng-30-00621504-v_en01

4.2. Morphological description

(Morph1) Lemma of the VMWE (non-abstract lemma) and a lemmatized form of each component of the VMWE (abstract lemma) (Savary, 2008): this is useful for those types of MWEs whose components have (rich) inflection. Identifying all the occurrences of a MWE and of all its occurring forms in a lemmatized corpus will definitely be helped by access to a lexicon where such information is registered.

(Morph2) Regular morphosyntactic representation, which consists of the set of forms realized by the expression's head in combination with the set of forms realised by the non-head components. This description covers VMWEs with full paradigm of the head verb. In case an inflection lexicon exists and it contains all the forms of the components of the VMWEs, a link to this resource could be used instead.

(Morph3) Restrictions on the paradigmatic realization of the verb head. The forms of a MWE's paradigm may be restricted to certain morphosyntactic features of the head or of the dependents. The restrictions may concern various morphological features, such as person, number, tense, mood, polarity, etc. Such restrictions tend to be most typical of verb idioms (VID). Below we list a couple of examples:

- a) restricted use of the VMWE in (a) particular **person(s)** of the verb form, e.g. 3rd person, both sg. and pl., BG *idva mi naum / idvat mi naum* 's.th. comes to my mind' "it occurs to me / they occur to me";
- b) restricted use of the VMWE in a particular **number** form, e.g. pl., BG *broim se na prăstite na ednata rāka* "be counted on the fingers of one hand".
- c) restricted use of the VMWE in a **negative** form: e.g., RO *nu privi cu ochi buni* ('not watch with eyes good', "regard with disfavour") is always used with the negative marker *nu* "not"; the same goes for BG *ne iskam akāl nazaem* ('not want brain as a loan', "to not need unsolicited advice").

An extensive inventory of the types of restrictions found in Romanian and Bulgarian would emerge in the course of the analysis of VMWEs included in the lexicon.

(Morph3) Restrictions on the inflected forms of the dependent components in the VMWE. Such a field is defined for each dependent and is used to encode explicitly any restrictions on its possible forms as part of the VMWE.

For example, the Romanian light verb construction *face vizită* (‘make visit’, ‘pay a visit’) can occur in more than 150 forms in the language: the verb may have more than 40 forms, the noun has 7 forms, and each verbal form can combine with only 4 of the noun forms: it cannot occur in the oblique case (that is, in the forms *vizitei*, *vizitelor*). Having access to such information will help distinguish between a context such as (2a) and (2b), the latter being a mere co-occurrence of the verb and noun, not a VMWE:

(2a) RO: *El le-a făcut vizite numeroase acestor persoane.*

‘He CL.Dat.3.pl.-has made visits numerous to-these people.’

‘He paid many visits to these people.’

(2b) RO: *Criticile pe care el le-a făcut vizitelor lor au fost numeroase.*

‘Criticisms on which he CL.Dat.3.pl.-has made to-visits their have been numerous.’

‘The complaints against their visits were numerous.’

These restrictions may be cast either in positive (e.g., *vizită* can occur only in the direct case) or in negative terms (e.g., *vizită* cannot occur in the oblique case). Theoretically, the most economic way of formulating the restrictions should be used in each case.

4.3. Syntactic description

The syntactic description is based on the relations within the UD framework. We opted for this framework given its multilingual orientation and the concern for identifying what is universal in the syntax and morphology of all or most languages, while offering the possibility for defining language characteristics in the same framework⁴. The information will consist of:

(Synt1) Internal syntactic structure of the VMWE, which describes the number and the types of the components and the syntactic relations between them.

We have identified the syntactic structures that are common to Romanian and Bulgarian VMWEs. Table 2 presents the structures with relatively high frequency in the data.

⁴ <https://universaldependencies.org/u/dep/index.html>

Table 2: Types of internal syntactic structures across Romanian and Bulgarian data

Relation	Description of relation	Example RO	Example BG
V + obj	linking V to the entity acted upon or undergoing change	<i>da declarație</i> (give declaration, “declare”)	<i>vzemam reshenie</i> (“make a decision”)
V + obl	linking V to a nominal as a non-core (oblique) argument or adjunct	<i>înceta din viață</i> (cease from life, “die”)	<i>poemam v svoi rătse</i> (“take into one’s own hands”)
V+advmod	linking V to a (non-clausal) adverb or adverbial phrase to modify a predicate	<i>da afară</i> (give outside, “remove from job, fire”)	<i>vzemam prevedid</i> (“take into account”)
V + nsubj	the VMWEs is made up of a verb and a subject	<i>fura somnul</i> (steal sleep-the, “fall asleep”)	<i>zvezdata mi izgryava</i> (“one’s star is rising”)
V + nsubj + obl	linking V to a subject and a non-core (oblique) argument or adjunct	<i>îngheța sângele în vine</i> (freeze blood-the in veins, “get cold feet”)	<i>krāvta zamrāzva v zhilite mi</i> (blood freezes in my veins, “get cold feet”)
V + obj + obl	linking V, an object and a non-core (oblique) argument or adjunct	<i>găsi drumul în viață</i> (find road-the in life, “find one’s way in life”)	<i>tsepya stotinkata na dve</i> (split the penny in two, “be stingy”)

The typology presented here does not aim at a complete inventory of possible structures, but rather at covering the most frequent cases and at showing that the framework can be extended further to include specific structures. Moreover, parallel analysis of Bulgarian and Romanian VMWEs can reveal types specific to either of the languages.

(Synt2) Possible dependents of the elements of the VMWE. Some MWE components may have dependents, others may not. These dependents are either arguments, that is, obligatory dependents of the VMWE components, or adjuncts, *i.e.* dependents that are not obligatory for the grammaticality of the sentence. Consider the RO *citi printre rânduri* or its BG equivalent *cheta mezhdu redovete*: the verb can take dependents of the type subject (nsubj or csubj in UD terminology) or direct object (obj in UD), but the noun cannot be modified by any word.

In contrast with other MWEs, a dependent within light-verb constructions in particular may and readily takes a dependent of its own, e.g. BG *vzemam reshenie* (“make a decision”) > *vzemam vazhno reshenie* (“make an important decision”), where the dependent noun, which is an object, may be modified (nmod in UD). We would like to keep track of the ability of the MWE elements to take modifiers of their own, especially ones that intervene in the internal structure of the VMWE as this may provide useful

information about the distance between the individual elements of a MWE in running text – a feature that affects the automatic recognition of MWEs.

(Synt3) The word order of the VMWE components and of the possible dependents. Whenever there is no restriction on the order of the VMWE components, it means that they occur in any order, although the pragmatics of each of the resulting construction could differ. Any restrictions on the order of the VMWE components are formulated in this field. For example, in the RO VMWE *da ortul popii* (‘give coin-the to-priest-the’, “die”) the obj *ortul* always precedes the indirect object *popii* (iobj in UD); in the BG VMWE *na star krastavichar krastavitsi prodavam* (‘to an old cucumber-seller cucumbers sell’, “to try to cheat someone with experience”) the obl *na star krastavichar* precedes the obj *krastavitsi* while the verb can be either at the front or at the end.

Similarly, whenever an external component can be inserted within the VMWE, its possibilities of occurrence are clearly indicated. For example, the Romanian VMWE *da foc* takes an iobj, but it must occur to the right of the whole expression, that is, after the dobj *foc*, in spite of the fact that dobj and iobj may occur in any order in Romanian.

4.4. Semantic description

(Sem1) The MWE type. For VMWEs the types are those defined within PARSEME shared task edition 1.1 (Ramish *et al.*, 2018), which are applicable to the respective language. Besides the universal types – verbal idioms (VID) and light verb constructions (LVC), the latter with its two subtypes (LVC.full and LVC.cause), both Bulgarian and Romanian have the following quasi-universal types: inherently reflexive verbs (IRV) and inherently adpositional verbs (IAV).

(Sem2) Usage information. This field provides real examples and relevant restrictions on the usage of VMWEs, which may be automatically retrieved from the respective wordnet, if available, or added by the lexicographer. For example, many idioms are specific to the informal use: RO *bate la ochi* ‘beat at eyes’ “catch someone’s eye”; BG *udryam kyoravoto:1* ‘hit the blind’ “hit the jackpot”.

(Sem3) Connotation. In this field we mark the positive, negative or neutral connotation of a given MWE, a feature which is all the more relevant as verbal idioms are a means of linguistic expressivity (Coşeriu, 2000). This information is either obtained from available resources, such as SentiWordNet (Baccianella *et al.*, 2010), or supplied manually by the two teams. In the former case the connotation values are ascribed from the respective synsets which have been assigned values from SentiWordNet (transferred automatically to BulNet and RoWN). For instance, the counterpart synsets BG: *{puka mi:1, dreme mi:2, davam pet pari:1, dam pet pari:1, davam puknata para:1, dam puknata para:1}*, RO: *{da doi bani:1, da două parale:1}*, EN: *{care a hang:1, give a hoot:1, give a hang:1, give a damn:1}* are assigned a positive value of +0.125 and a negative value of −0.375. Even so, attention needs to be paid to the fact that languages may behave differently and the same meaning may be rendered by MWEs with a different connotation (positive or negative, as the values in the example show) in each language or the connotation may differ from one literal to the other in the same synset.

4.5. Derivation

(Deriv1) The derivational information is presented as a list of the possible derivatives for each VMWE lexicon entry. Derivatives are indicated in their form occurring in the respective wordnet and accompanied by the ID of the synset to which they belong. If the derivatives are not implemented in the wordnet yet, then the ID remains unspecified.

A VMWE may be derivationally related to another MWE: RO *bate la ochi* – *bătător la ochi*, the latter being an adjectival MWE obtained from the VMWE *bate la ochi* by means of adding the suffix *-ător* to the verb *bate*. A MWE can also be derivationally related to a simple literal, e.g. RO *face mușama* (‘make tablecloth’, ‘cover up’) – *mușamaliza*, the latter verb being derived from the noun *mușama* and having the same meaning as the whole VMWE *face mușama*. Similar derivational patterns may be found in Bulgarian, e.g. BG *mălcha kato păn* – *mălchaliv kato păn* (‘keep as mute as a poker’, ‘as mute as a poker’), where the latter is an adjectival MWE related to the verbal MWE and formed with the adjectival suffix *-liv*. Similarly, verbal MWEs may be related to single verbs: BG *izvadya na svetlo* (‘bring to light’, ‘legalize’) – *izsvetlya* (‘make brighter’, ‘legalize’) illustrates the derivation of the verb *izsvetlya* from the adjective *svetlo* which is part of the MWE. The relationship between a VMWE and a derivationally related single verb will be accounted for by assigning a special relation.

Our focus is on the verb-to-noun derivation. The syntactic properties of the derived NPs depend mainly on the properties of the source VMWE. Derivation depends on the morphological characteristics of the VMWE and its head, the internal structural components and their restrictions; irregularities in the derivation (e.g., additional restrictions on otherwise possible forms or imposed changes on forms) are listed in this field as well, e.g. BG *razbivam (nechie) särtse* (‘break someone’s heart’ – the noun is in the singular) > *razbivatch na särtsa* (‘breaker of hearts’, ‘heart-breaker’ – the noun is in the plural). A comprehensive list of semantic and syntactic differences between derivationally related MWEs is presented by Barbu Mititelu and Leseva (2018).

4.6. Language-specific features

As each language may exhibit certain restrictions or peculiarities that may be too specific, at each level of the model we provide a separate field for describing language specific features. The thorough description of such features based on rich data will provide further grounds for comparative analyses and for streamlining the entire model.

5. Conclusions

We have proposed an outline of an entry in an electronic MWEs lexicon which combines linguistic information of various kinds, all of them relevant for the task of MWEs identification in texts. Our proposal makes use of several widely accepted and used frameworks: aligned wordnets, UD formalism, PARSEME VMWEs types. Our future work is aimed at the extensive description of VMWEs in RoWN and BulNet

following the proposed scheme. More fine-grained typology of syntactic and semantic types of VMWEs is also possible for each language depending on the research purposes.

The proposed framework is meant to be language independent and thus will facilitate cross-linguistic studies on VMWEs and will have applications in various NLP tasks such as semantic analysis, word sense disambiguation, syntactic parsing, information extraction, text summarization, machine translation.

Acknowledgements

This work has been undertaken under the bilateral project *Enhancing Multilingual Language Resources with Derivationally Linked Multiword Expressions* (2018-2020) between the Institute for Bulgarian Language at the Bulgarian Academy of Sciences and the Research Institute for Artificial Intelligence at the Romanian Academy. We would also like to thank two anonymous reviewers for their helpful suggestions.

References

- Attia, M., A. Toral, L. Tounsi, P. Pecina and van Genabith, J. (2010). Automatic extraction of Arabic multiword expressions. In *Proceedings of the Workshop on Multiword Expressions: From Theory to Applications (MWE 2010)*, August, 2010, Beijing, China, 19-27.
- Baccianella, S., Esuli, A. and Sebastiani, F. (2010). SENTIWORDNET 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the 7th Conference on Language Resources and Evaluation (LREC 2010)*, May 17-23, Valletta, Malta, 2200-2204.
- Baldwin, T., Kim, S.N. (2010). Multiword expressions. In Indurkha, N. and Damerau, F.J. (eds.), *Handbook of Natural Language Processing*, 2nd edition. Boca Raton, FL, USA: CRC Press, Taylor and Francis Group, 267-292.
- Barbu Mititelu, V. (2013). Increasing the effectiveness of the Romanian WordNet in NLP applications. *Computer Science Journal of Moldova*, 21(3):320-331.
- Barbu Mititelu, V., Leseva, S. (2018). Derivation in the domain of multiword expressions. In Manfred Sailer and Stella Markantonatou (eds.), *Multiword Expressions: Insights from a Multi-lingual Perspective*. Berlin, Germany: Language Science Press, 215-246.
- Barbu Mititelu, V., Stoyanova, I., Leseva, S., Mitrofan, M., Dimitrova, T., Todorova, M. (2019). Hear about verbal multiword expressions in the Bulgarian and the Romanian WordNets straight from the horse's mouth. In *Proceedings of the Joint Workshop on Multiword Expressions and WordNet (MWE-WN 2019)*, August 2, Florence, Italy 2-12.
- Calzolari, N., Fillmore, Ch., Grishman, R., Ide, N., Lenci, A., MacLeod, C., Zampolli, A. (2002). Towards best practice for multiword expressions in computational

- lexicons. In *Proceedings of the 3rd Conference on Language Resources and Evaluation (LREC 2002)*, 29 May - 2 June, Las Palmas, Spain, 1394-1940.
- Constant, M., Eryiğit, G., Monti, J., van der Plas, L., Ramisch, C., Rosner, M., Todiraşcu, A. (2017). Multiword expression processing: A survey. *Computational Linguistics*, 43(4): 837-892.
- Coşeriu, E. (2000). *Lecții de lingvistică generală*. Chişinău: ARC.
- Declerck, T., M. Siegel, S. Racioppa. (2019). Using OntoLex-Lemon for representing and interlinking German multiword expressions in OdeNet and MMORPH. In *Proceedings of the Joint Workshop on Multiword Expressions and WordNet (MWE-WN 2019)*, August 2, Florence, Italy, 22-30.
- Dyvik, H., Smørdal Losnegaard, G. and Rosén, V. (2019). Multiword expressions in an LFG grammar for Norwegian. In Yannick Parmentier and Jakub Waszczuk (eds.), *Representation and Parsing of Multiword Expressions: Current Trends*. Berlin, Germany: Language Science Press, 69-108.
- Fellbaum, C. (ed.). (1998). *WordNet: An Electronic Lexical Database*. MIT Press.
- Gralinski, F., Savary, A., Czerepowicka, M., and Makowiecki, F. (2010). Computational lexicography of multiword units. How efficient can it be? In *Proceedings of the Workshop on Multiword Expressions: From Theory to Applications (MWE 2010)*, August, 2010, Beijing, China, 2-10.
- Gregoire, N. (2010). DuELME: A Dutch electronic lexicon of multiword expressions. *Language Resources and Evaluation*, 44:23-39.
- Kahane, S., K. Gerdes, M. Courtin. (2018). Multi-word annotation in syntactic treebanks: propositions for Universal Dependencies. In *Proceedings of the 16th International Conference on Treebanks and Linguistic Theories (TLT 2018)*, January 23 - 24, Prague, Czech Republic, 181-189.
- Koeva, S., S. Leseva, I. Stoyanova, T. Dimitrova, and M. Todorova. (2016). Automatic Prediction of Morphosemantic Relations. In *Proceedings of the Eighth Global Wordnet Conference (GWC 2016)*, Research Institute for Artificial Intelligence – Romanian Academy, January 27 - 30, Bucharest, Romania, 168-176.
- Lichte et al. (2019). Lichte, T., Petitjean, S., Savary, A. and Waszczuk, J. Lexical encoding formats for multi-word expressions: the challenge of “irregular” regularities. In Yannick Parmentier and Jakub Waszczuk (eds.), *Representation and Parsing of Multiword Expressions: Current Trends*. Berlin, Germany: Language Science Press, 2019, 1-33.
- Markantonatou, S., Samaridi, N. and Minos, P. (2019). Issues in parsing MWEs in an LFG/XLE framework. In Yannick Parmentier and Jakub Waszczuk (eds.), *Representation and Parsing of Multiword Expressions: Current Trends*. Berlin, Germany: Language Science Press, 109-126.
- Ramisch, C., Cordeiro, S.V., Savary, A., Vincze, V., Barbu Mititelu, V., Bhatia, A., Buljan, M., Candito, M., Gantar, P., Giouli, V., Güngör, T., Hawwari, A., Iñurrieta, U., Kovalevskaitė, J., Krek, S., Lichte, T., Liebeskind, C., Monti, J., Parra Escartín, C., QasemiZadeh, B., Ramisch, R., Schneider, N., Stoyanova, I., Vaidya, A., Walsh, A. (2018). Edition 1.1 of the PARSEME Shared Task on

- automatic identification of verbal multiword expressions. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, August 25 -26, Santa Fe, USA, 222-240.
- Rizov, B., Dimitrova, T., Barbu Mititelu, V. (2015). Hydra for Web: A Multilingual Wordnet Viewer. In *Proceedings of the 11th International Conference "Linguistic Resources and Tools for Processing the Romanian Language" (ConsILR 2015)*, November 26 -27, Iași, Romania, 19-30.
- Sag, I. A., Baldwin, T., Bond, F., Copestake, A., Flickinger, D. (2002). Multiword expressions: a pain in the neck for NLP. In *Computational Linguistics and Intelligent Text Processing*. Berlin Heidelberg, Germany: Springer Berlin Heidelberg, 2002, 1-15.
- Savary, A. (2008). Computational inflection of multiword units, a contrastive study of lexical approaches. *Linguistic Issues in Language Technology*, 1(2).
- Savary, A., Candito, M., Barbu Mititelu, V., Bejček, E., Cap, F., Čéplö, S., Cordeiro, S.R., Eryiğit, G., Giouli, V., van Gompel, M., HaCohen-Kerner, Y., Kovalevskaitė, J., Krek, S., Liebeskind, C., Monti, J., Parra Escartín, C., van der Plas, L., QasemiZadeh, B., Ramisch, C., Sangati, F., Stoyanova, I., Vincze, V. (2018). PARSEME multilingual corpus of verbal multiword expressions. In Markantonatou, S., Ramisch, C., Savary, A., Vincze, V. (eds.), *Multiword Expressions at Length and in Depth: Extended Papers from the MWE 2017 Workshop*. Berlin, Germany: Language Science Press, 87-147.
- Savary, A., Cordeiro, S., Ramisch, C. (2019). Without lexicons, multiword expression identification will never fly. A position statement. In *Proceedings of the Joint Workshop on Multiword Expressions and WordNet (MWE-WN 2019)*, August 2, Florence, Italy, 79-91.
- Tufis, D., Cristea, D., Stamou, S. (2004). BalkaNet: aims, methods, results and perspectives: a general overview. *Romanian Journal on Information Science and Technology* 7: 9-34.
- Villavicencio, A., Copestake, A., Waldron, B., and Lambeau, F. (2004). The lexical encoding of MWEs. In Tanaka, T., A. Villavicencio, F. B., and Korhonen, A., eds., In *Proceedings of the ACL 2004 Workshop on Multiword Expressions: Integrating processing*, July 2004, Barcelona, Spain, 80-87.
- Vintar, Š. and Fišer, D. (2011). Enriching Slovene WordNet with domain-specific terms. *Translation: Computation, Corpora, Cognition* [online]. 2011. Vol. 1, no. 1, 29-44.

FRAME SPECIALISATION MOTIVATED BY INTER-FRAME RELATIONS IN FRAMENET

SVETLOZARA LESEVA, IVELINA STOYANOVA, MARIA TODOROVA,
HRISTINA KUKOVA

*Department of Computational Linguistics, Institute for Bulgarian Language,
Bulgarian Academy of Sciences*

{zarka, iva, maria, hristina}@dcl.bas.bg

Abstract

The paper deals with the conceptual and syntactic specialisation that takes place between more general and more specific or constrained semantic and lexical descriptions as represented in two previously linked resources – FrameNet and WordNet – where lexical units (LUs) instantiating a particular frame in FrameNet are linked to a semantically corresponding synonym set in WordNet. Using the inheritance of conceptual and lexical features as well as other regularities, we observe the modifications that occur in the conceptual and semantic structure between more general and more specific frames and synsets: (1) reducing the number of core frame elements by incorporating one of them in the verb's meaning; (2) reducing the scope of the frame through imposing stricter selectional restrictions; (3) profiling a different frame element from the one profiled by the hypernym; (4) inclusion/exclusion of frame elements. These types of specialisation are being studied as a point of departure for defining more narrow-scope frames that would allow for more precise predictions about the selectional restrictions and the syntactic realisation of frame elements.

Key words — conceptual structure, frame semantics, FrameNet, WordNet

1. Introduction

The research presented in this paper is an essential part of our work on enhancing WordNet with information about the conceptual structure of verbs based on the mappings between two large semantic resources, WordNet (WN) and FrameNet (FN). This information includes linking WN synsets with FN frames and providing description of the conceptual elements that receive expression as verbs' arguments and adjuncts as well as of the selectional restrictions imposed on these elements.

Our analysis focuses on the structural features of the resources and the possible structural improvements of FrameNet based on systemic features (e.g., inheritance, causativity, etc.), including definition of new conceptual frames.

Section 2 outlines the linguistic resources employed in the proposed research, along with their relevant features as well as the linguistic preliminaries, the motivation for the presented work and how it is situated against the background of previous studies. Section 3 discusses the main factors and procedures for the specification of frames in

FrameNet, including the incorporation of arguments, frame profiling, the introduction of specific restrictions to frame elements and the consideration of frame-to-frame relations. In the conclusion of the paper, we provide some directions for future work, including how the developed resource is applicable to other languages besides English, including Bulgarian and Romanian (Section 4).

2. Prerequisites

2.1. Linguistic resources

WordNet (Miller, 1995; Fellbaum 1998) is a large lexical semantic database that represents comprehensively the conceptual and lexical knowledge in the form of a network (graph) whose nodes denote cognitive synonyms (synsets) interconnected through a number of conceptual-semantic and lexical relations. The main relation that determines WordNet's structure (as reflected in the hierarchical treelike organisation of nouns and verbs) is a relation of inheritance of conceptual and lexical features between synsets. The respective pairs of synsets are linked through the relation of hypernymy/hyponymy.

FrameNet (Baker *et al.*, 1998; Baker, 2008) represents lexical and conceptual knowledge as conceptual structures (frames) which describe particular types of objects, situations, or events along with their components, called frame elements, or FEs (Baker and Ruppenhofer, 2002; Ruppenhofer *et al.*, 2016). Depending on their status, FEs may be core, peripheral or extra-thematic (Ruppenhofer *et al.*, 2016). For our purposes, we deal particularly with core FEs, which instantiate conceptually necessary components of a frame, and which in their particular configuration make a frame unique and different from other frames. To a lesser degree, we touch upon peripheral FEs, which mark notions such as Time, Place, Manner, Means, among others, and may be instantiated in any semantically appropriate frame. A lexical unit (LU) in FrameNet is a pairing of a word with a meaning; each LU is associated with a frame describing its conceptual semantics.

2.2. Related work

This proposal is inspired by previous work of different authors and teams aimed at the mapping of WordNet and FrameNet. Existing research may be grouped into the following main trends: (a) linking the two resources on the basis of the existing lexical and semantic information; (b) expanding the coverage of the mapping through various procedures.

Many previous efforts have aimed at maximising the advantages and contribution of the conceptual and lexical information encoded in the two resources. A number of proposals have been made towards their mutual enhancement, including the mapping of WordNet, FrameNet and VerbNet (Shi and Mihalcea, 2005), the elaboration of WordFrameNet¹ (Laparra and Rigau, 2010) and MapNet² (Tonelli and Pighin, 2009),

¹ <http://adimen.si.ehu.es/web/WordFrameNet>.

² <https://hlt-nlp.fbk.eu/technologies/mapnet>.

other FrameNet-to-WordNet mappings (Fernandez *et al.*, 2010), as well as more enhanced proposals, such as Semlink³ (Palmer, 2009), which unifies WordNet, FrameNet and VerbNet with PropBank, and its follow-up Semlink+ that brings in mapping to Ontonotes (Palmer *et al.*, 2014). Most of these efforts generally fall into the first strand of research noted above. As a result, they suffer from limited coverage due to the fact that they rely for the most part on the lexical mapping between FrameNet lexical units and WordNet literals (synset members). In the domain of verbs, in particular, out of 14,103 verb synsets, only 4,306 (30.5 %) have been successfully mapped through finding existing semantic equivalent (or near equivalent) pairs: <FN lexical unit: WN literal> (Leseva and Stoyanova, 2019).

An alternative or complementary approach is to try to ‘dig up’ non-explicit information about the frame membership of WordNet literals from the two resources. For instance, Burchardt *et al.* (2005) approach the expansion of the inter-resource coverage (mapping WN literals to FN frames) by weighing the candidate frames evoked by literals related to a given target literal through certain semantic relations (synonymy, hypernymy, antonymy). Another strategy which leads to substantial expansion of the coverage is to exploit the relational structure of the two resources – mapping frames to synsets on the basis of the inheritance of conceptual features in hypernym trees, i.e. assigning frames from hypernyms to hyponyms if other mapping procedures fail (Leseva *et al.*, 2018). A recent alignment that improves on this idea, including through validation procedures which result in refining the proposed mappings and enriching the FrameNet frame structure with systemic relations (e.g. causative, inchoative, etc. frame correspondences), was proposed by Stoyanova and Leseva (2019): this resource has been used in the observations made below.

The main drawback of the inheritance approach is that for deeper level WordNet synsets the inherited frames may be underspecified due to the lack of appropriately specific frames in FrameNet. Thus, a natural follow-up in the latter trend of research is to look for ways to define more elaborate frames as explained in detail in Section 2.3. At the current stage of research, we are exploring different possibilities by studying the data; in the long run our goal is not only to be able to extract relevant semantic information and to cast it in frame semantics terms but also to attempt automatic definition of frames.

2.3. Motivation

The proposed work is motivated by the observations on the frames that were automatically assigned to WordNet synsets. As noted above, when a synset could not be mapped directly to a FrameNet frame (i.e. the mapping procedure cannot find an existing mapping), this synset inherits the frame from its hypernym, as shown in Example 1. The frame assigned may either be appropriate for the semantic description of the hyponyms (e.g., Example 1), or it may be too general (e.g., Example 2).

Example 1. A WN subtree stemming from *{swell:2}*. FN frame assigned: Cause_expansion: ‘An Agent or a non-human Cause causes an Item to change its physical size.’

³ <https://verbs.colorado.edu/semLink/>

Synset	Definition (WN gloss)	Assigned frame	Steps ⁴
-- {swell:6}	cause to become swollen	Cause_expansion	0
----- {bulge:4}	cause to bulge or swell outwards	Cause_expansion	1
----- {bloat:2}	make bloated or swollen	Cause_expansion	1
----- {distend:2}	cause to expand by internal pressure	Cause_expansion	2
----- {tumefy:1}	cause to become very swollen	Cause_expansion	1

The synset {swell:6} was assigned the frame Cause_expansion on the basis of the lexical mapping between the verb in the synset and the lexical unit *swell.v* in the given frame (Leseva and Stoyanova, 2019). The remaining synsets are hyponyms of {swell:6}, either direct or indirect, and they have not been mapped in FrameNet. Therefore, they are assigned the frame Cause_expansion by transferring the frame from their hypernym on one or more steps. The definition of the frame describes sufficiently the semantics of {swell:6}'s hyponyms, which denote different manners or degrees of swelling. In contrast, when the frame assigned to the hypernym and transferred to its hyponyms is very general, it may not describe sufficiently the conceptual structure of the hyponyms. Such a case is illustrated below (Example 2) by the frame Cause_change, which specifies only that some change is being brought about to an entity. However, a more specific relevant frame may be available in FrameNet.

Example 2. A WN subtree stemming from {change:1}. FN frame assigned: Cause_change: ‘An Agent or Cause causes an Entity to change, either in its category membership or in terms of the value of an Attribute. In the former case, an Initial_category and a Final_category may be expressed, in the latter case an Initial_value and a Final_value can be specified.’

Synset	Definition (WN gloss)	Assigned frame	Steps
-- {change:1}	cause to change; make different	Cause_change	0
----- {gelatinize:3}	convert into gelatinous form or jelly	Cause_change	1

More specific frames involving the causation of change are usually distinguished in terms of the change of: (a) a particular attribute, such as *temperature* (Cause_temperature_change), *consistency* (Cause_change_of_consistency), *phase* (Cause_change_of_phase), *strength* (Cause_change_of_strength), *size* (Cause_expansion); or (b) the resulting state or category, such as *dryness* (Cause_to_be_dry), *wetness* (Cause_to_be_wet), *sharpness* (Cause_to_be_sharp), *fragmentation* (Cause_to_fragment).

In the particular case of {gelatinize:3}, we may argue that by assigning the frame of the hypernym, the attribute underlying the conceptual structure of the verb under discussion remains unspecified. As the process described by the verb involves the change of a substance from liquid to a jelly-like state, the attribute affected is *consistency*. On inspecting more specific frames involving the causation of change, we find out that a relevant frame is already available in FrameNet: Cause_change_of_consistency, which has the following definition: ‘An Agent or Cause changes the consistency of a Patient

⁴ The number in the last column shows whether the FN frame is directly assigned (0) or transferred from the hypernym and on how many steps (1, 2, etc.).

from its Initial_state to a Result'. The lexical units in this frame include verb such as *thin.v* 'cause to have a thinner consistency', *thick.v* 'cause to become thicker', among others. When, as in this case, an appropriate frame is found in FrameNet, we assign it, thus replacing the one mapped from the hypernym.

In other cases, there is no suitable frame that captures the conceptual structure of a particular verb or it is too general so a more specific one is needed. For instance, in Example 3 the synset $\{oxidize:2, oxidise:1, oxidate:2\}$ ⁵ is assigned the frame Cause_change from its hypernym; however, it does not account for the type of change that takes place.

Example 3. A WN subtree stemming from $\{change:1\}$. FN frame assigned: Cause_change.

Synset	Definition (WN gloss)	Assigned frame	Steps
-- $\{change:1\}$	cause to change; make different	Cause_change	0
----- $\{oxidize:2\}$	add oxygen to or combine with oxygen	Cause_change	1

The analysis supports the possibility to define a new frame that accounts for the Patient's attribute being changed, which in this case is its chemical composition. The new frame's definition is modelled on frames of similar type of change such as Cause_change_of_consistency and Cause_change_of_phase and may be stated as follows: 'A Cause or Agent causes a Patient to undergo a chemical reaction.' The difference between these particular frames lies in the specification of a different type of attribute for the Patient, which, in turn, is reflected in the different selectional restrictions imposed on the lexical items that may be selected as Patients.

2.4. Frame-to-frame relations in FrameNet

Frames in FrameNet are organised in a network whose nodes are related by means of frame-to-frame relations; in addition, some of them correspond to similar structural relations in WordNet. The relational structure of FrameNet is important in the evaluation of the transfer of frames from a hypernym to its hyponyms, as a more relevant or specific frame (than the one assigned) may be found among frames related to it. In particular, when the assigned frame is considered too general, the probable replacement candidates may be among the frames that are in an inheritance-based relation with the general frame. In addition, the types of frame specialisation, discussed in the next section often involve some of the relations sketched below:

(1) Inheritance: strong inheritance (based on the relations Is Inherited by – Inherits from) and **weak inheritance** (as defined by the relations Is Used by – Uses) posit a relationship between a more general (parent) frame and a more specific (child) frame (Ruppenhofer et al. 2016). The frame elements in the parent frame have correspondences in the child frame, which may be the same or more specific.

⁵ For consideration of space, in the examples we give only the first or most appropriate literal in the synset; some of the glosses are also abbreviated.

(2) Perspective (Is Perspectivized in – Perspective on) is a relation which indicates that a given situation viewed as neutral may be further specified by means of perspectivised frames representing different possible points-of-view on this neutral situation. Perspectivised frames of the same neutral frame can show different profiling of frame elements in the general frame.

(3) Causativity (Is Caused by – Causative of) is a systemic non-inheritance relation where one of the frames represents the causative counterpart of the other, stative or inchoative, frame (Ruppenhofer *et al.*, 2016: 85).

The specialisation of frames discussed in the next section are currently being studied as a point of departure for defining more narrow-scope frames that would allow for more precise selectional restrictions in terms of the syntactic realisation of frame elements. The new frames that are being defined need to be incorporated into FrameNet's structure by determining their frame-to-frame relations.

3. Specialisation of frames

Using the inheritance of conceptual and lexical features as well as other regularities, we observe the modifications that occur in the conceptual and semantic structure between more general and more specific frames and synsets, including, though not limited to:

- (1) Reducing the number of core frame elements by incorporating one of them in the verb's meaning, e.g. *{whip:4}* incorporates the FE Instrument ('whip') of *{strike:1}* in the frame Cause_harm assigned to both;
- (2) Profiling a different FE from the one profiled by the hypernym, e.g. *{rob:1}* (Robbery) profiles the Victim, while its hypernym *{steal:1; rip off:2; rip:4}* (Theft) profiles the stolen Goods;
- (3) Reducing the scope of the frame through imposing stricter selectional restrictions, e.g. *{drive:1}* (Operate_vehicle frame) as a hyponym of *{operate:3; control:3}* (Operating_a_system) applies only to land vehicles; other verbs in the frame impose other restrictions on the FE Vehicle;
- (4) Inclusion/exclusion of a causative/agentive FE, corresponding to a causative subevent in a pair of frames, such as *{break:5}* (Cause_to_fragment) and *{break:2, separate:10, fall apart:4, come apart:1}* (Breaking_apart).

3.1. Frames with incorporated arguments

Some frame elements in the conceptual frame of a verb might be incorporated in the verb's meaning; FEs of this kind include internal arguments such as Theme, Instrument, Patient, Container, etc. Although the incorporation does not trigger the need for modifying the frame or for defining a new one, it nevertheless poses restrictions in the syntactic realisation – the incorporated argument is rarely expressed explicitly in text unless it introduces a more specific concept of the same semantic class. Example 4 shows several cases of incorporated arguments.

Example 4. Incorporated arguments:

(4a) Theme: v *{butter:1}* ‘spread butter on’ and n *{butter:1}* ‘an edible emulsion of fat globules made by churning milk or cream’

(4b) Instrument: v *{knife:1}* ‘use a knife on’ and n *{knife:2}* ‘a weapon with a handle and blade with a sharp point’

(4c) Container: v *{bottle:2}* ‘put into bottles’ and n *{bottle:1}* ‘a glass or plastic vessel used for storing drinks or other liquids; typically cylindrical without handles and with a narrow neck’

(4d) Medium: v *{placard:2}* ‘publicize or announce by placards’ and n *{placard:1}* ‘a sign posted in a public place as an advertisement’

3.2. *Frames profiling particular frame elements*

One of the central notions in frame semantics is *profiling*, a profiled entity being the component of a frame that integrates directly into the semantic structure of the surrounding text or sentence (Fillmore et al. 2001). Profiling is thus a way of specialising the meaning of a more general frame. Notable examples of profiling between large classes of verbs are the closely related verbs in the frames Placing (5a) and Filling (5b), where the former profiles the Theme and the latter profiles the Goal; or between verbs in the frames Statement and Telling where the former profiles the Message (6a) and the latter profiles the Addressee (6b):

Example 5.

a) He_{AGENT} smeared [grease]_{THEME} [on the table cloth]_{GOAL}.

b) He_{AGENT} smeared [the table cloth]_{GOAL} [with grease]_{THEME}.

Example 6.

a) He_{SPEAKER} explained [the theory]_{MESSAGE} [to me]_{ADDRESSEE}.

b) They_{SPEAKER} told [her]_{ADDRESSEE} [another lie]_{MESSAGE}.

The study of profiling gives meaningful insights into some of the changes in the conceptual structure that take place in the process of lexical specialisation. We leave the further study of profiling between pairs of frames outside the scope of this paper.

3.3. *Subdividing frames by introducing restrictions to the frame elements*

We may decide on the degree of granularity of frames and may introduce new frames if such are needed to describe a set of lexical entities in a more precise and informative way. Among the main arguments for introducing subdivisions of frames is that this would enable us to define narrower, more relevant selectional restrictions for the frame elements.

Case 1. The FrameNet frame Filling

On the basis of the observations on verb synsets assigned the frame Filling and the analysis of the semantic features of their core FEs, we can derive several narrower conceptual descriptions. Sporleder et al. (2011) have proposed a more granular division

of this particular frame into: *Filling_container*, *Wrapping_an_object* and *Besmearing_a_surface*. Similarly, we observe several distinct subtypes of *Filling* which can be reformulated as new, more specific frames. Two of them are illustrated in Examples 7 and 8 below: *Covering_surface* (with the semantic restriction *Goal: Surface*) and *Decorating* (a specification of the *Purpose* of filling as one contributing attractiveness).

Example 7. A WN subtree stemming from *{coat:1}*. Original FN frame assigned: *Filling*; new FN assigned *Covering_surface*.

Synset	Definition (WN gloss)	New assigned frame
-- <i>{coat:1}</i>	cover the surface of	<i>Covering_surface</i>
----- <i>{glaze:1}</i>	coat with a glaze	<i>Covering_surface</i>
----- <i>{bonderize:1}</i>	coat with a substance that prevents corrosion	<i>Covering_surface</i>
----- <i>{gelatinize:1}</i>	coat with gelatin	<i>Covering_surface</i>

Example 8. Part of the WN subtree stemming from *{decorate:1}*. Original FN frame assigned: *Filling*; new FN assigned *Decorating*.

Synset	Definition (WN gloss)	New assigned frame
-- <i>{decorate:1}</i>	make more attractive by adding ornament	<i>Decorating</i>
----- <i>{color:4}</i>	decorate with colors	<i>Decorating</i>
----- <i>{miniate:1}</i>	paint with red lead or vermilion	<i>Decorating</i>
----- <i>{encrust:2}</i>	decorate or cover lavishly (as with gems)	<i>Decorating</i>
----- <i>{embroider:1}</i>	decorate with needlework	<i>Decorating</i>
----- <i>{spangle:2}</i>	decorate with spangles	<i>Decorating</i>

Case 2. Subdivisions of the FrameNet frame *Cause_change*

As discussed above, there are a number of frames which are specialisations of the frame *Cause_change* and refer to a change along a particular Attribute such as: *size*, *temperature*, *consistency*, etc. On the other hand, large and consistent WordNet subtrees cannot be mapped to particular frames as this part of the lexicon has not been described in FrameNet yet. Many of these sets of synsets involve clear-cut selectional restrictions, imposed on their frame elements, which are of the same level of granularity as those of *size*, *temperature*, *consistency*, among others. We, therefore, propose the definition of new frames that account for the specialisation of the meaning that takes place in these sets of verbs. As discussed in relation to Example 3 above, where possible, the new frames are modelled on relevant, already existing frames. The difference between the classes under discussion lies primarily in the specification of a different type of attribute for the Patient frame element. Examples 9 and 10 illustrate the cases where the Attribute is *Taste* and *Colour* respectively.

Example 9. A WN subtree stemming from *{change_taste:1}*. Original FN frame assigned: *Cause_change*; new FN assigned *Cause_change_of_taste*.

Synset	Definition (WN gloss)	New assigned frame
-- <i>{change_taste:1}</i>	alter the flavor of	<i>Cause_change_of_taste</i>
----- <i>{bitter:1}</i>	make bitter	<i>Cause_change_of_taste</i>

```

----- {sweeten:1}    make sweeter in taste    Cause_change_of_taste
----- {sugar:1}     sweeten with sugar    Cause_change_of_taste
----- {honey:1}     sweeten with honey    Cause_change_of_taste
----- {sugarcoat:1}  coat with something sweet    Cause_change_of_taste
----- {mull:2}      heat with sugar and spices    Cause_change_of_taste
----- {sour:2}      make sour or more sour    Cause_change_of_taste
    
```

Example 10. A WN subtree stemming from *{color:1}*. Original FN frame assigned: Cause_change; new FN assigned Cause_change_of_colour.

Synset	Definition (WN gloss)	New assigned frame
-- {color:1}	add color to	Cause_change_of_colour
----- {embrown:2}	make brown in color	Cause_change_of_colour
----- {silver:2}	make silver in color	Cause_change_of_colour
----- {reddden:2}	make red	Cause_change_of_colour
----- {encrimson:1}	make crimson	Cause_change_of_colour
----- {rubify:1}	make ruby red	Cause_change_of_colour
----- {madder:1}	color a moderate to strong red	Cause_change_of_colour

3.4. Defining new frames to reflect systematic frame relations

Apart from hierarchical relations, e.g. Inherits from – Is inherited by, there are other systemic relations in FrameNet, such as Causativity, which are reflected through corresponding WordNet relations. The Causativity relation in WordNet creates relatively parallel trees stemming from pairs of causative and inchoative or stative synsets, as in the case of *{change:1}* ‘cause to change; make different; cause a transformation’ and *{change:2}* ‘undergo a change; become different in essence’, respectively. Many of the members down the two subtrees are linked by a corresponding causative relation. This feature enables us to align the FrameNet frames assigned to such synsets and automatically identify: (a) cases where a more specific frame is available to one of the synsets; (b) cases where relevant frames are not defined in FrameNet, which creates a misalignment in the FrameNet-to-WordNet structure. These judgments are based on the assumption that for each (causative or stative/inchoative) frame an equally specific counterpart is likely to exist in the opposite domain.

In practice, a clear-cut correspondence between causative and inchoative frames is missing in a number of cases, for instance, with some classes of motion verbs, which results in the assignment of incorrect or non-specific frames. Consider Example 11 where a frame evoking directed motion verbs is described:

Example 11a.

Frame: Motion_directional

Frame definition: A Theme moves in a certain Direction which is often determined by gravity or other natural, physical forces.

Core frame elements: Theme; Area; Direction; Goal; Source; Path

Example: [The vase]_{THEME} toppled [down the stairs]_{PATH}.

As in FrameNet there is no causative counterpart for this frame such that describes causative verbs of directed motion, a new one needs to be defined based on the conceptual description of Motion_directional and related to it through the Causative relation. The conceptual structure of stative/inchoative and causative counterparts is distinguished by the presence of a causative subevent in the latter frames (Van Valin and LaPolla, 1997) which is associated with a causative (Agent or an Agent-like) participant (frame element). In Example (11b) the newly defined frame Cause_motion_directional is derived from Motion_directional by enriching the set of the latter frame's FEs with the frame elements Cause and Agent, thus accounting not only for sentient volitional doers, but for natural forces and other non-sentient causes.

Example 11b.

Frame: Cause_motion_directional

Frame definition: An Agent or a Cause causes a Theme to move in a certain Direction which is often determined by gravity or other natural, physical forces.

Core frame elements: Agent | Cause; Theme; Area; Direction; Goal; Source; Path

Example: [The wind]_{CAUSE} toppled [the vase]_{THEME} [down the stairs]_{PATH}.

The regularity of the Causative relation and the correspondence in the conceptual structure of causative and inchoative/stative frames makes for an efficient means for regularising the conceptual description of large parts of the verb lexicon.

4. Conclusions and future work

In addition to the objectives specified above, a further goal is to improve WordNet-to-FrameNet mappings by introducing the newly defined frames into FrameNet's structure and designing automatic and semi-automatic procedures for mapping. Special attention needs to be paid to verbs and WordNet subtrees which have no frames assigned.

A venue of ongoing research that we are concerned with is the definition of precise selectional restrictions imposed on the core FEs of particular frames and their implementation as semantic relations between a verb synset and a set of noun synonyms that satisfy these restrictions. In such a way we will enrich WordNet with relations between verbs and nouns corresponding to participants in their conceptual structure, particularly ones realised as arguments and adjuncts.

The work proposed in this paper is to a great extent language independent. Although we have presented data for English, our analysis of the data is extended to Bulgarian as well. As a further proof of concept, we may argue that, with certain adjustments, the mappings and their extensions are valid cross-linguistically and could be used for Romanian among other languages, as shown in Example 12⁶. **Example 12.** An example of the cross-linguistic application of the newly-defined frame Cause_change_of_colour in English (the Princeton WordNet), Bulgarian (BulNet) and Romanian (RoWN).

⁶ The parallel wordnets may be viewed at: <http://dcl.bas.bg/bulnet/>. As the PWN sense numbers might be displayed differently in this interface, the gloss should be considered when looking up the examples.

Root: {color:1} ‘add color to’

Synset ID	en	bg	ro
eng-30-00283911-v	-- {color:1}	-- {otsvetyavam:1}	-- {colora:3}
eng-30-00289532-v	----- {silver:2}	----- {posrebryavam:1}	----- {albi:3}
eng-30-00534837-v	----- {redde:2}	----- {pochervyavam:1}	----- {înroși:2}

The research presented herein may be further extended towards the development of methods for identification of predicate-argument structure in text, which in turn has numerous applications in the field of NLP – semantic analysis, word sense disambiguation, syntactic parsing, information extraction, text summarisation, machine translation.

Acknowledgements

This study has been carried out as part of the project *Towards a Semantic Network Enriched with a Variety of Semantic Relations* funded by the National Science Fund of the Republic of Bulgaria under the Fundamental Scientific Research Programme (Grant Agreement 10/3 of 14.12.2016). We would also like to thank two anonymous reviewers for their insightful comments and suggestions which proved very helpful in the process of improvement of this paper.

References

- Baker, C. F. (2008). FrameNet, present and future. In Webster, J., Ide, N., and Chengyu Fang, A., (eds.), *The First International Conference on Global Interoperability for Language Resources*. City University, City University, Hong Kong.
- Baker, C. F., Fillmore, C. J., Lowe, J. B. (1998). The Berkeley FrameNet Project. In *Proceedings of COLING-ACL*, August 10-14, 1998, Montreal, Canada, 86-90.
- Baker, C. F., Ruppenhofer, J. (2002). FrameNet’s frames vs. Levin’s verb classes. In Larson, J. and Paster, M. (eds.) *Proceedings of the 28th Annual Meeting of the Berkeley Linguistics Society*, 27-38.
- Burchardt, A., Erk, K., Frank, A. (2005). A WordNet detour to FrameNet. In *Sprachtechnologie, mobile Kommunikation und linguistische Ressourcen. Computer Studies in Language and Speech*, vol. 8, Peter Lang, Frankfurt, Germany, 2005.
- Fellbaum, C. (1998). A semantic network of English Verbs. In Fellbaum, C. (ed.), *WordNet: An Electronic Lexical Database*, MIT Press, Cambridge, MA, 1998, 69-104.
- Ferrandez, O., Ellsworth, M., Munoz, R., Baker, C. F. (2010). Aligning FrameNet and WordNet based on semantic neighborhoods. In *Proceedings of the 7th Conference on Language Resources and Evaluation (LREC 2010)*, May 17-23, Valletta, Malta, 29 May - 2 June 2010, Las Palmas, Spain, 310-314.

- Fillmore, C., Wooters, C., Baker, C. F. (2001). Building a large lexical databank which provides deep semantics. In Tsou, B. and Kwong, O. (eds.) *Proceedings of the 15th Pacific Asia Conference on Language, Information and Computation*, February 2001, Hong Kong, China, 3-26.
- Laparra, E., Rigau, G. (2009). Integrating WordNet and FrameNet using a knowledge-based word sense disambiguation algorithm. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2009)*, September 14-16, 2009, Borovets, Bulgaria, 208-213.
- Laparra, E., Rigau, G. (2010). eXtended WordFrameNet. In *Proceedings of the 7th Conference on Language Resources and Evaluation (LREC 2010)*, May 17-23, Valletta, Malta, 29 May-2 June 2010, Las Palmas, Spain, 1214-1219.
- Leseva, S., Stoyanova, I. (2019). Enhancing conceptual description through resource linking and exploration of semantic relations. In *Proceedings of the 10th Global WordNet Conference*, July 23-27, 2019, Wroclaw, Poland, 229-238.
- Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM* Vol. 38, No. 11, 39-41.
- Palmer, M. (2009). Semlink: linking PropBank, VerbNet and FrameNet. In *Proceedings of the Generative Lexicon Conference (Gen-Lex 2009)*, Pisa, Italy, 9-15.
- Palmer, M., Bonial, C., McCarthy, D. (2014). SemLink+: FrameNet, VerbNet and event ontologies. In *Proceedings of Frame Semantics in NLP: A Workshop in Honor of Chuck Fillmore (1929–2014)*, June 27, 2014, Association for Computational Linguistics, Baltimore, Maryland, USA, 13-17.
- Ruppenhofer, J., Ellsworth, M., Petruck, M. R., Johnson, C. R., Baker, C. F., Scheffczyk, J. (2016). *FrameNet II: Extended Theory and Practice*. International Computer Science Institute, Berkeley, California.
- Shi, L. and Mihalcea, R. (2005). Putting pieces together: combining FrameNet, VerbNet and WordNet for robust semantic parsing. In Gelbukh, A. (ed.), *Lecture Notes in Computer Science: Vol. 3406 Computational Linguistics and Intelligent Text Processing*, Springer, Berlin, Heidelberg, 100-111.
- Sporleder, C., van den Bosch, A., Zervanou, K. (eds) (2011). *Language Technology for Cultural Heritage: selected papers from the LaTeCH Workshop Series, (Theory and Applications of Natural Language Processing)*, Springer Open, Berlin, Germany.
- Stoyanova, I., Leseva, S. (2019). A structural approach to enhancing WordNet with conceptual frame semantics. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, September 2-4, 2019, Varna, Bulgaria, 629-637.
- Tonelli, S., Pighin, D. (2009). New features for Framenet–Wordnet mapping. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009)*, June 4-5, 2009, Boulder, USA, 219-227.
- Van Valin, R. D., LaPolla, R. J. (1997). *Syntax: Structure, Meaning and Function*. Cambridge University Press.

CHAPTER 6

NLP TOOLS AND LANGUAGE-BASED APPLICATIONS

INTEGRATION OF ROMANIAN NLP TOOLS INTO THE RELATE PLATFORM

VASILE PĂIȘ, DAN TUFIȘ, RADU ION

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

{pais,tufis,radu}@racai.ro

Abstract

This paper presents the functionality and key challenges encountered during the development of the RELATE platform. This is a web based interactive platform allowing any user to interact with an array of processing tools for the Romanian language, including: part-of-speech tagging, word embeddings, wordnets, named entity recognition. Platform components were written in different languages, usually without much consideration for their further integration. Therefore, this paper also explores how they were adapted in order to function in a unified processing pipeline making sense for the end user.

Key words — web platform, tools integration, web services

1. Introduction

Natural language processing research requires the combination of different tools and algorithms, such as sentence segmentation, tokenization, part of speech tagging, word embeddings calculation, wordnet linking, named entity recognition, etc. Different such implementations exist for Romanian language, but they were never integrated into a single platform, allowing the scientist user to quickly define and run his required processing steps.

In the context of the ReTeRom project, a web service, known as TEPROLIN (Ion, 2018), was already developed bringing together different algorithms and exposing them using a common programming interface. Afterwards it became obvious that a web platform was required to be overlaid on top of the TEPROLIN web service in order to allow easy interaction with the service, without requiring advanced programming skills for consuming it as a pure web service. Furthermore, other tools, outside the scope of TEPROLIN, were available, thus the platform, and of course the end user, would benefit from a broader integration, without being restricted by the ReTeRom project requirements.

The RELATE platform was constructed as an extensible web application portal. Each available functionality was integrated as either a web component, a web service, or by writing a new web interface.

This paper presents RELATE’s features and the challenges faced when integrating different modules, written in different languages, exposing different kinds of interfaces, under a common platform.

2. Platform structure and design

RELATE has two main components: a web front-end and a web back-end. Any user visiting the platform will interact with the web front-end. This in turn, can either make calls to the web backend, use browser only logic to resolve the query or interact with another web application and expose its functionality in the interface.

The web backend is responsible to dynamically assemble pages visualized in the front-end component and to interact with other modules integrated in the platform.

These interactions are described in Figure 1.

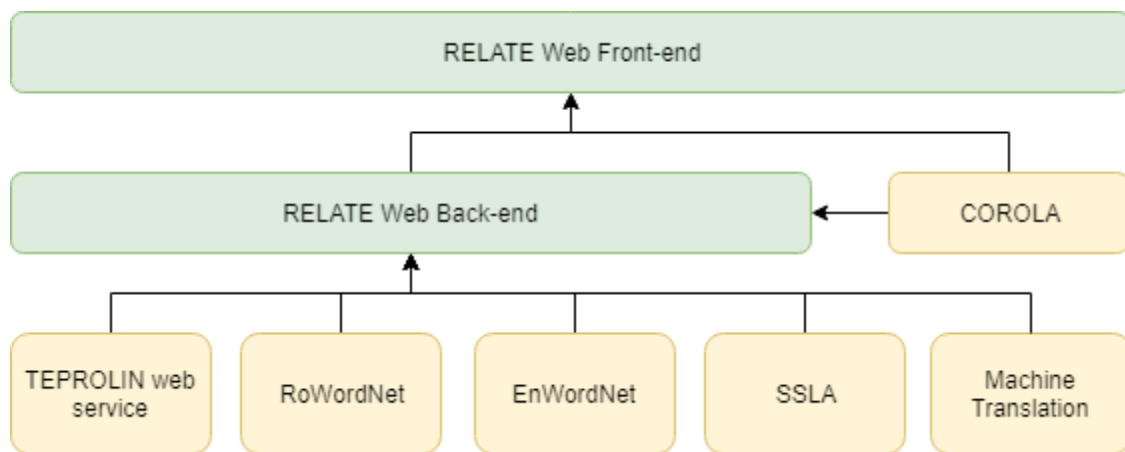


Figure 1: RELATE platform structure

Between the front-end and the back-end there is a communication channel implemented using JSON representation. JSON elements are transferred back and forth using AJAX. This allows the front-end to be highly responsive to user interaction.

The web front-end component is written using specific modern web technologies, such as HTML5, JavaScript, CSS. It implements generic library functions that allows it to interact with the different modules. All basic functionality is implemented as generic functions, thus allowing new modules to be integrated with ease.

The web back-end component is written in PHP and is responsible for interacting with the different modules being integrated in the platform and exposing their results to the end-user via the communication channel with the web front-end. This component implements a common set of library functions used throughout the platform, such as JSON and SOAP web service consumption, string manipulations, etc., and dedicated sets of files for each module being integrated.

Linking of new modules into the platform is accomplished by writing the dedicated interfacing files, if no common library method can be directly used, and by adding it to a platform configuration file, thus exposing its functionality.

Currently, the interfacing files must specify, for each page associated with a module, the HTML content used for rendering the page, additional CSS formatting rules, specific to the module, and any required JavaScript functionality, such as in-browser validations or user-friendly behaviour. Finally, one or more data endpoints can be exposed in the platform, if required by the module, usually in order to be consumed from JavaScript functions. Modules are registered in the platform by using the internal method “registerHandler” which allows for both pages and data endpoints to become available to end-users.

```

1 <?php
2
3 require "lib/lib.php";
4
5 require_once "securimage/securimage.php";
6
7 registerHandler("teprolinws", "data/teprolin/call.php", true);
8 registerHandler("teprolin/complete", "pages/teprolin/comple", true);
9 registerHandler("teprolin/custom", "pages/teprolin/custom.p", true);
10 registerHandler("teprolin/doc_dev", "pages/teprolin/doc_dev", false);
11 registerHandler("teprolin/stats", "pages/teprolin/stats.php", false);
12
13 registerHandler("corola/about", "pages/corola/about.php", false);
14 registerHandler("corola/korap", "pages/corola/korap.php", false);
15 registerHandler("corola/nlpcqp", "pages/corola/nlpcqp.php", false);
16 registerHandler("corola/ocqp", "pages/corola/ocqp.php", false);
17 registerHandler("corola/stats", "pages/corola/stats.php", false);
18 registerHandler("corola/we", "pages/corola/we.php", false);
19
20 registerHandler("ssla/synthesize", "pages/teprolin/synthesize", false);
21
22 registerHandler("rownws", "data/rown/rown.php", true);
23 registerHandler("rown/rown", "pages/rown/rown.php", true);
24 registerHandler("rown/query", "pages/rown/query.php", false);
25 registerHandler("rown/queryp", "pages/rown/queryp.php", false);
26

```

```

1 <?php
2
3 function getPageContent(){
4     $html=file_get_contents(realpath(dirname(__FILE__))."/complete");
5     $loading=file_get_contents(realpath(dirname(__FILE__))."/teprolin");
6     $output=file_get_contents(realpath(dirname(__FILE__))."/teprolin");
7
8     $text="";
9     if(isset($_REQUEST['text']))$text=$_REQUEST['text'];
10
11     $html=str_replace("{loading}", $loading, $html);
12     $html=str_replace("{output}", $output, $html);
13     $html=str_replace("{text}", htmlentities($text), $html);
14     return $html;
15 }
16
17 function getPageCSS(){
18     $css=file_get_contents(realpath(dirname(__FILE__))."/teprolin");
19     return $css;
20 }
21
22 function getPageJS(){
23     $js=file_get_contents(realpath(dirname(__FILE__))."/teprolin");
24     $js=file_get_contents(realpath(dirname(__FILE__))."/complete");
25     return $js;
26 }
27

```

Figure 2: Example interfacing files for the TEPROLIN module: a) Exposing functionality via the “registerHandler” method; b) actual interface file

In RELATE, one of the goals was to make a platform that could integrate potentially any web enabled application, regardless of how it is written: web service or not, Java, C++, PHP or any other language, designed to be integrated or not. Furthermore, the integration related work should be as little as possible. Therefore, we could not use existing integration platforms such as Apache UIMA¹, JBPM² or the approach taken by projects such as PANACEA³.

The following sections describe currently available RELATE modules as well as specific aspects related to their integration in the platform.

3. RELATE modules and their integration

3.1. TEPROLIN web service

TEPROLIN is a web service exposing basic NLP processing tools, such as:

- **Text normalization:** removal of multiple consecutive spaces and Romanian dash normalization;

¹ <https://uima.apache.org/index.html>

² <https://www.jbpm.org/>

³ <http://www.panacea-lr.eu/>

- **Diacritics restoration:** automatic detection of texts lacking Romanian diacritics and automatic diacritic insertion;
- **Word hyphenation** (Stan *et al.*, 2011);
- **Word stressed syllable identification** (Stan *et al.*, 2011);
- **Word phonetic transcription** (Stan *et al.*, 2011) using the SAMPA phonemes for Romanian;
- **Numeral rewriting** (Stan *et al.*, 2011): automatic transformation of number to their written form (e.g. 93 → “ninety-three”);
- **Abbreviation rewriting** (Stan *et al.*, 2011): automatic expansion of abbreviations or acronyms to their full form (e.g. art. → “article” or AI → “Artificial Intelligence”);
- **Sentence splitting** (Ion, 2007; Boroș *et al.*, 2018);
- **Tokenization** (Ion, 2007; Boroș *et al.*, 2018);
- **POS tagging** (Ion, 2007; Boroș *et al.*, 2018) using the Morpho-Syntactic Descriptors (MSD) for Romanian tag set;
- **Lemmatization** (Ion, 2007; Boroș *et al.*, 2018);
- **Named entity recognition (NER)** with four labels: person (PER), location (LOC), organization (ORG) and time (TIME);
- **Biomedical NER** (Boroș *et al.*, 2018) with four labels: disorder (DISO), anatomical part (ANAT), medical procedure (PROC) and chemical (CHEM). The sequence labeler was trained on the MoNERo corpus (Mitrofan *et al.*, 2018);
- **Chunking** (Ion, 2007) with four types of non-recursive syntactic phrases: noun (Np), verb (Vp), adjectival/adverbial (Ap) and prepositional (Pp);
- **Dependency parsing** (Boroș *et al.*, 2018) with the Romanian Universal Dependencies label set.

This is already a step in integrating different tools for Romanian language and exposes everything into a single JSON API.

TEPROLIN is written in Python, but since it already exposes the functionality via JSON REST APIs it was considered easy to integrate into a web platform.

The RELATE platform allows the user to specify required operations (or all of them if needed) and transforms this selection into appropriate web service calls. Finally, results are displayed back to the user and additional links to other platform modules are created.

All interactive parts in the platform related to TEPROLIN are constructed automatically based on web service results. Therefore, the platform is adaptive in the sense that new operations added to TEPROLIN will become available in the platform as well.

Integration of this component required writing a web service proxy that can handle all the platform's requirements, compose multiple calls when needed, for example in the case of obtaining a list of all operations with their options, extending the results to allow linking additional modules, and finally making all this available for AJAX calls.

Processing results can be visualized and exported in a number of formats, including: pure JSON, text, CoNLL-U⁴, CoNLL-X (Buchholz and Marsi, 2006), XML (this is the same format used in the COROLA corpus), graphical dependency tree, named entities highlighted using the BRAT rapid annotation tool (Stenetorp *et al.*, 2012).

Since the dependency tree visualization is the most complex and interactive one, most of the linked features from other platform components are available here. An example is presented in Figure 2.

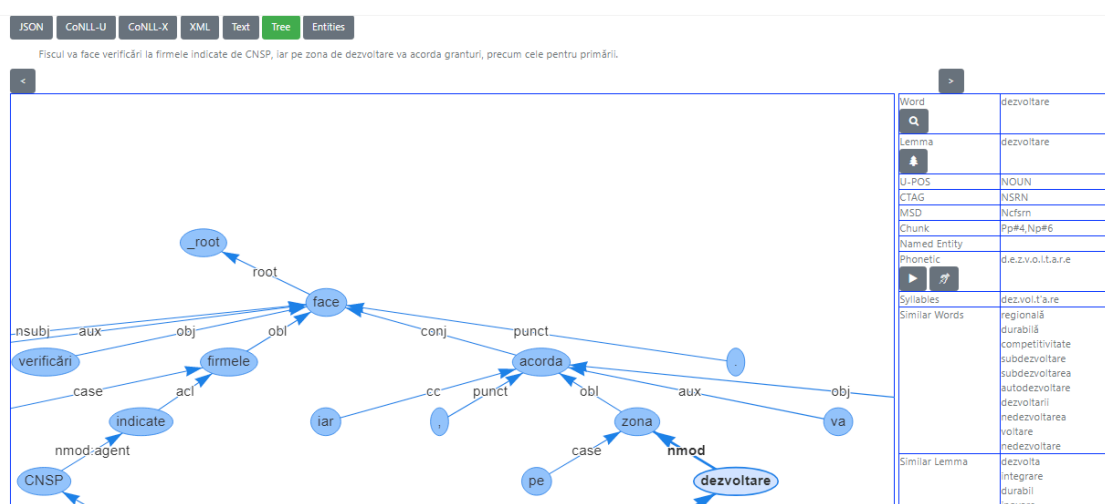


Figure 3: Example dependency tree visualization from RELATE with buttons activating features from various platform components

3.2. COROLA

The Reference Corpus for Contemporary Romanian Language (CoRoLa) (Mititelu *et al.*, 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), CoRoLa⁵ is one of the largest fully IPR-cleared Reference Corpus in the world.

At the time of this writing, COROLA is searchable via three interfaces: KorAP corpus management platform, developed at the Institute for German Language (Leibniz-Institut für Deutsche Sprache) in Mannheim (Bański *et al.*, 2014; Diewald *et al.*, 2016), NLP-

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

⁵ www.corola.racai.ro

CQP, a friendlier interface allowing queries to be specified in a more natural language format, OCQP, a dedicated interface for querying the oral part of the corpus.

These interrogation interfaces were directly integrated into the RELATE platform by exploiting the way URLs are being constructed. For example, the URL used to interrogate the KorAP instance exposing COROLA corpus for the word “dezvoltare” (“development”) is shown below:

<http://89.38.230.10:5555/?q=dezvoltare&collection=&ql=poliqarp&cutoff=1>

By manipulating the URL and changing the word with any other desired word, it is possible to include the results presented by KorAP directly in the interface and make it appear when clicking the corresponding button on any word. Apart from this integration, the user is able to perform additional queries directly in KorAP, if so desired, using any of the features presented by KorAP.

Similarly, it is possible to listen for a word spoken by a human if it is present in the COROLA spoken part. In this case, the OCQP interface is interrogated using this URL (given for the sample example word “dezvoltare”):

http://89.38.230.23/corola_sound_search/index.php?search_type=0&search=dezvoltare&search2_type=0&search2=&start=0&count=20&show_word=on&show_pos_ctag=on&context=5&embedded=y

As in the previous case, the result is integrated in the RELATE interface when clicked on a particular word.

Recent work on COROLA involved the creation of distributional representation of words, also known as “word embeddings”, constructed based on the entire corpus (Păiș and Tufiș, 2018). These are available for download and are also searchable directly on the COROLA server using a web service call.

As described in (Bengio *et al.*, 2003) and (Mikolov *et al.*, 2013), word embeddings can be used to identify words appearing in similar contexts as a given word, by computing the cosine distance between the vector representation of the given word and of all other words.

The RELATE platform makes use of these word embeddings web services in order to show similar words, based on either the complete word, lemma or a combination of lemma and part of speech. For lemmatization and part of speech, the TEPROLIN web service is being used.

This is an example where the backend component of the RELATE platform combines multiple web services from different modules in order to enhance the user experience and show all the relevant information in a single place without requiring further user interactions.

3.3. RoWordNet

Work on the Romanian wordnet started within the BalkaNet project (Tufiș, *et al.*, 2004). With recent enrichment of RoWordNet, as described in (Tufiș and Mititelu, 2014) it now contains 59348 synsets covering 85277 literals.

The previously used interface for interacting with the Romanian wordnet is represented by a Java API⁶.

However, for the purpose of the RELATE platform, we had to enhance this API and expose it via a web server, in the form of a JSON REST web service. This new web service allows the wordnet to be interrogated via the word lemma or by using the synset id.

The result of the web service can be accessed in the RELATE interface either by direct query or by accessing the link behind a word previously processed by the TEPROLIN web service. In the first case, the user must enter the corresponding lemma, while in the second case the TEPROLIN web service already lemmatized the word.

Romanian wordnet is aligned with the English wordnet (Miller, 1995). This means that the same synset id is used in both Romanian and English wordnets. This presents the opportunity to make an “aligned query”. Basically, the user queries one wordnet and gets the results from both of them, based on the resulting synset id.

In order to implement the aligned query feature in RELATE, we had to implement a similar web service functionality for the English word net as well. This was achieved by converting the available English word net files to our format and start a new service instance based on those files. Finally, the link between the two wordnets was accomplished in the RELATE web front-end by executing a separate AJAX call to the English web service instance. This is an extensible approach allowing in the future for other word nets to be queried from within the platform.

An example web service call is presented below. The service response was reduced for clarity.

URL: *http://relate.racai.ro/index.php?path=rownws&word=dezvoltare&sid=&wn=ro*

Response:

```
{ "senses": [
{
  "id": "ENG30-00367768-n", "pos": "Noun",
  "definition": "acțiunea de a se extinde dobândind
proporții, însemnătate, forță.",
  "literal": "extindere, dezvoltare",
  "relations": [
    { "tid": "ENG30-00365709-n",
      "rel": "hypernym", "tliteral": "mărire" },
    ...
  ]
}, ... ] }
```

⁶ Freely downloadable at <http://www.racai.ro/tools/text/rowordnet/>.

A similar query, this time using a synset id in the English wordnet exposed in the RELATE web service is presented below. Again for clarity the response was reduced:

URL: <http://relate.racai.ro/index.php?path=rowwnws&word=&sid=ENG30-00367768-n&wn=en>

Response:

```
{
  "senses": [
    {
      "id": "ENG30-00367768-n", "pos": "Noun",
      "definition": "act of expanding in scope; making more
widely available; 'extension of the program to all in need'
",
      "literal": "extension",
      "relations": [
        { "tid": "ENG30-00365709-
n", "rel": "hypernym", "tliteral": "expansion, enlargement" },
        ...
      ]
    }
  ]
}
```

As it can be seen by analysing the two example URLs, the RELATE wordnet web service allows interrogation of a specific word net, specified via the “wn” parameter (in our case either “ro” or “en”) and by specifying either the “word” parameter or the “sid” synset id parameter.

Figure 3 presents an example result displayed in the platform based on an aligned query request for the word “dezvoltare” / “development”.

RoWordNet	EnWordNet
Synset: ENG30-00249501-n <i>înaintare, dezvoltare, evoluție, mers înainte, progres, propășire</i> (Noun) Acțiunea de a (se) dezvoltași rezultatul ei • hypernym ENG30-00250259-n dezvoltare • near_eng_derivat ENG30-00252990-v spori • near_eng_derivat ENG30-00248659-v avansa, evolua, progresa, [se] contura • near_eng_derivat ENG30-00558371-v avansa, evolua, progresa, [se] dezvolta • hyponym ENG30-00249780-n avansare, înaintare, înălțare, promovare, ridicare • hyponym ENG30-00250143-n flux_de_lucru	Synset: ENG30-00249501-n advancement, progress (Noun) gradual improvement or growth or development; 'advancement of knowledge'; 'great progress in the arts' • hypernym ENG30-00250259-n development • derivationally_related_form ENG30-00252990-v build_up, work_up, build, progress • derivationally_related_form ENG30-00248659-v progress, come_on, come_along, advance, get_on, get_along, shake_up • derivationally_related_form ENG30-00558371-v advance • hyponym ENG30-00249780-n forwarding, furtherance, promotion • hyponym ENG30-00249987-n stride • hyponym ENG30-00250143-n work_flow, workflow
Synset: ENG30-00250259-n dezvoltare (Noun) mărire treptată; 'dezvoltarea economiei' • hypernym ENG30-00248977-n îmbunătățire • near_eng_derivat ENG30-01738774-v produce, dezvolta, realiza • near_eng_derivat ENG30-00411020-v [se] moderniza • hyponym ENG30-00249501-n înaintare, dezvoltare, evoluție, mers înainte, progres, propășire • hyponym ENG30-00250597-n lărgire, extindere	Synset: ENG30-00250259-n development (Noun) act of improving by expanding or enlarging or refining; 'he congratulated them on their development of a plan to meet the emergency'; 'they funded research and development' • hypernym ENG30-00248977-n improvement • derivationally_related_form ENG30-01738774-v develop • derivationally_related_form ENG30-00411020-v

Figure 4: Example aligned query result for the word “dezvoltare” / “development”

3.4. SSLA

Previous work at RACAI included the realisation of speech synthesis systems. One such system is the Speech Synthesis for Lightweight Applications (SSLA), described in (Boroş and Dumitrescu, 2015). This is available as a web endpoint, allowing the production and download of a wav file with the synthesized text.

Given the example word used before throughout this paper, “dezvoltare”/”development”, the endpoint URL for synthetization is:

<http://slp.racai.ro/services/tts/synth.php?lang=ro&coder=straight&voice=anca&text=dezvoltare>

In the RELATE platform this service was integrated and is available for each word after the initial text analysis. It is meant as a backup feature in case the word is not present in the oral part of the COROLA corpus, or in case there are not enough examples in there. By using this feature, the user can listen for the approximate pronunciation of any Romanian word.

Figure 4 shows an example of how SSLA is integrated in the RELATE platform.

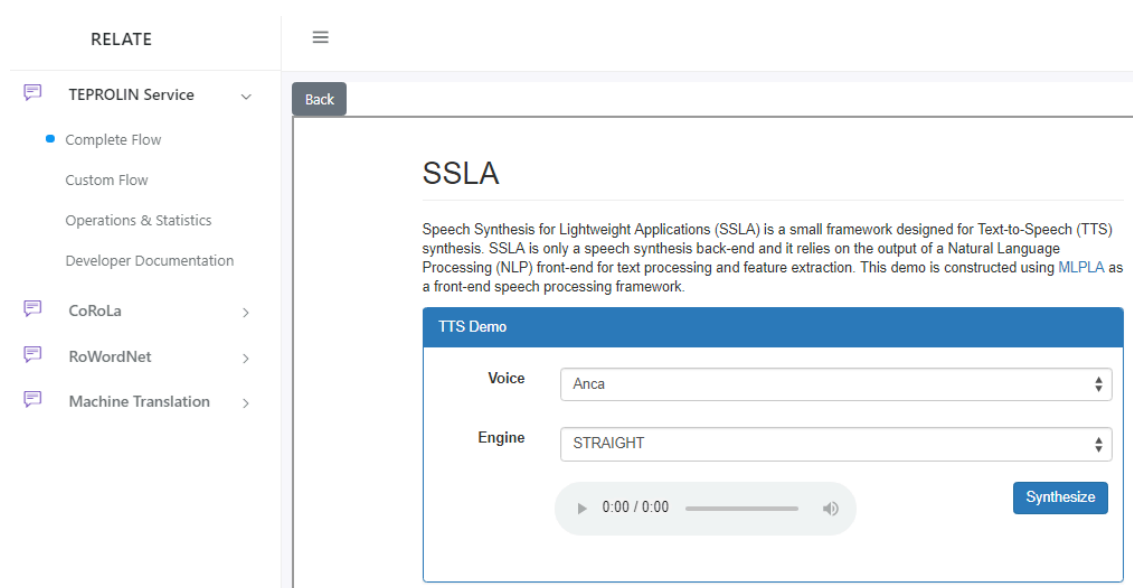


Figure 5: Example speech synthesis integration using SSLA

3.5 Machine Translation

The translation system has been developed within the project “CEF Automated Translation toolkit for the Rotating Presidency of the Council of the EU”, TENtec no. 28144308 and is addressing the languages of the trio Romania-Finland-Croatia. This project, led by TILDE, a linguistic technology company specialising in neural automatic translation, is a follow-up of the project which considered the previous trio Latvia-Austria-Bulgaria, led also by TILDE. The core translation system builds on EC

multilingual platform eTRANSLATION and is significantly improved by the plugged-in neural engines developed by TILDE. Being machine learning based, the neural engines and the core system require a training phase (large data needed, from hundreds of thousands to millions of translation units). During the translation process, neural automatic translation systems analyse the entire context of a sentence, producing high quality translations that are close to the style of human translations. The translation system (Ro-En and En-Ro), deployed in partnership with the Institute of Research for Artificial Intelligence “Mihai Drăganescu”, is a component of a wider system for the Presidency of the Council of the European Union. The Complete System⁷ allows users not only to translate text fragments but also entire documents and local websites with the European Commission platform (including secure automated translation systems for all EU official languages) CEF eTranslation.

Integration in the RELATE platform was realized through the TILDE Machine Translation API⁸ and selecting explicitly the system developed as mentioned above. This involved integrating several JSON calls in the backend and exposing them to the front-end as a single unitary API.

Currently it is possible to translate from English to Romanian and vice-versa. These are the only direct translations for Romanian language available from the project mentioned above. Technically it would be possible to implement other indirect translations by using English as a “pivot” language. Nevertheless, this was not integrated in the platform and consideration must be given to possible translation errors introduced by such an indirect mechanism.

4. Conclusions

RELATE is an integrated platform allowing the natural language researcher easy access to a wide range of tools for the Romanian language. Everything is made available through a simple and easy-to-use web interface.

This paper explored the components currently integrated in the platform and the technical challenges and solutions faced during the integration process. This is of importance especially since there is a mix of several technologies, depending on each module development strategies, such as: PHP, Java, Python, combined with the web specific technologies such as HTML5, JavaScript, CSS.

One of the key observations with regard to integrating multiple components is that it's always easier to integrate those that already expose their functionality as web services, especially JSON REST APIs. This happens because browsers natively can interact with such services, via AJAX calls. Nevertheless, due to RELATE's approach to mix multiple such services in order to provide more meaningful experiences to the end user, we chose to create “proxy” services, relying on the back-end functionality to call the needed component services. This allows for more powerful processing to take place

⁷ <https://ro.presidencymt.eu>

⁸ <https://www.tilde.com/developers/machine-translation-api>

inside the backend process in order to compose the different results and return a single, unified, response to the front-end.

RELATE was designed from the ground up to be extensible on all levels. This means that future tools can be easily integrated, thus increasing its complexity and usefulness.

Apart from the obvious usefulness within the natural language processing research community, we consider that RELATE can also be useful for any student of Romanian language, including foreign students who have the chance to explore and even listen to the pronunciation of words in Romanian.

Acknowledgements

Part of this work was conducted in the context of the ReTeRom project and “CEF Automated Translation toolkit for the EU Council Presidency”, TENtec no. 28144308.

References

- 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).
- Boroş, T., Dumitrescu, S. D., (2015). Robust deep learning models for text-to-speech synthesis support on embedded devices. In *Proceedings of the 7th International Conference on Management of computational and collective Intelligence in Digital EcoSystems (MEDES’15)*, Caraguatuba, Brasil.
- Boroş, T., Dumitrescu, Ş. D. and Burtică R. 2018. NLP-Cube: End-to-End Raw Text Processing With Neural Networks. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics, 171-179.
- Buchholz, S., Marsi, E., (2006). CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, 149-164, June 08-09, New York, USA.
- 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).
- Ion, R. 2007. Word Sense Disambiguation Methods Applied to English and Romanian. PhD Thesis, Romanian Academy, 148 pages (in Romanian).

- Ion, R. (2018). TEPROLIN: An Extensible, Online Text Preprocessing Platform for Romanian. In *Proceedings of the International Conference on Linguistic Resources and Tools for Processing Romanian Language (ConsILR 2018)*, November 22-23, 2018, Iași, Romania.
- Mikolov, T., K. Chen, G. Corrado, J. Dean. (2013). *Efficient Estimation of Word Representations in Vector Space*, arXiv:1301.3781.
- Miller, G. A. (1995). WordNet: A Lexical Database for English. *Communications of the ACM*, Vol. 38, No. 11: 39-41.
- Mitrofan, M., Barbu Mititelu, V. and Mitrofan, G. 2018. Towards the Construction of a Gold Standard Biomedical Corpus for the Romanian Language. *Data* 3(4): 12 pages.
- Păiș, V., Tufiș, D. (2018). Computing distributed representations of words using the COROLA corpus. In *Proceedings of the Romanian Academy, Series A, Volume 19, Number 2/2018*, 403-409.
- Păiș, V., Tufiș, D. (2018). More Romanian word embeddings from the ReTeRom project. In *Proceedings of the International Conference on Linguistic Resources and Tools for Processing Romanian Language – CONSILR 2018*, 91-100, November 22-23, 2018, Iași, Romania.
- Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou and Jun'ichi Tsujii. (2012). Brat: a Web-based Tool for NLP-Assisted Text Annotation. In *Proceedings of the Demonstrations Session at EACL 2012*.
- 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* 53(3): 442-450.
- Tufiș, D., Cristea, D., Stamou, S. (2004). BalkaNet: Aims, methods, results and perspectives. A general overview. *Romanian Journal of Information Science and Technology Special Issue*, 7(1-2): 9-43.
- Tufiș, D., and Mititelu, V.B. (2014). The Lexical Ontology for Romanian. In Gala, N., Rapp, R., Bel-Enguix, G. (eds.), *Language Production, Cognition, and the Lexicon. Text, Speech and Language Technology*, vol. 48, Springer, 491-504.
- Y. Bengio, R. Ducharme, P. Vincent. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137-1155.

README – MORPHO-SYNTACTIC ANALYSES FOR ROMANIAN LANGUAGE

ANDA-MĂDĂLINA FLOREA¹, IRINA TOMA^{1,2}, ROBERT BOTARLEANU³,
MIHAI DASCĂLU^{1,2,3}, ȘTEFAN TRĂUȘAN-MATU^{1,2,3,4}

¹*University Politehnica of Bucharest, Computer Science Department*

²*Cognos Business Consulting S.R.L., Bucharest*

³*Academy of Romanian Scientists, Bucharest*

⁴*Research Institute for Artificial Intelligence “Mihai Drăgănescu”,
Romanian Academy*

*{anda_madalina.florea, robert.botarleanu}@stud.acs.pub.ro,
irina.toma@rocketmail.com, {mihai.dascalu, stefan.trausan}@cs.pub.ro*

Abstract

Writing has become the dominant means of communication in the online environment, alongside the evolution of technology. Therefore, it is imperative to produce high quality writing, correct in terms of spelling, grammar, syntax and semantics. Identifying mistakes in texts is a time-consuming task and requires deep knowledge of the language rules and exceptions. Romanian language has complex morphological and syntactic rules that usually require declension analysis. Also, it has three basic tenses with forms for present, perfect and future which are combined with two types of moods, predicative and non-predicative. The current work focuses on addressing several relations between the subject and the predicate, as well as noun relations to other parts of speech. These corrections are integrated in the ReadME system as an additional text analysis step.

Key words — Natural Language Processing, Grammar checking, ReadME system, Text analysis.

1. Introduction

Writing is one of the most common means of communication in the online environment, therefore it is important for people to be able to write texts that are correct in terms of spelling, syntax, and semantics. Even if this seems straightforward, it is not an effortless process as languages have vast, complicated rules and exceptions. Because this study focuses on the Romanian language, we will take it as example to demonstrate the complexity of the morpho-syntactic rules.

Romanian is an inflected language, as presented by Cojocaru (2003), meaning that the form of words such as nouns, pronouns, articles, adjectives, numerals and verbs changes when expressing grammatical relations. Nouns and nominal parts of speech (i.e. pronouns, articles, adjectives, and numerals) are declinable, and the nominals are declined according to five cases: nominative, genitive, dative, accusative, and vocative. Also, the definite and indefinite articles also change through declension. Verbs change their form through conjugation according to the tense (past: imperfect, perfect,

pluperfect, compound perfect, present and future – future 1, future 2, future 3, future in the past, future perfect), mood (predicative and non-predicative), person, number, gender (feminine, masculine and neuter) and voice (active, passive and reflexive). Thus, it is indisputable that a person can easily write errored texts and that it would be beneficial to have a framework that automatically identifies these mistakes for Romanian language.

The aim of the current work is to enhance the ReadME framework with rules for identifying the relations between subject and predicate, the relation between nouns and other parts of speech, and provide feedback for manually correcting them.

Regarding the structure of the current paper, the next section describes the existing work in terms of automated text correction and introduces the ReadME framework. Section three describes the method and the rules used to identify mistakes for the Romanian language. Section four describes the integration into ReadME, whereas the Section 5 presents conclusions and future work.

2. Related Work

2.1. Automated Text Correction

Numerous methods for correcting texts have been proposed, ranging from manual rule definition to the use of Machine Learning models trained to detect and suggest corrections for various errors. This chapter presents a selection of relevant systems that take various approaches to correct errors in natural language documents.

De Felice *et al.* (2008) propose a method for identifying errors related to the improper use of prepositions and determiners, since these had been shown to be a significant source of mistakes for non-native English speakers. Using a maximum entropy classifier and a set of hand-crafted linguistic features which were selected to encompass relevant semantic and syntactic information, the authors report an accuracy of 92.15% for correct determiner detection task on an L1 (native) English corpus and a 70% accuracy in the task of detecting whether a preposition is incorrect in a text for an L2 (non-native) English corpus.

In some cases, methods of generating errors in text according to certain requirements have been explored. For example, McCoy *et al.* (1996) investigate the generation of “mal-rules”, rules which introduce errors into correct English texts. In this work, “mal-rules” are tailored for the errors commonly generated by hearing impaired American Sign Language users that are learning written English as a second language. In another study (Brockett *et al.*, 2006), authors generate artificial errors by analyzing mass noun errors produced by L2 English Chinese learners and then apply a machine translation model in order to correct the mistakes, achieving 61.81% accuracy. A different approach introduced by Wagner *et al.* (2007) presents methods of detecting whether a sentence is grammatically correct, with the errors being introduced by inserting various kinds of common errors such as removing or adding words, agreement errors and spelling mistakes. By training a decision tree classifier, authors achieve 66% accuracy on the problem of classifying the grammatical correctness of a sentence.

The use of the noisy channel model (Shannon, 1948) has been explored in the case of spelling correction. Brill and Moore (2000) improve the noisy channel by generating more impactful spelling errors, including mistyping and cognitive errors. By training a model to learn string edits, their probabilities are then used to correct errors in a common English spelling errors corpus, achieving accuracy of up to 98.8% in the case where context windows of 5 are used and the accuracy is measured using the top 3 suggestions returned by the system.

Shaalan *et al.* (2003) introduce a system capable of generating suggestions for mistakes in Arabic texts. The method takes into account various aspects of the Arabic language, such as its high degree of inflection, its strong structure and derivational nature and attempts to tackle issues regarding computational morphology (i.e. adding affixes to words), weak and consonant characters (i.e. some characters change in response to diacritic signs of surrounding characters) and morpho-graphemic rules (i.e., when morphological rules are applied some spellings change). The authors propose a method that utilizes heuristics to correct errors by analysing words isolation, with proposed changes ranging from editing characters (i.e. removal, addition, replacement) to adding whitespaces in order to split words.

Mangu and Brill (1997) propose the use of a transformation-based learning algorithm that aims to perform spell correction by learning a set of rules that can be understood, and improved on, by humans. The system is allowed to perform transformations of the form “switch one word with another” if they can either be found in a co-occurrence relationship, or if collocation patterns can be detected around the targeted word, with the possibility of having wildcards in the pattern also being considered.

For Romanian language, a common task is the introduction of diacritics to written text. Commonly, Romanian speakers avoid placing diacritics in typed text in favor of typing speed. Methods for introducing diacritics exist, such as the method described by Tufiş and Ceaşu (2007) where a hidden Markov model is used to model the highest probability character in a word. In contrast, Mihalcea (2002) uses n-grams for varying ranges. Additionally, Mihalcea and Năstase (2002) investigate the difficulty of diacritics restoration across a multitude of European languages and find that they can achieve high accuracies with the number of diacritics present in the language having little impact. As such, for Romanian the reported accuracy was 98.30% while the Czech model achieved 97.83% accuracy.

2.2. ReadME Framework

ReadME is an online interactive application created to improve the quality of users' writing. The application is designed for Romanian language but supports also English and can easily be extended to other languages. The evaluated texts are diverse and vary from student essays, to magazine texts, or blogs.

ReadME blends the features of an AWE (Automated Writing Evaluation) system and an LMS (Learning Management System). AWEs, as presented by Grimes and Warschauer (2010), automatically evaluate written texts, providing feedback and scoring. Therefore, the systems are divided in two components, one that provides scoring and a second one that generates personalized feedback. Scoring is performed in ReadME using the textual

ANDA MĂDĂLINA FLOREA, IRINA TOMA, ROBERT BOTARLEANU, MIHAI DASCĂLU,
ȘTEFAN TRĂUȘAN-MATU

complexity indices provided by the ReaderBench framework, calibrated either for Romanian, or for English (Sirbu *et al.*, 2018). The feedback is generated in a wizard-style interface, on different levels: diacritics (for Romanian language), morpho-syntax and semantics. On the morpho-syntactic analysis level, the following type of errors are already identified: dissonances, repetitions, and punctuation errors (Florea *et al.*, 2019).

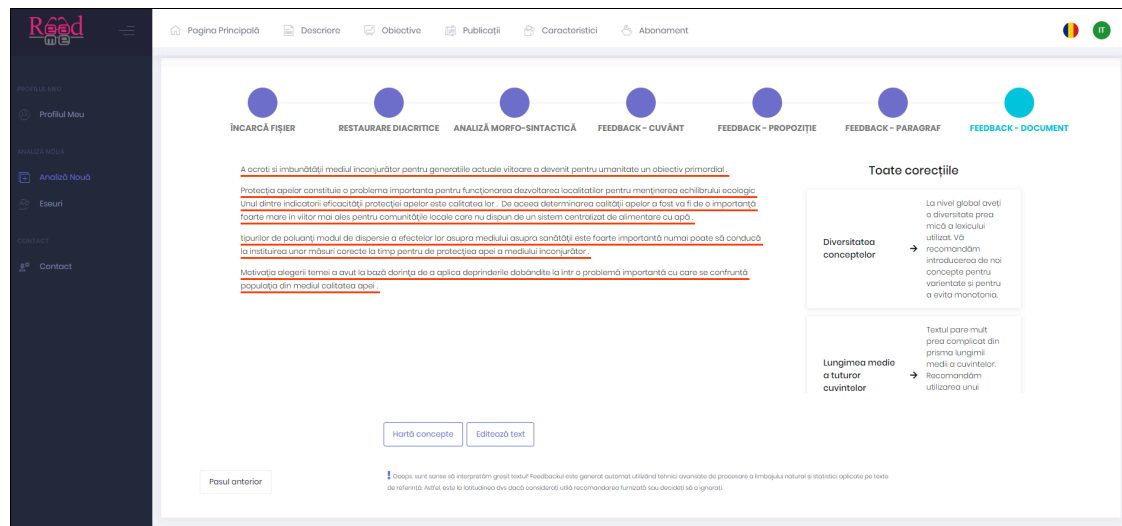


Figure 1: ReadME system – feedback at document level.

The semantic analysis is divided into several granularity levels: word, sentence, paragraph and document.

Figure 1 displays the ReadME wizard representation for feedback at document level. The left part contains the original text, with the identified mistakes underlined, while the right side contains the associated feedback.

From an LMS perspective, ReadME offers three types of users: administrator, mentor and trainee. Each type of user has a particular set of permissions and roles in the application. For example, the administrator oversees user management, mentors supervise the learning process (add and check homework, manages trainee lists) and trainees upload the essays and go through the feedback pipeline. Figure represents the graphical interface for the mentor where he/she adds homework for the trainees.

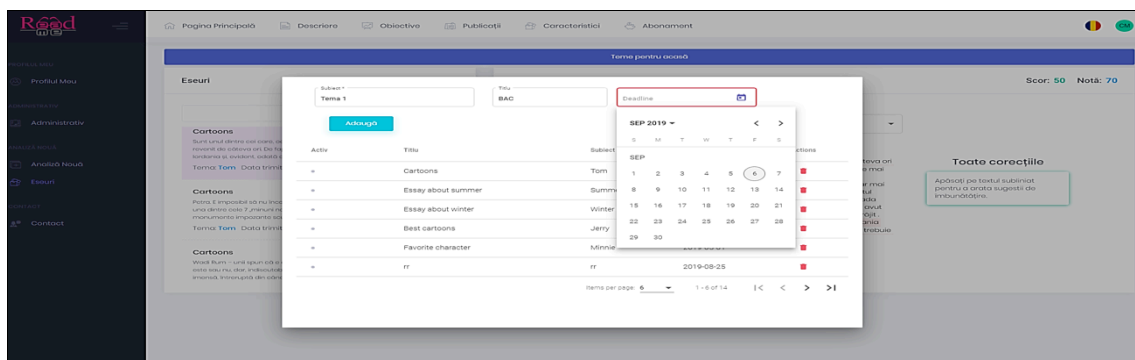


Figure 2: ReadME system – Adding homework.

3. Method

The purpose of the system is to provide targeted feedback for common morpho-syntactic mistakes found in Romanian texts based on a set of rules. These rules are defined by the relations between subject and predicate and by the relations between nouns and other parts of speech. Even though the rules are tailored for Romanian language, they can be generalised for other languages as well.

3.1. Relations Between Subject and Predicate

In Romanian language, the most important constituents at sentence level are the subject and the predicate. These are closely connected, so any miscorrelations cause flagrant grammatical mistakes. The corresponding parts of speech are identified using spaCy, a free open-source library for Natural Language Processing¹, and have the following syntactic dependencies or tags between them: “nsubj” – for the subject, ROOT – for the verb and “aux” – for the auxiliary part of the verb (e.g. “am făcut”, “had done”). For Romanian language, we have obtained the following accuracies using the Barbu Mititelu et al. (2016) corpus: part-of-speech tag accuracy – 97.29%, Unlabeled Attachment Score (UAS) for syntactic dependency parsing – 88.59% and Labeled Attachment Score (LAS) - 81.17%.

Verbal predicate. A Romanian language rule states that the person and the number should be the same for both the subject and the predicate. Since subjects realized by nouns have an indefinite person, they are by default assigned the third person.

The verbal predicates have three forms, depending on the tense and the mood.

- For the present, imperfect, simple perfect, pluperfect, conjunctive and imperative tenses, the verbal predicate is tagged as ROOT and consists of a single word (except for the conjunctive: “să fac”, “to do”). Here, the verbal predicate should change its form according to the subject’s person and number;
- For the compound perfect, perfect conditional and future in the past tenses (e.g. “am făcut”, “had done”), the verbal predicate has as ROOT the participial form of the verb. The first word, “am”, is the auxiliary part that changes its form depending on the subject and is tagged as “aux”;
- For the future and conditional tenses (e.g., “vom face”, “will do”) the verbal predicate has as ROOT the infinitive part of the verb. This case is similar to the previous one.

Nominal predicate. In this case the predicate has two main entities: the copulative verb and the predicative name. The following conditions have to be met for the correlation between the subject and the nominal predicate to be correct: a) the copulative verb must have the same number and person as the subject, and b) the predicative name must have the same gender and number as the subject.

For the nominal predicate, the ROOT of the sentence will be the predicative name, so the copulative verb will be selected as one of the children of the predicative name in the

¹ <https://spacy.io>

dependency graph. The next steps for the copulative verb will be the same as in the case of predicative verb. For the predicative name, the system has rules only for noun, adjective and ordinal/cardinal number. The case of a predicative name realized by a participial form of a verb is omitted because the system cannot differentiate between a verbal predicate and a predicative name (e.g., “cântecul este cântat”, “the song is sung” and “am cântat un cântec”, “I sang a song”).

Relations Between Nouns and Other Parts of Speech

Adjective. In English language, the adjective has only one form, but in Romanian, the adjective (simple or multiple) should have the same number and gender as the noun with which it is linked. A problem that the system cannot resolve is in the next sentence due to an ambiguity: “Băiatul frumos citea” (“The nice boy was reading”) because the system will determine that the word “frumos” is not an adjective that is linked with the noun, but it is an adverb that represented how the boy is reading, in this case – “nicely”.

Number and indefinite article. Like the case of the adjective, ordinal numbers, cardinal numbers and indefinite articles must have the same gender and number as the noun that they determine.

3.2.. Addressed Exceptions

An addressed exception within ReadME is the multiple subject. First, the system decides if in the sentence there is a multiple subject, and the following rules are applied:

- If the multiple subject consists of more than one noun, then only one of them will be linked to the predicate, while the others will be linked through a “conj” dependency to the noun;
- If the multiple subject consists of more than one personal pronoun, then all of them will be linked to the predicate through a “nsubj” dependency.

Second, if there is at least one subject that represents a person (as taken from a list with words that are synonyms with the word “person” or from a list with all pronouns in Romanian), then the following rules are applied, in order:

- If at least one of them is at the first person, then the multiple subject will have the first person;
- If at least one of them is at the second person, then the multiple subject will have the second person;
- Otherwise, the third person is used.

In terms of gender, the multiple subject will be feminine if all the words that compose it are feminine, otherwise if at least one of them is masculine, then the multiple subject will be masculine. If the parts of the multiple subject are things, then if there is at least one masculine plural noun, the nearest noun to a predicate will be accorded in number, person and gender with the predicate (e.g., “Munții și valea sunt împădurite”, “The mountains and the valley are forested”, but also “Valea și munții sunt împăduriți”, “The valley and the mountains are forested”). If there is at least one feminine noun, then the

multiple subject will be feminine plural (“Muntele și valea sunt împădurite”, “The mountain and the valley are forested”). Otherwise, the result will be masculine. The system does not take into consideration the neutral gender, since this flag is missing from the spaCy tagger.

The case of multiple predicates is simpler since each predicate is independent from the rest. As such, the system can iterate through each component predicate and check them individually.

3.3. Exceptions and Ignored Cases

There is a special situation in which the predicative name cannot have the same gender as the subject. For a subject with the feminine gender, the gender of the predicative name that represents a job does not matter. In order to treat this exception in the system, a list with all Romanian words that represent jobs has been integrated into the framework. For example, “Fata este doctor.” (“The girl is a doctor.”) is correct, even if the word “doctor” has the male gender, and “fata” has the feminine gender. Unfortunately, the case of feminine plural is not addressed as the generation of all plural forms of job-related words is time consuming.

The rule stating the subject and the predicate must have the same person and number has an exception, namely in the case of a subject that has the meaning of a collection, known as a collective subject. For example: “Majoritatea a votat.” (“The majority voted.”) is correct, but the form “Majoritatea au fost trimiși acasă.” (“The majority were sent home.”) is also correct, because in the last sentence the meaning of this construction is that “the majority of them” was sent home. This special case is called a dependency after meaning and the system does not correct this mistake.

3.4. Integration within ReadME

The current system is integrated in the ReadME framework in the morpho-syntactic analysis step, as presented in Figure. In the left part of the screen is the inputted user text, which has already passed through the diacritics restoration step. The identified mistakes are directly underlined in the left part. In the right part, all the correction feedback is displayed as a list. There is a correlation between the corrections on the left and the associated feedback. Clicking the correction highlights the corresponding feedback message(s) and vice versa.



Figure 3: ReadME morpho-syntactic analysis for Romanian language.

In Figure 3, the following suggestions are correctly identified:

- “Bunicii avea” (“Grandparents has”) – the suggestion is correct as it matches the rule stating that the subject and the predicate should have the same number;
- “Prelungire sinistru” (“sinister extension”) – the suggestion is correct as it matches the rule stating that the noun and the adjective should have the same gender.

However, other corrections are wrongly identified in the same example due to incorrect part of speech tags or syntactic dependencies from spaCy. The following list presents the motives of the incorrect suggestions, with examples:

- The subject of the sentence is wrongfully identified: e.g., “uși dădea” (“doors gave”) instead of “una dădea” (“one gave”);
- The numeral – noun pair is wrongfully identified: “două bunicii” (“two the grandparents”) instead of “două vaci” (“two cows”);
- The noun – adjective pair is wrongfully identified: “mare Cristi” (“big Cristi”) instead of “mare sperietură” (“big scare”).

Two other mistakes are not pinpointed by the system, “Acoperișul ... făcute” (“the roof ... made”) and “pereții ... era lipiți” (“the walls ... was glued”) because the associated noun from the pair is not correctly identified.

In terms of feedback presentation, the user interface needs adjustments, even though the representation of mistakes for the morpho-syntactic step is consistent with the rest of the application. Most rules are applied on two parts of speech, for example a noun and an adjective, but in the user interface these words are highlighted separately, with no means to visually identify the pair. In addition, the feedback messages are doubled for each element of the pair. In a future implementation, the elements that match a rule will be highlighted as a pair, to easily identify the mistakes. Also, the associated feedback will be set on the pair, not on the individual elements.

4. Conclusions and future work

The article presents a rule-based method to identify the morpho-syntactic errors found in texts written in Romanian language. Even though these rules are designed especially for Romanian, the system can be generalized to any language using pre-imposed rules. This mechanism is integrated into the README framework in a dedicated step, namely the morpho-syntactic analysis. With this addition, README has become an advanced tool to improve texts written in Romanian language by supporting diacritics restauration, morpho-syntactic and semantic analyses.

However, the new analysis step encountered problems related to the complexity of the Romanian language, as the process of finding generalizable rules is time consuming and complicated. For example, some categories of mistakes, mainly relating to context and declension, are not considered. In future implementations, several steps will be carried out to enhance the provided feedback. First, enhancements to part of speech tagging and dependency parsing for spaCy to better suit Romanian language need to be performed.

Second, new rules will be added to address other parts of the sentence, such as attributes and complements. Third, a larger corpus of human errors will be collected in order to consider the usage of new approaches, such as sequence-to-sequence models for automated text correction.

Acknowledgements

This research was supported 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.

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 the Tenth Int. Conf. on Natural Language Processing (HrTAL2016)*. Dubrovnik, Croatia.
- Brill, E., and Moore, R. C. (2000, October). An improved error model for noisy channel spelling correction. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 286-293.
- Brockett, C., Dolan, W. B., and Gamon, M. (2006, July). Correcting ESL errors using phrasal SMT techniques. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 249-256.
- Cojocaru, D. (2003). Romanian grammar. Slavic and East European Language Research Center (SEELRC), Duke University.
- De Felice, R., and Pulman, S. G. (2008, August). A classifier-based approach to preposition and determiner error correction in L2 English. In *Proceedings of the 22nd international conference on computational linguistics (Coling 2008)*, pp. 169-176.
- Florea, A.-M., Dascalu, M., Sirbu, M.-D., and Trausan-Matu, S. (2019). Improving Writing for Romanian Language. In *4th Int. Conf. on Smart Learning Ecosystems and Regional Development (SLERD 2019)*, Springer, Rome, Italy.
- Grimes, D., and Warschauer, M. (2010). Utility in a fallible tool: A multi-site case study of automated writing evaluation. *The Journal of Technology, Learning and Assessment*, 8(6).
- Mangu, L., and Brill, E. (1997). Automatic rule acquisition for spelling correction. In *ICML*, Vol. 97, 187-194.
- McCoy, K. F., Pennington, C. A., and Suri, L. Z. (1996). English error correction: A syntactic user model based on principled mal-rule scoring. In *Proceedings of the Fifth International Conference on User Modeling*, 59-66.

- Mihalcea, R. F. (2002). Diacritics restoration: Learning from letters versus learning from words. In *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, Berlin, Heidelberg, 339-348.
- Mihalcea, R., and Nastase, V. (2002). Letter level learning for language independent diacritics restoration. In *Proceedings of the 6th Conference on Natural Language Learning-Volume 20*. Association for Computational Linguistics, 1-7.
- Shaalan, K., Allam, A., and Gomah, A. (2003). Towards automatic spell checking for Arabic. In *Proceedings of the 4th Conference on Language Engineering, Egyptian Society of Language Engineering (ELSE)*, Cairo, Egypt, 21-22.
- Shannon, C. E. (1948). A mathematical theory of communication. *Bell system technical journal*, 27(3), 379-423.
- Sirbu, M.-D., Botarleanu, R., Dascalu, M., Crossley, S., and Trausan-Matu, S. (2018). ReadME – Enhancing Automated Writing Evaluation. In *18th Int. Conf. on Artificial Intelligence: Methodology, Systems, and Applications (AIMSA 2018)*, Springer, Varna, Bulgaria, 281–285.
- Sutskever, I., Vinyals, O., and Le, Q.V. (2014): Sequence to sequence learning with neural networks. In: *Advances in neural information processing systems*, 3104-3112.
- Tușiș, D., and Ceașu, A. (2007). Diacritics restoration in Romanian texts. In *A Common Natural Language Processing Paradigm for Balkan Languages*, 49-55.
- Wagner, J., Foster, J., and van Genabith, J. (2007, June). A comparative evaluation of deep and shallow approaches to the automatic detection of common grammatical errors. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 112-121.

QUERY EXPANSION - AUTOMATIC GENERATION OF SEMANTIC SIMILAR PHRASES USING WORDNET

DIANA LUCACI, ADRIAN IFTENE

“Alexandru Ioan Cuza” University of Iași, Faculty of Computer Science
{diana.lucaci, adiftene}@info.uaic.ro

Abstract

The applications of finding similar phrases can be identified in different systems such as information retrieval applications (search engines), correction and suggestion systems or software that uses large amounts of data. An example from the medical domain would be an application storing the medical records of the patients to help the medical staff narrow the search to more specific diagnoses and treatments. Similar treatments could lead to improvements in the treatment that the specialist is considering when dealing with a case. By focusing the result list of the search system on both the exact match and a wider circle of concepts (analysis, education, prescriptions, etc.), it would increase the chances that a doctor finds a new treatment or a similar drug that can be used for the case he or she is dealing with. This paper aims to prove the importance of improving the recall of information retrieval systems in order to discover more important information when querying the system.

Key words – query expansion, information retrieval, word embedding, WordNet.

1. Introduction

Natural Language Processing can be defined as the computational modeling of human language, in computer science, relating to formal language theory, theorem proving, and machine learning. It is a field of research that covers computer *understanding* and manipulation of human language, trying to make the machine derive meaning from human language in such a way that it can perform difficult tasks such as information retrieval and extraction, question answering, exam marking, document classification, report generation, automatic summarization and translation, speech recognition, dialogs between human and machine, or other tasks currently performed by humans such as help-desk jobs.

The NLP applications are one of the most challenging and popular because of the impact they have on the end-user. Replacing help-desks with artificial intelligence, automatic translation, and virtual assistants are some of the best-known usages of this domain.

The progress of the domain regarding synonyms and near-synonyms has advanced during the last years, gaining researchers' attention due to the wide range of applications that can benefit from it (text mining, information retrieval, query expansion, machine translation), but it lacks accuracy especially for phrases with more than two words (Gupta *et. al.*, 2015; Khosmood and Levinson, 2011). This particular task can have a big impact on information retrieval systems (a specific application of these type of systems

would be a search engine for the medical domain, which is known for its large amount of information that is available and that should be considered before making a decision regarding a diagnosis). Moreover, correction systems could take advantage of similar phrases (not necessarily synonyms), providing alternatives for scientific or grammatical mistakes.

As the language is evolving rapidly in this age and new words are being introduced in the vocabulary, this project aims to propose a new strategy of automatically generating similar phrases using the relationships between the concepts from a large lexical database organized as a large graph. Since we are not focusing on a specific domain with very technical terms such as the medical domain, for which we could use the specialized database Medical WordNet – MWN (Smith and Fellbaum, 2004), we are using the general language database: WordNet. The application is structured using independent modules which use the results of previous modules, similar to a waterfall model. In addition, a critical analysis is performed on the results of different approaches, by combining different strategies for each module.

The rest of the paper is structured as follows: Section 2 presents a brief literature review focused on the query expansion task, followed by the presentation of the proposed solution in Section 3 that comprises the description of the architecture (3.1), and a detailed explanation together with the summary of the results for each module of the application (3.2, 3.3, 3.4). Section 4 contains the evaluation of the proposed method on the LISA collections IR dataset. In Section 5, we present a medical domain-focused use case to exemplify the targeted use cases that our approach improves. Section 6 presents the challenges that we have encountered and the limitations of the proposed implementation, followed by solutions and future research direction in Section 7. The conclusions for our paper are presented in the last Section: 8.

We open to the public access to the created dataset and to the implementation of the application via github¹.

2. Previous work

The Query Expansion task has had the researchers' attention for many years now, having a long history in the literature due to its impact on information retrieval systems. It has first been introduced in 1960 (Maron *et al.*, 1960) as a technique for indexing and searching in a mechanized library. Later on, the task became important for text retrieval, search engines, Question Answering over Linked Data – QALD (Unger *et al.*, 2016).

Previous classifications of query expansion tasks taking into consideration end-user involvement mention manual, automatic and interactive methods (Efthimiadis, 1996). In our research we focus on automatic approaches, using linguistic resources and statistical approaches to validate our results.

Other Query Expansion approaches that use WordNet², such as Improving Query Expansion Using WordNet (Pal *et al.*, 2013), try to filter a list of possible candidates (e.g. 5 extracted using top-ranked documents) based on the similarity obtained from

¹ <https://github.com/Dianna22/Query-Expansion>

² <https://wordnet.princeton.edu/>

WordNet. The majority of the latest approaches that use WordNet try to preserve the compromise between precision and recall (Zhang *et al.*, 2019; Li *et al.*, 2016; Shekarpour *et al.*, 2013; Gînsacă *et al.*, 2011), as it is also emphasized in the latest literature surveys (Azad *et al.*, 2017). In addition to using WordNet, we have used other semantic resources to improve the quality of the retrieval process, such as YAGO³, Wikipedia or DBpedia⁴ (Gabor and Iftene, 2017; Iftene and Baboi, 2016; Micu and Iftene, 2016; Șerban *et al.*, 2016).

In the similarity literature, we have identified a number of similar tasks such as measuring relational similarity (Turney *et al.*, 2003; Jurgens *et al.*, 2012), but we have not identified a dataset or task focused on the similarity between phrases (or multi-word expressions). This is the motivation behind creating our own dataset consisting of pairs of similar 2-word phrases, that aims to direct researchers' focus towards the similarities and analogies between phrases, rather than between individual tokens.

Our method detaches from the previous attempts by extracting new candidates from WordNet that can be missed by the existing methods that consider perfect synonymity or similar context co-occurrence. The generated candidates are filtered based on the frequency on a corpus (checking for the validity of a phrase) and also, based on the relevance feedback from an interface (future work). Moreover, the generated phrases serve as a training set for a machine learning model which can also improve its accuracy over time.

Our contributions are:

- Proposing the task of determining similar phrases, by relaxing the definition of synonymity;
- Creating a new dataset for pairs of similar 2-word phrases using WordNet;
- Providing a baseline machine learning model for predicting a similar phrase (trained on the proposed dataset), given a 2-word query;
- Proposing an approach (generation of similar phrases) that can be applied in IR systems for query expansion to improve the recall.

We claim that this approach would improve the recall of an IR system due to the nature of the similarity definition that extends the semantic meaning of the initial query. The machine learning module is an extension of the approach that aims to capture the semantic relation between the phrases. The algorithm will determine the similar phrase of a given phrase in a more efficient way, benefitting from the advantages of the neural network: firstly, the time required to predict the similar phrase using it is less than the time needed to perform all the operations described in section 3; secondly, the model can also capture other features such as user preferences or frequencies of the usage of the phrases, that would improve the prediction accuracy.

³ <https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/>

⁴ <https://wiki.dbpedia.org/>

3. *Proposed solution*

Since pseudo-relevance feedback, also known as blind relevance feedback tends to converge on the same topic when assuming that the relevant results are always the top k results, the proposed method overcomes this issue, by expanding the query result with suggestions from similar queries.

To enrich the result list of an IR system with relevant information, multiple NLP tools are needed for performing different tasks such as part-of-speech tagging (PoS tagging), word embedding through distributed representation generation (word2vec), definition extraction (using a probabilistic model that computes the distance between words in order to establish the boundaries of a definition in a context), extraction of similar words using WordNet and hypernyms and hyponyms, which are lexical relations that navigate in the ontology (the taxonomy used to identify similar concepts), determining path similarity, measuring generality (in order to determine how far from the query concept, the generated one is in terms of semantics).

After identifying the key concepts in the given phrase using POS tagging, the identification of relations between those is performed using WordNet, from which we can extract the similar concepts and their properties that are in the same relation as in the given phrase. The number of generated results is determined based on a generality measure and a threshold parameter that establishes which results are still relevant to the queried domain. This parameter can be given as a parameter or it can be learned by a machine learning model. Furthermore, a corpus can be used to determine new properties of the used concepts.

The validation phase consists of filtering out the inconsistent data (the invalid properties of the mentioned concepts, senseless phrases), using a large and structured set of texts (text corpus). This validation has been performed using Wikipedia dumps⁵, indexed using Elasticsearch⁶ and performing phrases search for an exact match.

3.1. *System architecture*

The project is modular, being organized in 3 stand-alone modules described in detail below, which can also be seen in Fig. 1. This approach has been chosen for research purposes, in order to find easier the possible weaknesses of the application. In other words, the results of this research are influenced by each module, so replacing or improving one of them can lead to better results, without influencing the others.

⁵ <https://dumps.wikimedia.org/>

⁶ <https://www.elastic.co/>

QUERY EXPANSION - AUTOMATIC GENERATION OF SEMANTIC SIMILAR PHRASES USING WORDNET

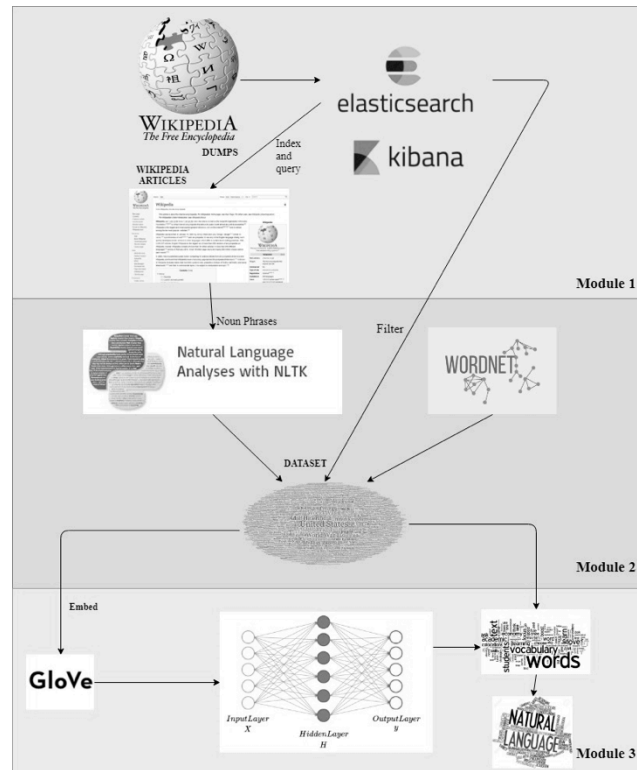


Figure 1: System architecture

3.2. Module 1 - Create a corpus by indexing web articles

The first module of the application is responsible for indexing 420.000 Wikipedia article revisions (Wikipedia dumps - available for downloading) that are used for:

- Extracting noun phrase candidates and creating a database of phrases;
- Validating the generated phrases through an exact matching of a phrase in the corpus.

We have performed data collection and data indexing using the Elasticsearch⁷ Python library.

Creating our own corpus for this task brings along a number of benefits. First of all, we can control the content of its information in order to improve precision and accuracy of the collected data for specific areas of study. A general corpus not only may not contain technical terms, but it can also reduce the accuracy of the system due to the polysemy of words that are used in different contexts.

Secondly, this method allows us to test the system in multiple fields and evaluate its ability of adaptability and generalization. Moreover, it is one of the most convenient ways of using a small corpus for a proof of concept on a very specific domain. We have

⁷ <https://elasticsearch-py.readthedocs.io/en/master/>

also adopted this method considering, as future work, to replace the data source with domain-specific text data.

Although this method seems to be a straightforward way of managing text data, the source of information can be problematic due to the noise it contains and also due to the reliability of the content. The usage of Wikipedia revisions came along with some challenges.

The extracted text contains HTML tags and custom punctuation that is removed as part of the data cleaning process, but words like *ref*, *Sac.png*, *px*, *File*, *category*, *reflist* remain and alter the end results. They influence the number of the resulting similar phrases, thus, the average being higher than in reality. This improves the recall (by gathering more results than necessary), but loses precision (by considering phrases that contain these kinds of words as similar to the initial query). Another impact they have on the results is that they can be part of the phrases that represent candidates (queries) of the system, further having a bad impact upon the training set that is created based on these extracted phrases.

In order to overcome this issue, cleaning the data is necessary, but since this system tries to improve the recall of an information retrieval system, cleaning the data used for indexing has not been performed because using regular expression rules specific for some of the articles might remove some possible and useful candidates.

The indexed corpus is further used for both extracting noun phrases and validating the generated similar ones. The noun phrase extraction has been done using *nlk*⁸, which is a natural language open-source toolkit for Python.

The validation has been performed using *elasticsearch*, by performing search and exact matching operations of phrases in a fast and efficient manner. Although a web search engine has the advantages that it is a simple monolingual corpus-searching tool which uses the extensive amount of information that is available on the web, this approach was not feasible due to the query time.

3.3 Module 2 - Generate and filter similar phrases

This module is responsible for generating similar phrases using WordNet. Apart from its stand-alone purpose, the dataset that will be generated will serve as a training set for a machine learning module (described in section 2.3).

The degree of similarity used in our approach utilizes a slightly different definition from synonymity, closer to near-synonymity, which can be defined using the explanation of the approach of generating the dataset: for 2-word phrases, a similar phrase is considered to be a phrase that is formed of the meronyms, holonyms and “sibling concepts” for each of the words in the initial query and that is an exact match in our corpus. For example, our approach leads to the following list of similarities for the query *research results*: *calculation formula*, *analysis result*, *experiment summary*, *search end*, *experiment explanation*, *research summary*, and *examination results*.

⁸ <https://www.nltk.org/>

QUERY EXPANSION - AUTOMATIC GENERATION OF SEMANTIC SIMILAR PHRASES USING WORDNET

Another example is *research steps*: *research position*, *research trial*, *research career*, *search output*, *research capacity*, *experiment set*, *research base*, *research tools*. Applying this approach for an IR system leads to newly and unexplored search directions that are still related to the initial query, but provide additional information from similar concepts (for example, when searching for *research results*, expanding the query with *experiment explanation* will lead to also returning the rationale behind the sought research results, thus creating a broader overview over the relevant documents to the initial query).

In order to obtain a similar phrase with a given one, we take into consideration different levels of generality for each of the word's similarity degree. With the generated similar terms for each of the initial phrase's words, we perform the Cartesian product that allows us to avoid missing any potential important similar phrase. Out of these phrases, a number of them will not be semantically meaningful. To overcome this, we use elasticsearch to filter them out automatically, by looking for exact matches in our corpus. The phrases that have never been used in any of the indexed Wikipedia articles are discarded.

In order to reduce the number of results for a phrase, we distinguish the following generality levels of similarities and present the obtained results after the filtering phase.

[1] Siblings, parents, and children

- "Siblings" - all the hyponyms of the hypernyms;
- "Parents" - all the hypernyms (relation of the type: part-of, component-of, element-of, sort-of);
- "Children" - all the hyponyms of the current term.

[2] Siblings and children

Out of 1.999 phrases extracted using nltk from the Wikipedia articles, for 1.539 phrases more than 1 similar phrase have been generated, obtaining the following metrics (Fig. 2) regarding the number of results for each phrase:

- minimum 1 similar phrase found for each phrase;
- a maximum of 5.625 similar phrases have been obtained for a given query;
- an average of 193.5211 similar phrases per query.

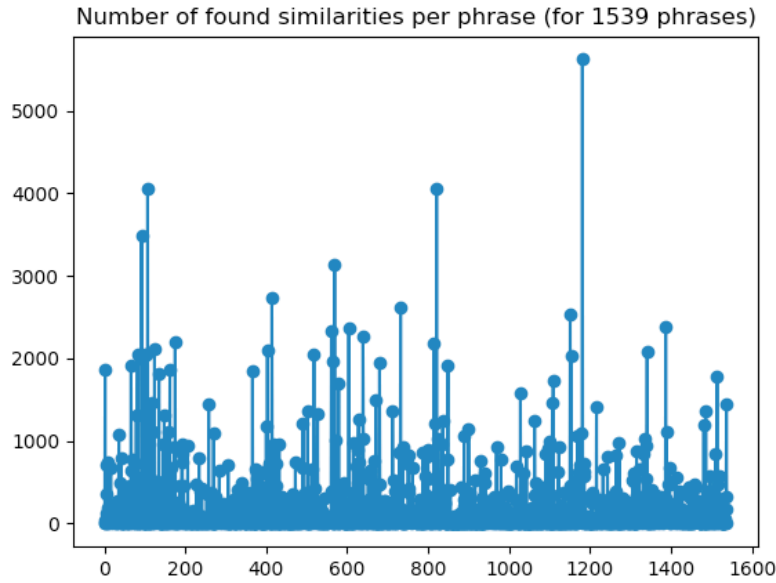


Figure 2: Number of similarities for each phrase (siblings and children)

[3] Siblings

For 1.526 phrases extracted using nltk from the Wikipedia articles, similar phrases have been generated, obtaining the following metrics (Fig. 3) regarding the number of results for each phrase:

- (1) minimum 1 similar phrase found for each phrase;
- (2) a maximum of 5.071 similar phrases have been obtained for a given query;
- (3) an average of 158,416120577 similar phrases per query.

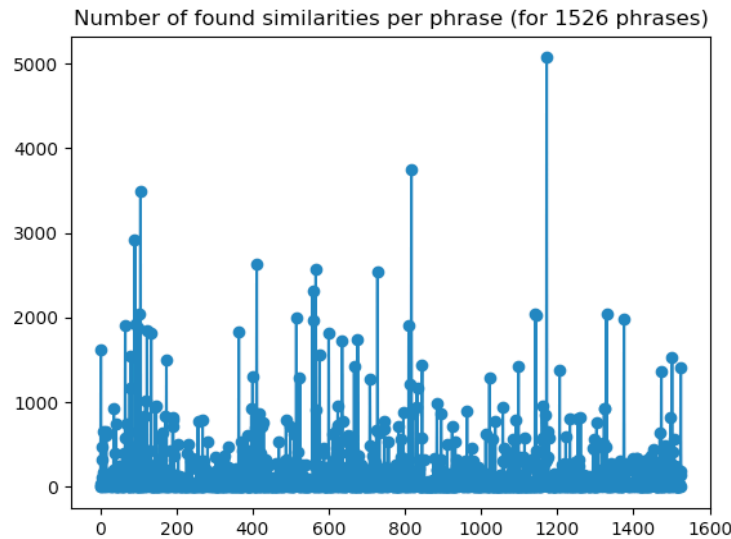


Figure 3: Number of similarities for each phrase (siblings)

3.4. Module 3 - Word embedding. Training set. Neural Network

The automatic detection of word similarity could be based on corpus analysis (e.g., word embedding like word2vec) or user behavior (queries, clicks from those queries,

etc.). Ideally, we use all available signals to train a machine-learned model, but this module is based on corpus analysis only. The result list can be ranked and improved using methods like **relevance feedback** or **indirect relevance feedback** (not the subject of this research project). We have performed a qualitative analysis of the generated examples and provided a few examples in section 3.3.

The dataset used for this module consists of 2.788 phrases, for which similar phrases were generated using module 2, obtaining a total of 241.743 pairs of 2-word phrases, which were shuffled and then split into a training set (80% - 193.394) and a testing set (20% - 48.349). We have used the Multilayer Perceptron (MLP) artificial neural network with one hidden layer to generate an embedding for the predicted similar phrase. The query candidate for which we want to predict the similar phrase is embedded using GloVe (Pennington *et al.*, 2014), with the dimension 50. The concatenation of the 2 embeddings represents the input of the neural network, having a dimension of 100.

We then decode the predicted embedding (output of the neural network) by using cosine similarity against our phrase dictionary and thresholding the similarity distance between the embeddings.

We have performed experiments with different numbers of neurons for the hidden layer, using as activation function the hyperbolic tangent (*tanh*) and the loss function: mean squared error. The training objective is to minimize the distance between the generated embedding and the expected phrase (gold truth obtained from the dataset). Since the distributions of the word embeddings are very different from each other, the training objective is difficult to obtain. We use early stopping on the validation loss and train the network for 500 epochs.

We have also empirically determined the threshold value of the similarity between the generated embedding of the phrase and the identified phrases from the dictionary, by ranging it between 4.5 and 5. The reported results from Table 1 use 5 as a threshold value.

Table 1: MLP results

Hidden layer	Batch size	Accuracy
300	100	85.57
300	50	78.72
500	50	86.63

4. Results

To test the impact on the recall measure for the proposed query expansion method, we have used the LISA⁹ (Library and Information Science Abstracts) Document Collection (Davies, 1983). The dataset contains 6004 documents taken from the Library and

⁹ http://ir.dcs.gla.ac.uk/resources/test_collections/lisa/

Information Science Abstracts database. For testing, we have used 32 queries. The queries contain an average of 19.4 terms per query and are linked to a mean of 10.8 relevant documents per query. Out of the 6004 documents, there are 379 relevant documents. Table 2 reports the average precision and recall of the tested queries using tf-idf for both the raw query (without expansion) and using our approach of query expansion. For our approach, we have extracted 2-word noun phrases from the query and used them as input for the neural network. For each of the extracted noun phrases of the query, we have used the top predicted similar phrases that have a semantic distance of at most 3 between the output embedding distribution of the network and the embedding from the vocabulary (at decoding phase).

Table 2: LISA collections - results

	TF-IDF (no query expansion)	TF-IDF (query expansion)
Average recall	0.1533	0.1706
Average precision	0.03020	0.0202

There are a number of improvements that can be made on the testing phase, such as adding lemmatization as a preprocessing step, removing stop words, and spelling correction. Moreover, we have ignored out of vocabulary words, which might also influence the results due to a loss of semantics. The experiment proves that the method improves the recall of a model (given the same setup, data, and preprocessing), regardless of the model's baseline performance. The improvement is by 2% in recall.

5. Practical application – use case example

An example that inspired the idea behind this approach comes from the medical domain, where often cases there are similar concepts that are mistakenly interchanged. We present below a use case where a user can greatly benefit from adjacent information (as we also reference as similar information using our custom definition), but without being aware of the exact terminology he or she is looking for.

To give a little bit of context, *osteoporosis* is a disease in which the bones become weak and are more likely to break. Calcium is an essential mineral for the proper development of the bones. To help the body absorb the bone-boosting calcium, vitamin D is needed.

Suppose a person wants to find out how he or she can prevent osteoporosis, searching for *strengthen bone health*. The system would normally answer with relevant information such as *Calcium is a key nutrient*. The user might not know that vitamin D is needed for calcium absorption and there is no way to find this piece of information without searching for it explicitly. It would be extremely helpful for the user to also find out this additional information from the initial query. This would be possible if the query expansion would bring this semantically related information, apart from managing the correct spelling and grammar of the initial query (which is out of this paper's scope).

This is a simple example illustrating that sometimes, adjacent information that, at the first sight, is not relevant to the query or it is not directly related to it, might be really useful and hard to find without knowing the exact term or idea.

Furthermore, for complex classifications of concepts that have different properties, generating similar phrase, can be handy, making possible the detection of possible errors and new properties of the concept or similar concepts from the same class and their properties.

For example, the following pair of sentences: *aorta carries oxygen-rich blood* and *pulmonary artery carries blood laden with carbon dioxide* would, on the one hand, clarify a possible mistake (for a query like *aorta blood co2*) and, on the other hand, provide additional information (for a query like *type of blood carried by aorta*, providing the similar answer having the complementary value of the blood property, naming *deoxygenated*). The two types of concept association can be better understood from the diagram in Fig. 4, which comprises the information used for illustrating the error that can be done.

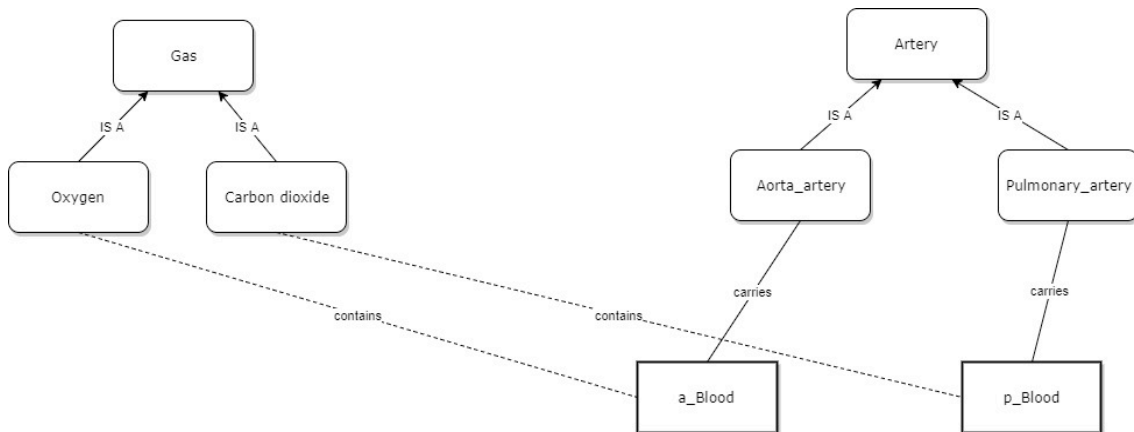


Figure 4: Similarity example from the medical domain

6. A critical review of the approach

The proposed approach aims to increase the recall for the information retrieval task, also serving as a correction system by suggesting similar phrases to the initial query. Although this is a popular task in the NLP community, the novelty of the approach draws attention over a category of similarities that are often overlooked. Thus, our work proposes a method of query expansion using WordNet. We use this approach to build a dataset of pairs of 2-word phrases that serves as a training set for machine learning algorithms that learn to capture this semantic relation between phrases.

There are a number of considerations that have been tackled in this research project and that could benefit from some of the latest progress in the NLP domain. The dataset that we have created is a good starting point for experimenting the latest advances in deep learning architectures such as the transformer model (Vaswani *et al.*, 2017) for sequence to sequence learning, along with state of the art word embedding's.

Overall, the accuracy of the proposed solution can be improved by narrowing down the considered domain for the corpus that we have used (e.g. using phrases only from the medical domain). Having a dataset for a specific topic or area of expertise will reduce the search space and considerably improve the results.

7. Future work

As mentioned in the previous section, the approach can benefit from a number of improvements that are considered as future work.

Firstly, we are considering directing the research towards building a dataset for the medical domain. These preliminary experiments aimed to proof the assumptions that query expansion could benefit from this novel approach of using a lexical resource. To assess the quality of the dataset and of the generated pairs of phrases, we have chosen to start with a general-purpose corpus and knowledge base that will allow us to understand and establish the threshold values used in our methodology. Transferring this approach to a domain-specific application can be easily achieved once the method was proven to yield relevant results.

Secondly, the machine learning module of the application can be further improved by using more complex architectures of deep learning models that currently obtain good results on different NLP downstream tasks. Last but not least, as we have also mentioned in section 2, integrating this solution in a front-end application and gathering user input can not only improve the quality of the dataset, but also customize the results towards the most used query phrases.

8. Conclusions

Our query expansion idea is based on real use cases where people can benefit from adjacent information in information retrieval systems. Our experiments prove the feasibility of building an application that solves this challenge by bringing as a novelty an approach for automatically generating a training dataset that can be further used by the state of the art machine learning algorithms for sequence to sequence natural language processing. Although the focus of this research work was not on optimizing the prediction model, we have managed to obtain good results with a simple MLP architecture. This incipient research application aimed to draw attention over the potential impact of a novel approach of query expansion over IR systems, and motivate the research community to improve this idea and extend it to domain-specific applications.

Acknowledgements

This work was 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

- Azad, H., and Deepak, A. (2017). Query Expansion Techniques for Information Retrieval: A Survey. In *Information Processing & Management*, Elsevier, 56:5, 1698-1735, <https://doi.org/10.1016/j.ipm.2019.05.009>.
- Davies, A. (1983). *A Document Test Collection for Use in Information Retrieval Research*, (Dissertation). Department of Information Studies. University of Sheffield.
- Efthimiadis, E. N. (1996). Query expansion. In *Annual review of information science and technology (ARIST)* 31, 121-187.
- Gabor, S., and Iftene, A. (2017). MIR a System for Medical Image Retrieval. In *Curative Power of Medical Data – Selected Papers of the First International Workshop MEDA 2017*. Publishing house of “Alexandru Ioan Cuza” University, Iași, 9-19.
- Gînscă, A. L., Boroș, E., and Iftene, A. (2011). Adapting Statistical Language Identification Methods for Short Queries. In *Notebook Paper for the CLEF 2011 LABs Workshop*, 19-22 September, Amsterdam, Netherlands.
- Gupta D., Carbonell J., Gershman A., Klein S., and Miller D. 2015. *Unsupervised phrasal near-synonym generation from text corpora*. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI'15)*. AAAI Press, 2253-2259.
- Iftene, A., and Baboi, A.M. (2016). Using semantic resources in image retrieval. In *Proceedings of 20th International Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES2016*, Elsevier. 5-7 September 2016, York, United Kingdom, 96, 436-445,
- Jurgens, D., Turney, P., Mohammad, S., and Holyoak, K. 2012. SemEval-2012 task 2: measuring degrees of relational similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval '12)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 356-364.
- Khosmood, F., and Levinson, R. (2011). *Automatic Synonym and Phrase Replacement Show Promise for Style Transformation*. 958 - 961. 10.1109/ICMLA.2010.153.
- Li, W., Ganguly, D., and Jones, G. J. (2016). Using WordNet for Query Expansion: ADAPT @ FIRE 2016 Microblog Track. *FIRE*, 1737, 4 pages.
- Maron, M. E., and Kuhns, J. L. (1960). On relevance, probabilistic indexing and information retrieval. *Journal of the ACM (JACM)* 7:3, 216-244, <http://dx.doi.org/10.1145/321033.321035>.
- Micu, A., and Iftene, A. (2016). Semantic diversification of text search results. In *Proceedings of 8th International Conference on Computational Collective Intelligence*. Lecture Notes in Computer Science (LNCS), Springer International Publishing. 28-30 September 2016, Halkidiki, Greece, 9876, 78-88.

- Pal, D., Mitra, M., and Datta, K. (2013). *Improving Query Expansion Using WordNet*. 18 pages, *arXiv:1309.4938*.
- Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global Vectors for Word Representation. In *EMNLP*, 1532-1543.
- Shekarpour, S., Hoffner, K., Lehmann, J., and Auer, S. (2013). Keyword query expansion on linked data using linguistic and semantic features. In *Proceedings of the 7th IEEE International Conference on Semantic Computing (ICSC)*, Springer, 191-197.
- Smith, B., and Fellbaum, C. (2004). *Medical WordNet: a new methodology for the construction and validation of information resources for consumer health*. In *Proceedings of the 20th international conference on Computational Linguistics (COLING '04)*. Association for Computational Linguistics, Stroudsburg, PA, USA, Article 371. DOI: <https://doi.org/10.3115/1220355.1220409>.
- Șerban, C., Alboaiie, L., and Iftene, A. (2016). Image and user profile-based recommendation system. In *Workshop on Social Media and the Web of Linked Data (RUMOUR 2015) at EUROLAN 2015 Summer School on Linguistic Linked Open Data*. 18 July 2015, Sibiu, Romania. Springer International Publishing Switzerland. D. Trandabăț and D. Gîfu (eds.): EUROLAN 2015, CCIS 588, 1-16, DOI: 10.1007/978-3-319-32942-0_5.
- Turney, P.D., Littman, M.L., Bigham, J., and Shnayder, V. (2003). Combining independent modules to solve multiple-choice synonym and analogy problems. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP-03)*, Borovets, Bulgaria, 482-489.
- Unger, C., Ngomo, A. C. N., and Cabrio, E. (2016). 6th open challenge on question answering over linked data (qald-6). In *Semantic Web Evaluation Challenge*, Springer, 171-177.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is All you Need. In *Advances in Neural Information Processing Systems 30*, Curran Associates, Inc., 5998-6008.
- Zhang, J., Deng, B., and Li, X. (2019). Concept Based Query Expansion Using WordNet. In *International e-Conference on Advanced Science and Technology*, 52-55, DOI: 10.1109/AST.2009.24.

INDEX OF AUTHORS

Barbu Mititelu, Verginica 3, 153	Lucaci, Diana 205
Bibiri, Anca-Diana 13, 87	Marinescu, Ioana 3
Bobicev, Victoria 139	Martinez, Maria Adelaida Gil 25
Boghiu, Șerban 13	MărânduC, Cătălina 87, 139
Boroș, Tiberiu 75	Meterez, Alexandru 75
Botârleanu, Robert 195	Mitrea, Adrian 51, 99
Coman, Andrei 39	Mitrofan, Maria 3, 39
Cotaie, Andrei 75	Moruz, Maria 115
Cristea, Dan 13, 127	Moruz, Mihai Alex 115
Cristescu, Mihaela 153	Oneață, Dan 65
Cucu, Horia 65	Onofrei, Mihaela 13
Dascălu, Mihai 195	Pădurariu, Constantin Cristian 13, 127
Dinu, Alexandru 51, 99	Păiș, Vasile 181
Florea, Anda-Mădălina 195	Pistol, Ionuț Cristian 13
Georgescu, Lucian 65	Scutelnicu, Andrei 13
Gîfu, Daniela 13	Stoyanova, Ivelina 153, 167
Gînsac, Ana-Maria 115	Todorova, Maria 153, 167
Hanu, Bogdan 51, 99	Toma, Irina 195
Iftene, Adrian 205	Trandabăț, Diana 13
Illoaica, Paul 75	Trăușan-Matu, Ștefan 195
Ion, Radu 181	Tufiș, Dan 39,181
Iordache, Florin 65	Ungureanu, Mădălina 115
Kukova, Hristina 167	Untilov, Roman 139
Leseva, Svetlozara 153, 167	VLAD, Adriana 51, 99