PROCEEDINGS
OF THE 12TH INTERNATIONAL CONFERENCE
"LINGUISTIC RESOURCES AND TOOLS FOR PROCESSING THE ROMANIAN LANGUAGE"
MĂLINI, 27-29 OCTOBER 2016

Editors:

Maria Mitrofan

Daniela Gîfu

Dan Tufiș

Dan Cristea

Organisers

Faculty of Computer Science
"Alexandru Ioan Cuza" University of Iași

Research Institute for Artificial Intelligence "Mihai Drăgănescu"
Romanian Academy, Bucharest

Institute for Computer Science
Romanian Academy, Iași

Under the auspices of the Academy of Technical Sciences

This volume was published with the support of
the National Authority for
Scientific Research and Innovation (ANCSI)

ISSN 1843-911X

PROGRAM COMMITTEE

Verginica Barbu Mititelu, Research Institute for Artificial Intelligence "Mihai Drăgănescu", Romanian Academy, Bucharest
Costin Bădică, Faculty of Automation, Computers and Electronics, University of Craiova
Tiberiu Boroș, Research Institute for Artificial Intelligence "Mihai Drăgănescu", Romanian Academy, Bucharest
Mihaela Colhon, Faculty of Mathematics and Natural Science, University of Craiova
Dan Cristea, Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași and Institute for Computer Science, Romanian Academy, Iași
Ștefan Daniel Dumitrescu, Institute of Artificial Intelligence "Mihai Drăgănescu", Romanian Academy, Bucharest
Corina Forăscu, Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași and Research Institute for Artificial Intelligence "Mihai Drăgănescu", Romanian Academy, Bucharest
Daniela Gîfu, Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași
Adrian Iftene, Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași
Andreea Macovei, Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași
Cătălina Mărănduc, Institute of Linguistics "Iorgu Iordan - Al. Rosetti", Romanian Academy, Bucharest and Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași
Mihai Alex Moruz, Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași
Ionuț Cristian Pistol, Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași
Octavian Popescu, IBM Research Center, USA
Elena Isabelle Tamba, "A. Philippide" Institute for Romanian Philology, Romanian Academy, Iași
Diana Trandabăț, Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași
Dan Tufiș, Research Institute for Artificial Intelligence "Mihai Drăgănescu", Romanian Academy, Bucharest
Marius Zbancioc, Institute for Computer Science, Romanian Academy, Iași

ORGANISING COMMITTEE

Mihaela Colhon, Faculty of Mathematics and Natural Science, University of Craiova
Dan Cristea, Faculty of Computer Science, "Alexandru Ioan Cuza" University and Institute for Computer Science, Romanian Academy, Iaşi
Lucian Gâdioi, Faculty of Computer Science, "Alexandru Ioan Cuza" University, Iaşi
Daniela Gîfu, Faculty of Computer Science, "Alexandru Ioan Cuza" University, Iaşi
Adrian Iftene, Faculty of Computer Science, "Alexandru Ioan Cuza" University, Iaşi
Andreea Macovei, Faculty of Computer Science, "Alexandru Ioan Cuza" University, Iaşi
Maria Mitrofan, Research Institute for Artificial Intelligence "Mihai Drăgănescu", Romanian Academy, Bucharest
Diana Trandabăț, Faculty of Computer Science, "Alexandru Ioan Cuza" University, Iaşi
Dan Tufiș, Research Institute for Artificial Intelligence "Mihai Drăgănescu", Romanian Academy, Bucharest

TABLE OF CONTENTS

FOREWORD

From its inception, in 2001, the ConsILR Conference (traditional acronym for *Consorțiul pentru Informatizarea Limbii Române*, an initiative born in the Section for Information Science and Technology of the Romanian Academy) was meant as a meeting place for linguists and computational linguists, but also for 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 series of events have run, with few exceptions, once every year, first in the format of a workshop, and since 2010 – as a conference. In order to reach wider visibility, the organisers decided to make the Conference itinerant and to publish its Proceedings in English. Thus, ConsILR is not strictly addressed to researchers working on Romanian language but also to other scientists, from any part of the world, which could find sources of inspiration in the models and techniques developed for our language and apply them for their own languages. Opening the gate for researchers working on languages other than Romanian to participate in the Conference and publish their work in this Proceedings, a reverse influence is also facilitated, namely that their work inspire scientists working on the Romanian language.

This year the Conference was organised at Mălini, a lovely place in the heart of Bucovina, the historical province in the North part of Romanian Moldova. It is the 12th in the series and, as usual, it aimed to advance the level of computerisation of the Romanian language with new resources and improved processing tools. The contributions were clustered in 5 chapters: Corpora Acquisition and Annotation, Tree-banks ad Dependency Parsing, Speech Data Processing, Lexical and Semantic Resources, and Practical Sessions.

The research results presented this year cover some new areas: diachronic corpora acquisition, new types of annotation for Romanian texts (temporal mark-up in linked data), a large and heavily annotated medical corpus (included in the reference corpus for contemporary Romanian language – CoRoLa, a priority project of the Romanian Academy, still under development in both Bucharest and Iaşi).

A number of articles presented in this volume report on significant advances in two areas which were mentioned as under-developed in the MetaNet White Paper Studies, namely syntactically annotated corpora and speech processing. The tree-banks and dependency parsing issues are described in Chapter 2 (the largest) and several parsers are evaluated on the already available UD-compliant tree-banks. Although the accuracy of the dependency parsers (including the one based on the TensorFlow platform, recently released by Google) is still below the state-of-the-art results for major languages, the progress is steady and we hope that soon a competitive dependency parser for Romanian will be announced.

The use of neural networks and deep learning for Romanian language processing is best exemplified in the progress on speech processing, described in Chapter 3.

The 4th chapter, a traditional section of our volumes, is dedicated to the creation, management and use of lexical and semantic resources (lexical ontologies and terminologies).

An innovation for the ConsILR volumes, in the edition of this year, is the inclusion of a special section, Short Papers, reporting on evaluation of language resources and language technology in practical analytical studies as well as informal communications based on empirical studies.

This year organisers of the Conference *Linguistic Resources and Technologies for Romanian Language* are the Faculty of Computer Science of the "Alexandru Ioan Cuza" University of Iași and two institutes of the Romanian Academy: the Research Institute for Artificial Intelligence "Mihai Drăgănescu", in Bucharest, and the Institute for Computer Science, in Iași.

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

November 2016
The editors

CHAPTER 1
CORPORA ACQUISITION AND ANNOTATION

DIGITIZATION OF ROMANIAN PRINTED TEXTS OF THE 17TH CENTURY

ALEXANDRU COLESNICOV, LUDMILA MALAHOV, TUDOR BUMBU

Institute of Mathematics and Computer Science, Chișinău, Moldova

acolesnicov@gmx.com; {lmalahov, bumbutudor10}@gmail.com

Abstract

Problem of digitization of historical-literature heritage is a domain of priority in digital agenda for Europe supported by EU with a lot of European projects. In Moldova and Romania old books were published mainly in the old Romanian Cyrillic script. This script was definitely formed in the 17th century, then from 1830 and until the official introduction of the Latin alphabet for Romanian in 1862 several mixed of Cyrillic and Latin letters transitional scripts were used. In the 20th century a different Cyrillic script was used in Bessarabia. We had described earlier the technology for digitization of printed Romanian Cyrillic texts of the 18th–20th centuries. This work refers to the 17th century as the first books in the Romanian language were printed. The typography repeated the standard of the Romanian and Slavonic Cyrillic manuscripts like insertion of some letters over the line, use of letters for numbers, etc. Therefore OCR of these texts by ABBYY FineReader meets some difficulties whose nature and overcoming is discussed.

Key words — historical and cultural heritage, OCR, Romanian Cyrillic script of the 17th century.

1. Introduction

The EU strongly supports the accessibility of cultural heritage. One more step to this aim is the Namur Declaration of 24 April 2015 that stated a European Cultural Heritage Strategy for the 21st century. The Strategy "aims to re-define the place and role of cultural heritage in Europe with a view to furthering good governance and participation in identifying and managing heritage".[2] It promotes the support of heritage activities on the national, territorial, and local authority levels.

Three components the Strategy is built around are: the social component; the territorial and economic development component; the knowledge and education component. The social component supposes to use heritage to promote diversity. The second component should reinforce heritage to contribute to economic and territorial development based on local resources, tourism and employment. The third component promotes education, training and research programs in heritage, and supposes the creation of heritage knowledge centers.

[2] http://www.herein-system.eu/strategy-xxi

Our team at the Institute of Mathematics and Computer Science in Chişinău works to widen the electronic access to old Romanian printed books and documents. Historically, due to the influence of the Orthodox Church, the Romanian language used the Cyrillic script, that was replaced by the Latin script in 1862 with the intermediary 30-year period of transition when the mix of Cyrillic and Latin scripts was used. In Bessarabia, under the USSR, a different variant of the Cyrillic script was used in the 20th century.

We worked over the recognition of Romanian Cyrillic scripts starting from the 20th century and returning back in time. Our efforts on the Romanian Cyrillic texts of the 18th–20th centuries and the initial approach to the texts of the 17th century were discussed in (Cojocaru *et al*., 2016). In this article we described in details the problems in OCR of the Romanian Cyrillic books of the 17th century, and our approach to their solution.

2. Romanian book printing in the 16th–17th centuries

The Romanian book printing began in the first decade of the 16th century in Wallachia that was one of three principalities that formed the modern Romania (Wallachia, Moldavia or Moldova, and Transylvania). It was only 60 years after the Gutenberg's invention of printing press and the printing technology. The Wallachian ruler *Radu cel Mare (Radu the Great)* initiated book printing to print Slavonic books for churches not only in his country, but also for other Orthodox Churches that used the Slavonic language in the worship. The first book printed in Wallachia in 1508 was *Liturghierul (Euchology)*. Two more Slavonic books were printed in 1510 and 1512. As to the site of the typography for these three books, most researchers name the Dealu monastery in Târgovişte. Wallachian printing activity was resumed during the reign of Radu Paisie (1535–1545) by *Dimitrie Liubavici* from Serbia. In 1544–1551 the typography produced several Slavonic books including *Tetraevanghel (Four Gospels)* that was ordered for Moldova.

Another printing house functioned in the sixteenth century in Bucharest. It has long been thought that the first printing press came into being in Bucharest only in the last quarter of the seventeenth century; the book printed in 1678 was *Cheia înţelesului (The key to understanding)*. The latest research showed that since 1573, an hieromonk named *Lavrentie* with his disciple worked in the printing house in Bucharest. In 1582 they printed two editions of *Tetraevanghel (Four Gospels)* and *Psaltire (Psalter)*.

The most important representative of the Romanian printing activity was *deacon Coresi*. Originated from Târgovişte, he learned the art of printing from *Dimitrie Liubavici*. (Other sources say that he was a Greek from Chios and his Greek name was Coresios, and that he was somehow confused with another Coresi who was an official in Târgovişte.) In 1557–1558 he printed in Târgovişte *Triod-Penticostar (Pentecostarion)*. In 1559 he crossed the mountains and settled in Braşov. With this, the printing in Târgovişte ceased for almost 90 years. Coresi found in Braşov a fruitful ground for printing books in Romanian because these circulated in

Transylvania for a long time, and the Slavonic tradition was not as strong here as it was in the two Transcarpathian principalities. The very first Romanian book, *Tetraevanghelul (Four Gospels),* appeared in 1560–1561. The publisher was Hans Benkner. The demand in Wallachia and Moldova was little because of stronger Slavonic traditions, and Coresi printed Slavonic *Four Gospels* in 1562. *Apostol (Apostle)* of 1566 was printed in Romanian. *Pravila Sfinţilor Părinţi (Rules of Saint Fathers)* end the first period of Coresi's activity in Braşov, that of association with Hans Benkner. After Benkner's death (1565) Coresi was associated with Hungarian nobleman Forro Miklos, but later he became the editor on his own. The last book printed by Coresi, and at the same time the most important of all his books, was *Evanghelia cu învăţătură (Gospel with Learning)*, or *Cazania (Homiliary)*, set to printing in 1580 and finished in 1581. The number of Coresi's printings is not exactly known; in any case, they are over 25, which represent more than half of the entire production of the books in Slavonic and Romanian of the sixteenth century (over 11,000 pages).

The 17th century can be characterized by the collision of old and new book tradition. In the 17th century the manual book copying continued: 45 names of monk scribes of the seventeenth century are known, and even a school of copyists existed at Râmnicu Vâlcea. The typography tried to reproduce the look and feel of Slavonic manuscripts, that included, in particular, the wide use of abbreviations and overline marks, denoting of numbers by letters, printing of proper names in lowercase letters including the first one, etc.

Nevertheless, the new trends found their ways. In the seventeenth century the monopoly of religious books ended. The seventeenth century is the starting century of the Romanian literature (Călinescu, 2001). Books of learning, philosophy, etc. occur, for example, *Bucoavna (ABC-book)* from Alba Iulia, Dimitrie Cantemir's *Divanul (Divan)*. Church book undergo changes that start to make a militant content, as in Varlaam's *Cazania (Homiliary)*, a gift of the Romanian language, which was printed and reprinted 17 times. In the seventeenth century printing spreads to more printing centers. While in the sixteenth century 80% of books were produced in Transylvania, in the seventeenth century most of books were printed in Wallachia. The largest share is that of Bucharest and Snagov (75%). There was diversity in graphic art, fonts, and varied ornaments. The art of book decoration by artistic elements like head ornaments and drop caps was extremely refined and developed in Romanian typographies, being later adopted by European and Russian publishers. The edition of books increased. The patronage of the book printing extended, with the most important role of rulers. Book circulation intensified: books from Wallachia and Moldavia were increasingly present in Transylvania. Moreover, Slavonic books from Braşov are found even in Russia. In the seventeenth century books were accessible and spread in wider social strata.

The rise continued not only by encouraging book production, but there were rulers who supported the development of literature (sec. XVII–XVIII). The name of *Matei Basarab* (1632–1654) is linked with the reintroduction of printing in Wallachia. In his time the first stories were written, and *Îndreptarea legii (Commentary on law)*

was published. During his reign 23 books were published in four Wallachian typographies, 12 were Slavonic, 9 were Romanian, but two were mixed Slavonic-Romanian. During the reign of *Constantin Brâncoveanu* (1688–1714) five printing presses worked at Snagov, diocese of Buzău, Râmnic, Târgovişte, and Bucharest.

In Moldova, Vasile Lupu (1634–1652) was a great supporter of book production. The Moldavian ruler addressed, in this regard, for help to Petru Movilă of Kiev. The latter sent an entire typography, and the master engraver Ilia. This printing press was installed in Iaşi, at the Three Hierarchs Monastery. The typography at the Three Hierarchs in Iaşi produced during the reign of Vasile Lupu five Romanian books and one Slavonic book.

At the same time, to propagate the teachings of the Reformation among Romanians in Transylvania, religious books in Romanian were needed. To this end, the prince of Transylvania Gheorghe Rakoczi 1 (1630–1648) brought to Wallachia not only fonts, but the whole typographies. Thus, a printing company was founded in Alba Iulia. There the Catehismul (Catechism) Calvinized was printed in 1640, which was spread not only in Transylvania but also in Wallachia.

3. Process of retro-digitization

Digitization of old books counts four stages.

The first stage is scanning that produces images of pages in one of graphical formats. The desirable quality is 600 DPI or more, with 24-bit color depth. Special book scanners and software are used during this stage.

The second step in the process of the retro-digitization is obtaining the digital version of the full text. The full text can be obtained automatically by OCR, or it can be entered manually.

Automated character recognition by the OCR program is performed using feature analysis, or by pattern matching. Modern languages can be recognized with accuracy more than 99% that is a very good result. For historical texts, irregularities, mud spots, and font mix don't permit to guarantee a high OCR quality. That's why many OCR programs provide the training feature to improve the result. The user can tune the program by loading special functions that will read the text under specific adjustments but this helps only partially.

The efficiency of the text recognition is directly depending both on quality of the input image (scan density, color depth), and on patterns (font type, font size, sharpness, deformations). Not the last factor for getting optimal text quality is the implementation and the development level of the used OCR software. Some programs are restricted in their features that cease their usage for hard OCR cases. For example, IRIS that is available free with most scanners doesn't permit to change glyph borders in its training mode. We used in our work ABBYY FineReader 12 Professional (AFR) to perform OCR.

Another method of text input is its manual retyping. It is extremely labor-intensive and implies considerable expenses for personnel.

The third step of digitization is so-called *tagging,* or selecting elements of the text structure and annotating them with some codes. This information can be presented in different formats. For example, the Text Encoding Initiative (TEI) is a consortium, which collectively develops and maintains a standard for the representation of texts in digital form.[3] TEI is oriented to the development of humanitarian sciences and publishing. The corresponding format guidelines are constantly developing. They were introduced in 1988 as SGML format, and in 2002 redefined as XML. Now TEI contains 21 modules with more than 400 elements. Therefore, TEI permits accurate semantic encoding of texts of different kind.

The final stage of the technology of digital encoding is the data dissemination. Standard metadata are indispensable for reliable access to the digital texts. Search machines can therefore find the necessary information more effectively through multiple links in the documents. The main information on the whole text is included in the header of the TEI file (metadata, bibliographic description, file history, etc.). More detailed information can be found inside tags.

Quality of the text produced by OCR is comparable with that of the manually retyped text for modern languages. For historical texts, the technology depends on the layout analysis, patterns, and historical lexicons. The error ratio can be 1–2%.

4. Problems of OCR for Romanian texts of the 17th century

As it was said, the printed books of the 17th century imitated the manuscripts with wide use of abbreviations, accents, over line signs, ligatures, monograms, etc. The typical technique was skipping a letter in a word and setting it over this word. Therefore, each book needs a time consuming process of AFR training, possibly for the whole text. AFR supports ligatures, and thus we process as a ligature each accented letter, each number written with Cyrillic letters, each abbreviation or monograph. The process includes manual setting of the ligature border, and the retyping of the text denoted by the ligature. See examples on Fig. Borders of ligatures are shown on images and with brackets in the accompanying transcription in the Latin script.

[3] http://www.tei-c.org/

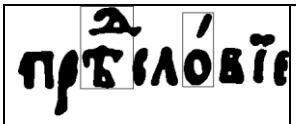
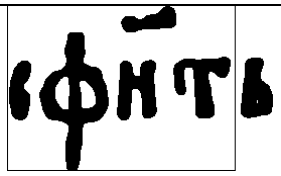
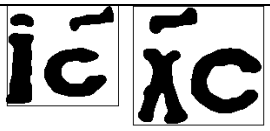
	*Pr*[*ed*]*sl*[*o*]*vie (Introduction)* Letter *d* over *e* Regular abbreviation		[*Duh*]*ului (of the Spirit)* The first *u* omitted Regular abbreviation
	Number [5]		[*Sfânt*]*i (Holy)* Letter *â* omitted Regular abbreviation
	*Ţin*[*em*] *(we hold)* Letter *m* over *e* Casual abbreviation at the line end		[*Iisus*] [*Hristos*] *(Jesus Christ)* Regular monogram

Figure 1. Examples of abbreviations in the Romanian Cyrillic books of the 17th century

Along these difficulties originated from the specific texts, we met some problems produced by the used software, namely AFR. During the training AFR doesn't permit to overpass the borders of the line. In many cases, AFR divided one line with overline signs in two lines, and we could not select true borders of ligatures to include overline signs with letters. We ask AFR developers from ABBYY for the feature to control line separation manually. This proposal was taken into account, but for the time being a roundabout solution was found: to get desirable line separation, AFR developers proposed to increase the nominal density of the image to a value greater than the optimal one recommended by the picture editor included with AFR. In our test case, the density was increased to 1200 DPI. You don't need to change the real resolution of the image that implies re-scan, but you should make if virtually changing the numerical value of the resolution stored with the image (Figure). With the unchanged size in pixels, this could be interpreted as shrinking the image to lower dimension in linear units.

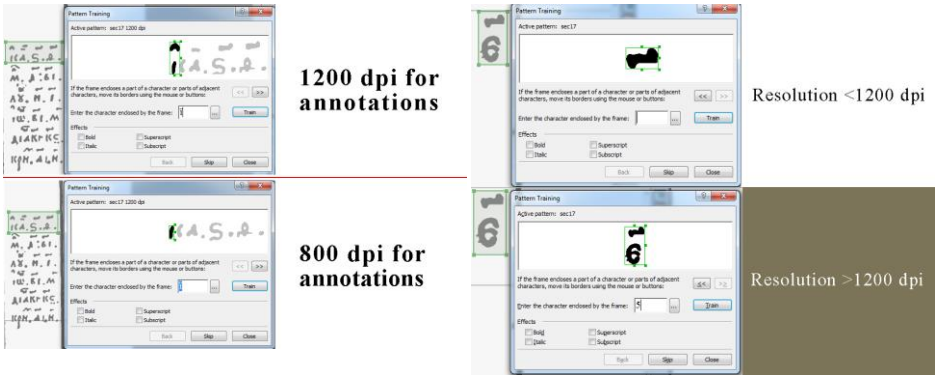


Figure 2. Image resolution and line separation in AFR

The corresponding tools are included with AFR. First of all, the user selects the page. Then he accesses the image editor through the menu PAGE | Edit image… On the right vertical pane there is the item Resolution that expands on mouse click to a dialog. This dialog shows current resolution and permits to detect optimal resolution or to introduce new nominal resolution manually, and to apply it to the image. Hint: to restore current resolution, the user should to shrink the dialog by expanding another item on the pane, and then to re-expand Resolution. The AFR image editor and Resolution dialog are shown in Fig. 3.

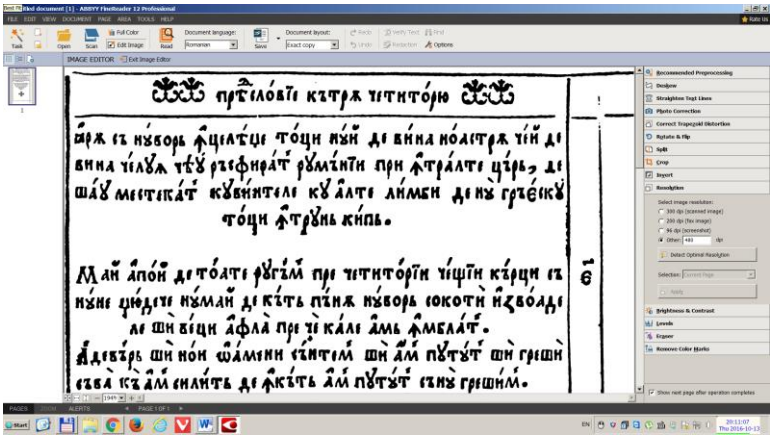


Figure 3. AFR image editor with expanded Resolution dialog (right pane)

With these procedures and the corresponding dictionary produced with the help of our Romanian colleagues from Bucharest, we got satisfactory results of OCR for Romanian texts of the 17th century but we needed the training over the whole text and further proofreading by philologists. Nevertheless this is less labor-consuming than the manual retyping. As to statistics, we got 40% of recognized text (word

level) without dictionary and training, 60% with dictionary, and more than 85% with dictionary and training. The proofreading is therefore obligatory.

Except for the feature of manual control of line separation, we contacted AFR developers with several proposals on AFR interface improvement like restoration of the now discarded possibility to select language for a block on the image, controlled AFR interface font, screen keyboards, etc.

5. Conclusion

Romanian texts of the 17th century printed in the Cyrillic script have their specificities caused mainly by the strong imitation of manuscripts in the printed books. ABBYY FineReader with the corresponding procedures based mainly on ligature identification, the spelling dictionary of the old language, and the thorough training over the whole text permits to obtain satisfactory results of OCR.

Acknowledgements

We sincerely acknowledge the company "Est Computer SRL" (str. Iazului 2/4, MD 2020 Chisinau, Moldova), and, in particular, dr. Ioachim Druguș, which granted us with the license for ABBYY FineReader 12 Professional ESD, and the developers from ABBYY for their extremely useful consultations.

References

Cojocaru, S., Burtseva, L., Ciubotaru, C., Colesnicov, A., Demidova, V., Malahov, L., Petic, M., Bumbu, T., Ungur, Ș. (2016). On Technology for Digitization of Romanian Historical Heritage Printed in the Cyrillic Script. In *Proceedings of the Conference on Mathematical Foundations of Informatics MFOI-2016*, July 25–29, 2016, Chisinau, Republic of Moldova, 160-176.

Călinescu, G. (2001). *History of the Romanian literature* (Istoria literaturei române: compendiu). Chișinău.

TEMPORAL ANNOTATION IN LINKED DATA SOURCES

IOANA IONAȘCU, TIBERIU BOROȘ

Research Institute for Artificial Intelligence, Romanian Academy

{ioana, tibi}@racai.ro

Abstract

The identification of temporal expression is crucial in natural language processing applications such as text summarization, question answering and general information extraction. We previously focused on developing a scalable NLP framework designed to work in low-resourced environments. In this paper we described how we introduced temporal annotation capabilities in our platform and evaluate how our ID3-based classifier performs on this task.

Key words — linked data, temporal annotations, natural language processing, *information retrieval*

1. Introduction

Since its emergence, the World Wide Web (WWW) has known an exponential growth leading to a large number of information sources which, more often than not, come in the form of unstructured data. The term unstructured data defines any free-form or semi-constrained text which describes a process, event or provides any type of information that one might find useful. The main issue with unstructured information is the absence of means to extract relevant answers to questions. This lead to an increasing interest in the development of automatic information extraction methods from unstructured data and the challenges involved in this process have sparked interest in the entire Natural Language Processing (NLP) field. One often employed solution is the enrichment of the text with semantic annotations which helps machines identify keywords such as locations, person names, organizations, temporal expressions etc. – commonly referred to as named entities (NEs). The usage of these named entities is two-fold: (a) they can be directly used in the construction of knowledge bases which serve as means in question answering (QA) systems (i.e.: when asked "what is the capital of Romania?" the Google QA system provides the direct answer "Bucharest"); (b) identical NEs from distinct documents are matched and linked, allowing fast navigation and retrieval of references (i.e., the Wikipedia page for Romania contains the following sentence: "Its capital and largest city, Bucharest, is the sixth largest city in the EU.", where the underlined word "Bucharest" leads directly to a page that is focused on information about the city).

The later mentioned usage describes what is known as linked data and aside from NE annotation involves an additional step in which an external reference is introduced.

Temporal expressions are particularly difficult to handle and equally important in linked data-sources because they provide important information that cannot be directly inferred from shallow parsing the data. For instance, the reference to "last Monday" does not hold any usable information unless the document date is known. The importance of recognition and classification of temporal events has been proved in practical NLP applications such as text summarization (Daniel *et al*., 2003), QA (Pustejovsky, 2002). This paper focuses on handling temporal annotation within unstructured information and in what follows we will offer a general view over our processing framework (section 4), an insight on temporal annotation challenges (section 2), present our automatic temporal annotation approach (section 4) and discuss the results and future development plans (sections 5 and 6).

2. Temporal information

TimeML represents a series of rules that aid into electronically encoding documents, with the goal of marking time expressions, which was mainly developed by the Laboratory for Linguistics and Computation at Brandeis University. The TimeML project's goal was to create a language that will mark temporal events in documents accurately, ultimately becoming an event markup standard.

TimeML manages to resolve four issues that arise for the attempt to mark events described in a document, out which worth mentioning would be, but is not the full extent: Time Stamping that allows linking an event to a time of occurrence and Event ordering by respecting one another.

It will achieve the aforementioned goals by using four base tags (**EVENT**, **TIMEX3**, **SIGNAL** and **LINK**, which is a set of tags and contains TLINK, ALINK and SLINK tags) and seven event classes (Reporting, Perception, Aspectual, I_Action, I_State, State and Occurence) which will be discussed later.

For a better understanding, we used the text "Johanna left on Monday." and applied to it the aforementioned tags and connection links which are presented in Figure 1. In this particular example, the verb "left" (EVENT) represents an occurrence which is associated to the temporal expression "Friday" (TIMEX3) by using a temporal link (TLINK), in which the temporal signal "on" (SIGNAL) makes the connection between them.

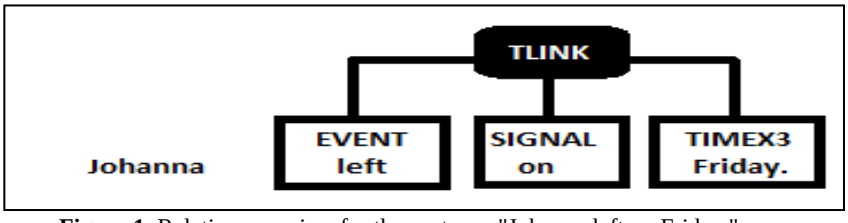


Figure 1. Relation overview for the sentence "Johanna left on Friday."

TimeML accomplishes the interpretation of the expressions by (a) identifying signals within the given document, based on several semantic aspects: temporal prepositions (*during*, *on*), temporal connectives (*before*, *while*); (b) identifying all classes of event expressions: tensed verbs (*has left, was captured, will resign*), static adjectives and other modifiers (*sunken, stalled, on board*), events nominals (*merger, Military Operation, Gulf War*); (c) creates dependencies between events and times: anchoring (*Jean left on Saturday*.), orderings (*The party happened after midnight.*), embedding (*Lucy said Travis left*.).

To better understand how TimeML works, we will continue with some annotation examples:

a) The EVENT tag – used for temporal expressions (Figure 2);

Example 1:
Sentence: *She has finally reached the goal for which she strove so bravely.*
Tagged: *She has finally* <EVENT id='1'>reached<EVENT> *the goal for which she strove so bravely.*

Figure 2. Tagging example for the EVENT tag

b) The TIMEX3 tag - used for marking explicit temporal expressions like times, dates, durations, etc. (*Monday evening*, *Thursday the 18th*, *five o'clock, January 3, 1984, October of 1963, summer of 1973*) (Figure 3);
There are four types of temporal expressions (TIMEX3) which offer the possibility to express different granularities of the events like DATES (*on the 27th of September 1990*, *last Sunday*), DAY TIMES (*this afternoon*), DURATION (*for two days, two weeks ago*) and SET (*twice a week*, *every eight hours*);

Example 1
Sentence*: I'm going on a cruise two months from next Friday.*
Tagged: <TIMEX3 tid="t1"> two months </TIMEX3> from <TIMEX3 tid="t2"> next Friday </TIMEX3>
Example 2
Sentence: *Tracy left 2 weeks before yesterday.*
Tagged: Tracy left <TIMEX3 tid="t1">2 weeks </TIMEX3> before <TIMEX3 tid="t2">yesterday. </TIMEX3>

Figure 3. Sample usage of the TIMEX3 tag

c) A SIGNAL tag is an element that makes the connection between two entities (a timex and an event, a timex and another timex, an event and another event). SIGNAL tags are, generally temporal prepositions (*on*, *in*, *at*, *from*, *to*, *during*), temporal conjunctions (*before*, *after*, *while*, *when*), special characters like "-" and "/" used in temporal

expressions which reflect a period of time (*October 1-2*, *Apr. 1957/Jul. 1957*) (Figure 4).

Example 1
Sentence: *They will repeat the dance before the competition.*
Tagged: *They will repeat the dance <SIGNAL sid="s1"> before </SIGNAL> the competition.*

Figure 4. Example of text annotated with the SIGNAL tag

As we said before, there are seven classes of events defined for TimeML:

(a) Reporting - describes an action of a person or organization making a declaration or informing about an event (*say*, *report*, *tell*, *explain*).

*No robberies were **reported** in the last month.*

(b) Perception – this class includes events that imply the physical perception of other events (*see*, *watch*, *view*, *hear*, *listen*);

*People told the police that they **saw** prisoners hiding in an old building.*

(c) Aspectual – in languages like English and French there is an aspectual grammatical device which focuses on different aspects of the history of the event like initiation (*begin*, *start*, *initiate*), reinitiation (*restart*, *reinitiate*), termination (*stop*, *cancel*, *end*, *terminate*), culmination (*finish*, *complete*), continuation (*continue*, *keep*, *persist*, *go on*);

*Lucy **completed** the test.*

(d) I_Action – introduces an event argument which is not happening in the moment that action happens. In the example below, the bolded word is an I_Action and the underlined one is the event introduced by the I_Action.

*They were **asked** to <u>accompany</u> the band.*

(e) I_State – I_State events are very alike to the ones in the previous class. This class includes statements referring to alternative or possible worlds;

*There is no reason why she would be **prepared** for [a battle].*

(f) State – describes statements presenting the truth about something or someone.

*He was in a **coma** for six years.*

(g) Occurence – this class encloses all types of events describing facts that happen in the world.

*When the **war** started all men were called to duty.*

d) LINK is a set of tags that describes different types of connections between temporal expressions in a text. The tags belonging to LINK are:

i. TLINK (-used for marking time relationships-); Lucy **went** to Rome **from the 27th to the 31st of May**

ii. ALINK (-used for annotating aspectual relationships-):

Joan **will begin** to work at the book **at 3 noon**;

iii. SLINK (-used for modal or evidentiality aspects-): Gary said he **would go** to France **in August**.

3. Automatic annotation of text with temporal information

The temporal annotation module is part of a large natural language processing framework. Before we proceed with the description of our approach we will introduce the natural language processing framework.

3.1. The MLPLA Framework

The Modular Language Processing for Lightweight Applications (MLPLA) is a language processing tool designed to work in low-resourced environments. Its primary focus is to provide an extensible framework which allows the creation of NLP-dependent applications that require low-level text processing. The basic toolset included in MLPLA consists of: (a) a text normalizer responsible for the initial processing and tokenization of the input text, (b) a part-of-speech tagger, (c) a rule-based chunker, (d) a word syllabifier, (e) a phonetic transcriber, (f) a lexical stress predictor and (g) a shallow prosodic analyzer. All these components are implemented using two basic classifiers: (i) an ID3-based decision tree classifier and (ii) a Deep Neural Network (DNN) based classifier. The usage of these classifiers was primarily based on the small foot-print models they create and the individual results at each task have been thoroughly explained and presented in Zafiu *et al*., (2015) (the ID3 classifier) and in Boros and Dumitrescu (2016) (the DNN classifier). While MLPLA was primarily intended for text-to-speech (TTS) analysis and synthesis, its extensible nature allowed us to introduce a new module in the processing pipeline which was designed for automatic labelling with time-related events.

The modular architecture of MLPLA, enables the user to create his own processing modules and to include them directly in the backbone of the processing framework. The entire code is written in JAVA and adding new modules is done by creating a JAR file which contains compiled sources that implement one of three interfaces: (a) input; (b) processing and (c) output. The role of input processors is to pre-process the input data into a list of sentences and tokens. Each token is then feed to the processing pipeline which is composed of any number of processors. Once all the processors are run on the sentences and tokens, the results are fed to the output layer, which is designed to convert the data into an interpretable format for other applications. The MLPLA framework comes with a basic chain of processors. The list of processors is controlled using an external configuration file. The format of the configuration file is very simple (see figure 5): it is divided between 3 sections (input, pipeline and output) and every section contains the JAVA packages and class names that implement the corresponding interfaces.

```
[Input]
com.ineo.nlp.language.preprocessing.BasicTokenizer
[Pipeline]
com.ineo.nlp.language.baseprocessors.BasicTagger
com.ineo.nlp.language.baseprocessors.BasicLemmatizer
com.ineo.nlp.language.baseprocessors.BasicChunker
com.ineo.nlp.language.baseprocessors.BasicParser
com.ineo.nlp.language.baseprocessors.BasicSyllabifier
com.ineo.nlp.language.baseprocessors.BasicLTS
com.ineo.nlp.language.baseprocessors.BasicStress
[Output]
com.ineo.nlp.language.formats.TabFeatureOutput
```

Figure 5. MLPLA configurable pipeline

During runtime the classes are instantiated and they are sequentially called by the framework. When creating a new processor, one can choose from a list of standard processor types: tagging, lemmatizing, chunking, parsing, syllabification, letter to sound, stress prediction and miscellaneous. The miscellaneous processor type is used to implement non-standard functionality. For example, annotation with temporal events was not anticipated as a standard functionality, thus the new module we created currently uses the miscellaneous type. The order in which these processors are used on the data is given by the order in which they are listed in the configuration file.

The standard output processors include a tab-feature output print processor, which prints the results to the standard output one word per line a HTS feature output, which converts data to HTS format for speech synthesis (see figure 5 for examples).

```
TAB FEATURE OUTPUT:
Acesta  DMSR a-'ces-ta  a ch e s t a
este  V3 'es-te  e s t e
un TSR  un u n  Np1
simplu  ASN  'sim-plu  s i m p l u  Np1
test  NSN  test  t e s t Np1
.  PERIOD  .  pau
HTS FEATURE OUTPUT (limited context provided):
#^#-pau+a=ch:/SYL:a/NSYL:3/NPHON:1/SIW:start/SYLI:0/PI:0/…
#^pau-a+ch=e:/SYL:a/NSYL:3/NPHON:6/SIW:start/SYLI:0/PI:0/…
pau^a-ch+e=s:/SYL:ces/NSYL:3/NPHON:6/SIW:mid/SYLI:1/PI:1/…
a^ch-e+s=t:/SYL:ces/NSYL:3/NPHON:6/SIW:mid/SYLI:1/PI:1…ch^e-
s+t=a:/SYL:ces/NSYL:3/NPHON:6/SIW:mid/SYLI:1/PI:1/…e^s-
t+a=e:/SYL:ta/NSYL:3/NPHON:6/SIW:end/SYLI:2/PI:2/…s^t-
a+e=s:/SYL:ta/NSYL:3/NPHON:6/SIW:end/SYLI:2/PI:2…t^a-
e+s=t:/SYL:es/NSYL:2/NPHON:4/SIW:start/SYLI:0/PI:0/…a^e-
```

```
s+t=e:/SYL:es/NSYL:2/NPHON:4/SIW:start/SYLI:0/PI:0...e^s-
t+e=u:/SYL:te/NSYL:2/NPHON:4/SIW:end/SYLI:1/PI:1...s^t-
e+u=n:/SYL:te/NSYL:2/NPHON:4/SIW:end/SYLI:1/PI:1/...
```

Figure 6. Tab feature output for the sentence "Acesta este un simplu test." (en. "This is a simple test.")

Automatic annotation of text with TimeML events is hard to achieve, especially when we consider large amounts of text or application which require that such annotations are performed ad-hoc on a previously unseen text.

While most NLP tasks can be successfully performed using data-driven classifiers, the high complexity of temporal annotation requires that hybrid approaches are used. Thus, many methods isolate between (a) identification of temporal expressions and (b) classification of temporal expressions and extraction of values. The first task can easily be performed using classical ML methods, but the later mentioned task requires carefully crafted hand-written rules which are able to cover many types of expressions.

The data-driven identification approach is inspired by the method presented in (Llorens *et al*., 2010). In their paper, the authors investigate the importance of various text-based features in both the identification and classification stage of temporal expressions. While their approach is based in CRFs, we resume to using an ID3 classifier, mainly based on the small foot-print of the resulting model. In their paper, the authors use a large number of features extracted using several external tools and resources, such as WordNet (Fellbaum, 1998), Charniak parser (Charniak and Johnson, 2005), TreeTagger (Schmid, 1994) etc. However, given the restricted resourced environment in which the MLPLA framework has to work, we had to resume to using the following features (extracted from a 5-word centered window): part-of-speech, chunk, lemma, numeric expressions, identified dates, dayOfWeek, monthOfYear, possibleYear etc.

We constructed our training and testing data using the AQUAINT TimeML corpus (Verhagen and Moszkowicz, 2008). The corpus is composed of 73 news articles which were annotated with EVENT, TIMEX and LINKS. The evaluation of the ID3 classifier was performed by training on 90% of the data and testing on the other 10%, and calculating the F-score for each individual tag: TIMEX3 (65%), EVENT (71%). The effective assignment of the value for each TIMEX3 tag is performed in the rule-based fashion. As up now, our rules fully cover the cases found in AQUAINT corpus, but more rules will be added in the future, depending on the exception cases we will encounter.

4. Conclusions and future work

The identification of temporal expression is crucial in natural language processing applications such as text summarization, question answering and general information extraction. We previously focused on developing a scalable NLP

framework designed to work in low-resourced environments. In this paper we described how we introduced temporal annotation capabilities in our platform and evaluated how our ID3-based classifier performs on this task. Future development plans include an evaluation of the DNN classifier on the same task and the extension of the platform for other languages (our primary focus will be Romanian).

References

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*, ACM, 98-102

Charniak, E., Johnson, M. (2005). Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 173-180

Fellbaum, C. (1998). WordNet. Blackwell Publishing Ltd.

Daniel, N., Radev, D., Allison, T. (2003). Sub-event based multi-document summarization. In *Proceedings of the Workshop "HLT-NAACL 03 on Text summarization"*, 9-16

Llorens, H., Saquete, E., Navarro, B. (2010). Tipsem (english and spanish): Evaluating crfs and semantic roles in tempeval-2. In Proceedings of the *5th International Workshop "Semantic Evaluation",* 284-291

Pustejovsky, J., Castano, J. M., Ingria, R., Sauri, R., Gaizauskas, R. J., Setzer, A., Radev, D. R. (2003). TimeML: Robust specification of event and temporal expressions in text. In *New directions in question answering*, 3, 28-34.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the international conference on new methods in language processing*, vol. 12, 44-49.

Verhagen, M., Moszkowicz, J. L. (2009). Temporal annotation and representation. *Language and Linguistics Compass*, 3(2), 517-536.

AN INTRODUCTION TO TIME TRACKS

ANDREEA MACOVEI

Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași, Romania

andreea.gagea@info.uaic.ro

Abstract

This paper describes a preliminary research regarding the necessity to rethink the way we look at books: the main purpose is to highlight the presence of time tracks in texts and their role as these time tracks can be used for establishing the real chronological order of a text that includes temporal switches in the form of flashbacks, flash forwards, embedded fabulae, temporal ruptures, and transitions. In this study, some guidelines related to an annotation process of time tracks are considered and also, several examples are exposed in order to exemplify these temporal phenomena and to facilitate the process of understanding those mise-en-scenes for reordering the events appeared in a book according to what stories or substories belong to the main storyline.

Key words — temporal relations, time tracks, temporal reordering

1. Introduction

Temporal information continues to be a challenging subfield of natural language processing: template based question answering, multi-document text summarization and information extraction for temporal event tracking are only few applications that involve temporal information. And reconstructing the temporal ordering covers various domains as literature, clinical narratives, news, etc.

Ordering events may cause some problems when it comes to belletrist texts; trying to find a computational representation of temporal phenomena occurring in literary texts with the aim of reordering the cursive thread of the action (the belletrist texts are chosen because there are various changes of the current direction of time that readers perceive, also flashbacks and flash forwards, temporal ruptures, switches between different stories, etc.) seems to be a milestone quite difficult to reach.

Once starting to discover these unusual temporal phenomena, the first task concerns the modality to identify and collect them for previous research. They are unusual phenomena because they cannot be represented as linear timelines as long as very often, a story is interrupted by a flashback or is told by many characters - in this case, the perspective from which the events are told changes - there are discontinuities in narrations, disparate stories develop without any apparent link between them, followed by surprising encounters of characters and merging of destinies, and the other way round is also possible, characters with common or parallel lives can disappear or their stories continue separately.

The studies regarding timelines and storylines highlight the idea that storylines include one or more timelines: and this is possible because a storyline can be established once merging the individual timelines with two or more different entities, but considering the fact that they are co-participants of at least one relevant event (Laparra *et al.*, 2015).

Despite the fact that a timeline chronologically exposes a sequence of events, there are also linear-structured timelines that focus on a single entity or character without considering the information of relevant interactions with other participants (Brants *et al.*, 2003). But, there are also systems capable to represent the story development (Shahafe *et al.*, 2013) or more complex storylines (Hu *et al.*, 2014) in an explicit matter using maps of connection between events and temporal markers. Some experiments of creating timelines have been done and the start point is represented by the news articles in English, but this type of texts differ from other narrative structures (Chambers and Jurafsky, 2009).

Considering the standards for temporal annotation, TimeML (Pustejovsky *et al.*, 2003) and ISO-TimeML (Pustejovsky *et al.*, 2010) should be mentioned in order to have an overview of the temporal markup language; TimeML and ISO-TimeML are a robust specification language for events, temporal expressions and relations in natural language processing. These standards contain a set of rules for encoding documents electronically in order to automatically identify an event and anchor it in time (time stamping of events), to order events with respect to one other (it covers the lexical and discourse properties of ordering), to resonate with contextually underspecified temporal expressions (for examples, temporal expressions as *last week, two days ago,* etc.) and to resonate about the duration of events (aspects about how long may an event or the outcome of an event last).

2. Defining time tracks

The proposed notion of time tracks (Cristea and Macovei, 2016) reveals a sequence of events or statements that an author, a narrator or a character exposes voluntarily along the storyline; called time frames or temporal plans (Macovei and Cristea, 2016) in a first stage, those connected sequences of events can be considered specific modalities where diverse narrative threads can intersect and split with the purpose of capturing the reader's attention regarding the story or the stories that develop.

This means that despite timelines (the representation of all the events chronologically exposed in a story), there is a natural order of events that the reader seems to perceive once he continues to read or even, at the end of the book. So, a first step could concern the identification of this order of events using the model of time tracks: certainly, this order will lead to a better understanding of the entire story.

The time track model includes time tracks and time segments: time tracks are made out of one or more time segments. Also, these time tracks have start points, end

points, join points and split points that reveal where they start and finish, split and join.

On the other side, time segments are groups of events exposed in a coherent order although they may commute from one time track onto a different one; this is possible without specific markers that may delimitate this specific transition.

This model also contains temporal relations (so-called times) between time segments and time tracks that reveal a certain order of time tracks and time segments that may help in deciphering the right order of the stories and substories included in a text. For all these elements (time tracks, time segments and times), an XML annotation scheme was proposed.

In Ex. 1, there can be distinguished 2 time tracks as the action of the novel is diverted to another story that occurred quite recently:

Exemple 1:

[$_{TT1}$ *Camionul a dat înapoi, apoi a ţâşnit înainte, stârnind nisipul şi ridicând un nor de praf. [..] Toată luna fuseseră semne prevestitoare ale unei nenorociri de neocolit, însă parcă numai el le văzuse. Marea se agitase săptămâni de-a rândul, iar pământul fremăta atât cât să stârnească suspiciunea unui cutremur.* $_{TT1}$]

[$_{TT2}$ *Într-o noapte, Adam fusese trezit de o asemenea clătinare, iar când se dusese la uşă şi privise afară, cocotierii unduiau în toate părţile, cu toate că nu se simţea nici o adiere de vânt. Simţea că nu mai e sigur pe picioare, iar pentru o clipă n-a mai ştiut dacă nu cumva era chiar el cel care se leagănă, şi nu copacii.* $_{TT2}$]

[$_{TT1}$ *The truck turned back, then moved forward, blewing the sand into the air and stirring up a cloud of dust. [..] During the whole month there were warning signs of an unavoidable misfortune, but only he saw them. The sea was rough weeks in a row and the land vibrated enough to arouse the suspicion of an earthquake.* $_{TT1}$]

[$_{TT2}$ *One night, Adam was roused by such a shaking, and when he went to the door and looked out, coconut trees waved in all directions, although he did not feel any breeze. He felt that he was not steady on his feet, and for a moment he did not know anymore if he was the one who sways, and not trees.* $_{TT2}$] [4]

2.1. Time track model far from TimeML conventions

TimeML is an annotation scheme specifically designed for marking events, times, and their temporal relations in text, and ISO-TimeML is a formal specification standard, also for temporal information markup in natural language. This standard aims the standardization of principles and methods relating to the annotation of temporal events.

The annotation conventions covered by TimeML aim to capture and represent temporal information using four tag types: TIMEX3, EVENT, SIGNAL and LINK.

[4] The example is back translation into English from the Romanian version of Tash Aw's book (done by the author).

TIMEX3 tag is used for dates, times, durations, and sets of dates and times and EVENT tag for annotating those elements in a text that mark the semantic events described by it (with one or more MAKEINSTANCE tags for including information about a particular instance of the event).

SIGNAL tag is necessary for temporal function words such as "after," "during, "and "when" in order to represent a temporal relationship. LINK tag is for relations; there are three relations (temporal, subordination, and aspectual relationships) with the following tags: TLINK tag (temporal link relating two temporal expressions, two event instances, or a temporal expression and an event instance.), SLINK tag (subordination relationship that involve event modality, evidentiality, and factuality) and ALINK tag (aspectual connection between two event instances). All these details provide an overview of the annotation schema of time tracks necessary in order to move on with this research.

3. Determining time tracks

In order to see if different annotators share the same ideas about time tracks and time segments, several students received a fragment of Tash Aw's book, *Map of the Invisible World*[5] and were asked to identify the time track or the time tracks that appear in that fragment.

Also, they had to mark borders for each time track and to indicate whether cue phrases announcing the transition between time tracks exist. The results show that in most cases, two out of three students had a similar representation of time tracks including time segments covered by those time tracks. The text was specially chosen for its richness of time tracks and frequent switches between them. As this is an incipient research, these preliminary results are encouraging for continuing to annotate a larger corpus within the scheme shown in this paper.

At the moment of writing this paper, one important problem resides in the fact that there are no Romanian language resources that may support temporal annotation (Cristea and Forăscu, 2006), but for annotating some fragments of Tash Aw's book, "Map of the Invisible World" Multi-document Annotation Environment (MAE) tool (Stubbs, 2011) has been used.

In the future, a new tool for annotation time tracks will be implemented and this will allow the exact configuration of the proposed scheme. So far, fifty pages from Tash Aw's book, Map of the Invisible World (the first four chapters) were annotated and ninety time segments, seven time tracks and seventy times (temporal relations) have been found throughout the text. Figure 1 shows the MAE tool used for the process of

[5] We thank to the Humanitas Publishing House for offering us the Romanian version of the book for research purposes.

annotation:

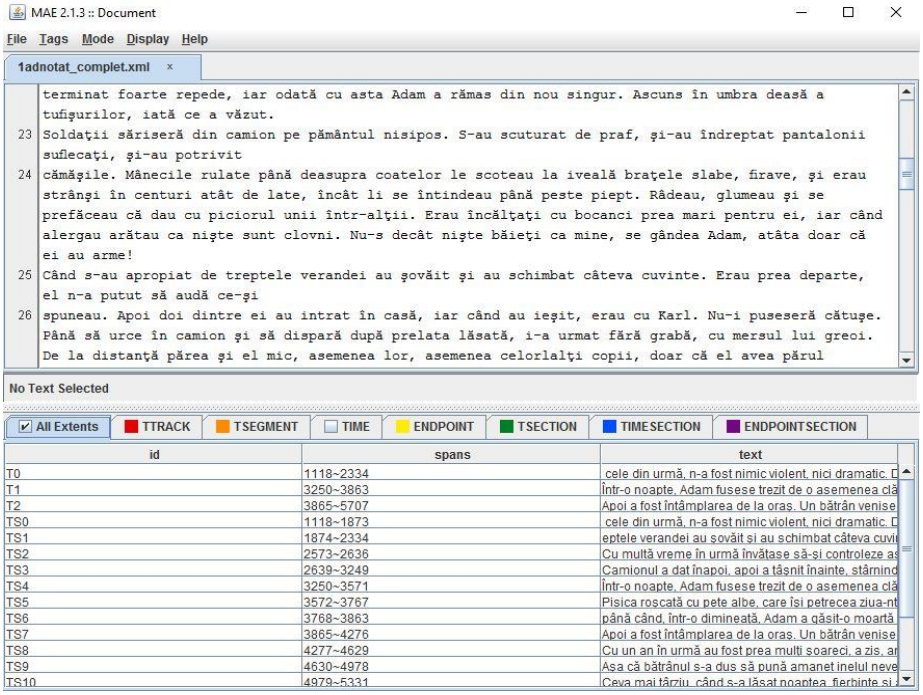


Figure 7. Annotating time tracks using MAE

4. Annotation scheme of time tracks

All the conventions of TimeML can be considered a starting point for this research as the aim is to see if there are more than one timeline in a story and to decipher those timelines (there could be various timelines as long as different stories are exposed throughout a single narrative thread, are told by different characters or include flashbacks, flash forwards, temporal ruptures, etc.).

This new scheme considers a time track (TT) with the following attributes: a specific ID, a NAME attribute given by the annotator in order to summarize the narrative thread, LEFT-ENDOINT and RIGHT-ENDOINT attributes; LEFT-ENDOINT and RIGHT-ENDOINT are attributes with the values START or STOP related to the story moments when two time tracks join or split in order to provide one or two new time tracks. ENDPOINT tags mark the endpoints of time tracks: these tags include an ID, a TYPE attribute (JOIN or SPLIT, as indicated by the annotator), a NAME attribute and the IDs of the two time tracks that get to join or split.

Then, the time segment also has an ID, an attribute IN-TT that signalizes the time track to which the time segment belongs, a NAME attribute, a TYPE attribute (the annotator chooses from a classification of time segments (Macovei and Cristea,

2016): NARrative, REMembers, SUPposition, GENeral knowledge, FICtion), and a PER attribute which comes from perspective (that indicates the person who relates the story).

On the other hand, this research regarding the time tracks does not consider the event as a minimal unit, but the time segment: this time segment regroups one or more events that are interconnected and capture a particular moment of the whole action. Annotating at the level of event requires paying attention to special details and in a literary text, there are thousands of events narrated, presented and exposed by a narrator, one or more characters. Such a time segment is exemplified in Ex.2; in Ex. 3, there is the same time segment with the afferent attributes and values resulted after the annotation process:

Example 2:

[$_{TS1}$ *Când s-a întâmplat în cele din urmă, n-a fost nimic violent, nici dramatic. De altfel, s-a şi terminat foarte repede, iar odată cu asta Adam a rămas din nou singur. Ascuns în umbra deasă a tufişurilor, iată ce a văzut. Soldaţii săriseră din camion pe pământul nisipos. S-au scuturat de praf, şi-au îndreptat pantalonii suflecaţi, şi-au potrivit cămăşile. Mânecile rulate până deasupra coatelor le scoteau la iveală braţele slabe, firave, şi erau strânşi în centuri atât de late, încât li se întindeau până peste piept. Râdeau, glumeau şi se prefăceau că dau cu piciorul unii într-alţii. Erau încălţaţi cu bocanci prea mari pentru ei, iar când alergau arătau ca nişte sunt clovni.* $_{TS1}$]

[$_{TS1}$ *When it finally happened, it was nothing violent or dramatic. Moreover, it also finished very quickly, and once again Adam was left alone. Hidden in the dense shade of cracks, here is what he saw. The soldiers jumped out of the truck on the sandy soil. They shook off the dust, straightened rolled-up trousers, and matched their shirts. Sleeves rolled-up above the elbows showed their weak and slender arms and they have so wide belts as they are spread over their chest. They laughed, joked and pretended to hit each other on feet. They were wearing boots too big for them and when they ran, they seemed to be clowns.* $_{TS1}$][6]

Example 3:

<TSEGMENT id="TS1" spans="1118~1873" text="Când s-a întâmplat în cele din urmă, n-a fost nimic violent, nici dramatic. De altfel, s-a şi terminat foarte repede, iar odată cu asta Adam a rămas din nou singur. Ascuns în umbra deasă a tufişurilor, iată ce a văzut. Soldaţii săriseră din camion pe pământul nisipos. S-au scuturat de praf, şi-au îndreptat pantalonii suflecaţi, şi-au potrivit cămăşile. Mânecile rulate până deasupra coatelor le scoteau la iveală braţele slabe, firave, şi erau strânşi în centuri atât de late, încât li se întindeau până peste piept. Râdeau, glumeau şi se prefăceau că dau cu piciorul unii într-alţii. Erau încălţaţi cu bocanci prea mari pentru ei, iar când alergau arătau ca nişte sunt clovni." in-

[6] The example is back translation into English from the Romanian version of Tash Aw's book (done by the author).

tt="T0" name="Adam_ascuns_și_apariția_soldaților"
type="NAR" PER="Narator"/>

TIME tag is used for establishing the chronological order of time tracks and time segments: this tag includes an ID, a temporal RELation (defined as BEFORE, IMMEDIATELY-BEFORE, AFTER, IMMEDIATELY-AFTER, SIMULTANEOUS), with FROM and TO attributes that are IDs of TTs or TSs and a TRIGGER attribute added by the annotator in order to give explanation about his or her choice regarding the corresponding order of time tracks and time segments.

Future work

There is much work to be done in the future. First of all, it is very important to continue with the annotation process: having an entire corpus annotated according to the above-exposed scheme means having some concrete results of our research and also doing some experiments that will reveal some statistics or perhaps, some markers that can be used for automatic extraction of time tracks.

Also, creating a visualization tool that may catch the real order of the adventures of one or more characters in a book is a further step. This tool will give a visual representation of all the stories and substories covered by a narrator, a character or an author (for example, a time sequence or a narrative thread).

In the end, developing an application that may identify the time tracks in a text will help any reader to have an overview of the entire action, once he or she finished reading the book or during the reading. Such tools will make reading an interactive activity: the reader will not have to return to various passages in the book as long as she or he has a handy instrument that can show her or him the background of a character, the climax of a story, the present time of a story, etc.

Conclusions

This paper presents a new scheme of temporal annotation having as a start point the conventions of TimeML standard: it is a new way of looking at a text with elements of time analysis and text structure.

Annotating literary texts is extremely challenging as the authors and the narrators enjoy complete freedom in order to indicate a storyline to the readers; the lack of temporal indications, the presence of flashbacks, the temporal ruptures, the embedded substories can lead the action thread in a completely different time direction from the exposed order in a book.

Although these time tracks could be interleaved and interrupted, the flow of the text is evident: the reader gets to discern quite late or quite early these temporal changes in the storytelling act as often, the switching from a time track to another one is almost imperceptible. The reader is forced to return to reading and to establish if the story follows the same narrative thread or there exists another story or stories unrelated to that thread.

As some experiments have showed, readers tend to discern the existence of these time tracks in a story and try to find some clues that mark the presence of many time tracks in a book (temporal adverbs, change of verb tenses, temporal phrases, the number of characters presented in a sequence of events, change of the perspective from which the story is told – narrator, characters, etc.). The possibility to represent them as graphs or diaphragms can contribute to a modality to restructure the story and this is a significant step in deciphering the structure of the discourse.

Acknowledgements

This survey was published with the support of the PN-II-PT-PCCA-2013-4-1878 Partnership PCCA 2013 grant, having as partners „Alexandru Ioan Cuza" University of Iași, SIVECO Romania, and „Ștefan Cel Mare" University of Suceava and of the grant of the Romanian National Authority for Scientific Research and Innovation, CNCS/CCCDI – UEFISCDI, project number PN-III-P2-2.1-BG-2016-0390, within PNCDI III.

References

Macovei, A. and Cristea, D. (2016). Time Frames: Rethinking the Way to Look at Texts. In *Proceedings of isa-12, the Twelfth Workshop on Interoperable Semantic Annotation in conjunction with LREC*, Portoroz, Slovenia, 59-62.

Cristea, D. and Macovei, A. (2016). Why time resembles rail yards? A way to look at Time in Text Books, *ROMJIST* (under publication).

Cristea, D., and Forăscu, C. (2006). Linguistic resources and technologies for romanian language. *Computer Science Journal of Moldova*, *14*(1), 40.

Stubbs, A. (2011). MAE and MAI: lightweight annotation and adjudication tools. In *Proceedings of the 5th Linguistic Annotation Workshop* (pp. 129-133). Association for Computational Linguistics.

Laparra, E., Aldabe, I., and Rigau, G. (2015). From TimeLines to StoryLines: A preliminary proposal for evaluating narratives. *ACL-IJCNLP 2015*, 50.

Brants, T., Chen, F., and Farahat, A. (2003). A system for new event detection. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval* (pp. 330-337).

Shahaf, D., Yang, J., Suen, C., Jacobs, J., Wang, H., and Leskovec, J. (2013). Information cartography: creating zoomable, large-scale maps of information. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 1097-1105).

Hu, P., Huang, M. L., and Zhu, X. Y. (2014). Exploring the interactions of storylines from informative news events. *Journal of Computer Science and Technology*, *29*(3), 502-518.

Chambers, N., & Jurafsky, D. (2009). Unsupervised learning of narrative schemas and their participants. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2* (pp. 602-610).

Pustejovsky, J., Castano, J. M., Ingria, R., Sauri, R., Gaizauskas, R. J., Setzer, A., and Radev, D. R. (2003). TimeML: Robust specification of event and temporal expressions in text. *New directions in question answering*, *3*, 28-34.

Pustejovsky, J., Lee, K., Bunt, H., & Romary, L. (2010). ISO-TimeML: An International Standard for Semantic Annotation. In *Proceedings of LREC*, La Valette, Malta.

BUILDING AND EVALUATING THE ROMANIAN MEDICAL CORPUS

MARIA MITROFAN, DAN TUFIȘ

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

{maria, tufis}@racai.ro

Abstract

In this article we present up-to-date statistics about the structure of the medical corpus of electronic documents collected in the framework of a large national project, CoRoLa, aiming at building a reference corpus for the contemporary Romanian language. This resulted in a high-quality resource that can be used in areas of natural language processing and biomedical text mining. All the texts are cleaned and transformed in a format compatible with the tools used for automatic processing.

Key words — corpus annotation, medical sub-corpus, Romanian resources, statistics

1. Introduction

In the international community there is a growing interest in managing and exploring huge amounts of medical data, clinical and research tasks (Patel *et al*., 2009). Also biomedical research has shown that there is a great need for computational techniques, due to the increasing rate of published information every year (Coleman, 2009; Gabbay, 2010).

Natural Language Processing (NLP) represents one of the technologies that is largely used for extracting valuable information from biomedical texts which include medical reference books, research papers, discharge summaries, etc. Various NLP techniques have been applied to enhance the research process (e.g. information search), having a direct impact on quality of care. For example discharged summaries have been used for detecting diabetes and obesity (Mishra NK et. all, 2012), and also to provide efficient ways to enhance the identification of depression cases, thus increasing the number of identified cases by almost a third in general population (Lucy R. Fischer, 2008).

Many NLP tools when trained in one domain and applied in different domains, may suffer performance degradation. For example the accuracy of a tagger is always influenced by the percentage of the unknown words (i.e. missing from its lexicon) and specific tokenization (as required by biomedical documents).

The standard action to overcome the performance degradation of NLP tools is their statistical models adaptation to the new domain (especially when this is very distant from the initial training domain). In case of a tagger this comes at least to updating the lexicon and name-entity recognition module.

At international level many biomedical corpora have been developed outlining the domain-specific features of texts and the types of events aimed to be recognised (e.g. GENIA is a corpus of medical literature related to human blood cells and transcription actors, BMC corpus contains full text articles provided by BioMed Central, Anatomy Corpora is a collection of corpora that contains manually annotated anatomical entities).

At national level there is a lack of linguistic resources specific to certain domains (biomedical area among others). In order to fill this gap, in 2012, the Romanian Academy Research Institute for Artificial Intelligence "Mihai Drăgănescu" (RACAI) and Institute for Computer Science in Iași started a project, that aims to create a reference corpus of the contemporary Romanian language (CoRoLa) (Barbu Mititelu, 2014). In this context we constructed an important sub-corpus for medical domain.

2. Building the medical sub-corpus

2.1. Data Collection

Collecting the texts was a difficult task, as always, because of the intellectual property restrictions and also due to lack of medical texts published in Romanian language. The main effort that we made in this direction was to contact publishing houses, editorial offices representatives to find solutions for collaborations. So far (September 2016), based on the agreements that already have been signed, the main providers, gratefully acknowledged here, of medical texts are Romanian Academy Publishing House and Polirom from which we received files in .pdf and .doc formats. Besides the texts received from the publishing houses we managed to collect an important amount of the medical corpus from different free medical online resources such as Romanian Medical Journal (http://rmj.com.ro/), medical blogs (www.pentrudiabet.ro), courses made for medical students and, books (http://federatiaromanadiabet.ro/). We considered only texts written with diacritics (because otherwise, the linguistic annotation will be incorrect) even if not the standard ones (e.g. ş vs. &scomma;, ţ vs. &tcomma;, ã vs. ă etc.) as they can be deterministically corrected.

2.2. Metadata Creation

We have created manually metadata that contain specific information like the author, source, type and genre of the text, etc. Some of the information specified in the metadata at the document level is essential for the indexing of the corpus and the facilitation the searching process for the end users. A total of 445 files have been manually associated with metadata descriptors, each of them in concordance with the metadata scheme used for CoRoLa, figure 1.

```
<?xml version="1.0"?>
-<root>

-<Metadata>
<DocumentTitle>Factori genetici implicaţi în etiopatogenia diabetului zaharat de tip 1 (insulinodependent)</DocumentTitle>
<AuthorName>Cristian Guja</AuthorName>
<PublicationDate>2006</PublicationDate>
<Source>Editura</Source>
<SourceName>Editura Academiei Române</SourceName>
<TranslatorName>-</TranslatorName>
<Medium>Written</Medium>
<DocumentTextStyle>Science</DocumentTextStyle>
<DocumentTextDomain>Science</DocumentTextDomain>
<DocumentTextSubDomain>Medicine</DocumentTextSubDomain>
<CollectionDate>2015</CollectionDate>
<SubjectLanguage>Romanian</SubjectLanguage>
<ISSN-ISBN>978-973-27-1299-3</ISSN-ISBN>
</Metadata>
</root>
```

Figure 1. Metadata scheme used in CoRoLa

2.3. Data Cleaning

The next step in the medical corpus creation was to extract the raw texts in order to be easy to process and annotate. The textual resources (usually DOC and unprotected PDF files), were converted into text format which fits our pre-processing tools (Tufiș *et al.*, 2008).

The boilerplate removal process was partially automated (Moruz and Scutelnicu, 2014). The automatic cleaning and processing steps consisted in: retrieving and extracting the texts from the .pdf files, paragraph limits recovery, deleting column marking newlines as well as hyphens at the end of the lines. Nevertheless, the texts cleaning step required additional manual processing: removal of headers, footers, page numbers, figures, tables, separating articles with different authors.

The correction of diacritics was another challenge that we met during this text cleaning process. Due to the initial formatting several texts had no correct type of

diacritics and all of them needed to be automatically replaced (*"~n aceast\ perioad\, un num\r de 12 mediciromâni [i-au sus]inut la Bucure[ti, Ia[i, Paris [i Montpellier teze de doctorat cu subiecte de endocrinologie*."). Furthermore we used regular expressions to remove the chapter titles that were interspersed with text exported from PDF files *("Sistemul endocrine și cel nervos sunt mijloace 40 ENDOCRINOLOGIE majore prin care informația circulă între diferite componente ale organismului.*").

We also decided to eliminate the sentences which were written in English ("*Test strips for blood glucose monitors are not always accurate. Diabetes Care, 26:3190,2003 8.GlucoWatch Biographer, accesat in 19.12.2003").* Various types of misspellings were also corrected: extra spaces ("*nozocomi ale*" instead of "*nozocomiale*"), missing spaces ("*implantareainvazivă*" instead of "*implantarea invazivă*"), missing letters ("*mecan*" instead of "*mecanic*"), missing dashes ("*acidobazic*" instead of "*acido-bazic*"). However at the end of the cleaning process roughly 2-3% of the words still need to be adjusted.

After this step was completed, the content was converted into UTF-8 encoding and saved as plain text documents.

3. Corpus Analytics

At the moment, we extracted from the corpus both quantitative (represented by general statistics over the corpus), as well as qualitative data related to each medical sub-domain (finer classification of the documents, according to their sub-domains). Outlines of the categories can be seen in the figure 2 and figure 3.

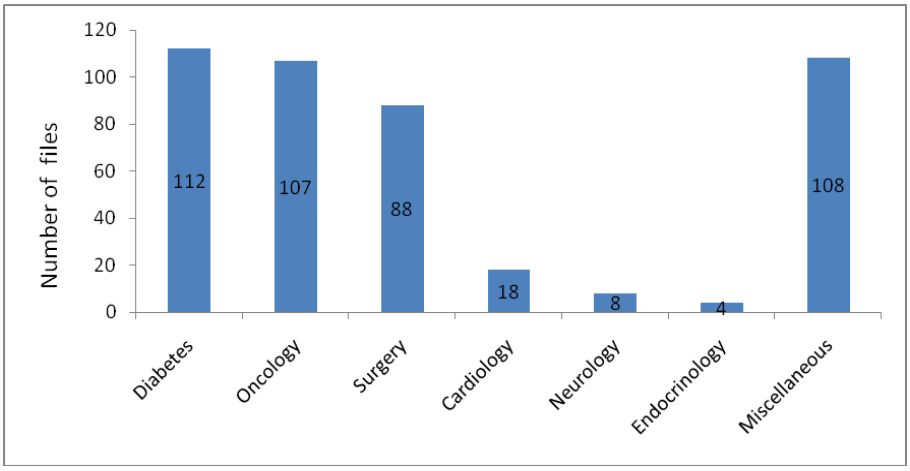


Figure 2. The distribution of the medical sub-domains in the corpus

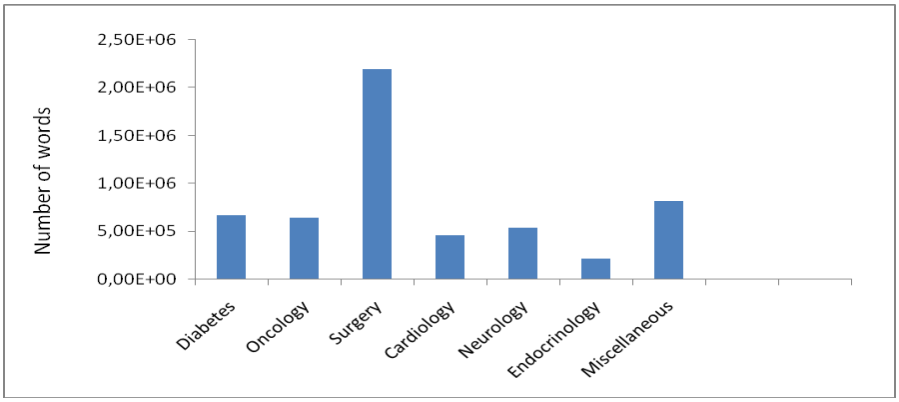


Figure 3. Number of words differentiated by sub-domains

In order to compute the general statistics over the corpus, we counted all the tokens (words plus punctuation), words (functional words and content words), lemmas and sentences. The count of the punctuation was obtained by subtracting the words count from the tokens count. Table 1 presents relevant statistics of the corpus.

Table 1. General statistics over the corpus

# tokens, punctuation included	7,173,396
# words	6,287,246
# unique lemmas	136,330
# sentences	309,948
# tokens per sentence	23.14
# words per sentence	20.28
# punctuation per sentence	2.8

The high frequency of out-of-vocabulary words is an important feature of the medical corpus that we want to highlight. Skimming through technical literature reveals a rich, field-specific vocabulary. For example, consider the following sentence taken from the medical corpus:

"Hemodinamica renală este ajustată de rezistența *microvasculară*, la rândul sau controlată de fenomenul de autoreglare și eliberare locală de factori *autocrini* și *paracrini*." ("Renal hemodynamic is adjusted by *microvascular* resistance, which is

controlled by self-regulation phenomenon and local release of *autocrine* and *paracrine* factors.")

Out-of-dictionary terms are more common in technical literature than in more-conventional text types (Baldwin *et al*., 2013). Moreover, many domain-specific technical terms are not included in general-purpose dictionaries and lexical resources. Domain-specific corpora are therefore particularly rife with out-of-vocabulary terms. This linguistic phenomenon also appeared in the medical corpus that we built. Comparing the words from the dexonline.ro database with the medical texts, we found that approximately 10.000 are not part of this lexical resource and 2% of these words come from a foreign language, most often English.

4. Corpus Annotation

Natural Language Processing (NLP) systems consists of a processing chain that ensures specific functionalities: tokenization, sentence delimitation, part-of-speech (POS) tagging, lemmatising, chunking, parsing and concept mapping. As one of the initial pipelined components, POS tagging represents the process of marking up a token in a sentence to a part of speech tag (such as noun, verb, adjective, preposition, etc.) and it is important to have an accurate POS tagger, most of the taggers targeting biomedical literature involve annotation of a training corpus. The medical corpus was undergone to a part-of-speech annotation process and for this step we used the text processing platform (TTL) developed at RACAI (Ion, 2007; Tufiș *et al.*, 2008). During the annotation process the punctuation and parenthesis were splitted from adjoining words in order to make separate tokens. The TTL tool performs specific functionalities such as: name entity recognition, sentence splitting, tokenization, tiered-tagging (Tufiș, 1999), lemmatising, POS tagging and chunking. In table 2 we present the statistics for the part-of-speech tags contained in the Romanian medical corpus.

Table 2. Number of part-of speech tags in the Romanian medical corpus

POS	**Number**
Nouns	1,832,344
adjectives	802,617
Verbs	654,449
Articles	130,706
pronouns	189,935
Adverbs	125,412
determiners	76,880
conjunctions	247,275
adpositions	790,010

numerals	203,297
abbreviations	51,556
residuals	6,891
Others	2,062,024

5. Conclusions

In this paper we presented a quantitative and qualitative description of the Romanian medical sub-corpus, which complements the previsioned structure of the CoRoLa corpus (see Barbu Mititelu and Irimia, 2014) and represents an important resource for domain adaptation of the NLP tools already available at RACAI (Tufiş *et al.*, 2008). The technical and scientific texts make use of a significant amount of domain specific vocabulary. The TTL resources (statistical lexicon and tokenizer's gazetteer) were updated accordingly and the language model (HMM3) has been re-constructed. We also want to emphasize the fact that specialised corpora allow a much closer link between the contexts in which the texts that are in the corpus were produced and the corpus, also reflect contextual features of the domain.

References

Baldwin, T., Cook, P., Lui, M., MacKinlay, A., and Wang, L. (2013). How noisy social media text, how different social media sources? In *Proceedings of the 6th Iternational Joint Conference on Natural Language Processing (IJCNLP 2013)*, Nagoya, Japan, 356–364.

Barbu Mititelu, V., Irimia, E. (2014). The Provisional Structure of the reference Corpus of the Contemporary Romanian Language (CoRoLa). In *Proceedings of the 10th International Conference "Linguistic resources and Tools for Processing the Romanian Language"*, Sept. 18-19, Iaşi, Romania, 57-66.

Coleman, K., Austin, B.T., Brach, C., Wagner, E.H. (2009). Evidence on the chronic care model in new millennium. Health Aff., 28, 75-85.

Ion, R. (2007). Word Sense Disambiguation Methods Applied to English and Romanian. PhD thesis (in Romanian). Romanian Academy. Bucharest. 138 p

Gabbay, A.R., Adelman, A.M. (2010). Future Models of Diabetes Care in *Textbook of Diabetes*. 4-th ed., Oxford, Wiley-Blackwell.

Ficher, L.R, William, A.R., Kluznik, J.C., O'Connor P.J, Hanson, A.M. (2008). Clinical Medicine & Research, 125-126

Moruz, A., Scutelnicu, A. (2014). An Automatic System for Improving Boilerplate Removal for Romanian Texts. In *Proceedings of The 10th International Conference "Linguistic resources and Tools for Processing the Romanian Language"*, Sept. 2014, Craiova, Romania, 163-170.

Ninad, K., Mishra, M., Roderick, Y., James, J. (2012). Automatic Diabetes Case Detection and ABCS Protocol Compliance Assessment, Clin Med Res. 2012 Aug; 10(3): 106-121

Patel, V.L., Shortliffe, E.H., Stefanelli, M., Szolovits P., Berthold, M.R., Bellazzi R. (2009). The coming of age of artificial intelligence in medicine. ArtifIntell. Med., 46, 5–17.

Tufiş, D. (1999). Tiered Tagging and Combined Classifiers In *Proceedings of 6th International Conference on Text,* Sept 2013, Pilsen, Czech Republic, 28-33.

Tufiş, D., Ion R., Ceaușu, A., and Ștefănescu, D. (2008). RACAI's Linguistic Web Services. In *Proceedings of the 6th Language Resources and Evaluation Conference – LREC'08*, Marrakech, Morocco.

CHAPTER 2
TREE-BANKS AND DEPENDENCY PARSING

FOLK POETRY FOR COMPUTERS: MOLDOVAN CODRI'S BALLADS PARSING

VICTORIA BOBOCEV[1], TUDOR BUMBU[2], VICTORIA LAZU[1], VICTORIA MAXIM[1], DANIELA ISTRATI[1]

[1]*Technical University of Moldova, Chişinău*
victoria_bobicev@rol.md, lazu_vica@mail.utm.md, maxivica@yahoo.com, danielaistrati.p@gmail.com
[2] *Institute of Mathematics and Computer Science, Chişinău, Academy of Science of Moldova*
tudor.bumbu@estcomputer.com

Abstract

The paper presents the on-going work on syntactic processing of folk poetry from the Moldovan Codri region. The processed ballads have been collected from rural population by Academy of Science folklorists during the 1965 – 1970 period and published in 1973 as an anthology. 100 pages from the book were scanned and recognized using the OCR (Optical Character Recognition) tool ABBYY FineReader. For a better result of recognition, we attached a dictionary and performed some pattern training. As the text was written using Cyrillic alphabet, it was transliterated in modern Romanian Latin script by applying AAConv tool, developed by Institute of Mathematics and Computer Science of ASM. Transliterated texts were processed by hybrid part of speech tagger and by automate parser which used Dependency grammar. As the folk poetry sentences present rather unusual structures, most of them were parsed with errors. Thus, the next step of our work is the manual correction of automate analysed texts. We used the TreeAnnotator tool which provides the graphical interface for the manual correction of dependency structures. The final objective of our work is to create a Treebank with at least 1000 sentences with correct dependency structures. This part will be added to the corpus of old texts in order to train the automate parser for such types of texts.

Key words — folk poetry, optical character recognition, transliteration, parsing, dependency grammar, Treebank, manual correction.

1. Introduction

Modern linguistics pays the increasing attention to diachronic and dialectal variations of language which need adaptation of the existing natural language processing tools. Historical documents are being digitized on a vast scale in cultural heritage and digital library projects in many countries. The need for historical lexical resources is increasingly recognized by the historical research community. These resources could be obtained by digitalization and implication of language technology in their preparation.

The paper presents the on-going work on all steps of historical text's digitalization. The old books are scanned, recognized by OCR tool, corrected and transliterated. Part 2 of the paper describes these steps, the difficulties in this work and the results. Part 3 reports the process of the morphological tagging and part 4 contains discussions about the last step: syntactic parsing. The final aim of our effort is the historical text's Treebank creation. This Treebank will be further used for the syntactic parser training. About 10000 of correctly tagged and parsed sentences are enough to train the automatic parser and after the training it will be able to parse old texts with minimal quantity of errors.

2. Related works

Our work is a part of the collaborative effort on the creation of UAIC-RoTb Treebank (Mărănduc and Perez, 2015). During the work, described in (Perez, 2014), the corpus was extended starting from 600 to the 1400 syntactically annotated sentences with manual correction. The latest publications reported on 10,920 sentences, containing 200,764 words and punctuation elements. The corpus contains complex sentences from all language registers including complex sentences. Their length varies from 4 to over 100 components. The further work is the creation of complex resources such as Ro-PAAS linked to the dependency Treebank. Ro-PAAS is a resource which contains the Romanian verbs and their specific argument and adjunct structures. The connection allows to reach the semantic level of text analysis and may be used for word sense disambiguation.

Several attempts were undertaken for Romanian corpora creation: (Tufiş and Irimia, 2006) presented RoCo News corpus PoS tagged automatically without manual validation; (Ion *et al*., 2012) contained a brief description of ROMBAC corpus automatically PoS tagged and chunked; (Barbu Mititelu *et al*., 2014) described a CoRoLa corpus with morphological and possible further syntactic annotation. In most cases automate annotation needs manual correction to be fully reliable. Another problem of these corpora was the lack of syntactic annotation.

A syntactically annotated corpus for Romanian was created in the framework of RORIC-LING[1] project (Hristea and Popescu, 2003). It contained around 4000 sentences manually annotated by a linguist using Dependency Grammar formalism. One of the major flaws of the corpus is the absence of the five Romanian diacritics (ă, â, î, ş and ţ) in the annotated texts. The Treebank, described in (Mărănduc and Perez, 2015) is coded in utf-8 which solves the diacritics presentation problem.

A RACAI-RoTb treebank containing 5000 sentences was presented in (Irimia and Barbu Mititelu, 2015). This corpus contains 5 sub-sections from various parts of the Romanian balanced corpus ROMBAC (Ion *et al.*, 2012). The Treebank was annotated using a reduced set of dependency relations in the attempt to keep them

[1] http://www.phobos.ro/roric/

consistent with Universal Relations (UR)[2]. UAIC-RoTb used larger set of dependency relations in order to better reflect the specific of Romanian syntax. Lately, an attempt has been undertaken to map the annotations of both corpora UAIC-RoTb and RACAI-RoTb to the Universal Dependencies set (Barbu Mititelu *et al.*, 2015).

A part of UAIC-RoTb creation is the annotation of old Romanian texts. In order the corpus be fully balanced, it has to contain text from various time frames, including previous centuries. The start of this work, the project RoDia was presented in (Mărănduc *et al*., 2016). The paper reports on annotation of New Testament of Bălgrad, downloaded from the site of the Library of the University of Cluj-Napoca; 230 sentences in popular and regional style of the language and 650 sentences in the old Romanian.

Our paper describes the further efforts to create and annotate the historical part of UAIC-RoTb.

3. The book, OCR and transliteration

3.1. The source

The book we worked with was a collection of folkloristic texts of different genres collected in villages in central region of the Republic of Moldova, so called Codri. The book was published by "Ştiinţa" publishing house in 1973 in Chişinău. It contained legends, ballads, fairytales, funny stories, jokes, poetry, songs, and other works collected by the folklore expeditions during the whole decade, approximately in 1962 – 1972. These expeditions were organised by Academy of Science of Moldova, the folkloristic department; the collection was organised by Botezatu G. G., Băeşu N. M., Junghietu E. V., Savina M. G., Tolstenco E. V., Hâncu A. C., Cirimpei V. A., Ciobanu I. D. and edited by Botezatu G., Cirimpei V. A., and Ciobanu I. D.

Every work published in the book was accompanied by the short description which included name, ages and education of the person from whom this composition has been collected, the place and time of collection, and the name of the collector who wrote the composition down. An example is presented in Figure 1. It is seen that the text was written in Cyrillic.

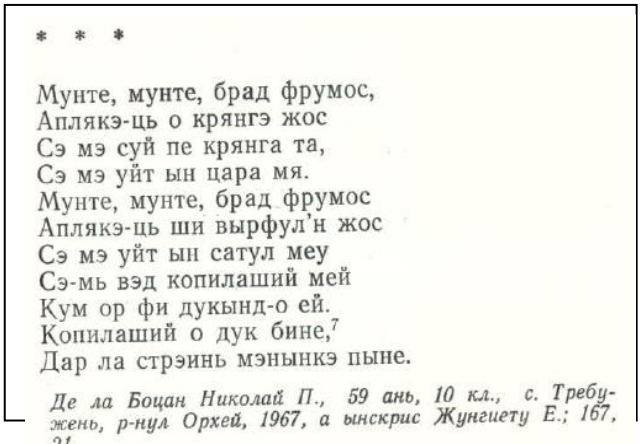
We selected the ballads and songs sections for further processing. These two types of folklore occupied almost one hundred pages of the book. Scanning was the first step of text processing. The next step was Optical Character Recognition (OCR).

3.2. OCR

Optical character recognition was performed using ABBYY Finereader tool. As the book was published in 1973, the quality of printing was quite good and the scanned

[2] http://universaldependencies.org/docs/u/dep/index.html

text is satisfying. In that period the script used for printing was Moldavian Cyrillic. Almost all letters from Russian alphabet were used excluding letters **ъ, ё, щ**. An additional character **ӂ** was used for **j** sound. For OCR process, in Finereader we chose Russian alphabet, added letter **ӂ** to it and attached a dictionary. Every language in Finereader has its integrated pattern set where all the letters of a language have their models already trained. In our case, we realized pattern training for letter **ӂ** because Russian alphabet doesn't contain models for this character. It is a supervised process, so, we found 5 words which contain this letter, trained it and made our own user pattern. In order to recognize all the letters, in the settings of FineReader we have set the option to use built-in and user patterns. The used dictionary was extracted from the first 15 pages which were recognised automatically by FR. All the mistakes form the extracted dictionary were corrected manually. The most common mistakes were generated by the messy areas of the scanned text and by confusing letters, for example: **т** recognized as **г** (бэе**т**е→ бэе**г**е), **п** recognized as **н** (**п**эскут→**н**эскут), **а** recognized as **д** (лэиц**а**→лэицд), **е** recognized as **с** and **иц** recognized as **щ** (ф**ети**ц**е**ле→ ф**стщ**еле), etc. Finally, the dictionary contained 936 unique words. After all described preprocessing, the accuracy of recognition was about 95%.



* * *

Мунте, мунте, брад фрумос,
Аплякэ-ць о крянгэ жос
Сэ мэ суй пе крянга та,
Сэ мэ уйт ын цара мя.
Мунте, мунте, брад фрумос
Аплякэ-ць ши вырфул'н жос
Сэ мэ уйт ын сатул меу
Сэ-мь вэд копилаший мей
Кум ор фи дукынд-о ей.
Копилаший о дук бине,[7]
Дар ла стрэинь мэнынкэ пыне.

Де ла Боцан Николай П., 59 ань, 10 кл., с. Требужень, р-нул Орхей, 1967, а ынскрис Жунгиету Е.; 167,

Figure 1. An example of a poem collected from Nicolai Boţan, aged 59, graduated from 10 year school. The poem has been collected and written down by Junghietu E. V. in Trebujeni village, Orhei district in 1967

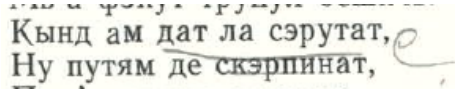
Munte, munte, brad frumos,
Apleacă-ţi o creangă jos
Să mă Sui pe creanga ta,
Să mă uit în ţara mea.
Munte, munte, brad frumos
Apleacă-ţi şi vârful'n jos
Să mă uit în satul meu
Să-mi văd copilaşii mei
Cum or fi ducând-o ei.
Copilaşii o duc bine,7
Dar la străini mănâncă pâine.
De la Boţan Nicolai P., 59 ani, 10 cl., s. Trebu-
ieni, r-nul Orhei, 1967, a înscris Junghietu E.;

Figure 2. An example of a poem after OCR and transliteration

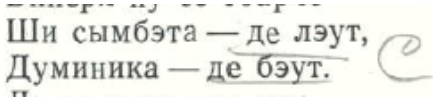
3.3. Transliteration

For transliteration we used AAConv application (Cojocaru *et al.*, 2016). It is a very useful tool for converting Romanian Cyrillic and Moldavian Cyrillic scripts into Romanian Latin script and vice versa developed by Institute of Mathematics and Computer Science of AŞM. This tool works using transliteration rules. Finally, transliterated text was of a good quality as it can be seen on Figure 2. In this example only one letter is wrongly written in upper case (line 3) and the footnote number (7, line 10) should be removed, otherwise it would confuse part of speech tagger.

Thus, transliterated text had around 3% of errors at word level and less than 1% of errors at character level. Some examples of errors are presented on Figure 3. It is seen that the errors were caused by some pencil marks made in the book.



Când am dat la sărutat,
Nu puteam detgcărşpgit,



Şi sâmbătă — de lăut,
Duminica — dë~5ăutG

Figure 3. Examples of wrong OCR

4. Morphological tagging

Morphological tagging was performed automatically using hybrid PoS tagger created by Radu Simionescu (2011) at Alexandru Ioan Cuza University, Iaşi. The PoS tagger used in this tagger is the common MSD (Morpho-Syntactic Description)

tag set created in the framework of the MULTEXT-East project for six East-European languages. MSD tag set for Romanian contained 623 tags, each coding one form of the word. For example, the tag Ncfsry is coding a noun (N), common (c), feminine (f), singular (s), direct case (r), definite (y), each letter of the tag representing one characteristic of the word. The number of tags in Radu Simionescu's tagger was reduced to 406 tags by removing less important morphological characteristics for some words.

The tagger used a morphological dictionary which contained about 1.255.200 word forms accompanied by their tags and lemmas extracted from the online source dexonline.ro. The tagger's hybrid model combined the Maximum Entropy statistical model and a system that allows writing context-based rules for constraining the possible sets of labels. The reported accuracy of the tagger was about 95%.

It is well known that any tagger accuracy heavily depends on the training and testing texts. The tagger was trained on NAACL 2003 newspaper texts and JRC-ACQUIS European Commission laws and decisions translated in Romanian. We, however, used the tagger to tag folkloric ballads and songs. Obviously, the accuracy dropped considerably for this type of texts. There were several sources of PoS tagging errors. First, we already described scanning errors which leaded to the wrong part of speech detection. Second, some errors appeared in the transliteration process. Third, some words were written with errors in the book due to local or rural pronunciation while collectors tried to preserve the original form of the words as they sounded. Some of them were simply written wrong; for example *Ce-ai **bia**, puică, ce-ai mânca?* should probably be *Ce-ai **bea**, puică, ce-ai mânca?*or ***lumânarea-n ţinea*** should be ***lumânarea-i ţinea***. Examples of transliteration errors are: ***sloi de ghiaţă,*** should be ***sloi de gheaţă***; ***Stânge,*** *maică, lumânarea* - ***Stinge****, maică, lumânarea*; ***s'a vindicat - s'a vindecat***; *în timpul **ernii** - în timpul **iernii***. The next source of errors presented different grammatical rules during the period of book editing. For example, Romanian is well known for its multiple clitics when two or even more words are merged, loosing some letters and connected with hyphen as in *mă-nvăţat* or *într-una*. In our text in many cases the apostrophe was used in place of hyphen as in the following example: *ş'apoi adă vinu'ncoace, Când m'oi mărită*. The parser was not trained to tokenize such type of words. The easiest way to solve this problem was to replace all apostrophes with hyphens and then parser tokenized the corrected texts without errors. The issue is that if we change the old texts we actually "modernize" them and they loose their originality. Ideally, they should be written as they were in the book. However, we already transliterated them; otherwise we could not process them by the tagger and compare with the dictionary.

The next type of tagging errors was made by the tagger itself due to its imperfection. The most typical errors were:

The capitalized words at the beginning of lines in the poems were tagged as proper nouns. There were cases when words connected with a hyphen were not divided and analysed apart. In such cases, the parser did not recognize the word and also considered it proper noun.

Figure 4. An example of automatically tagged text

Figure 4 presents a fragment of a poem with three capitalized words: ***La***, ***La***, and ***Drumul***. Only the first ***La*** is correctly tagged as **preposition** (Sp); the next ***La*** and ***Drumul*** were tagged as **proper nouns** (Np).

And, of course, word ambiguity was also frequent cause of parsing errors. For example, in the text: *Frunză verde pai-secară* the word ***pai*** can be a noun (*paie - straw*) or an interjection (*păi - well*). In this case it was used in sense of straw but automatic annotation labelled it as interjection. The other word *mărita* has two senses: *to marry* as verb and *greater* as adjective. In most folklore songs this word was used in the first sense whilst in modern texts the second sense was more frequent. The clitic *-i* can be either a verb or a pronoun and it also was tagged wrongly.

The problem of verb in their participle form which in most cases can be an adjective (*she was **married** (verb) – **married** (adjective) woman*) added their errors which leaded to wrong syntactic relations.

In spite of so many causes of parsing errors we obtained around 3-5% of PoS tag errors in final annotation which we had to correct manually changing lemmas and PoS tags in xml files. This result was obtained due to rigorous pre-processing: replacing apostrophes with hyphens, correction of grammatical, OCR and transliteration errors, changing punctuations and adding diacritics.

5. Syntactic parsing

Automatic syntactic parsing was performed by the variation of the Malt parser (Hall *et al*., 2006, Hall, Nilsson, 2007) trained on UAIC-RoDepTb (Mărănduc *et al*., 2015). FDG (Functional Dependency Grammar) is the formalism which is implemented in Malt parser. Its functioning is based only on dependency relations, on the training and on the morphological annotation and it can be used for any language. The accuracy of parser is determined by the amount of the training corpus.

At least 10,000 sentences with absolutely correct morphological and syntactic annotation should be provided for training in order to obtain satisfactory accuracy of automatic parsing. We used the parser trained on UAIC-RoDepTb to produce the first version of parsed texts. In some cases, especially for short sentences, the parser produced correct structures which did not require manual intervention. However, long sentences were parsed with multiple errors due to wrong morphological tagging.

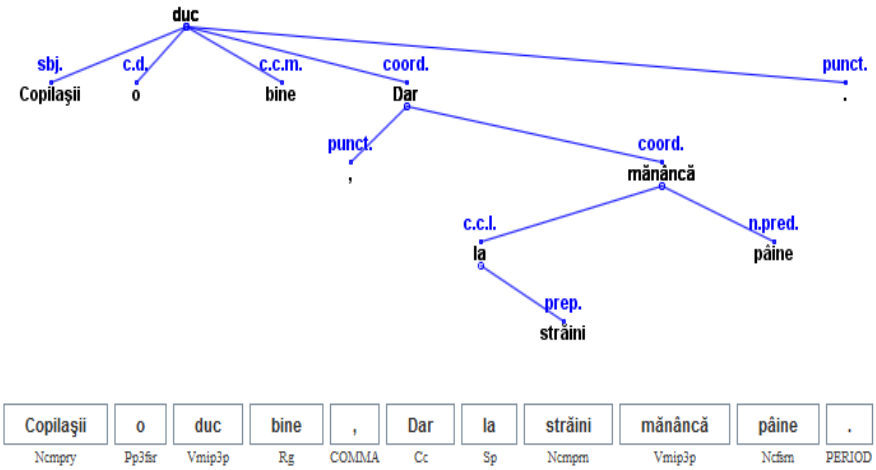


Figure 5. An example of correctly tagged and parsed sentence

Figure 5 contains a screenshot from the annotation visualization and correction tool which was used for editing the wrong annotation. Capitalized word ***Dar*** has the right tag **Cc** meaning **conjunction**. In the syntactic tree, the verb, main predicate is the head of the sentence and the second predicate is connected by coordinating conjunction.

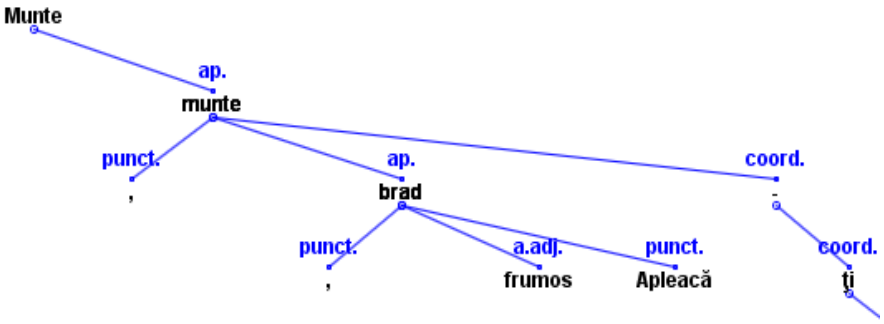


Figure 6. An example of incorrectly parsed sentence

Figure 6 presents a wrongly formed dependency tree as the head of the tree is a noun. Although ***Apleacă*** was tagged as **verb**, it is connected to ***brad*** with **punctuation** relation, which is not possible. The corrected fragment of the tree is presented in the Figure 7. The main predicate was made a head; nouns were linked to it as subordinate elements.

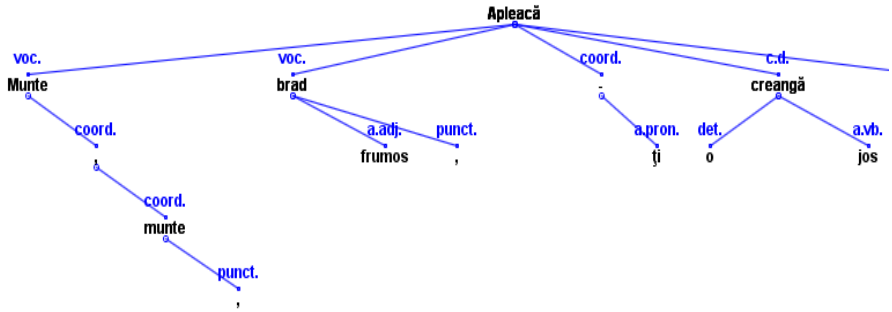


Figure 7. An example of a sentence with corrected parse tree

However, correctness of longer sentences annotation in many cases raised hot discussions among linguists. For example the sentence

Eu călare-am alergat,
Calu de gard am legat,
Pe fereastră m'am uitat,
Ce-am văzut m'am speriat:
Mă-sa, lumânarea-n ținea
Tat-so masa îi gătea,
Meșterii sicriu făcea.

This sentence contained one subject and multiple predicates in its first part. This was considered as a cause for making the subject the head of a sentence.

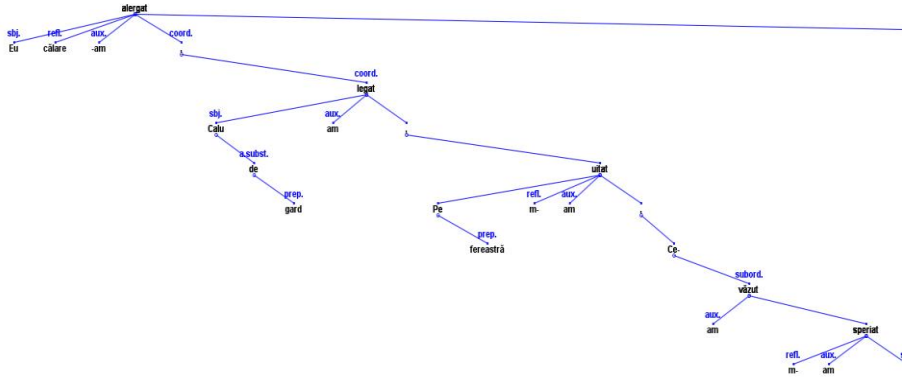


Figure 8. An example of a sentence with corrected parse tree.

One of the solutions is presented in Figure 8 where the head is ***alergat***, all the following predicates are connected in a chain with the coordination relation realized by commas.

The correction of annotation for such long sentences is rather time consuming but we cannot work with only short and simple ones. At least one third of the Treebank should be comprised by the long sentences. This provides us with a Treebank of good quality, appropriate for training.

The difficult problem presented rather loose punctuation in analyzed texts. It did not correspond to modern rules; it often violated even the punctuation rules of the time the texts were written. Probably, the publishers considered to preserve the original writing of these texts as they were transcribed by collectors from the people. However, commas and periods in the texts were in disorder and this affected drastically the structures of sentences. For example, in the fragment:

Eu bădiță, ce-aș mânca
Aceea nu mi-i căta!
Strug de mure în timpul ernii
Sloi de ghiață-n timpul verii!

The second part after the exclamation mark is not actually a sentence as it does not contain a verb which should be the head in the syntactic structure. In this case the head remained the noun ***strug.***

6. Conclusions

The presented paper discussed the problems and possible ways of their solution in our work on folk poetry from the Moldovan Codri region digitalization. The processed ballads have been collected from rural population by Academy of Science folklorists during the 1965 – 1970 and published in 1973 as an anthology. The text was written using Cyrillic alphabet and transliterated in modern Romanian Latin script by applying AAConv tool, developed by Institute of Mathematics and Computer Science of ASM. The hybrid part of speech tagger and automate parser were provided by Al. I. Cuza University, Iași. The work on the folk poetry parsing is a challenge for our linguists because of rather unusual syntactic structures. The manual correction of automate analysed texts is not a tedious repetitive work. Each sentence in each poem has its specifics which require considerable creativity in annotation. The TreeAnnotator tool provides us the graphical interface for the manual correction of dependency structures and made the correction easier. The final objective of our work is to create a Treebank with at least 10000 sentences with correct dependency structures. This corpus will allow us to train the automate parser for such types of texts.

Acknowledgements

We are grateful to Cătălina Mărănduc, who helped us with tools, examples and suggestions during the whole workflow.

References

Cojocaru, S., Burtseva, L., Ciubotaru, C., Colesnicov, A., Demidova, V., Malahov, L., Petic, M., Bumbu, T., Ungur, Ș. (2016). On Technology for Digitization of Romanian Historical Heritage Printed in the Cyrillic Script. In *Proceedings of the Conference on Mathematical Foundations of Informatics*, MFOI'2016, Chisinau, Moldova.

Simionescu, R. (2011). Hybrid POS Tagger. In *Proceedings of the workshop "Language Resources and Tools with Industrial Applications"*, Eurolan, 2011.

Hall, J., and Nilsson, J. (2007). CoNLL-X Shared Task: Multi-lingual Dependency Parsing, MSI report 06060, Växjö University, School of Mathematics and Systems Engineering.

Hall, J., Nivre, J. and Nilsson, J. (2006). Discriminative Classifiers for Deterministic Dependency Parsing. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, (COLING-ACL).

Mărănduc, C., Perez, C.-A., Balmuș, R.-Ș. (2015). Aligned Dependency Treebank English-Romanian-French. In *Proceedings of ConsILR 2015*, Iasi, Romania.

Barbu Mititelu, V., Irimia, E., Tufis, D. (2014). CoRoLa - The Reference Corpus of Contemporary Romanian Language. *LREC*, 2014, 1235-1239.

Ion, R., Irimia, E., Ștefanescu, D., Tufiș, D. (2012). ROMBAC: The Romanian Balanced Annotated Corpus. *LREC*, 2012, 339-344

Tufiș, D., Irimia, E. (2006). RoCo-News: A Hand Validated Journalistic Corpus of Romanian, *LREC*, 2006.

Hristea, F., Popescu, M. (2003). A Dependency Grammar Approach to Syntactic Analysis with Special Reference to Romanian. In *Building Awareness in Language Technology*, 9-34. University of Bucharest Press, Bucharest.

Mărănduc, C., and Perez, A.-C. (2015) A Romanian Dependency Treebank. In the *International Journal of Computational Linguistics and Applications* Vol. 6 No. 2 July-December, 25-40.

Irimia, E.,Barbu Mititelu, V. (2015). Building a Romanian Dependency Treebank. *Corpus Linguistics 2015*, Lancaster, UK, 21-24, July 2015.

Barbu Mititelu, V., Mărănduc, C., Irimia, E. (2015). Universal and Language-specific Dependency Relations for Analysing Romanian. *DepLing 2015*, 28-37.

Perez, A.-C. (2014). Linguistic Resources for Natural Language Processing. PhD thesis, Al. I. Cuza University, Iași.

Perez, A.-C., Mărănduc, C., Simionescu, R. (2015). Ro-PAAS - A Resource Linked to the UAIC-Ro-Dep-Treebank. In *Proceedings of MICAI (1) 2015*, 29-46.

Mărănduc, C., Malahov, L., Perez, C.-A., Colesnicov., A. (2016). RoDia project of a regional and historical corpus for Romanian. In *Proceedings of MFOI*, Chișinău, 2016, 268-284.

DEPENDENCY PARSING WITHIN NOUN PHRASES WITH PATTERN-BASED APPROACHES

MIHAELA COLHON[1], DAN CRISTEA[2,3]

[1] *Department of Computer Science, University of Craiova*
mcolhon@inf.ucv.ro
[2] *Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iasi*
[3] *Institute for Computer Science, Romanian Academy – Iaşi,*
dcristea@info.uaic.ro

Abstract

In this article we present a method for applying syntactic patterns to detect dependency relations within noun phrases. The patterns are extracted from a corpus automatically annotated for tokens, sentence borders, part of speeches and noun phrases, to which dependency relations between words were manually added. The patterns have been generalized in order to cover more instances that were present in the corpus and to reduce the size of the acquired model. The evaluation shows promising results for the development of a dependency relation recognizer for Romanian.

Key words — dependency relations, pattern-based approaches, Morpho-Syntactic Description.

1. Introduction

Two of the most used formalisms to describe syntactic structures are in terms of hierarchies of constituents and dependency relations. Constituents are contiguous sequences of words grouped under non-terminal symbols, part of context free grammars, while dependency relations (Mel'čuk, 1987) are asymmetrical functional relations between pairs of words, considered head and modifier. While these two traditions have sometimes been presented as competing with each other, there also seems to exist a straightforward correspondence between a projective dependency analysis (in which there are no crossing links) and a constituent structure analysis (Brody, 1994).

Reliable dependency parsing is a notorious difficult problem in Natural Language Pro-cessing. We describe in this paper a pattern-based approach in dependency parsing that addresses only Nominal Phrases (NPs) that display a rather limited recursivity and that can be identified automatically by reliable chunkers. The idea is that, generally, NP constituents cover a significant part of a sentence and if their dependency structures would be correctly identified, the whole structure of a sentence could be more easily retrieved. Moreover, rather often, NPs, sometimes augmented to prepositional phrases (PPs), fulfil semantic roles around verbs. The ambiguity of attaching an NP/PP to a verb can be reduced by benefiting from a

semantic roles parser. Therefore, an approach for depicting the hidden dependency structure of NPs can be combined to other syntactic and semantic methods, on the way to build the whole dependency structure of a sentence.

To build a dependency treebank, the human experts must decide for each word which is the one it depends on. However, rather often there is no consensus among annotators on what the correct dependency structure for a particular sentence should be, because the decision regarding dependencies involves a deep interpretation process. In contrast, in building phrase-structures, human annotators are confronted to much less ambiguity. This is because only a sequence of constituents should be indicated in the right hand side of a grammar rule, not also their relative roles/functions in the parent constituent, as indicated by the left symbol. It is therefore normal that the effort of establishing correct dependency structures be paid back, and the difference stays in a much closer resemblance of a dependency structure to a semantic interpretation then in the case of a constituent structure. This aspect is even more important in the case of languages that have a free word order (as are for instance Czech, Romanian, etc.), where dependency treebanks are preferred to constituent structure representations (see, for instance the Prague Dependency Treebank (Hajič *et al.*, 2000)).

Using treebank data for training and evaluation of parsing systems is identified under the name of treebank parsing, a methodology that has been used to construct robust and efficient parsers for several languages over the last ten years (Marinov and Nivre, 2005). For this kind of parsing, the treebank data is used to train the parser but also to evaluate the quality of the resulted parser with respect to accuracy as well as efficiency. Călăcean and Nivre (2009) report early results on a MaltParser-based dependency (Nivre *et al.*, 2006) for Romanian, trained on a manually annotated Romanian Treebank[1] built as result of the RORIC-LING project (Hristea and Popescu, 2003). Their precision for recognising labelled relations are between 60.8% and 95.9%, depending on the length of the link, while the recall is in the range 71.3% - 96.3%. Their corpus includes only short sentences (with an average of 8.94 tokens per sentence) and a gold standard part-of-speech annotation.

In this paper we present an on-going research on a treebank dependency parsing mechanism that is restricted to NP chunks. We work on Romanian, but the method is general enough to be applicable to other languages as well. In Section 2 we present the Treebank. Section 3 describes the organisation of a collection of patterns extracted from an automatic annotation to NP chunks over the Treebank. Section 4 describes the evaluation method and the results and Section 5 formulates a number of conclusions.

[1] http://www.phobos.ro/roric/texts/xml/

2. The Treebank

Presently, two independently built dependency treebanks for the Romanian language exist, one developed in the NLP-Group of the Faculty of Computer Science of the "Alexandru Ioan Cuza" University of Iași (Mărănduc and Perez, 2015) and the other in the Romanian Academy Research Institute for Artificial Intelligence, in Bucharest (Irimia and Barbu Mititelu, 2015). We have used the treebank built at UAIC-FII in the training and evaluation stages of our proposed mechanism. The corpus contains Romanian texts selected from a wide range of genres and registers of language[2]. Three levels of annotation have been added to the raw text and encoded in XML, by adopting a simplified form of the XCES standard (Ide *et al.*, 2000):

- Level-1: segmentation and lexical information. Sentences have their boundaries marked and each token has attached its part of speech, lemma, and morpho-syntactic information, by running an automatic processing chain that includes: segmentation, tokenisation, POS-tagging and lemmatisation (Simionescu, 2012a);

- Level-2: noun phrases. By exploiting Level-1 information, an NP-chunker (Simionescu, 2012b) adds information regarding noun phrase boundaries and their head words;

- Level-3: syntactic dependency data. During a manual annotation phase (Perez, 2012) each token of all sentences has been complemented with its head-word and the dependency relation towards the head.

The Treebank thus acquired contains 2,630 sentences, in which 13,038 NP structures were identified by the NP-chunker and manually corrected. Let's note also that all NPs are contiguous and if two NP spans intersect then they are necessarily nested.

3. The database of NP patterns

The primary data extracted from the corpus are used to associate dependency structures with each sequence of MSD[3] tags corresponding to NPs extracted from the corpus, which we will call in the following morphological structures. As we have already said, the corpus used in this study puts in evidence three levels of annotations: POS tags, NP chunks and dependency relations. For each NP chunk, we extracted the morphological structure and the configuration of dependency relations manually marked among the words of the NP chunk.

To syntactic constituents of the sentence correspond dependency structures organized in sub-trees (with respect to the global tree of the sentence they are part of). Each node of such a sub-tree is a word, connected with a dependency relation to its head word, above it. The roots of these sub-trees are the only elements related to words outside the constituents themselves. The only exception is the very root of the

[2] The corpus mainly includes Romanian translations of the first chapter of George Orwell's novel "1984", Romanian parts from the JRC-Acquis corpus, Romanian Wikipedia texts, grammar texts used in high-schools, etc.

[3] Morpho-Syntactic Description – notation used in the Multext projects (Erjavec, 2010).

sentence which is not related to another word. Since an NP is a sentence constituent, its head word is the only word of the NP related outside the NP itself.

For example, if we take the following bracket representation for a Romanian noun phrase "*lamele de ras tocite*" (En. "*the blunt razor blades*"):

```
[NP [Ncfpry lamele] [Spsa de] [Ncfsry ras] [Afpfp-n tocite]]
```

its MSD structure is: `Ncfpry Spsa Ncfsry Afpfp-n`[4] and its internal dependencies are:

```
a.subst.(lamele-1,de-2)
prep.(de-2,ras-3)
a.adj.(lamele-1,tocite-4)
```

In the notations above the name of relation is placed in front of a pair of words, the first one being the head and the second – the modifier word. The number attached to a word denotes its position inside the chunk.

4. Generalizing the NP patterns

Our method for generating and applying corpus-based patterns in dependency parsing works as follows: starting from flat NP sequences and using a set of syntactic patterns extracted from the training corpus, we identify dependency links between the words of the NPs, based on their MSD fingerprints. By this, the flat NP sequences become dependency sub-trees. In order to do that, the dependency relations, considered independent one of the others, are decoupled from the particular morphological structures they occur in. A generalisation is then attempted on the set of contexts of each relation. The aim of the generalization is to reduce the number of patterns for dependency relations extracted from the corpus, but also to infer deducible sequences not instantiated in the corpus. During the pattern-generalization process, the following steps are repeated for all records of the database in which a relation *R* occurs between identical tags: for each `<seq-rels>` sequences in the database, three types of contexts are marked:

- *left context*: is represented by the sequence of MSD tags in `<seq-msd>` appearing to the left of the position of the first element involved in the dependency *R*;
- *middle context*: is represented by the sequence of MSD tags appearing in between the two tags involved in the targeted dependency relation (Figure 1);
- *right context*: is represented by the sequence of MSD tags appearing to the right of the second element involved in the targeted dependency relation *R*.

[4] See the Appendix for the meaning of the MSD tags in this paper.

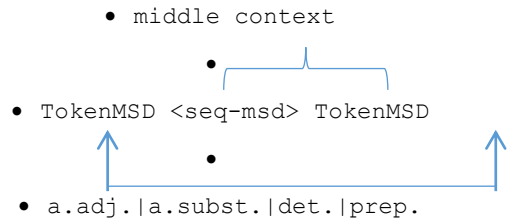


Figure 1. Considered MSD sequences for the evaluation task

For each dependency relation, the dependents are represented by two MSD tags: the MSD tag for the head is noted with `1:` and the modifier MSD tag is marked with `2:`. The contexts (if they are present in the pattern structure) may be optional or mandatory: the optional ones are marked with `?` and the mandatory ones with `{1}`.

Let us now take three examples of NPs:

NP#1: "*Fetele acestea frumoase*" (En: "*These beautiful girls*")

NP#2: "*Mâinile lui curate*" (En: "*His clean hands*")

NP#3: "*Lumânările aprinse*" (En: "*candles lit*")

In all these NPs the same relation (*adjectival attribute*, noted with "*a.adj.*") is found between words displaying identical MSD tags (that is `Ncfpry` as head and `Afpfp-n` as modifier). The following patterns for this dependency relation are found:

From NP#1:

```
<1:Ncfpry Dd3fpr- 2:Afpfp-n>
```

From NP#2:

```
<1:Ncfpry Pp3mso- 2:Afpfp-n>
```

From NP#3:

```
<1:Ncfpry 2:Afpfp-n>
```

After the generalization process, all these representations will be merged into a single one with an optional middle context:

```
<1:Ncfpry (Dd3fpr-|Pp3mso)? 2:Afpfp-n>
```

The middle context in the generalized pattern is optional because it can either contain one of the tags separated by the "or" operator ("|") or can be an empty one like in the last example.

Also, multiple MSDs that are identical modulo one position and the variation in that position covers all possible values, are generalised by replacing the values in the position with un underscore:

`Pp3mso` and `Pp3fso => Pp3_so`

After generalisation, the patterns extracted from the corpus were grouped into several sets, conforming to the dependency relations annotated in the entire span of each pattern. We have kept from the resulted sets the first four, which correspond to the most frequent dependency relations found in the corpus. We obtained thus four sets of distinct NP patterns, as follows: a set of 776 patterns for the *adjectival attribute* relation (*a.adj*.), 863 patterns for the *nominal attribute* relation (*a.subst.*), 280 patterns for the *determiner* relation (*det*.) and 599 patterns for the *prepositional* relation (*prep.*).

5. Evaluation

The goal of this research is to improve the accuracy of a dependency parser. In order to evaluate the usefulness of the extracted patterns for recognising dependency relations, we tested several classifiers on the obtained dataset using the Weka toolkit (Witten *et al.*, 2011). Weka is a collection of machine learning algorithms for data mining tasks. In order to use the Weka software package, it was necessary to transform the NP patterns dataset into ARFF files, one for each dependency relation type. In this transformation, for the time being, we ignored the left and the right contexts of the NP patterns, thus the ARFF files contains the MSD sequences that result after expanding the NP patterns, where between the first and the last token of the sequence there is always realized one of the four dependency relations considered for this study, mentioned above.

In this manner the problem of recognizing a dependency relation is reduced to a classifier with two classes: recognize *the realization or the non-realization of a certain dependency relation between the first and the last token of the given MSD sequences*. For this reason, we enriched each of the four ARFF files previously built and which correspond to a certain dependency relation with an equivalent number of patterns from the other three ARFF files. These last patterns will represent the false cases for the classifiers.

In Table 1 it is presented the dataset used for evaluation: the first column contains the relation types, the second column shows the number of distinct patterns which were created in the pattern-based relation extraction phase; the length in tokens of the MSD sequences resulted by expanding the NP patterns is given in the third column. The size of the ARFF files created in order to evaluate the classifiers is given in the forth column, together with the percentages of positive cases which show how many of the candidates really hold the relation type – they represent the accuracy of pattern-based classification, and can be used as baseline of the classifiers accuracy.

Table 1. The dataset

Dependency relations	# of distinct patterns	Average no. of tokens in MSD sequences	The dataset (positive cases)
a.adj	776	2.41	1537 (52, 4%)
a.subst.	863	3.12	1760 (51%)
det.	280	2.23	533 (52, 5%)
prep.	599	2.25	1182 (50, 7%)

The dataset was split into a training set and a testing set. The recognizer was trained on approximately 90% of each class of MSD sequences and evaluated on the remaining 10% using a 10-fold cross-validations policy, which guaranteed no intersection between training and evaluation sentences. No restrictions of projectivity of the generated dependency structures have been included at this moment. Also, no normalisation of the data was considered in the training or testing data.

We applied three of the Weka two-class classifiers on our dataset: Naïve Bayes MutinomialText (under Weka *bayes* classifiers group), SGDText (Weka *functions* classifiers group) and CVParameterSelection. (Weka *meta* classifiers group).

Multinominal Naïve Bayes is a version of Naïve Bayes that is commonly used in text classification problems. This classifier takes each text as a collection of words in which the words' order is considered irrelevant. *Stochastic Gradient Descent for Text* (SGDText) is a classifier specifically designed for working on texts. SGDText employs the *Stochastic Gradient Descent* (SGD) algorithm (Zhang, 2004) combined with the Weka *String To Word Vector* (STWV) filter in order to build the dictionary and update it in successive iterations. Weka's meta-learner *CVParameterSelection* searches for the best parameter settings by optimizing cross-validated accuracy on the training data (Witten *et al.*, 2011), without being a specialized text classifier.

As expected, the best evaluation results are obtained with the *Naive Bayes MultinomialText* and *SGDText* classifiers, as it is illustrated in Table 2. From these two, *SGDText* shows the highest accuracy. The least precise classification is obtained for the *a.subst.* relation. A possible explanation for this could be the length of the sequence, which, on average, is the longest (see Table 1).

Table 2. Evaluation of the different classification techniques

Dependency relations	Naïve Bayes MultinomialText			SGDText			CVParameterSelection		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
a.adj	0.93	0.93	0.93	0.94	0.94	**0.94**	0.27	0.52	0.36

a.subst.	0.66	0.64	0.63	0.79	0.79	**0.79**	0.26	0.51	0.35
det.	0.92	0.92	0.92	0.93	0.93	**0.93**	0.28	0.52	0.36
prep.	0.88	0.87	0.87	0.93	0.92	**0.92**	0.25	0.50	0.34

6. Conclusions

It is well-known that there is a difficult trade in designing proper generalization patterns, because making them too lax could trigger false instances. On the other hand, making them too straight will imply low recall scores. There is perhaps more to be done in this direction.

We are aware that the model would gain in precision if lexical information would be included, by enriching the MSD tags of the generated patterns with lemmas. The disadvantage is that lexical information presupposes a much larger training corpus.

It remains for further work to enrich the model by taking into consideration also the left and right contexts, as described in Section 4. Same as for lexicalised patterns, more data will be needed in this case.

Another issue to concentrate on in the future are the errors made by the NP-chunker in detecting the boundaries of NPs, for instance in cases of wrong PP-attachments. As said in the Introduction, the disambiguation of the head of a PP (a preceding noun or a preceding verb) should be tried on semantic grounds, for instance a semantic role labeller. In all, three processes should collaborate to achieve a correct parsing: an NP-chunker, a semantic role labeller and a syntactic parser. The decision of each of them, taken independently, may corroborate well or may disagree with the others. As such, a strategy able to integrate more processes' decisions into a coherent whole, would only produce the correct output. This also is an interesting subject of reflection.

The study, as presented here, has more a theoretical value then a practical, applicative, one. Its practical utility (for instance, as part of a syntactic parser) should be proved by comparing it with the accuracy of a classical (MALT) parser that would be trained on identical conditions. This is also planned for future work.

Appendix

Table3. The following table gives the notation used in this paper

MSD tag	**The meaning of the notation (according to MULTEXT-East lexical specifications)**
`Afpfp-n`	Adjective qualifier positive feminine plural -definiteness
`Dd3fpr-`	Determiner demonstrative third feminine plural direct
`Ncfpry`	Noun common feminine plural direct +definiteness

Ncfsry	Noun common feminine singular direct +definiteness
Pp3mso-	Pronoun personal third masculine singular oblique
Spsa	Adposition preposition simple accusative

References

Brody, M. (1994). Phrase structure and dependence. Working papers in the theory of grammar. In *Theoretical Linguistics Programme*, Budapest Univ, 28 pages

Călăcean, M. Nivre, J. (2009). A Data-Driven Dependency Parser for Romanian. In *Proceedings of TLT-7*.

Erjavec, T. (2010). Multexteast version 4: Multi-lingual morphosyntactic specifications, lexicons and corpora. In *Proceedings of LREC 2010*.

Ide, N., Bonhomme, P. and Romary, L. (2000). Xces: An xml-based encoding standard for linguistic corpora. In *Proceedings of the Second International Language Resources and Evaluation Conference. Paris: European Language Resources As-sociation*.

Hajič, J., Böhmová, A., Hajičová, E., Vidová-Hladká, B. (2000). The prague dependency treebank: A three-level annotation scenario, A. Abeillé, editor. In *Treebanks: Building and Using Parsed Corpora*, Amsterdam: Kluwer, 103-127.

Hristea, F. and Popescu, M. (2003). Building Awareness in Language Technology, University of Bucharest Publishing House.

Irimia, E., Barbu Mititelu, V. (2015). Building a Romanian Dependency Treebank. In *Corpus Linguistics 2015*. Lancaster University, UK.

Marinov, S., Nivre, J. (2005). A Data-Driven Dependency Parser for Bulgarian. In *Proceedings of TLT 2005*, 89-100.

Mel'čuk, I. (1987). Dependency Theory: Syntax and Practice, Albany, NY:SUNY Press.

Nivre, J., Hall, J. Nilsson, J. (2006). MaltParser: A data-driven parser-generator for dependency parsing. In *Proceedings of LREC-2006*, 2216-2219.

Perez, C. A. (2012). Casuistry of Romanian functional dependency grammar. In *Proceedings of the 8th International Conference "Linguistic Resources And Tools For Processing Of The Romanian Language"*, 19-28.

Mărănduc, C., Perez, C.-A. (2015). A Romanian Dependency Treebank. In *International Journal of Computational Linguistics and Applications*, vol. 6, no. 2, 25–40.

Simionescu R. (2012). Graphical grammar studio as a constraint grammar solution for part of speech tagging. In *Proceedings of the 8th International*

Conference "Linguistic Re-sources And Tools For Processing Of The Romanian Language", 109-118.

Simionescu R. (2012). Romanian deep noun phrase chunking using graphical grammar studio. In *Proceedings of the 8th International Conference 'Linguistic Resources And Tools For Processing Of The Romanian Language"*, 135-143.

Witten, I. H., Frank, E., Hall, (2011) Data Mining. Practical Machine Learning Tools and Techniques – 3rd ed., ISBN 978-0-12-374856-0.

Zhang, T. (2004) Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the twenty-first international conference on Machine Learning*, ICML '04, ACM, New York, NY, USA.

SYNTAXNET FOR ROMANIAN: RESULTS AND POTENTIAL

PAULA GRADU, RADU ION

Research Institute for Artificial Intelligence "Mihai Drăgănescu" Bucharest, Romania

paula.gradu@gmail.com, radu@racai.ro

Abstract

This paper presents the applicability of SyntaxNet, an open-source neural network framework created by Google Inc. for sequence labelling, to the task of Romanian dependency parsing. Its parsing performance is compared to the performances of other three state-of-the-art dependency parsers, showing that global optimization is superior to local optimization. Training and validation data were based on the RoRefTrees, an UD compliant treebank developed within the SSPR project.

Key words — Romanian language, parsing, part-of-speech tagging, parser

1. Introduction

Syntactic analysis is a key problem of Natural Language Processing and Natural Language Understanding that has seen major improvements due to recent advances in (deep) neural networks learning. Over the past decade, deep neural networks with different architectures have been used for part-of-speech tagging and sentence parsing, achieving notable results. Among the best-known state-of-the-art dependency parsers we find the Malt Parser (Nivre, 2007) and the Stanford Parser (Chen and Manning, 2014), but recently, SyntaxNet (Andor *et al*., 2016) made its debut into the world of dependency parsing. Its authors claim that it is the best dependency parser for English among the published state-of-the-art performances.

Being engineered for the English language, these parsing models perform significantly different on other languages, including Romanian. Therefore, it is important that we carefully analyse the results obtained from the various available frameworks and their particularities when applied to Romanian corpora, taking into consideration both grammatical and structural differences between languages and the size and the annotation quality of corpora. This paper studies SyntaxNet adaptability to Romanian, the results obtained when using the model (compared to the results produced by other three dependency parsers) and the potential for an increase in current accuracy of Romanian sentence parsing and part-of-speech tagging.

2. Training a SyntaxNet Parsing Model

2.1. POS Tagging

To compute the POS tags, SyntaxNet processes the sentences left-to-right. For every word in the input, the tagger extracts the features from a window around it, stacks them in an input vector and passes this vector to a feed-forward neural network classifier that computes a probability distribution over the possible POS tags. Building on the fact that the tagger makes the decisions in a left-to-right order, previously-made decisions are used as features for subsequent classifications. The trained model of the POS tagger is itself used for feature computation in dependency parsing.

2.2. Parsing

SyntaxNet is a globally optimized, incremental, transition-based neural network parser that uses both local optimization and global optimization with beam search to achieve state-of-the-art results for English. It constructs a parse incrementally, processing the words left-to-right. The unprocessed input is placed into a *buffer* from which words are shifted, one by one, into a *stack*. At each step, the parser either pushes another word onto the top of the stack or links the top two words from the stack (see Nivre (2007) for additional details on transition-based dependency parsing).

Parsing with SyntaxNet involves two steps: local pre-training and global training with a beam search. For local pre-training, SyntaxNet trains a softmax layer to predict the correct action given the dependency annotations from the train set. Because the parser's decisions are all independent, this step is performed very efficiently. For global training, the softmax scores are summed in log space, and are not normalized until the parser reaches a final decision. When the parser stops, the scores of each hypothesis are normalized against a small set of possible parses, depending on the beam size. During training, if the gold path at a given time falls off the beam, the parser is penalized (see equation 6 from Andor *et al*., (2016)). This last step of the pipeline is three or four times more time-consuming than the previous one. SyntaxNet's use of a simple feed-forward neural network with beam search optimization, built over the Google's multi-threaded and highly-efficient TensorFlow[1] platform, allows it to perform very efficiently, training in a matter of hours on a train set with approximately 175K tokens.

2.3 Parameters

SyntaxNet allows for in-depth parameter tuning and selection. It is important that the parameters are set accordingly for each step of the pipeline (*i.e.* part-of-speech tagging, local pre-training and global training) and suit the data used. Therefore,

[1]https://www.tensorflow.org/

special attention should be given to the number of decay steps and learning rate (see Figure 1), which should both be much smaller for the global training step than for POS tagging and local pre-training. It is recommended that training is run with multiple seeds for best (and stable) results.

```
  --batch_size=8 \
  --decay_steps=100 \
  --graph_builder=structured \
  --hidden_layer_sizes=200,200 \
  --learning_rate=0.02 \
  --momentum=0.9 \
  --output_path=models \
  --seed=0 \
  --training_corpus=projectivized-training-corpus \
  --tuning_corpus=tagged-tuning-corpus \
  --params=200x200-0.02-100-0.9-0 \
  --pretrained_params_names=\
embedding_matrix_0,embedding_matrix_1,embedding_matrix_2
,\
bias_0,weights_0,bias_1,weights_1
```

Figure 8. Example parameters for SyntaxNet training

2.4. Observations

SyntaxNet is built on top of TensorFlow, an open-source software library developed by Google Research for Machine Intelligence. Due to its internal functionality, SyntaxNet is currently only usable on platforms that run either Linux or OSX but with Docker virtualization[2], SyntaxNet can successfully be run on Windows 10 as well.

3. Data

The treebank used to train the SyntaxNet was developed in the SSPR project (Barbu Mititelu *et al.,* 2016), based on two existing treebanks created in the dependency grammar formalism: UAIC-RoDepTb (Perez, 2014) and RACAI-RoTb (Irimia and Barbu Mititelu, 2015). The two treebanks were combined into a single resource, RoRefTrees that complies with the principles of Universal Dependencies[3], an initiative dedicated to a unified cross-linguistic annotation standardisation for syntactic parsing. To the existing relations used by the UD project, new subtypes for universal relations that capture linguistic phenomena in Romanian have been added to increase accuracy and performance.

[2]https://www.docker.com
[3]http://universaldependencies.github.io/docs/introduction.html

A substantial part of the treebank (6347 sentences) and a thorough description of the Romanian syntax within the UD framework are accessible online, freely available for download since May 2016, through the last public release of the UD initiative, version 1.3[4]. Since then, the development of RoRefTrees continued such that the treebank now contains 9521 sentences, the targeted size in the SSPR project. The treebank has been continuously improved along the following lines:

- POS tagging has been corrected such that the labels are all compliant with TTL's (Ion, 2007) inventory and all lemmas are in the TTL's lexicon;
- Most of the syntactic annotations inconsistencies reported by the „content validation" UD check tool[5] have been fixed;

As a result, at the time of the writing, there are several thousands differences between the version 1.3 of RoRefTrees and the current, development version. The current version will form the basis of the next UD release (1.4).

3.1. Format and Sources

The data used for training, validation and evaluation was provided in the CoNLL-X format (see Figure 2). The dataset contains the following tab-separated attributes: ID, word form or punctuation symbol, lemma of the word form, coarse-grained part-of-speech tag, full, language-specific part-of-speech tag (which we call an MSD[6]), list of morphological features, head of the current token, universal dependency relation to the head and a list of secondary dependencies (head-dependency pairs). Columns that are not set have a default value of "_".

The experiments used both the updated version of the RoRefTrees (designated as 'v1.4' below) as well as the extended, released version, 'v1.3' (i.e. the full, 9521 sentences treebank, *before* any POS tagging/syntactic corrections).

3	)	)	RPAR	RPAR	_	2	punct	_	_
4	Medicul	medic	NSRY	Ncmsry	_	6	nsubj	_	_
5	veterinar	veterinar	ASN	Afpms-n	_	4	amod	_	_
6	administrează	administra	V3	Vmip3	_	0	root	_	_
7	personal	personal	R	Rgp	_	6	advmod	_	_
8	tratamentul	tratament	NSRY	Ncmsry	_	6	dobj	_	_
9	animalelor	animal	NPOY	Ncfpoy	_	6	iobj	_	_
10	de	de	S	Spsa	_	11	case	_	_
11	fermă	fermă	NSRN	Ncfsrn	_	9	nmod	_	_
12	care	care	RELR	Pw3--r		15	nsubjpass		
13	au	avea	VA3P	Va--3p	_	15	aux	_	_
14	fost	fi	VA	Vap--sm	_	15	auxpass	_	_
15	identificate	identifica	VPPF	Vmp--pf	_	9	acl	_	_
16	în	în	S	Spsa	_	15	advmod	_	_

[4]https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-1699
[5] http://universaldependencies.org/svalidation.html
[6] http://nl.ijs.si/ME/V4/msd/html/index.html

```
17  mod     mod     NSN     Ncms-n  _   16  mwe     _   _
18  clar    clar    R       Rgp     _   16  mwe     _   _
```

Figure 2. The CoNLL-X format of the treebank (excerpt from a sentence)

4. Results

4.1. Methodology

In order to achieve greater reliability, the sentences were shuffled and split into ten equal folds, which were alternatively used as training (80% of the data), tuning/development (10%) and testing (10%) data. By averaging the results obtained independently for every step of the 10-fold cross-validation, the final accuracy is more indicative of how SyntaxNet generalizes to Romanian corpora. We also compare results on the versions 1.3 and 1.4 (the current development one) of the treebank to showcase the increase in parsing accuracy when POS tagging and dependency annotation inconsistencies are removed.

4.2. Part-of-Speech Tagging

Table 1. Part-of-speech tagging accuracy on the test set

Fold No.	**Accuracy (v1.3)**	**Accuracy (v1.4)**
01	92.37%	92.89%
02	93.05%	93.57%
03	93.03%	93.96%
04	93.55%	93.13%
05	93.30%	92.81%
06	92.34%	94.33%
07	93.11%	94.27%
08	92.91%	93.94%
09	93.58%	93.63%
10	93.14%	94.21%
Average	**93.04%**	**93.67% (+0.63%)**

We were able to increase the accuracy of the learned POS tagging model by 0.63% by running the SyntaxNet's POS tagger on the more accurately POS tagged treebank. The current version of the treebank (v1.4) is not fully correct at the POS-tagging level, as only the lemmatization and the tagging of the unknown words has been manually checked. We expect to see a much larger value (close to 98%) when the treebank has been checked at word level.

4.3. Parsing

Parsing accuracy can be measured in Unlabeled Attachment Scores or UAS (i.e. only the tree structure is checked for matching against the gold standard) and Labeled Attachment Score or LAS (i.e. both the tree structure and the edge labels

are checked against the gold standard). Depending on the application, UAS or LAS can be optimized in parser training.

Table 2 presents the evolution of the UAS score of the SyntaxNet's Local and Global models on the two versions of the treebank as well as the comparison against the baseline Malt Parser. The Malt Parser has been used in the treebank development and it is a fair baseline since it has been optimized to work well on Romanian using the arc-eager algorithm and a linear regression model for choosing the best parsing decision. The features for the latter model have been adapted to Romanian by laborious hand crafting and they use lemmas and coarse-grained tags instead of default word forms and extended POS tags.

Table 2. Parsing UAS (including punctuation) for the Local Model and the Global Model

Fold No.	**Local (v1.3)**	**Global (v1.3)**	**Local (v1.4)**	**Global (v1.4)**	**Malt (v1.4)**
01	84.76%	85.14%	84.93%	85.23%	84.4%
02	84.58%	85.12%	85.05%	85.32%	84.3%
03	83.63%	84.46%	84.54%	84.63%	83.9%
04	84.33%	85.30%	84.24%	84.74%	83.3%
05	84.42%	85.00%	84.58%	85.29%	83.5%
06	84.82%	84.62%	84.96%	85.84%	84.9%
07	84.33%	84.71%	84.25%	84.76%	84.1%
08	83.87%	84.14%	84.18%	85.40%	83.9%
09	85.12%	85.50%	85.61%	85.89%	84.6%
10	84.85%	85.11%	85.58%	85.77%	85.3%
Average	**84.47%**	**84.91%**	**84.79% (+0.32%)**	**85.28% (+0.37%)**	**84.22% (-1.06%)**

Version 1.4 of the treebank brings an improvement on UAS of 0.37% for the Global model. The Global model is better than the Malt baseline by over 1% (1.06%) convincingly showing that global (i.e. sentence-level) optimization of the dependency tree is better than local (i.e. word-level) parsing decision.

Table 3. Parsing LAS (including punctuation) of the SyntaxNet's Global Model vs. other three open-source parsers

Fold No.	**Global (v1.4)**	**Malt (v1.4)**	**Turbo (v1.4)**	**RBG (v1.4)**
01	77.9%	77.8%	78.5%	80.1%
02	77.5%	77.4%	77.9%	78.7%
03	76.7%	77.4%	77.3%	78%
04	77.5%	77%	77.1%	78.3%
05	77.8%	76.9%	77.6%	78.2%
06	78.1%	78.2%	78.1%	78.7%
07	77.3%	77.6%	78%	78.4%
08	78%	77.1%	77.9%	78%
09	78.1%	77.8%	78.1%	79%

10	78%	78.7%	78.6%	79.5%
Average	**77.69%**	**77.59% (-0.1%)**	**77.91% (+0.22%)**	**78.69% (+1%)**

With LAS scores we see a slightly different picture: SyntaxNet with the default parameters cannot surpass the other, very good, open-source dependency parsers which have been trained with the default parameters as well. Turbo Parser[7] (Martins *et al*., 2013) has been trained with the standard model which took 12 hours to finish the training/testing of our 10-fold cross-validation set (a full model is also available) and the RBG Parser[8] (Lei *et al*., 2014) has been trained with the basic model (standard/full models are also available but a standard training takes one day for a single train set). Both of them have tens of parameters that can be tuned.

5. Conclusions

We trained SyntaxNet with the default parameters (the ones that have been used for English) but, as Table 3 shows, this is not enough to obtain state-of-the-art results (this is also true with respect to the other open-source parsers, with the exception of Malt Parser which, in our opinion, has reached its top performance on this treebank). Here are some ways in which we can definitely improve the output of SyntaxNet: (a) use coarse-grained tags instead of MSDs and lemmas instead of word forms as features (this strategy helped a lot with Malt Parser); (b) change the number of neurons in the hidden layer; and (c) modify the C++ code to allow for the insertion of arbitrary features which both Turbo and RBG parsers accept by default.

For all parsers, we need to develop grid-searching parameter optimization methods, which run in parallel environments (many CPUs, large amounts of RAM) with which we can obtain top results on RoRefTrees.

Acknowledgements

This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS – UEFISCDI, project number PN-II-RU-TE-2014-4-1362.

References

Andor, D., Alberti, C., Weiss, D., Severyn, A., Presta, A., Ganchev, K., Petrov, S., and Collins, M. (2016). Globally normalized transition-based neural networks. 1603.06042.

Barbu Mititelu, V., Ion, R., Simionescu, R., Irimia, E., Perez, C.-A. (2016). The Romanian Treebank Annotated According to Universal Dependencies. In *Proceedings of the Tenth International Conference on Natural Language Processing (HrTAL 2016)*, Croatia.

[7] https://github.com/andre-martins/TurboParser
[8] https://github.com/taolei87/RBGParser

Chen, D., and Manning, C. D. (2014). A Fast and Accurate Dependency Parser Using Neural Networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, October 25–29, Doha, Qatar, 740–750.

Ion, R. (2007). Word Sense Disambiguation Methods Applied to English and Romanian. Ph.D. dissertation, Romanian Academy.

Irimia, E., and Barbu Mititelu, V. (2015). Building a Romanian Dependency Treebank. In *Proceedings of the Corpus Linguistics 2015 Conference*, July, 21-24, Lancaster, UK.

Lei, T., Xin, Y., Zhang, Y., Barzilay, R., and Jaakkola, T. (2014). Low-Rank Tensors for Scoring Dependency Structures. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, June 23-25, Baltimore, Maryland, USA.

Martins, A. F. T., Almeida, M. B., and Smith, N. A. (2013). Turning on the Turbo: Fast Third-Order Non-Projective Turbo Parsers. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, August 4-9, Sofia, Bulgaria.

Nivre, J. (2007). Algorithms for Deterministic Incremental Dependency Parsing. *Computational Linguistics*, 34:4, 513-553.

Perez, C.-A. (2014). Resurse lingvistice pentru prelucrarea limbajului natural, Ph.D. dissertation, A.I. Cuza University of Iași.

TWO RESOURCES DEVELOPED IN THE PROJECT SEMANTICS-DRIVEN SYNTACTIC PARSER FOR ROMANIAN

ELENA IRIMIA, VERGINICA BARBU MITITELU

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

{elena,vergi}@racai.ro

Abstract

The paper describes in detail two essential resources created for the purpose of developing a performant hybrid (statistical and rule-based) parser for Romanian: 1. a treebank, designed by merging two existent dependency treebanks and by mapping their existent annotation to the one adopted in this project; it contains 9522 sentences and will be used as a benchmark for testing the parser and a valuable resource to promote within the community; given the value of such a linguistic resource, it is publicly released and maintained for research purposes; 2) a collection of valency frames for all the verbs in the treebank (and others), which contain, for each sense of a verb, the possible arguments and the morpho-syntactic (case), lexical (selected preposition, conjunction) and semantic restrictions on their lexicalization (formulated using top concepts from the Romanian wordnet).

Key words — treebank, valency frames, Romanian.

1. Introduction

SSPR (Semantics-driven Syntactic Parser for Romanian) is a project that aims to develop a syntactic parser for Romanian; but it also targets a higher objective: finding solutions for errors typical to statistical parsing by employing additional – mainly semantic – information. At the moment, researchers manifest sustained interest in dependency parsing, a framework very convenient for those that need syntactic information as a means to solve more difficult NLP tasks: it allows for gain in efficiency by avoiding complex annotation.

Data-driven (statistical) techniques that aim at modelling the syntactic and lexical properties of words as reflected in large corpora are extensively used in the field, including in parsing. Various attempts of using semantic information to improve the statistical syntactic parse are reported as successful by Carrol *et al.* (1998) – who used subcategorisation frames to re-rank analyses after parsing –, Zeman (2002) – who showed that subcategorisation-aware lexicalisation improves the performance of a dependency parser –, Aguirre *et al.* (2008, 2011) – who proved that semantic classes help to obtain significant improvement in both parsing and prepositional phrase attachment tasks –, Ciaramita and Attardi (2010) – who used features extracted by a named entity tagger –, etc. However, the improvements observed by these experiments are moderate and the different strategies have not yet been

assembled together to prove their combined strength. We consider that there is place for testing and even improvement here, by using different types of semantic information (word classes, word vectors, subcategorisation frames, etc.) as features of a statistical parser.

The biggest challenge for parsing is represented by the ambiguous constructions, mostly prepositional phrases (PP). For example, in the sentences "The student reads the paper with interest for the topic" and "The student reads the paper with information about the results", the attachment of the PP following the noun phrase (NP) direct object ("the paper") must be done to the verb in the first sentence, respectively to the NP direct object in the second. Different parsers try to solve the PP-attachment problem by using lexically-conditioned features (lexicalisation). E.g., the parser can learn from the training data that the verb "to read" occurs with the manner adjunct "interest" in prepositional phrases headed by "with", which will reduce the probability of attaching the PP to "the paper". However, if the parser is confronted with a similar situation in another sentence – e.g. "I read the book with pleasure –, it will be unable to reproduce the attachment preference in the previous example, since the lexico-syntactic model has no means to associate "interest" and "pleasure" as semantically alike. The solution to this limitation is the extension from lexical constraints to semantic class constraints (each content word receives a feature in the form of its semantic class in a predetermined set of semantic classes). In the previous examples, if both "interest" and "pleasure" have the semantic class STATE as associated feature, the parser can relate them and infer the same attachment preference.

Subcategorization (or valency[1]) frames for verbs, especially if they are augmented with semantic class restrictions for their arguments and, even more, lexical constraints on the prepositions for prepositional arguments, are a powerful resource to use in boosting a statistical parser's performance: e.g. they are able to solve the syntactic ambiguity between the subject and the object of a verb, difficult to deal with in languages with (relatively) free word order and morphologic homonymy like Romanian. In the sentence "Citește studentul articolul cu mare interes." (*Reads student-the paper-the with big interest. "The student reads the article with great interest."*), the morpho-syntactic descriptions associated to "studentul" and "articolul" are the same (given the nominative-accusative homonymy in Romanian). In this situation, the valency frame for the verb "a citi" ("to read") presented in Ex. 1, together with a mechanism that identifies, at runtime, the semantic classes associated to "studentul" – *+person:1*[2] – and "articolul" – *+written_communication:1* –, can solve the ambiguity and identify "studentul" as subject (the *nom* (nominative) feature in the frame) and "articolul" as direct object (the *ac* (accusative) feature in the frame).

[1] The valency of a verb is a structural description containing the number and type of complements it requires (arguments), as opposed to non-obligatory complements (adjuncts). The semantic restrictions on the arguments are supplementary; they do not usually belong in the valency frame.

[2] The number after ":" is the sense number of the word in RoWordnet.

Example 1:

a citi
GN[nom, +persoană:1] GN[ac, +comunicare_scrisă:1]
to read
NP[nom, +person:1] NP[acc, +written_communication:1]

In our endeavour to develop an efficient hybrid (statistical and rule-based) parser for Romanian, we will rely on: i. a syntactically annotated corpus (treebank) that will be used to train the statistical parser; ii. the Romanian Wordnet (Tufiș *et al.*, 2004) as a model for semantic representation; iii. a collection of valency frames for around 2600 verbs, enriched with semantic class restrictions on the arguments and lexical constraints on the prepositions for prepositional arguments. While it is widely known in the field what the Romanian Wordnet is, what information can be extracted from it and how it can be freely accessed by the members of the research community, this is not the case for the other two resources, recently developed in SSPR project. The following sections of the paper will present (i) and (iii) in detail, focusing on the linguistic information they provide and on their accessibility for research purposes.

2. A reference Romanian treebank

The treebank (RoRefTrees) we will use to train our parser was developed in the SSPR project based on two existing treebanks, UAIC-RoDepTb (Perez, 2014) and RACAI-RoTb (Irimia and Barbu, 2015), both annotated according to the dependency grammar formalism. The two resources were subject to a laborious process of harmonisation, since: 1. they were tokenized, lemmatised and part-of-speech (POS) tagged with different tools (UAIC POS tagger (Simionescu, 2011) and, respectively, TTL (Ion, 2007)); 2. slightly different tagsets were employed in the POS-tagging task; 3) they use slightly different syntactic annotation principles and sets of syntactic relations. For a detailed description of the common annotation principles and main annotation differences see (Mititelu *et al*., 2016). The POS-tagging incongruences were solved by re-tagging the whole UAIC-RoDepTb with TTL, whose tagset is MSD-based (MULTEXT-East morpho-syntactic descriptors (Erjavec, 2012)).

We compiled the two treebanks into a single resource (RoRefTrees) that comply with the principles in Universal Dependencies[3] (UD), an initiative dedicated to a unified cross-linguistic annotation standardisation for syntactic parsing. For this purpose, we made a thorough description of the Romanian syntax within the UD framework (Mititelu and Irimia, 2015). Since the set of relations (both the universal and the language-specific relations introduced by other languages in the project) used by UD did not capture all linguistic phenomena in Romanian, we extended it by proposing new subtypes for some universal relations (which is the standard procedure in UD).

[3] universaldependencies.github.io/docs/introduction.html

The actual conversion from UAIC-RoDepTb and RACAI-RoTb to RoRefTrees was done in a bootstrapping manner, starting from a manually annotated UD (small) treebank of 500 sentences, extracted from the sub-corpora to be mapped. The MaltParser (with the arc-eager parsing algorithm (Nivre *et al*, 2007)) was trained on this first set of sentences and used to parse the whole remaining corpus. From this parse and the original annotation of the respective sub-corpora, a mapping table was derived. The mapping function was applied on a new set of 600 sentences that were manually corrected and added to MaltParser training set. The process was resumed until we automatically mapped and manually validated the whole corpus. The sub-corpora were mapped subsequently, first RACAI-RoTb and then UAIC-RoDepTb, and we noticed that the mapper improves significantly with the size of the UD training corpus. For a detailed description of this process, see (Mititelu *et al*., 2016).

At this point, RoRefTrees contains 9522 sentences that are unevenly distributed over the following domains:

Table 1. The distribution of sentences per corpus domains

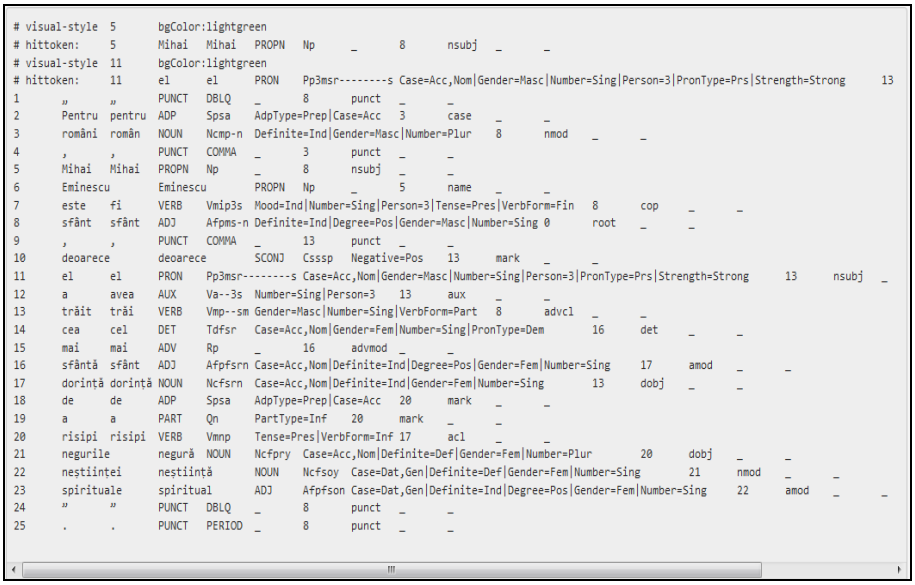
Domain	Sentences	Tokens	Average length
Prose	1819	37308	20 tokens/sentence
Juridic	1606	48295	30 tokens/sentence
Medical	1210	27654	23 tokens/sentence
Academic	950	19991	21 tokens/sentence
Encyclopedic	611	14046	23 tokens/sentence
Journalistic	933	23356	25 tokens/sentence
Miscellanea	2393	47715	20 tokens/sentence
TOTAL	**9522**	**218365**	**23 tokens/sentence**

A substantial part of RoRefTrees (6347 sentences, since May 2016) is already accessible online, freely available for download, through the last public release of the UD initiative, at https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-1699. The UD platform recommends two manners of querying the treebank online: using SETS querying system[4] and using PML Tree Query[5]. We consider the first option the most easy to use, with a very intuitive interface and user-friendly query language (see Figures 1, 2 and 3).

[4] For quering: http://bionlp-www.utu.fi/dep_search; SETS tutorial: http://bionlp.utu.fi/searchexpressions-new.html.

[5] For quering: http://lindat.mff.cuni.cz/services/pmltq/#!/home; PML Tree Query tutorial: https://ufal.mff.cuni.cz/pmltq/doc/pmltq_doc.html;

Figure 1. SETS treebank search for all words serving as nominal active subjects (nsubj) in the Romanian treebank. The dependencies are expressed using "<" (x < y, x is governed by y) and ">" (x > y, x governs y) and "_" signifies "any token". Below the query form, the user can opt to see the context of each returned result and also the annotated sentence in conllu format. This capture shows the context for the first query hits ("Mihai" and "el" in the bolded sentence).



```
# visual-style  5       bgColor:lightgreen
# hittoken:     5       Mihai   Mihai   PROPN   Np      _       8       nsubj   _       _
# visual-style  11      bgColor:lightgreen
# hittoken:     11      el      el      PRON    Pp3msr--------s Case=Acc,Nom|Gender=Masc|Number=Sing|Person=3|PronType=Prs|Strength=Strong      13
1       „       „       PUNCT   DBLQ    _       8       punct   _       _
2       Pentru  pentru  ADP     Spsa    AdpType=Prep|Case=Acc   3       case    _       _
3       români  român    NOUN    Ncmp-n  Definite=Ind|Gender=Masc|Number=Plur    8       nmod    _       _
4       ,       ,       PUNCT   COMMA   _       3       punct   _       _
5       Mihai   Mihai   PROPN   Np      _       8       nsubj   _       _
6       Eminescu        Eminescu        PROPN   Np      _       5       name    _       _
7       este    fi      VERB    Vmip3s  Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin   8       cop     _       _
8       sfânt   sfânt   ADJ     Afpms-n Definite=Ind|Degree=Pos|Gender=Masc|Number=Sing 0       root    _       _
9       ,       ,       PUNCT   COMMA   _       13      punct   _       _
10      deoarece        deoarece        SCONJ   Csssp   Negative=Pos    13      mark    _       _
11      el      el      PRON    Pp3msr--------s Case=Acc,Nom|Gender=Masc|Number=Sing|Person=3|PronType=Prs|Strength=Strong      13      nsubj   _
12      a       avea    AUX     Va--3s  Number=Sing|Person=3    13      aux     _       _
13      trăit   trăi    VERB    Vmp--sm Gender=Masc|Number=Sing|VerbForm=Part   8       advcl   _       _
14      cea     cel     DET     Tdfsr   Case=Acc,Nom|Gender=Fem|Number=Sing|PronType=Dem        16      det     _       _
15      mai     mai     ADV     Rp      _       16      advmod  _       _
16      sfântă  sfânt   ADJ     Afpfsrn Case=Acc,Nom|Definite=Ind|Degree=Pos|Gender=Fem|Number=Sing     17      amod    _       _
17      dorință dorință NOUN    Ncfsrn  Case=Acc,Nom|Definite=Ind|Gender=Fem|Number=Sing        13      dobj    _       _
18      de      de      ADP     Spsa    AdpType=Prep|Case=Acc   20      mark    _       _
19      a       a       PART    Qn      PartType=Inf    20      mark    _       _
20      risipi  risipi  VERB    Vmnp    Tense=Pres|VerbForm=Inf 17      acl     _       _
21      negurile        negură  NOUN    Ncfpry  Case=Acc,Nom|Definite=Def|Gender=Fem|Number=Plur       20      dobj    _       _
22      neștiinței      neștiință       NOUN    Ncfsoy  Case=Dat,Gen|Definite=Def|Gender=Fem|Number=Sing       21      nmod    _       _
23      spirituale      spiritual       ADJ     Afpfson Case=Dat,Gen|Definite=Ind|Degree=Pos|Gender=Fem|Number=Sing    22      amod    _       _
24      ”       ”       PUNCT   DBLQ    _       8       punct   _       _
25      .       .       PUNCT   PERIOD  _       8       punct   _       _
```

Figure 2. Conll format for the first hit of the query in Figure 1. First four line in the image mark the hit nsubj in position 5 and 11 in the sentence.

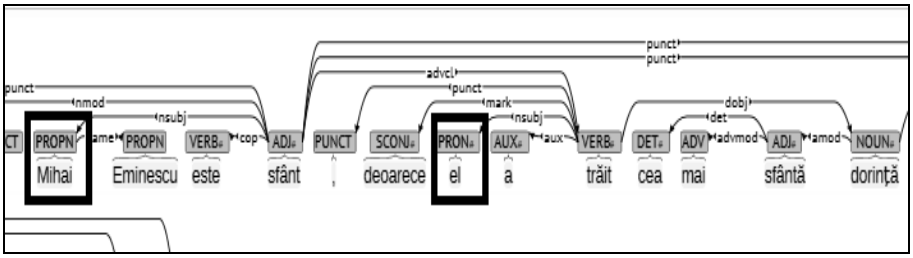


Figure 3. The treelike representation of the first query hit for the query in Figure 1.

The image was truncated due to limited space in the page. The original format is coloured, with the green colour highlighting the returned nodes (framed in this figure by bolded squares) and the blue colour marking the rest of the nodes in the sentence.

Another option for querying the treebank is at http://clarino.uib.no/iness/page?page-id=iness-main-page, with the INESS infrastructure (Rosén, 2012). The interface is very similar to the one described above, but the treelike representation of the hits is more user friendly. Both search interfaces offer powerful query languages, able to search for wordforms, lemmas, parts of speech, dependency relations, direction of the dependency relations, negation, combination of queries, universal quantification, etc.

(Mititelu and Irimia, 2016) elaborates on the types of syntactic relations (dependencies) that can be found in RoRefTrees, discussing the linguistic relevance of their relative frequencies in the treebank. It reports on what can be found and what cannot be found in the treebank with respect to the linguists' needs and on the resource qualities and deficiencies resulting from the annotation formalism and framework we opted for.

3. A collection of valency frames for Romanian verbs

Historically, valency frames collections (or lexicons) have been built within different theoretical frameworks and have either been created based on linguistic intuition supported by naturally occurring or artificially designed examples or have been automatically extracted from annotated corpora and later manually validated by linguists. We combined the two approaches and constructed the valency lexicon in two steps:

Step 1: **Creating valency frames for some verbs inside Romanian Wordnet.**
The Romanian Wordnet (RoWN) contains 529 verb synsets (1031 literals) manually enriched with valency frames that look as in Table 2 below, illustrating a specific sense of the literals "a conserva" and "a păstra" ("to preserve"). While machine-readable, the xml format in which the wordnet is encoded is not human-friendly, and therefore not convenient for further human validation and for creating new valency frames for other verbs; moreover, the manner of representing the case as numbers and the order of their association (1-nominative, 2-dative, 3-genitive, 4-accusative,

following a Czech model) are not natural for the Romanian grammar; finally, some information in the xml format is redundant, since "cineva" is a reference for a "person", while other parts are not refined enough, since "ceva" could be anything, including FOOD.

Table 2. A Romanian Wordnet synset containing a valency frame

```
<SYNSET>
    <ID>ENG20-00205023-v</ID>
    <POS>v</POS>
    <SYNONYM>
        <LITERAL>conserva<SENSE>1</SENSE></LITE
      RAL>
        <LITERAL>păstra<SENSE>2</SENSE></LITERA
      L>
    </SYNONYM>
    <DEF>A conserva alimentele în bună stare, ferindu-le de
   alterare</DEF>
    <VALENCY>
        <FRAME>cineva1*AG(person:1)=ceva4*FOOD(f
      ood:1.1)
        </FRAME>
    </VALENCY>
    <BCS>2</BCS>
    <ILR>ENG20-016
    16879-v<TYPE>hypernym</TYPE></ILR>
    <ILR>ENG20-
   02654727v<TYPE>verb_group</TYPE></ILR>
    <DOMAIN>gastronomy</DOMAIN>
    <SUMO>Cooking<TYPE>+</TYPE></SUMO>
</SYNSET>
```

Hence we opted for a human-readable format that we obtained by automatically converting the wordnet valency frames, living aside the uninteresting data. For the synset in Table 2, which contains two literals, two different entries in the valency lexicon (with the same valency frame) were created (see Ex. 2 below).

Example 2: valency frames for the verbs „a conserva" and „a păstra"
a conserva
1. GN [nom, +persoană: 1] GN [ac, +hrană: 1.1]
a păstra
2. GN [nom, +persoană: 1] GN [ac, +hrană: 1.1]

There are two situations in which a synset in the RoWN can contain more (up to 6) than one frame:

- a specific literal has more than one valency frame for a specific sense:

Example 3: a deranja
1. GN [ac, +person: 1/animal: 1] GV [că]
2. GN [nom, +anything: 1] GN [ac, +person: 1/animal: 1]

- a specific synset has different frames for different literals in the synset; in Ex. 4, the literals "accentua, evidenția, marca, puncta, releva, reliefa, sublinia" have 2 common frames, while their synonim "pune_accent" has a different frame, requiring as obligatory argument a prepositional phrase introduced by the preposition "pe":

Example 4: a accentua, evidenția, marca, puncta, releva, reliefa, sublinia
1. GN [nom, +persoană: 1] GN [ac, +idee: 3]
2. GN [nom, +persoană: 1] GV [că]
pune_accent_1.2.x
3. GN [nom, +persoană: 1] GP [pe, +idee: 3]

Step 2: **Automatically extracting all the verbs from RoRefTrees together with their obligatory arguments and manually correcting and enhancing them with semantic restrictions for the arguments.**

For the verbs that already have frames, created in Step 1, the human validator checked if the verb sense occurring in the treebank is already described in the valency lexicon; if not, a new frame for the specific sense was created[6]. For the verbs that do not have any frames created in the lexicon, the linguists created them based on the frame skeleton extracted from the treebank (see Ex. 5 below). The syntactic structure of the frame skeleton was automatically generated by interpreting the syntactic dependencies in the treebank and the lexical information concerning prepositions and conjunctions.

We target a list of valency frames for all the verbs in the treebank. The valency frames are still in process of manual validation and they will be soon uploaded on the SSPR project site and free for using in research purposes. The valency frames that are defined in the RoWN are also subject to modification whenever necessary.

Ex. 5 below shows that 4 different frame skeletons were automatically extracted from the treebank for the verb "a sublinia". The first three frames were already present in our valency lexicons (extracted from RoWN), with a single difference in the second frame: there is a mention of the subordinating conjunction "să" occuring in the treebank with "sublinia" that has to be checked as it may be an error in the treebank annotation. The 4th frame skeleton is incorrect since the prepositional group introduced by "pe" is actually a place adverbial (like in "Elevul a subliniat pe caiet un singur cuvânt". / "The student underlined a single word in the notebook."), and therefore should not appear in the frame. It can be seen in Ex. 5 that the semantic

[6] In the valency lexicon we constructed, no sense distinction is explicitly made: RoWN is not complete and for verbs missing from RoWN it would be impossible to assign a sense. Therefore, an entry for a specific verb in the valency lexicon can contain: 1. different frames for different senses; 2. the same frame for different senses; 3. different frames for the same sense.

restrictions are missing: there is no information after the symbol "+" in the frame skeleton; the restrictions were inserted by the human validator using context analysis and introspection.

Example 5:
a sublinia
1. GN [nom,] GN [ac, +]/GP [pe, +]
*2. GN [nom,] GV [**să**/că]*
3. GN [nom,] GN [dat, +] GN [ac, +]/GP [pe, +]
*4. **GN [nom,] GP [pe, +] GN [ac, +]***

4. Conclusions

We presented two significant linguistic resources that were developed in the SSPR project and that will be integrally available to the research community. We explained their use in developing an efficient syntactic parser, but we also want to stress the importance of these resources for other applications in the field of Natural Language Processing and Computational Linguistics but also –especially in the treebank's case– for scientists in Theoretical Linguistics.

Acknowledgements

This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS–UEFISCDI, project number PN-II-RU-TE-2014-4-1362.

References

Agirre, E., Baldwin, T., Martinez, D. (2008). Improving parsing and PP attachment performance with sense information. In *Proceedings of the 46th Meeting of the Association for Computational Linguistics: Human Language Technologies*, ACL-08: HLT, Columbus, Ohio, 317–325.

Agirre, E., Bengoetxa, K., Gojenola, K. and Nivre, J. (2011). Improving dependency parsing with semantic classes. In *Proceedings of the 49th Meeting of the Association for Computational Linguistics: Human Language Technologies*, ACL-11: HLT, Portland, USA, 699–703.

Barbu Mititelu, V. and Irimia, E. (2015). Description of the Romanian Syntax within Universal Dependency Project. In *Proceedings of the 11th International Conference "Linguistic Resources and Tools for Processing the Romanian Language"*, CONSILR November 2015, Iași, Romania, 185-194.

Barbu Mititelu, V. and Irimia, E. (2016). Linguistic Data Retrievable from a Treebank, In *Proceedings of Conference on Computational Linguistics in Bulgaria*, CLIB-2016, Sofia, Bulgaria.

Barbu Mititelu, V., Ion, R., Simionescu, R., Irimia, E. and Perez, C.-A. (2016). The Romanian Treebank Annotated According to Universal Dependencies, Presented at The Tenth International Conference on Natural Language Processing (HrTAL2016), Sept. 29 – Oct. 1, Croatia, in press.

Carroll, J., Minnen, G.and Briscoe, T. (1998). Can subcategorisation probabilities help a statistical parser?. *arXiv preprint cmp-lg/9806013*.

Ciaramita, M. and Attardi G. (2010). Dependency parsing with second-order feature maps and annotated semantic information. In *Trends in Parsing Technology*. Springer Netherlands, 2010, 87-104.

Erjavec, T. (2012). MULTEXT-East: Morphosyntactic Resources for Central and Eastern European Languages. *Language Resources and Evaluation*, 46:1, 131-142.

Ion, R. (2007). Word Sense Disambiguation Methods Applied to English and Romanian. PhD thesis, Romanian Academy, Bucharest (in Romanian).

Irimia, E., Barbu Mititelu, V. (2015). Building a Romanian Dependency Treebank. In Proceedings of Corpus Linguistics conference, *CL 2015*, Lancaster University, UK, 171-174.

Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S. and Marsi, E. (2007). Malt-parser: A language independent system for data-driven dependency parsing. Natural Language Engineering, 13, 95–135.

Perez, C. A. (2014). Linguistic Resources for Natural Language Processing. PhD thesis, A.I. Cuza University of Iasi (in Romanian).

Rosén, V., De Smedt, K., Meurer, P. Dyvik, H. (2012). An open infrastructure for advanced treebanking. In Proceedings of the W*orkshop "Advanced Treebanking", LREC2012*, 22–29.

Simionescu, R. (2011). Graphical Grammar Studio as a Constraint Grammar Solution for Part of Speech Tagging. In *Proceedings of the 8th International Conference "Linguistic Resources and Tools for Processing the Romanian Language"*, Publishing House of "Al. I. Cuza" University, Iasi, 109-118.

Tufiş, D., Barbu, E., Barbu Mititelu, V., Ion, R. and Bozianu, L. (2004). The Romanian Wordnet. *Romanian Journal on Information Science and Technology*. 7:2-3, 105-122.

Zeman, D. (2002). Can subcategorization help a statistical dependency parser?. In *Proceedings of the 19th international conference on Computational linguistics*, 1, Association for Computational Linguistics, 1-7.

A RESOURCE FOR THE WRITTEN ROMANIAN: THE UAIC DEPENDENCY TREEBANK

CĂTĂLINA MĂRĂNDUC[1,2], CENEL-AUGUSTO PEREZ[1]

[1]*Faculty of Computer Science, "Alexandru loan Cuza" University of Iasi, Romania*
[2]*Academic Institute of Linguistics "Iorgu Iordan - Al. Rosetti" Bucharest, Romania*

{catalina.maranduc, augusto.perez}@info.uaic.ro

Abstract

The Romanian Treebank was created with automatic manually checked annotation, using the following tools: two annotation interfaces called Treeannotator and Treebank Annotator, a hybrid POS-tagger, and a statistical parser, variant of the Malt parser. The resource, having now 12,675 sentences and 233,169 tokens, is an opportunity to develop other researches. The creation of a multilingual aligned Treebank was initiated. The documentation for a semantic layer based on accurate logical criteria is being finalized and the transformation syntactic-semantic is made in 46%. Part of the treebank is in course of affiliating at the UD (Universal Dependencies) project. The authors are interested not only in the contemporary standardized Romanian, but also in the old, regional, and Social Media non-standardized Romanian. The automatic annotation of a New Testament (1648) began, checking them to build a gold corpus for the training of tools on old Romanian. The "New Testament" corpus will be aligned with similar books in old languages.

Key words — aligned corpora, annotation interfaces, conventions of annotation, dependency grammar, gold corpus, non-standardized language, old Romanian, regional variations, semantic layer, treebank**.**

1. Introduction

The first version of the paper was written with the occasion of the 100th anniversary of the founding of the Romanian Academy, with the intention of providing researchers with a result of our work. In this new version of the paper we reiterate the theme, adding new data about developments of the corpus in the period from March to October and the prospects opened.

A treebank is a corpus of sentences in natural language, with rich morphological annotation and with syntactic structures in form of trees, i.e. marking not only the syntactic relation, but also the head of each token.

The treebank described below was created during a project started in 2007, when

600 sentences were manually annotated in tree form by the students of Computational Linguistics Master. Later, the project was taken over by the Academic Institute of Computer Science Iasi, in partnership with the Faculty of Computer Science, Alexandru Ioan Cuza University of Iasi. During the research in his PhD thesis, Augusto Perez (Perez, 2014) had extended it to 4,600 sentences. The corpus was also presented in (Mărănduc and Perez, 2015).

This Treebank, called UAIC-RoDepTb (Dependency Grammar Treebank for Romanian created at Al. I. Cuza University of Iasi), is balanced, containing complex sentences forming sub-corpora from more language registers: the legal style, (the Acquis communitaire), the journal's style (the Frame-Net), the universal fiction style (Orwell's novel *1984),* popularizing science (Wikipedia) and a big number of quotations extracted from the Romanian Thesaurus Dictionary (Romanian Academy, 1913-2010) or from its bibliographical sources.

The sources of the big dictionary are carefully selected from all the styles of the language, and are written in the XIX and XX centuries or even earlier. These lexicographic sources are introduced in the treebank for their stylistic variety, and also with the intention to illustrate the Romanian specific verbal constructions.

To illustrate also the contemporary Romanian, three sub-corpora from chat communication were added, with 2,576 sentences and 39,291 tokens. This new style of the language is similar with the oral style, free and innovative. The non-standardized character of this style was a challenge for the text processing tools trained on Romanian standardized language (Perez *et al.* 2015a).

For the training of these tools, certain homogeneity of sub-corpora is required, and also a quantity of at least 1,000 sentences or 12,000 tokens in each stylistic variety. Some quotations were grouped into two separate documents, containing the first, regional folk lyrical texts, and the second, archaic and religious texts. These sub-corpora will be developed in future research projects.

2. Related Works

An important number of treebanks for all the natural languages were created in the XXI century, due to their more complex and well-structured information. The treebanks are generally created in the HPSG (Head-Driven Phrase Structure Grammar)[1] system of annotation (Pollard and Sag, 1994), which is more rigorous, or in the DG (Dependency Grammar) of FDG (Functional Dependency Grammar) conventions (Mel'Cuk, 1987).

FDG is chosen by the UAIC project due to its flexibility and economy of the system. The HPSG tree, based of inclusion relations, has a bigger number of nodes, whilst a FDG tree is similar of a Finite State Automaton. There are also automatic programs to transform a treebank annotated in HPSG conventions into a treebank annotated in DG conventions. For example, the Bulgarian Treebank[2] is created in HPSG

[1]http://hpsg.stanford.edu/ideas.html
[2] http://bultreebank.org/

convention and then transformed in a Dependency Treebank. (Simov and Ostenova, 2011).

UAIC-RoDepTb is not the first dependency treebank from Romanian: it is necessary to mention the example and the experience of previous woks (Hristea, and Popescu, 2003) and (Călăcean and Nivre, 2008) In these treebanks, the average of words per sentence is half as in the UAIC-RoDepTb, and the sentences, only in the journalistic style, are simplified. The performance of the statistical Malt parser is better in these conditions.

The languages that have created Dependency Treebanks were affiliated to the UD[3] (Universal Dependencies) project (Rosa *et al.*, 2014). A set of universal conventions of annotation is respected by all the participants, with the intention to facilitate comparisons between languages, aligned corpora for the machine learning projects, and also to build a new syntactic parser, language independent. The documents in UD are in CONLLU format, a variant of CONLLX (Hall, and Nilson, 2007).

The UD conventions are simplified, to keep information available for all the languages. The transposition of UAIC conventions onto UD ones is made with loss of information. There is a delicate balance between the need to be compatible with universal projects and the need to render the specifics of the Romanian language.

An interesting model to follow is the PDT[4] (the Prague Dependency Treebank), which being affiliated to the UD, has kept the additional information in a semantic annotation layer (Bohmova *et al.*, 2005, Hajic *et al.*, 2000). The semantic annotation, the next level of complexity, after the syntactic one, is increasingly the attention of computer scientists and linguists worldwide.

3. Tools used for the Processing of UAIC-RoDepTb texts

3.1. The hybrid UAIC RoBinPOS-tagger

After the syntactic annotation, the texts introduced in the treebank are pre-processed in a pipeline of tools: a splitter, a tokenizer and a POS-tagger, i.e. the text is segmented in sentences, then in words, and finally each word is morphologically analysed and related with a lemma. At the beginning of the activity, the texts were processed with the POS-tagger developed by the researchers of RACAI (Research Institute for Artificial Intelligence of Romanian Academy)[5], and since 2011, the texts are processed with the UAIC POS-tagger[6] (Simionescu, 2011a).

This hybrid POS-tagger is trained on NAACL corpus and evaluated on the Orwell's *1984* corpus, created in the MULTEXT-East[7], project (Erjavec, 2004). We have in intention to make a gold corpus for the training using our own conventions, without

[3]http://universaldependencies.org/
[4]http://ufal.mff.cuni.cz/pdt2.0/
[5]http://www.racai.ro/en/tools/text/
[6]http://nlptools.infoiasi.ro/
[7]http://nl.ijs.si/ME

inconsistencies. The POS-tagger for Contemporary Romanian has 446 morphological labels, and a lexicon of 1,500,000 forms for 230,000 lemmas, some proper nouns included. A clone of the POS-tagger, for Old Romanian processing, will have about 500 morphological labels and approximately 30,000 lemmas will be added in the first step. The POS-tagger is hybrid, i.e. it offers the possibility to introduce rules to eliminate frequent mistakes.

The rules for the annotation of verbs at composed modes and tenses, at various forms of future, and for the annotation of the auxiliaries of passive verbs were introduced or will be introduced.

3.2. Graphical Grammar Studio

This tool is developed with the intention to make searches in annotated corpora. It can search complex structures, with the form of little ”grammars”, i.e. symbols existents in the annotated corpus in which the search is made, related by simple or complex rules, e.g. nouns with imbricate determinations (Simionescu, 2011b).

It can also introduce annotations, and can be helpful for the formulation of rules in REGEX, to be introduced in the hybrid POS-tagger. The third version of the tool can make searches in a corpus hierarchically structured as trees. It extracted from the Treebank a list of verbs with their dependencies, to be used in the construction of the Dictionary of Romanian Verbal Patterns (Perez *et al.*, 2015b).

3.3 The Malt Parser Trained on Romanian Sentences

Finally, the text with morphological information is processed with the Malt parser (Hall, *et al*., 2006) trained on Romanian using the bootstrapping method. At the beginning, the corpus being too small for a statistical tool, a voting scheme with three parsers was used. The manually checked sentences were always added to the gold corpus, by the bootstrapping method, and the UAIC statistic parser was developed simultaneously with the treebank. The treebank is now a big gold corpus that can be used in order to train the UAIC tools for Romanian text processing.

Compared to other training corpora used until now, the present treebank is carefully supervised. The correctness of the annotation for the weak forms of personal and reflexive pronouns was carefully supervised and the participles were annotated as verbs, and not as adjectives, because they accept the same Romanian language-specific dependencies as the others verbal forms.

There must be no accepted concessions from the linguistic correctness to simplify the processing, because more complicated examples will be found in natural language and the entire system will be deregulated. It is better to start from smaller percentages of accuracy that will increase with the size of the corpus.

3.4. The Treeannotator

For creating a treebank, the annotating interface is very important. It checks the correctness of XML format and of the graph, according to the axioms of dependency grammar. It displays the automatic annotation and enables quick manual corrections of errors found. The Treeannotator was the first annotation interface used to create our treebank (Moruz, 2008). It has many qualities, importing a large XML document containing a sub-corpus with 1000 sentences, indicating the errors in XML format and their place. It sorts the features in XML, with the id and the word form at the beginning of the string, so that it is easier to read. It has two possibilities for the display (see Fig. 1 for the linear view). The tree view is shown in Fig. 3.

However, several researchers having worked at this tool, the codes are not available now. This tool is not flexible, does not allow changes of format, it is impossible to add or remove tokens in the sentence. It allows correction of POS-tag labels displayed at the bottom, but lemma is inaccessible, and it remains incorrect.

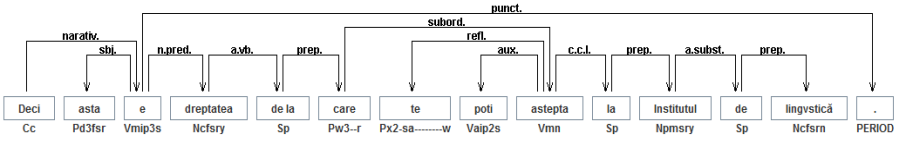


Figure 1. The standard linear view in Treeannotator interface.

For creating aligned treebanks, an interface that can simultaneously display more trees is needed. UAIC-RoDepTb includes also a corpus of 250 sentences taken out from the Orwell's novel *1984*, which were aligned with the corresponding phrases in the original English version and the translation of these phrases in French. Thereby, the 1984-EnRoFr sub-corpus was created (Mărănduc *et al.*, 2015).

3.5. The MaltEvalViz interface

In the article cited below, the trees in Romanian, then in French, were compared on the English ones, considered the original form, using MaltEvalViz[8]. However, this interface was created by the authors of the Malt Parser, with the intention to compare the version automatically annotated of a tree with the gold version checked by experts.

[8] http://www. maltparser.org/malteval.html.

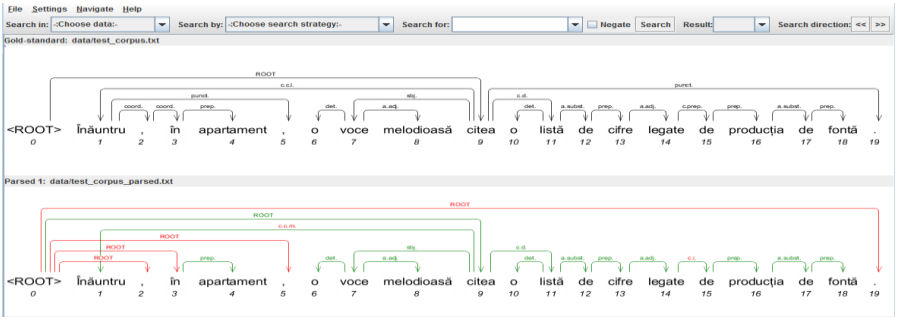


Figure 2. A tree automatically annotated and another checked by experts, compared by MalEvalViz.

The tool is useful for the intended purpose, and we frequently evaluate the automatically annotated trees with it. For example, in the figure 3 is easy to remark that the mistakes of the parser are in the top of the tree, because it works left to right and bottom to the top and this procedure is not suitable in the case of a language with free order of word, like Romanian.

3.6. The Treebank Annotator

This is our newest tool (Hociung, 2016), which is not yet available on the NLP tools page because it is still in work. It is a multifunctional interface, which can load now 5 formats of documents, and other formats can be introduced, each of them with their list of features, existing in the folder Configurations: ConllxConfig, PROIELConfig, UAIC-RoDepTbSemanticConfig, UAIC-RoDepTbSyntacticConfig and UDConfig. Each of them has dropdown lists of labels for most of Attributes.

The tool permits to add or to remove tokens or to change the word order. All the attributes in the XML or CONLLX format are shown in the interface and can be checked. It permits also to open more documents simultaneously, even if they use different formats, and to display more trees. The format is chosen for each document after to open it. The author wants also add functions to save the documents in a chosen format, i.e. to transform the documents from a format to another. For the moment, each document is saved in the same format in which it was open, the program works still slowly, consuming a lot of working memory and the view or trees is not standard.

So, a big number of work tools are needed to create a corpus type treebank. Our intention is to create different POS-taggers for the Old Romanian processing, written with Latin or Cyrillic characters, and the Romanian dialects spoken in the south of the Danube, and also different parsers for each format of the treebank.

4. Types of Information Annotated in the UAIC-RoDepTb

The text in XML format is separated in words, not also in letters. For each word, the XML contains an id, which shows the place of the word in the sentence. ”Form” is

the occurrence of the word in the text. The morphological analysis begins with ”lemma” and continues with the ”postag”, a sequence of letters that codifies the morphological categories. The first capital letter contains the information regarding the part of speech, and the following ones describe flexion properties for each category. For example:

form="craii" lemma="crai" postag="Ncmpry"

form="fac" lemma="face" postag="Vmip3p"

”Crai” and ’’face” are the dictionary entry for these words; ”Ncmpry” means: Nc = common Noun, m = masculine, p = plural, r = recte case (nominative, accusative), y = determined (having the definite article -i). ”Vmip3p” means: Vm = Main verb, i = indicative, p = present, 3p = third person of plural.

Syntactic information is depicted in the last tokens of the XML word tags: "deprel" and "head". Each word, except the root, has a single ascendant, determiner, or head, this tag contains a number, the id of the word determined by the word analyzed. The ”deprel” is the codification of the dependency relation that exist between the determiner and the determinant.

The entire rows for these words look like this:

<word id="4" form="craii" lemma="crai" postag="Ncmpry" head="3" chunk="" deprel="sbj."/>

<word id="5" form="fac" lemma="face" postag="Vmip3p" head="0" chunk=""/>

The second word has not a ”head” and a ”deprel”, because it is the root of the

Table 1. Syntactic Relations

Nr. crt.	**Syntactic relation**	**English name of relation**
1	a.adj.	adjectival attribute
2	a.adv.	adverbial attribute
3	ap.	apposition
4	a.pron.	attribute expressed by pronoun
5	a.subst.	attribute expressed by noun
6	aux.	auxiliary verb
7	a.vb.	verbal attribute

Table 2. Statistics of the syntactic relations

Label	Accuracy	Occurence
c.c.cz.	75	28
neg.	92.31	278
c.c.cons.	69.23	13
comp.	72.63	95
c.c.t.	78.31	189
prep.	81.13	231
c.c.soc.	11.11	9
c.c.instr.	64.29	14

The lists of the morphological labels and the dependency relations are placed on the NLP resources web page, because UAIC-RoDepTb is an open source. There are 44 syntactic relations. The parser returns after each training a statistic of their frequency.

In the table 1 there is a fragment of the list of ”deprel” explained in English, and in the table 2 there is a fragment of a such statistic of frequencies of the occurrences of the relations in the training corpus; there is a concordance between a big number of occurrences and the accuracy of their automatic detection.

The conventions of UAIC-RoDepTb respect the axioms of the Dependency grammars:

1. The primitive element of the syntactic description is the core (the root).

2. The connection is a binary functional relationship between a higher term (regent) and a lower term (dependent).

3. Each core is a node in the tree and has exactly one syntactic regent [14].

The nodes of the graph obtained are words and punctuation elements. The dependency relations are labels from the arcs of the graph. A challenge of this model is to find a convention for figuring the horizontal relations, as the coordination, quite frequent in the natural language.

There are more models of the coordination in FDG and also there are automatic programs to transpose the annotation from a convention onto another. In UAIC-RoDepTb, the coordination is an asymmetrical descendent relation. The connector is placed between the two related words, i.e. it is dependent of the first and regent for the second.

There are more situations in which the dependency relations are not in fact

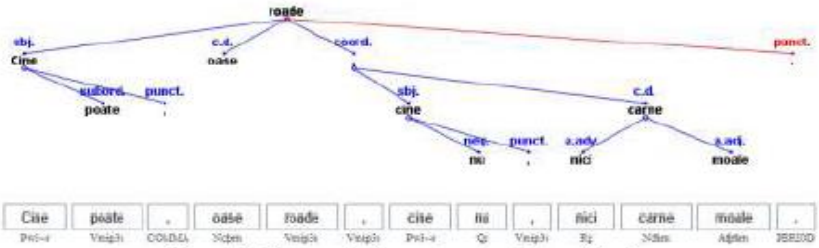


Figure 1. The graph of a sentence with two ellipsis.

dependences, but rather succession relations, between an antecedent and a consequent in time order: relations between the independent clauses in the sentence, between an incident clause and the rest of the syntactic structure, between the vocative noun and the core of the sentence, etc. In Figure 3, a tree is displayed in standard tree view.

The sentence meaning is a logical expression formed by the truth value of component clauses. The modality of figuring the horizontal logical relations appears in DG as oblique descendent lines and it is sufficiently explicit. The linguists that have created the treebank are also interested by the annotation of the ellipses (words not present in the text, which can be understood in the context). In figure 3 there is the graph of a sentence with two ellipses. *(Who can, gnaw bones, who [can] not [, = gnaw] nor soft flesh*.)

In UAIC-RoDepTb the ellipse is treated as a translation of the information by an existent element, to a coordinated or symmetrical item (punctuation, functional word). The second comma receives by coordination the information of the verb ”roade”*(gnaw)*, and the other ellipse is recoverable by symmetry, the sense is translated to the negation: *who can ... who [can] not.*

5. Conclusions and Future Work

The example of PDT and BulTb shows that a large corpus with rich information is usable and reusable for numerous other projects. In the last year, more projects were started. The Aligned EnRoFr treebank open the perspective to compare the different languages, and more rules for the Machine Translation were written.

Another project is RoPAAS (Romanian Predicate Arguments and Adjuncts Structure), a dictionary of the specific patterns for Romanian verbs, linked with examples from the treebank. The extraction of the verbs with their direct dependencies from the treebank provided a list of 1785 verbs and the treebank is about to be increased with quotes for a complete description of the Romanian verbs.

The documentation from the future semantic level of UAIC-RoDepTb is finished and applied to a 230-sentence sub-corpus. It replaces the syntactic dependency relations with semantic dependency-relations; forming parallel trees (see Fig. 4).

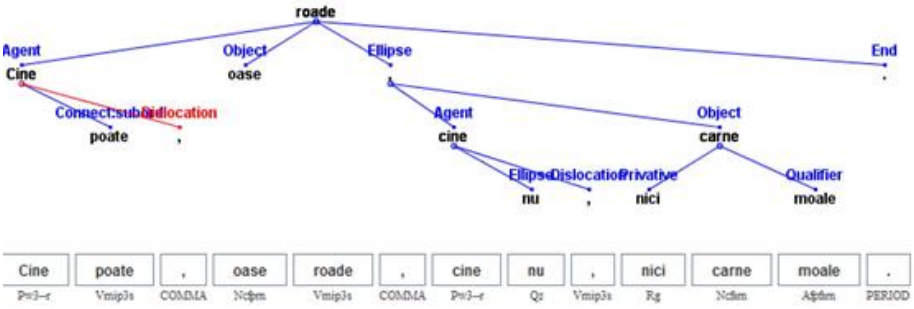


Figure 4. The tree of the figure 3, with semantic annotations.

The semantic layer was created and 44% of the syntactic relations are changed with semantic labels, including the four types of coordination and the subordination relations, the 14 types of circumstantial and other relations with unique correspondent like the agent complement, the vocative and so on. A program is in course to be experienced, built by Cătălin Mititelu, called Treeops, for the transposition of other relations, using more conditions, i.e. the syntactic label and the morphological classification (that include semantic values: past tense, definiteness, indefiniteness, future tense, and so on).

This is an example of a rule which change all the undefined articles, in the semantic label ”Undefined” whatever genre and number is morphologically analyzed in the postag label:

//word[@deprel='det.' and (@postag='Tifso' or @postag='Tifsr' or @postag='Timso' or @postag='Timsr' or @postag='Ti-po')]/@deprel => changeAttrValue('Undefined')

The annotation in UAIC conventions will be used without modifications for the training of the statistical parser. This annotation is also considered the pivot for semi-automated transformations in semantic, UD or PROIEL conventions.

Another project is the creation and the training of UAIC NLP tools for processing the old Romanian Language. We chose to annotate the New Testament (1648); the first 1600 sentences are checked and 630 mapped also with the Cyrillic characters obtained by the researchers from Chisinau. This book is the first printed New Testament in Romanian, edited by the bishop Simion Ştefan, in Alba Iulia. It will be introduced in the PROIEL project[9], which aligns the New Testaments in the old languages (Haug and Jøhndal, 2008). A large lexicon of archaic words and old flexion will be created adding the automatic extraction of the manually checked annotations from the New Testament.

The old corpus will be balanced, including texts from XVI, XVII, XVIII, and XIXth centuries, in legal style, fiction, religion, history, and cookery books.

[9] http://proiel.github.io/

These perspectives demonstrate that the UAIC-RoDepTb is a valuable resource with multiple applications and its value increases with the size and with the complexity of the layers of annotation.

References

Bohmova, A., Hajic, J., Hajicova, E., Hladka, B. (2005). The Prague Dependency Treebank: A Three-Level Annotation Scenario, Prague.

Călăcean, M., Nivre, J. (2008). Data-driven Dependency Parsing for Romanian, *Acta Universitatis Upsaliensis*, 65-76.

Erjavec, T. (2004) MULTEXT-East Version 3: Multilingual Morphosyntactic Specifications, Lexicons and Corpora. In *Proceedings of the Fourth Intl. Conf. on Language Resources and Evaluation, LREC*, ELRA.

Hajic, J., Bohmova, A., Hajicova, E., and Hladka, B. V. (2000). The Prague Dependency Treebank: A Three-Level Annotation Scenario, in A. Abeille, *Treebanks: Building and Using Parsed Corpora*, Amsterdam, Kluwer.

Hall, J., Nivre, J. Nilsson, J. (2006) Discriminative Classifiers for Deterministic Dependency Parsing. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, (COLING-ACL), Main Conference Poster Sessions.

Hall, J., Nilsson, J. (2007). CoNLL-X Shared Task: Multi-lingual Dependency Parsing, MSI, Växjö University, School of Mathematics and Systems Engineering.

Haug, D.T.T., Jøhndal, M.L. (2008) Creating a Parallel Treebank of the Old Indo-European Bible Translations, In *Proceedings of the Second Workshop "Language Technology for Cultural Heritage Data"*, 27-34.

Hociung, F. (2016). Treebank Annotator, dissertation, Faculty of Computer Science, Alexandru Ioan Cuza University, Iași.

Hristea, F., Popescu, M. (2003). A Dependency Grammar Approach to Syntactic Analysis with Special Reference to Romanian in Building Awareness. In *Language Technology*, 9-34, University of Bucharest Press, Bucharest.

Mărănduc, C., Perez, A.-C. (2015). A Romanian Dependency Treebank. In the *International Journal of Computational Linguistics and Applications*, Vol. 6 No. 2 July-December, 25-40.

Mărănduc, C., Perez, A.-C., Balmuș, R. (2015). Aligned Dependency Treebank English-Romanian-French. In *Proceedings of ConsILR*, Iași, Alexandru Ioan Cuza University Publishing, 39-52.

Mel'Cuk, I. A. (1987). *Dependency Syntax: Theory and Practice*, Buffalo, Suny Press.

Moruz, Al. (2008). Developing a FDG (Functional Dependency Grammar) annotator for Romanian, dissertation, Faculty of Computer Science, Alexandru Ioan Cuza University, Iaşi.

Perez, A.-C. (2014). Linguistic Resources for Natural Language Processing. PhD thesis, Alexandru Ioan Cuza University of Iasi.

Perez, A.-C., Mărănduc, C., Simionescu, R. (2015). Including Social Media – a Very Dynamic Style, in the Corpora for Processing Romanian Language. In *Proceedings at EUROLAN 2015*, CCIS 588, 1–15, Springer International Publishing, Switzerland, 1–15.

Perez, A.-C., Mărănduc, C., Simionescu, R. (2015b) Ro-PAAS – A resource linked to our UAIC-Ro-Dep-Treebank. In *Proceedings of Advances in Artificial Intelligence and Soft Computing 14th Mexican International Conference*, Sidorov, G., Haro, G., Sofía N. (Eds.) Springer Publishing Switzerland.

Pollard, C., and Sag, I. (1994). *Head-Driven Phrase,*University of Chicago Press.

Romanian Academy (1913-2010). *Thesaurus-Dictionary of Romanian Language* (DTLR) compound by (1913-1949) *Dictionary of Romanian Language,* (DA) Bucharest, Socec, Universul Publishing, and (1965-2010) *Dictionary of Romanian Language,* (DLR) New series, Bucharest, Romanian Academy Publishing.

Rosa, R., Masek, J., Marecek, D., Popel, M., Zeman, D., Zabokrtsky, Z. (2014). HamleDT 2.0: Thirty Dependency Treebanks Stanfordized. In *Proceedings of LREC.*

Simionescu, R. (2011). Hybrid POS Tagger. In *Proceedings of the Workshop "Language Resources and Tools in Industrial Applications"*, Eurolan.

Simionescu, R. (2011). Graphical grammar studio as a constraint grammar solution for part of speech tagging. In *Proceedings of The Conference on Linguistic Resources and instruments for Romanian Language Processing*.

Simov, K., Osenova, P. (2011). Towards Minimal Recursion Semantics over Bulgarian Dependency Parsing. In *Proceedings of the RANLP Conference*. 12th-14th September, Hissar, Bulgaria.

CHAPTER 3
SPEECH DATA PROCESSING AND STUDIES

THE TRANSCRIPTION OF ROMANIAN CORPORA BETWEEN WHAT IS SPOKEN AND THE GRAMMATICALY CORRECT WRITING

VASILE APOPEI, OTILIA PĂDURARU

Institute of Computer Science of the Romanian Academy, Iasi Branch

vasile.apopei@iit.academiaromana-is.ro; otilia@iit.tuiasi.ro

Abstract

This paper aims to highlight the importance for the linguistic research, of the existence for Romanian spoken language corpora, of the literary transcription with grammatically correct form and the transcription to what is spoken. In what is spoken, we encounter partly pronounced words, repetitions of words, hesitation sounds, and contraction through the apheresis, syncope and elision phonetic phenomena. This approach are based on recent papers from the literature in the field of transcriptions of spoken language corpora for English, German and French, with application in sociolinguistic analysis and in the speech technology for extraction of prosodic model and phonetic model for speech synthesis and speech recognition.

Key words — speech-to-text transcription, literary transcription, phonetic model, prosodic model, phonetic phenomena

1. Introduction

The evolution of the spoken language corpora is based on impressionist transcripts of the recordings of dialogue or spoken language using literary transcripts (Schmidt, 2012) or functional discursive-pragmatic transcripts (Hoarță Cărăușu, 2013). The literary transcription of the spoken corpora, called by some and transcription with standard orthography (Baude *et al*., 2010), provides a standard spelling of the speech and corpora become accessible and for non-experts. Another facility offered by literary transcription is that, it enables processing of linguistic text using the automatic natural language processing tools.

Beyond the advantages presented for literary transcripts, the potential of the spoken corpora transcribed with standard orthography remains largely untapped because these transcription "often normalizes the form of the words in the text to standard spellings, meaning that orthographically transcribed material is rarely a reliable source of evidence for research into variation in pronunciation" (McEnery and Hardie, 2012).

In the last years, more and more researchers from linguistics and speech technology work for developing of spoken language corpora with time-aligned multi-level annotations at orthographic transcription, word and phoneme (Bigi 2012, 2015; Blache *et al*., 2010, TEI, 2015, Schalley *et al*., 2014). These corpora are successfully used to access relevant data for analysing sociolinguistic and phonetic phenomena, and in speech technology for extraction of phonetic model, prosodic model and language model for speech synthesis and speech recognition applications.

In this paper, we propose a framework for alignment at the word and phoneme levels of the speech corpora with literary transcripts. For automatic alignment, we add to the literary transcriptions, supplementary information about hesitation sounds, overlapping talk, unclear speech segments and words of whose pronouncing are affected by phonetic contraction with phonetic phenomena as elision, syncope or apheresis (L. Pistol 2015).

2. Phonetic phenomena and non-orthographic words

A preliminary analysis of the causes that determines the desynchronizations in automatic time-aligned at words level of the large audio files has revealed the following aspects that appear in spoken language:

- significant differences of the vocal timbre between participants at dialogues;

- segments with speech overlaps between speakers or unclear that occur in multi-party conversation;

- hesitation sounds and non-standard pronunciations of words that occur during speech.

Considering these causes that hinder the automatic alignment of large files, we intend to extract from the large audio file, segments of speech without overlap and uncertainties. This speech segment we automatically aligned at words level using the SailAlign program, following as subsequently, to add this information to the TextGrid file manually annotated, the supplementary information about the alignment at word and phoneme levels.

Next, we present the most common phonetic phenomena that we have encountered in pronunciations of some words and particular pronunciations for some orthographic words and abbreviations.

Elision is a phonetic process where one or more final phonemes are omitted in pronunciation, usually in order to simplify the pronunciation. It may occur for vowels, consonants and syllables, although it is much more common for consonants:

- elision of vowels: *pe orizontală/ p-orizontală, mă ajută/m-ajută*;

- elision of consonants and syllables: *pot să/po-să, poa**te** să/ poa-să,* treize**ci** și/treize și*;*

- elision of Romanian definite article **l**: artistu**l**/artistu, omu**l**/omu, declicu**l**/declicu;

Apheresis is a phonetic process where one or more phonemes from the begining of word are dropped in pronunciation: *clasa întreagă/ clasa-ntreagă, numai în/ numa-n*;

Syncope is the phonetic phenomenon by which a vowel (usually unstressed) inside word is not pronounced: or**i**ce/orce, or**i**cît/orcît, or**i**cine/orcine, or**i**care/orcare, *vr**e**o/*vro, vr**e**odată/ vrodată, vr**e**un/vrun.

Contraheren (in Romamian *contragere*) is the tendency to speak together two adjacent phonemes, identical or different, by reduction to one phoneme or to a diphthong: *plecân**d d**e/ plecân de, făcu**t t**eologie/făcu teologie, su**b b**alconul/ su-balconul, var**a a**sta/var-asta, uit**e a**șa/uite-așa;*

Assimilation is a general term in phonetics for the process by which a speech sound becomes similar or identical to a neighbouring sound. Assimilations may happen inside a word, or between two words, when the final sound of a word touches the first sound of the next word.

Examples:

- t-d consonant context: *înto**td**eauna/întodeauna, to**td**eauna/todeauna, dinto**td**eauna/ dintodeauna, momen**t d**at/momen dat, su**nt d**e/sân de*;

- d-t consonant context: *cîn**d t**rec/cân trec;*

- t-m/n consonant context: *mul**t m**ai/ mul-mai, mul**t n**oroc / mul-noroc, as**tm**/asm, is**tm**/ism*;

- t-s consonant context: *făcu**t s**ub / făcu sub, cânta**t s**ub/cânta sub, po**t s**ă/po-să, o**pt**sprezece/ opsprezece*;

- d-c consonatic context: *fiin**dc**ă/fiincă*;

- n-m consonant context: *î**nm**ormîntare/îmormîntare, î**nm**ugurit/îmugurit, î**nm**agazinată/ îmagazinată, î**nm**oaie/îmoaie, î**nm**ulțire/ îmulțire*;

Numerals pronunciation: Cincisprezece/cinșpe

Pronunciation of abbreviation: *MJC/em-j-se, CD/si-diuri*.

Differences in pronunciation, because of change the norms of writing, such as: *sunt/sînt*.

Hesitation sounds such as: *î, ă, m.*

3. Frame work -Methodology

The research presented in this paper are part from research undertaken in the Institute of Computer Science in order to develop the phonetic and prosodic subcomponents of the spoken language component of the COROLA and SRoL corpuses (Jitcă *et al*., 2015; Apopei and Păduraru, 2015, Teodorescu *et al*., 2011).

In this endeavour, we started this study from the literary transcriptions (L. Pistol 2015), made by human operators using Praat, a publically available computer program for speech analysis and synthesis (Boersma and Weenink, 2012). These transcriptions are saved in TextGrid files together with following information: type of speech event, speakers and audio contents.

Based on information from TextGrid files, we extract from the input audio file, the speech segments corresponding to an audio content (Casey, 2001) and to a single speaker, and we will save this speech segments in wav files. Simultaneously with creating the wav files, we will create the text files with the corresponding transcription of the speech segment (Figure 1).

For save the transcriptions from TextGrid files in this text files we use the following conventions:

- The omitted phonemes by elisions, syncope, apheresis and assimilation are explicit marked by parenthesis ().
- The transcriptions of particular pronunciation for the orthographic words, pronunciation of abbreviation are marked by using square brackets [].
- pronunciation of abbreviated words
- Hesitation sounds, partly pronounced words and repetitions of words are marked by using curly brackets {}.

Comedie-farmacie spun {îî} strămoșii noștri, actori în comedie. Dacă nu ai o exactitate aproape de perfecțiune nu-(î)ți atingi scopu(l). ... Uită pentru o secundă, sper eu, că eu sunt actoru(l) ... și pentru o secundă sunt personaju(l) pe care eu îl interpretez. Ș(i)-atunci el se regăsește în personajul respectiv și uită că eu sunt [sînt] ăla pe care l-a văzut la televizor sau în alte scheciuri sau în alte {îî} ipostaze și primește personaju(l) care-l face să râdă de fapt. (a)

Comedy-pharmacy says our ancestors, actors in comedy. If you do not have an accuracy nearly of perfection, do not reach your purpose. … Forgets for a second, I hope, that I am the actor and for a second, I am the personage, which I play. And-then he is re-finds in the respectively personage, and forget that I am the one he saw him on television or in other sketches or in other hypostasis and receives the personage that makes him laugh actually. (b)

Figure 1. A sample of transcriptions saved in text files (a) Romanian, (b) English

For each pair of wav-text files, we save into a text file the information about wav file name and the time indexes for the beginning of the segments into the input large audio file (Table 1).

Table 1. A sample of information about from time indexes of wav files

File name	Time indexes [ms]
00bcecb1_seg1.wav	0.000000
00bcecb1_seg2.wav	22.825937
00bcecb1_seg3.wav	35.638313
00bcecb1_seg4.wav	52.762436
00bcecb1_seg5.wav	74.309189

00bcecb1_seg78.wav	3240.246826
00bcecb1_seg79.wav	3310.788574
00bcecb1_seg80.wav	3326.326416

In the next step, the audio and text files results in the previous stage are used as input for SailAlign tool (Katsamanis, 2011) in order to automatic speech-to-text alignment at the word and phoneme levels (Figure 2).

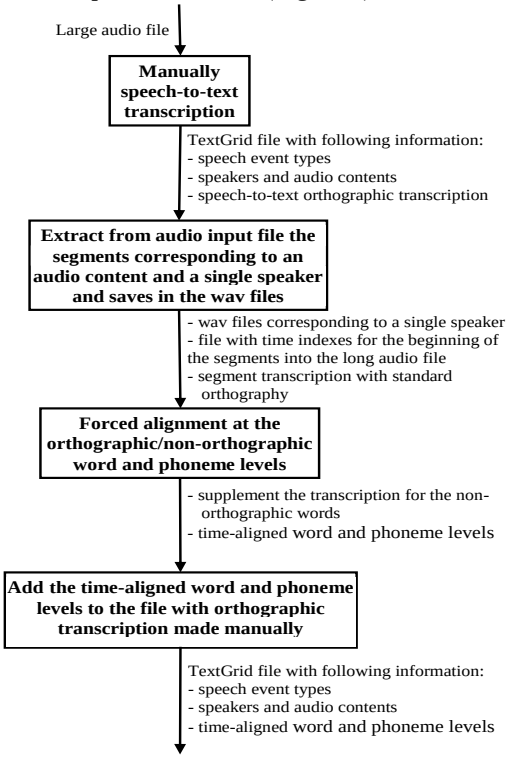


Figure 2. The diagram for develop the phonetic and prosodic subcomponents of the spoken language component of the COROLA corpus

At this stage of alignment at word level, we will correct literary transcription and the additional information about what is spoken (hesitation sounds and non-orthographic words), until the SailAlign tool makes a good alignment of the speech with the words. Even if the statement "correct the literary transcription" seems inappropriate, these is necessary because the human annotators may mistakes sometimes due to the fatigue inherent that occurs during manually transcripts.

After we have validated the automatic time-alignment of the audio files corresponding to a single speaker with word and phonetic transcription, we add these levels to the TextGrid file with the literary transcription made manually for the initial large input audio file.

4. *Conclusions*

The proposed framework for "correct the literary transcription" and alignment of the audio files at word and phonetic transcription, make possible to use the corpora for the sociolinguistic and phonetic phenomena analysis, and in speech technology for extraction of phonetic model and prosodic model.

Time-aligned of audio recordings with the phonetic pronunciation of words, opens new perspectives in the analysis of the phonetic phenomena, such as syncope, assimilation apheresis or elision, which occur in the Romanian language spoken.

Acknowledgments

The research presented in this paper has been conducted within the Institute of Computer Science of the Romanian Academy, Iasi branch. The study was started from the manually annotation, made by Laura Pistol and Otilia Paduraru, of audio files from spoken language component of the COROLA corpus.

References

Hoarţă Cărăuşu, L. (2013). *Corpus de limbă română vorbită actuală nedialectală,* Editura Universităţii „Alexandru Ioan Cuza".

Apopei V. (2014). About prosodic phrasing of the Noun Phrases. In *Proceedings of the Romanian Academy*. Volume 15, Number 2/2014, 200-207.

Jitcă D., Apopei, V., Păduraru O. Maruşca S. (2015). Transcription of Romanian intonation. In Sonia Frota & Pilar Prieto (eds), *Intonational in Romance*: Oxford University Press, 284-316.

Apopei, V., Păduraru, O. (2015). Towards Prosodic Phrasing of Spontaneous and Reading Speech for Romanian Corpora. In *Proceedings of . 8th Conf. Speech Technology and Human - Computer Dialogue (SpeD)*, Bucharest, Romania, Oct 14-16, 2015.

Apopei, V., Hoarţă Cărăuşu, L., Jitcă, D. (2015). Aspects of the speech transcription within the spoken language component of the COROLA corpus, *ConsILR* 2015.

Katsamanis, A., Black, M., Georgiou, P, Goldstein, L., Narayanan, S. (2011). *SailAlign: Robust long speech-text alignment*, Proc. of Workşop on New Tools and Methods for Very-Large Scale Phonetics Research.

Schalley, A.C.,Musgrave, S., Haugh, M. (2014). Accessing Phonetic Variation in Spoken Language Corpora through Non-standard Orthography. *Australian Journal of Linguistics*,vol 34, 139-170.

McEnery T., Hardie A. (2012). *Corpus Linguistics: Method, Theory and Practice*, Cambridge UK: Cambridge University Press.

Baude, O., Blanche-Benveniste, C. (2010). *Spoken Corpora Good Practice Guide 2006*, 2010.

Boersma, P., Weenink, D. (2012). "Praat: Doing Phonetics by Computer", version 5.3.1., <http://www.praat.org/>.

TEI (2015), *TEI P5: Guidelines for Electronic Text Encoding and Interchange*, edited by C.M. Sperberg-McQueen and Lou Burnard for the ACH-ALLC-ACL under the supervision of the Technical Council of the TEI Consortium.

Blache, P., Bertrand, R., Bigi,B., Bruno, E., Cela, E., Espesser, R., Ferré, G., Guardiola, M., Hirst, D. (2010). Multimodal annotation of conversational data. In *The Fourth Linguistic Annotation Workshop*, Uppsala (Sueden)

Bigi, B. (2015). *SPPAS tutorial: Methodology and software for the semi-automatic annotation of speech*.

Bigi, B., Péri, P., Bertrand, R. (2012). Orthographic Transcription: Which Enrichment is required for Phonetization?. *Language Resources and Evaluation Conference*, Istanbul Turkey, 1756-1763.

Casey, M. (2001). General sound classification and similarity. In *MPEG-7*, *Organised Sound*, Volume 6, Issue 02, 153-164, Cambridge University Press.

Thomas Schmidt, (2012). EXMARaLDA and the FOLK tools – two toolsets for transcribing and annotating spoken language. *Language Resources and Evaluation Conference*, Istanbul Turkey, 236–240.

Teodorescu, H.N., Zbancioc, M., Feraru, M. (2011). The analysis of the vowel triangle variation for Romanian language depending on emotional states. *Interntional Conference on Signals, Circuits and Systems - ISSCS Conference*, Iasi, Romania 30 June-1Jul.2011, 331-334.

Pistol, L. (2015). Colectarea de texte vorbite din "Dezvoltări cantitative şi calitative asupra corpusului COROLA. Includeri de texte şi înregistrări vocale, proiectarea şi implementarea funcţionalităţii platformei de achiziţionare, testarea unei platforme de interogare", Raport cercetare iunie 2015.

VOICE CONTROLLED HOME AUTOMATION SYSTEM

TIBERIU BOROȘ[1], ȘTEFAN DANIEL DUMITRESCU[1], HORIA CUCU[2]

[1] *Romanian Academy, Centre for Artificial Intelligence "Mihai Drăgănescu",(RACAI)*
{tibi, sdumitrescu}@racai.ro
[2] *Speech and Dialogue Research Laboratory, University "Politehnica" of Bucharest,*
horia.cucu@upb.ro

Abstract

Voice enabled human computer interfaces (HCI) that integrate automatic speech recognition, text-to-speech synthesis and natural language understanding have become a simple commodity, which was introduced by the immersion of smart phones and other gadgets in our daily lives. Smart assistants are able to respond to simple queries (similar to text-based question-answering systems), perform simple tasks (call a number, reject a call etc.) and help with organizing appointments. In this paper we introduce a newly created home automation platform which is designed to enable the user to control home appliances using a natural voice interface. We offer an overview over the state-of-the art technologies which enabled us to construct our system, mainly focusing on automatic speech recognition and text-to-speech synthesis and we also address the technical challenges involved in integrating them with standard home automation communication protocols.

Key words — automatic speech recognition, home automation, natural language processing, text-to-speech synthesis, voice interface

1. Introduction

This paper describes the technologies used to construct a natural voice assistive system for home automation. All the work presented in this paper was carried out during the implementation of the ANVSIB[1] national project. We cover aspects related to system architecture, challenges involved by individual tasks, performance figures of the sub-modules and technical decisions related to the development of a working prototype. The tools and technologies are divided in 3 main topics: (a) automatic speech recognition (ASR); (b) text-to-speech synthesis (TTS) and (c) integration with home automation services.

Before we proceed with the actual description of the system we will start with an overview of the context in which this system was developed. The immersion of smart devices into our daily lives has reshaped our expectations from high-end technology and re-modelled our expectations in terms of human-computer (device) interaction. The major differences refer to, but are not limited to: (a) **high level**

[1] Assistive Natural-language, Voice-controlled System for Intelligent Buildings

availability for the device (usually influenced by battery life) and for the provided services (some services require an active internet connection, which is not always sustainable), (b) **usability** (the interaction with any application should be intuitive simple, and, most importantly, the user should not be required to read any manuals in order to operate the software), (c) **multimodality** (devices provide multiple input methods such as touch, haptic, and voice – according to Google, in 2015, 30% of the queries on its search engine were issued via voice), (d) **natural interaction** (it has become common to allow the user to enter free-form queries via text or voice), (e) **response time** (lags in UI responsiveness have an negative impact on the user experience). Major mobile OS developers are currently including some form of personal assistants within their operating systems, which enable the user to control their device using natural voice queries (see Google Now in Android, Siri in Apple OSx, etc.). However, these systems offer only restricted interaction possibilities, the user being constrained to a limited personal digital (virtual) space. This actually means that the user (a) is able to control one or more smart devices, (b) has access to information from simple queries (QA systems included by major competitors in the mobile OS world are able to understand and automatically summarize answers to queries such as: "how is the weather today?", "who is the president of the United States" or "who was Michael Jackson") and (c) can organize his/her agenda according to the documents stored in his/her own cloud-hosted storage (Google is able to parse and obtain information from plane tickets and bookings and automatically provides calendar entries as well as general tips such as "your plane leaves tomorrow at 11 AM and you should be at the airport before 10 AM due to traffic").

The emergence of the Internet of Things (IoT) was primarily driven by the increasing standards imposed by consumers. It refers to a network of physical devices, vehicles, buildings, sensors and actuators which can be remotely accessed and controlled by users. Communication with these devices is regulated by standard communication protocols. Among the open standards we mention KNX (EN 50090, ISO/IEC 14543), which is a standardized network communication protocol used in our project primarily because of its simplicity and scalability.

These protocols offer control over IoT devices but most implementations are limited to legacy input methods (touch, keyboard and mouse). Based on the previously mentioned trends in HCI, we set our goals to create a fully voice controlled interface, KNX based home automation system. The primary language on which we focused our efforts is Romanian. However, our system can be extended to other languages using appropriate training data.

2. General system architecture

As previously mentioned the functional tasks of the project are divided between speech processing and home automation. The speech processing task is also separated into speech recognition (ASR) and speech synthesis (TTS). To ensure the scalability of the project we designed a system architecture in which computationally expensive tasks are handled by separate modules (see figure 1 for

details). The TTS (section 2.1) and ASR (section 2.2) modules act as standalone servers and they are queried over TCP/IP. The KNX home automation server is a node that is designed to relay messages from the Voice Controller (section 2.3) to any networked peripheral. The Voice Controller is a distributed application that processes the user queries and is able to balance the tasks between multiple ASR and TTS servers.

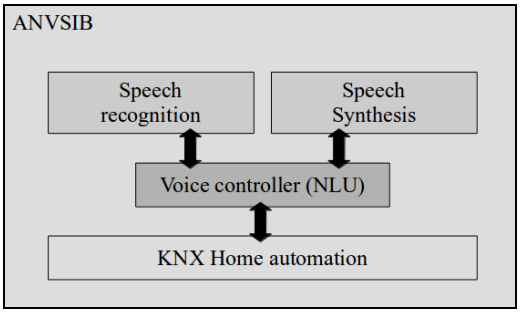


Figure 9. Architecture Overview

2.1. Text-to-speech synthesis

Text-to-speech synthesis refers to the conversion of any arbitrary (unbounded) text into audio signal. The unrestricted text requirement makes this task very difficult and, while state-of-the-art systems produce remarkable results in terms of intelligibility and naturalness, the recipe for producing synthetic voices which are undistinguishable from natural ones has not yet been found. This limitation is caused by the fact that natural language understanding still poses serious challenges for machines and the fact that the surface form of the text does not provide sufficient cues for the prosodic realization of a spontaneous and expressive voice (Taylor, 2009). However, when we refer to home automation, the utterances that will be synthesized can be anticipated because they are typical expressions that will often be used by the system. Thus, the quality of the synthetic voice is enhanced by employing either limited domain speech synthesis or performing domain adaptation on a general purpose TTS system.

TTS synthesis involves two major steps: (a) extraction of features from text and (b) conversion of symbolic representations into actual speech. **The text processing step** (step a) is usually composed of low-level text-processing tasks such as part-of-speech tagging, lemmatization, chunking, letter-to-sound conversion, syllabification etc. **The signal processing task** (step b) consists of selecting an optimal set of speech parameters (given the features provided by step a) and generating an acoustic signal that best fits these parameters.

Text processing is carried out by a natural language processing pipeline. Our natural language processing pipeline is provided by an in-house developed extensible framework – called Modular Language Processing for Lightweight Applications

(MLPLA) (Zafiu *et al*., 2015). Most of the individual modules have been thoroughly described in our previous work (Boroș, 2013; Boroș *et al*., 2013; Boroș, 2015). Though we experimented with many techniques and strategies, for the sake of clarity we will provide accuracies only for the top-performing methods. The part-of-speech tagger is a neural inspired approach (Boroș *et al*., 2013b) which achieved 98.21% accuracy on the "1984" novel by G. Orwell using morphosyntactic descriptors (MSDs) (Erjavec, 2004) in the tagset. For the other tasks our platform is able to switch between a MIRA classifier (Crammer and Singer, 2003) and a Deep Neural Network (DNN) classifier. The accuracy figures for each task are shown in table 1. One can easily see that the reason for being able to switch between the two classifiers is that, while MIRA provides higher accuracy values, the DNN classifier has a smaller model size and achieves up to par performance. The evaluation of the two methods was done using 10-fold validation and the accuracy rates are reported at word-level (not phoneme/letter).

Table 2. Individual accuracies for basic text pre-processing

Task	Mira		DNN	
	Accuracy	**Model size**	**Accuracy**	**Model size**
Syllabification	99.01%	9426.5KB	98.23%	36.7KB
Letter-to-sound	96.26%	1389.1KB	96.16%	43.4KB
Stress prediction	98.8%	6435.3KB	97.67%	110.3KB

We have to mention that for syllabification we used the onset-nucleus-coda (ONC) tagging strategy proposed in (Bartlett *et al*., 2009) and chunking (not evaluated here) is performed using a POS-based grammar described in (Ion, 2007).

The signal processing backend is also performed using an in-house built framework –Speech Synthesis for Lightweight Application (SSLA). Currently there are two major data-driven methods that handle this task: unit-selection and statistical parametric speech synthesis (Tokuda *et al*., 2000). For the ANVSIB project, the synthesis backend used by our TTS system employs the later mentioned method. The choice was driven by the fact that it provides a small footprint for the synthesis model and is able to produce stable results in terms of quality. The filter used in our analysis and voice re-synthesis process is STRAIGHT (Kawahara *et al*., 1999). SSLA also provides unit-selection speech synthesis which was described in (Boroș *et al*., 2013). Our concatenative system placed alongside the other synthesis systems in the final round of the Blizzard Challenge 2013.

There are two major contributions for the TTS system developed during the ANVSIB project. The first one is the implementation of the statistical parametric speech synthesis module and the second one is the creation of a new acoustic resource (corpus) for TTS synthesis. The corpus was recorded in studio conditions using two professional speakers and is composed of two sections:

(a) The first section (section A) is based on Wikipedia and contains a number of sentences that were chosen using a greedy algorithm in order to assure the completeness of the phonetic domain of the Romanian language. The sentences are treated as individual prompts (no larger context is provided), thus the

speaker had to record each individual sentence "out of the bloom" and limit his narrative interpretation to the utterance itself.

(b) The second section (section B) of the corpus is composed of sentences from the Romanian adaptation after Allen Carr's book "Easy way to stop smoking". The book contains a lot of motivational and persuasive passages which are carefully crafted by the author to convince smokers quit their habit. Additional to the prompts themselves, we also made use of an existing audiobook. Originally, this audiobook was recorded by a male actor and has approximately two and a half hours of high quality studio recordings at 48KHz. The original actor made use of highly prosodic rhetoric speech with the purpose of (a) reshaping the cognitive state of the listener (b) and relaying embedded messages to the smoker. Gaining access to the prosodic parameters (F0, phone duration and pauses) that make up such a speech is an asset to research in the field of natural TTS systems. The matching prompts (from the audiobook) were made available to our speakers in order to act as a baseline and a guide in their voice shaping process.

The Wikipedia section of the speech corpus is freely available for download and use in any research activity. It is composed of 4h:7m:23s (for the female speaker) and 4h:25m:46s (for the male speaker) and the archive contains the speech prompts (one file each), the corresponding audio files, the phonetic transcription lexicon and time-aligned phoneme sequences for each prompt-audio pair.

The audiobook section we collected is composed 3h:15m:07s (for the female speaker) and 3h:20m:17s (for the male speaker) but we will only provide pre-trained speech models (see section 4 for details) and statistics from this section in order to avoid any Intellectual Property Rights (IPR) violation.

2.2. Automatic Speech Recognition

The Automatic Speech Recognition (ASR) web service created by Speech and Dialogue Research Laboratory (http://speed.pub.ro) is a scalable and extensible online Speech-to-Text (S2T) solution. At the moment it supports two languages: Romanian and English, and it can transcribe speech from various domains, depending on the application requirements. For example, within the ANVSIB project, a new ASR domain was created, namely "Casandra Commands", in order to accommodate the particularities of the automatic speech recognition task required in the project. Similarly, the service can be potentially extended to any speech recognition task, in any language, provided that the speech and language resources are available.

The Speech-to-Text (S2T) system resides in the cloud or in SpeeD's IT infrastructure and can be accessed through the Internet or from SpeeD's local ***area network (as shown in Figure 2***). Client applications can be developed using any technology that is able to communicate through TCP-IP sockets with the server application. The communication between the client applications and the ASR service is based on a proprietary xml-based protocol. The server application can

communicate simultaneously with several client applications, serving them simultaneously or sequentially, in a first-in-first-out order.

The speech-to-text system can be configured to transcribe various types of speech, from various domains and various languages. It can be configured to instantiate multiple speech recognition engines (S2T Transcribers), each of these engines being responsible for transcribing speech from a specific domain (for example, TV news in Romanian, medical-related speech in Romanian, country names in English, etc.). The speech recognition engines are based on the open-source CMU Sphinx speech recognition toolkit.

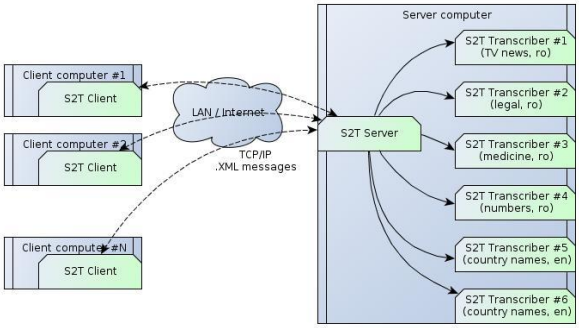


Figure 10. General overview of the ASR framework

The xml-based protocol used by the client application to communicate with the speech-to-text system involves the following steps:

- Authentication: the client authenticates itself with the server
- Configuration: the server sends information regarding the supported ASR configurations
- Data connection: the client and the server establish a data connection through which the client will send the audio data to be transcribed
- Automatic speech recognition: the client sends a transcription request along with the audio data to be transcribed and the ASR system responds with one or several transcriptions

The ASR domains supported by the current version of the speech-to-text system are Romanian news, Romanian cities, Romanian dates, Romanian forenames, Romanian surnames, Romanian numbers, English countries, English numbers, NAO commands and Casandra commands. Here are a few examples for Casandra: "casandra, porneşte sistemul de climatizare" ***(en. Casandra, turn on the air-conditioning system)***, "casandra, aprinde toate luminile" ***(en. Casandra, turn on all the lights)***, "casandra, activează stropitorile" ***(en. Casandra, start the irrigation system)***, "casandra, măreşte temperatura cu trei grade în sufragerie" ***(en. Casandra, raise the living room temperature with three degrees)***.

The ASR systems with small vocabulary and grammar language models were evaluated in depth in (Cucu H. *et al*., 2015), while the ASR system with large vocabulary and statistical language model was evaluated in depth in (Cucu H. *et al*., 2014). The first ones have word error rates between 0 and 5%, while the latter has a word error rate of about 16%.

For any new ASR domain the following resources are needed: an acoustic model, a language model (grammar or statistical) and a phonetic dictionary. Provided that the acoustic model and the phonetic dictionary for the Romanian language were available prior to the introduction of the "Casandra Commands" ASR domain, only the language model had to be created. In this case, all the commands for the smart home were defined in the Java Speech Grammar Format and then integrated in the speech-to-text system.

2.3. Voice Controller

The Voice Controller (VC) is the module responsible for the entire logic of the application. In the standard scenario, the VC receives a voice command from the user, it queries the ASR system for obtaining the transcription and decides what command to issue to the KNX automation system and what text to synthesize using the TTS system. The synthesized text is then played back to the user. It is also possible to instruct the VC system to automatically start an interaction with the user (i.e. the user returns home and he is asked by the system if he wants the temperature set to certain value). Regardless of who initiates the interaction process (the user or the system) the communication ends with a synthesized message.

The architecture of the system allows for integration of any number of TTS, ASR and VC systems. In fact, the VC system was designed as a standalone application which runs on any Android enabled devices. This allows extended flexibility and diminishes the workload of the servers (the NLP processing is distributed over the machines and also provides a user interface). Typically the application will reside in specially designed Android endpoints which will be placed on the walls of the home. However, the user is given the option to install the application on his mobile phone and he is able to access the automation services remotely.

The logic of the voice controller is provided by a number of rules which are designed and written during the smart-house setup. The set of rules is custom tailored for the users and the automation capabilities which are installed in his home. Later on, these rules can be changed and update by editing the configuration file. Currently, home automation is based on a strict grammar, thus the rules can be easily written. However, we intend to extend support for free language and enable integration with other standard systems, such as organizers, calendars, e-mail system, and weather and question answering.

3. Conclusions and future developments

Controlling devices and home appliances using speech is a natural step in the evolution of smart technologies. In this paper we described our efforts to integrate ASR, TTS and a simple NLU in order to build a voice controlled environment centred on a standard home automation system called KNX. Our work was concluded by building a working prototype of the home automation system with an extensible number of possible interactions. Currently our prototype supports reading data from sensors and issuing commands to smart appliances. However, it suffers from one major limitation of not being able to initiate interactions with the user (though VC provides this support).

Future development plans include reversed interaction, which will enable the system to automatically start a dialogue with the user and the extension of the NLP framework to support unbounded language queries and integration with standard services, other than those provided via KNX. Additional work will also be carried around the TTS and ASR systems in order to provide stable results in terms of recognition accuracy and voice synthesis quality.

Acknowledgements

This work was supported by UEFISCDI, under grant no PN-II-PT-PCCA-2013-4-0789, project "Assistive Natural-language, Voice-controlled System for Intelligent Buildings" (2013-2017).

References

Bartlett, S., Kondrak, G., & Cherry, C. (2009). On the syllabification of phonemes. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 308-316. Association for Computational Linguistics.

Boros, T. (2013). A unified lexical processing framework based on the Margin Infused Relaxed Algorithm. A case study on the Romanian Language. In *RANLP*.

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*, 98-102.

Boros, T., Ion, R., Dumitrescu, S. D. (2013). The RACAI Text-to-Speech Synthesis System. *Blizzard Challenge*.

Boros, T., Ion, R., Tufis, D. (2013). Large tagset labeling using Feed Forward Neural Networks. Case study on Romanian Language. In *ACL (1)*, 692-700.

Crammer, K. Singer, Y. (2003). Ultraconservative online algorithms for multiclass problems. In *Journal of Machine Learning Research*, 951-991.

Erjavec, T. (2004). MULTEXT-East Version 3: Multilingual Morphosyntactic Specifications, Lexicons and Corpora. In *LREC*.

Cucu, H., Buzo, A., Burileanu, C. (2015). The SpeeD Grammar-based ASR System for the Romanian Language. In *Romanian Journal of Information Science and Technology*, vol. 18, no. 1, 33-53, Jan 2015, ISSN: 1453-8245.

Cucu, H., Buzo, A., Petrică, L., Burileanu, D., Burileanu, C. (2014). Recent Improvements of the SpeeD Romanian LVCSR System. In *Proceedings of the 10th International Conference on Communications (COMM)*, Bucharest, 2014, 11-114.

Ion, R. (2007). Word sense disambiguation methods applied to English and Romanian. PhD thesis. Romanian Academy, Bucharest.

Kawahara, H., Masuda-Katsuse, I., De Cheveigne, A. (1999). Restructuring speech representations using a pitch-adaptive time–frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds. In *Speech communication*, 27(3), 187-207.

Taylor, P. (2009). *Text-to-speech synthesis*. Cambridge university press.

Tokuda, K., Yoshimura, T., Masuko, T., Kobayashi, T., & Kitamura, T. (2000). Speech parameter generation algorithms for HMM-based speech synthesis. In Acoustics, Speech, and Signal Processing. In *Proceedings of 2000 IEEE International Conference*, vol. 3, 1315-1318.

Zafiu, A., Dumitrescu, S. D., and Boros, T. (2015). Modular Language Processing for Lightweight Applications. In *Proceedings of Language & Technology Conference*.

A RECURRENT NEURAL NETWORKS APPROACH FOR KEYWORD SPOTTING APPLIED ON ROMANIAN LANGUAGE

SONIA PIPA, TIBERIU BOROȘ

Center for Artificial Intelligence, Romanian Academy

{sonia.pipa, tibi}@racai.ro

Abstract

Keyword spotting (KWS) is a technology that enables the detection of specific spoken words occurrences in audio or video streams. Typical approaches to KWS are based on custom designed HMM decoders. In this paper, we employ Bidirectional Long Short-Term Memory Networks (BDLSTMs) in performing the KWS task using phoneme-bases speech transcription. Our experiments are designed to establish if their capacity to access long-range context provides the necessary support in building a robust KWS system.

Key words — Automatic Speech Recognition (ASR), Keyword Spotting (KWS), and Natural Language Processing (NLP).

1. Introduction

Speech recognition technologies, such as Automatic Speech Recognition (ASR) and Keyword Spotting (KWS), have many applications in real life. Their development was primarily driven by the necessity to improve the human computer interaction experience. ASR systems are commonly used in transcription and dictation tasks, as well as automatic captioning systems and, not least, in speech to speech translation systems. In fact, the increasing requirement to merge ASR with machine translation systems is shown by the large number of papers and challenges which focus on this particular scenario (see International Workshop on Spoken Language Translation). More focused applications for ASR and spoken term detection systems refer to providing accessibility for people with disabilities and speech enabled security systems.

The Keyword Spotting (KWS) technology is primarily intended as a means of finding certain words occurrences in continuous speech records. This, combined with speaker identification, enables the use of voice for authentication purposes or, when integrated in smart security systems, can be used, say, to trigger an alarm on security systems when a specific combination of words, not necessarily consecutive, was detected. Another application for KWS systems is in the task of collecting statistics from media files and streams (TV shows, radio broadcasts, Internet Media etc.), allowing speech analytics to be applied.

Many approaches to KWS have employed the services of Large Vocabulary Speech Recognition systems (LVSR), mostly based on HMMs (see section 2 for details).

However, research has shown that using HMMs trained for ASR to spot keywords has several disadvantages, one of the most important ones being the fact that the LVSR is limited by a predefined vocabulary and it is trained to maximize recognition accuracy allowing its decoding hypothesis to be influenced by the language model (LM) (Keshet *et al*., 2009). As such, better results are obtained by methods which are primarily designed for the KWS task itself.

2. Related work

Standard HMM based KWS approaches perform the keyword detection task by combining a keyword model with a background model, which is also referred to as garbage model. Viterbi search is employed on the target audio file and if the best path moves from the garbage model into the keyword model and most often back to the garbage model, the system detects the occurrence of the word. The keyword model is either a whole word model or a phonetic based model.

Among whole world modelling approaches we count the keyword/non-keyword model introduced in Rahim *et al*. (1997) and Rohlicek (1989), which is useful when provided with appropriate training data. This means multiple recordings of the keyword and multiple recording of non-keywords for the garbage model. However, the bottleneck imposed by the pre-available recordings has enforced the need for phonetic based approaches (Bourlard *et al*., 1994; Manos and Zue, 1997; Rohlicek *et al*., 1993). These approaches model the keyword as a sequence of phonemes (or larger contexts such as triphones), which in turn are estimated using aligned audio recordings, but the recordings are not compelled to contain the word. The garbage model can also be removed by simply estimating the likelihood of the keyword model using a sliding window of data and focusing only on sub-sequences of the audio file/stream (Junkawitsch *et al*., 1997).

Recently, the improvements brought to the modeling and training of recurrent neural network architecture have made it possible to use these classifiers on classical tasks such as part-of-speech tagging (Perez-Ortiz and Forcada, 2001), intelligent character recognition (ICR) (Graves, A. and Schmidhuber, 2009)) and also ASR (Graves *et al*., 2013). One of the interesting models designed for sequence labeling uses Bidirectional Long Short-Term Memory (BDLSTM) networks. The power of these approaches is generated by the fact that given an ordered sequence of feature vectors $s_{1..n}$, for every feature frame s_k the network can learn to model and to access long range dependencies from both the previous feature frames $s_{l,l<k}$ and also the future feature frames $s_{m,m>k}$. This is obtained by employing a layer with two recurrent networks that are fed with data independently: one from left to right and the other from right to left. Of course, this is a feature that is most useful for offline processing of the data, but by introducing a response delay, it can easily be adapted to real-time applications, by buffering the data before applying the classifier.

Our KWS method is similar to that presented in (Wollmer *et al*., 2009). In their paper, the authors use a BDLSTM network architecture to convert words into their

phonetic representation and then use dynamic programming for locating most likely positions that resemble target keywords. The main difference is that we also employ a Connectionist Temporal Classification (CTC) which was recently introduced by Graves *et al*. (2013) for automatically aligning the audio features with their phonetic counterparts.

3. Keyword Spotting System

As previously mentioned, a KWS system requires a means of modelling the acoustic parameters of the target keyword. While HMM based approaches use whole-word modelling or phonetic modelling, using BDLSTMs as classifiers achieves this by transcribing the entire audio file into a sequence of phonemes (garbage or keyword). Thus, KWS with BDLSTMs is a two-fold task: (a) transcribe the audio file and (b) use some type of dynamic programming to locate likely positions of target keywords. Of course, one could argue that some speech frames make the discrimination between phonetic classes hard (i.e. the classifier may choose phoneme 'd' over 'b' by relying on small probability margin, say 51% versus 49%), and once the transcription is done, the search algorithm no longer has access to this information. However, this can be mitigated by using confusion matrices that reflect the chance of misclassification between output classes. In what follows we will discuss the acoustic modelling process that we used (including corpora preparation and classifier training – sections 3.1 and 3.2) and we will introduce and evaluate the actual KWS algorithm (section 3.3).

3.1. Acoustic Model

BDLSTM architecture implies two recurrent hidden layers, both connected to the same input layer and the same output layer which has access to information about the data frame before and after the current frame in the sequence. One hidden layer is used to process the training sequence forwards and the other one to process the same sequence backwards. This architecture enables access to context information that is learned during training and does not have to be specified beforehand. LSTM recurrent neural networks are used to avoid vanishing gradient problem of the simple RNNs. A LSTM memory block contains three multiplicative gates (forget gate, input gate and output gate) and one or more recurrently connected memory cells. The input gate activation multiplies the cell input, the output gate activation multiplies the cell output and the previous cell values are multiplied by the forget gate activation. The gates allow the network to store and retrieve information over long periods of time. If, for example the input gate remains closed, the activation of the cell will not be overwritten by new inputs and can therefore be made available to the net much later in the sequence by opening the output gate. The forget gate resets the memory cell. This principle overcomes the vanishing gradient problem and gives access to long range context information.

Generally classifiers are trained on feature-labels pairs. In speech processing that implies manually or automated aligning audio features with phoneme transcription.

Manually alignment is impossible to be done on huge amount of data. Commonly used approaches for automated alignment are including HMM Baum-Welch. A method we used before to align corpus is HMM Toolkit (HTK). Recently, Graves introduced Connectionist Temporal Classification (CTC), which is an output layer that not requires pre-segmented training data or post processing to transform its outputs into transcriptions, designed for sequence labelling with RNNs. It trains the network to predict a conditional probability distribution over all possible output label sequences, given the complete input sequence.

3.2. Corpus description

Our acoustic model was trained on 8 hours of speech. Normally, this would be an extremely small corpus for automatic speech recognition and it provides a good means of comparing how HMM based methods compare to neural approaches on low resourced languages. The corpus is composed by three sub corpora: (a) the Romanian Anonymous Speech Corpus (Dumitrescu *et al*., 2014); (b) the Romanian Speech Synthesis (RSS) database (Stan *et al*., 2011); (c) a manually aligned audiobook corpus after the Romanian adaptation of Allen Carr's "Easyway to stop smoking". All corpora contain audio files, each with a sentence utterance and text files with their phoneme based transcription. In turn, the RSS corpus consist of 2 hours and 15 minutes of recordings, including 1500 utterances (104 minutes) minutes in random newspaper section, 1000 utterances (53 minutes) in diphone section and 1000 utterances (67 minutes) in fairy-tale section. The RASC corpus contains 4 hours and 20 minutes of speech consisting in recordings of 7000 sentences from Wikipedia whose distribution is described in what follows: (1) **gender distribution**: 33.4 % male speakers, 66.6% female speakers; (2) **age distribution**: 75.8% age between 18 and 35 years, 24.2% between 35 and 60 years; (3) **dialect distribution**: 64.7% Muntenesc, 30.8% Moldovenesc, 3.3% Oltenesc, 1.3% Bucovinean. The third corpus of 1 hour and 20 minutes was obtained by manually aligning 700 audio files, obtained from the audio book, with their transcription.

The training corpus was normalized to 16 KHz, 16 bit, mono and the text was pre-processed by expanding numbers, abbreviations and acronyms to their spoken form. Also, the entire punctuation was stripped and the words were converted into a uppercase form. The phonetic realization of text is not straight-forward and, before we could build our acoustic model, we had to prepare a transcription lexicon in which we included all the unique words and we automatically converted them into phonemes using the grapheme-to-phoneme (G2P) method introduced in Boroș (2013).

In order to test our approach we divided the data into 3 subsets: we randomly extracted 10% of sentences from each corpus (RASC, RSS and audiobook) for a test set, another 10% for a validation set and the rest of 80% was used to train the classifier. The phoneme distribution in our corpora is presented in Table 3.1. The phonetic transcription standard is that which was used by Stan *et al*. (2011) in transcribing the RSS corpus.

Table 3.1. Representation of phoneme distribution through the training corpus

Phonemes	**Occ.**	**Distrib.**	**Train set**	**Validation**	**Test**
<PAU>	1843	1.65%	1474 (1.66%)	185 (1.58%)	184 (1.62%)
A	11118	9.96%	8867 (9.99%)	1150 (9.86%)	1101 (9.75%)
@	1707	1.53%	1357 (1.53%)	192 (1.64%)	158 (1.39%)
AI	3436	3.08%	2681 (3.02%)	399 (3.42%)	356 (3.15%)
B	1039	0.93%	818 (0.92%)	115 (0.98%)	106 (0.93%)
CH	1677	1.50%	1309 (1.47%)	192 (1.64%)	176 (1.55%)
D	3410	3.05%	2719 (3.06%)	339 (2.90%)	352 (3.11%)
DZ	306	0.27%	235 (0.26%)	36 (0.30%)	35 (0.30%)
E	11803	10.57%	9357 (10.55%)	1231 (10.56%)	1215 (10.76%)
EAI	1049	0.94%	838 (0.94%)	122 (1.04%)	89 (0.78%)
F	1698	1.52%	1340 (1.51%)	170 (1.45%)	188 (1.66%)
G	814	0.73%	654 (0.73%)	74 (0.63%)	86 (0.76%)
H	124	0.11%	92 (0.10%)	20 (0.17%)	12 (0.10%)
I	8535	7.65%	6795 (7.66%)	879 (7.54%)	861 (7.62%)
IJ	1091	0.98%	865 (0.97%)	135 (1.15%)	91 (0.80%)
J	2461	2.20%	1957 (2.20%)	252 (2.16%)	252 (2.23%)
K	4121	3.69%	3253 (3.66%)	432 (3.70%)	436 (3.86%)
L	5206	4.66%	4133 (4.66%)	553 (4.74%)	520 (4.60%)
M	3409	3.05%	2772 (3.12%)	322 (2.76%)	315 (2.78%)
N	6902	6.18%	5497 (6.19%)	712 (6.10%)	693 (6.13%)
O	4429	3.97%	3527 (3.97%)	441 (3.78%)	461 (4.08%)
OAI	472	0.42%	380 (0.42%)	49 (0.420%)	43 (0.38%)
P	3398	3.04%	2687 (3.02%)	346 (2.96%)	365 (3.23%)
R	7629	6.83%	6090 (6.86%)	795 (6.82%)	744

					(6.58%)
S	4382	3.93%	3493 (3.93%)	447 (3.83%)	442 (3.91%)
SH	1289	1.15%	1011 (1.13%)	149 (1.27%)	129 (1.14%)
T	7324	6.56%	5841 (6.58%)	732 (6.28%)	751 (6.65%)
TS	1346	1.21%	1036 (1.16%)	173 (1.48%)	137 (1.21%)
U	6402	5.73%	5063 (5.70%)	673 (5.77%)	666 (5.89%)
V	1352	1.21%	1071 (1.20%)	139 (1.19%)	142 (1.25%)
W	431	0.39%	337 (0.37%)	44 (0.37%)	50 (0.44%)
Z	1170	1.05%	927 (1.04%)	137 (1.17%)	106 (0.93%)
ZH	262	0.23%	212 (0.23%)	21 (0.18%)	29 (0.25%)

To train the model for this task we used Rnnlib, a recurrent neural network library, implemented by (Graves, 2013). The network architecture was designed with two layers: a bidirectional recurrent neural network with LSTM blocks of one memory cell on a layer of size 250 and a CTC layer as output. The CTC layer size is equal with the size of the phonetic inventory with two additional slots used for speech pause modelling and non-output (the network does not output any phoneme at a particular frame).

Each speech frame (5ms) was converted into a vector of size 39. The first 13 positions were used for Mel-frequency cepstral coefficients (MFCC) and the next 26 positions contain their first and second derivatives. Thus the network was built with an input shape 1x39. The network converged with momentum set at 0.9 and learning rate 0.0001, using mini-batch training with batch size 1% of total number of input sequences. Training was finished after 70 epochs in 3 days 3 hours and 22 minutes, the last 20 epochs bringing no improvements on error rate. The lowest CTC error was obtained at 30^{th} epoch and the lowest label error at 49^{th} epoch (13.96% on train set, 28.46% on validation set and 25.79% on test set). The variation of label error during the 70 training epochs can be observed in diagram presented in Figure 3.1.

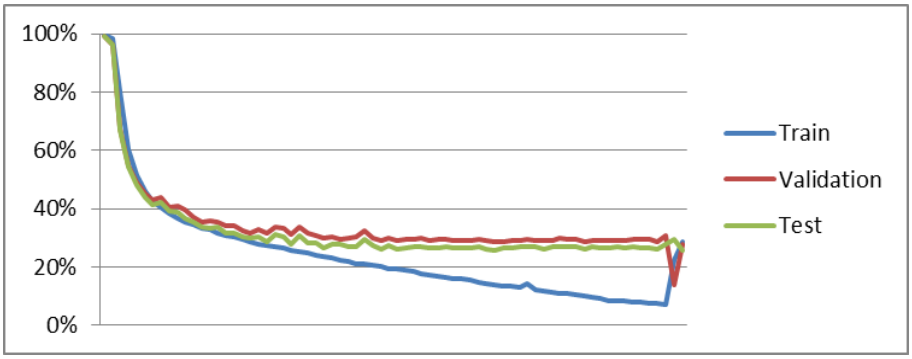


Figure 3.1. Evolution of label error during the model training

3.3. Dynamic search algorithm

So far, we have an acoustic model trained to transcribe audio files into phonemes. To perform the KWS task we have to run this model on audio files in which we want to find keywords, in order to obtain their phonetic transcription. At this point, we require the keyword spotter that searches in the output of the classifier for sequences representing the phonetic transcription of keywords. As previously mentioned in Section 3.2, the accuracy of our acoustic model is 74.21%. However, in keyword spotting it is important to know how this translates into substitutions, deletions and insertions. Substitutions refer to the fact that the classifier mistook one phoneme for another, insertions means that the classifier generated incorrect phoneme sequences between correct phoneme sequences and deletions means that the classier failed to recognize some of the correct phonemes. These scores influence how the alignment score is calculated. For instance, high insertion rates means that the deletion cost used in the alignment algorithm should be low, while low insertion rates mean that the alignment algorithm should use a high score when skipping phonemes. These values can be easily computed by calculating and interpreting the alignment matrix between the system's transcription and the actual (human validated) transcriptions. We evaluated these figures on all corpora (train, validation and test sets) and we obtained the values presented in table 3.3. One can easily see that the model is mostly liable to confuse between phonemes or to introduce incorrect phonemes into transcription and it is less likely not to recognize correct phonemes. At the first glance, one would expect to use dynamic alignments between similar length phonetic sequences. Because transcriptions are imperfect and we don't expect to find perfect matches, in order to obtain acceptable performance from the word spotter, when we extract sequences from the transcription we use a window of phonemes which is larger than the size of the keyword. For every keyword and transcription sequence pair, we dynamically compute the Levenshtein matrix. This matrix holds the partial distances between the two aligned sequences. Thereby, at position [i,j] in the Levenshtein matrix we have computed the distance between first sequence prefix of i phonemes and second sequence prefix of j phonemes. The main difference form standard Levenshtein distance is that we use as

the alignment cost the lowest value from the column (or row – depending on how you apply the algorithm) corresponding to the last phoneme of the keyword. This allows us to ignore the tailing phonemes which we added previously to the sequence of phonemes extracted from the transcribed sentence.

Table 3.3. The insertion, deletion and Substitution rates on all the three sets of corpus

	Train_set	**Validation_set**	**Test_set**
Insertions	5.07%	8.53 %	7.83%
Deletions	0.37%	0.83%	0.25%
Substitutions	8.51%	19.09%	17.69%

The system decides to keep a keyword candidate by thresholding the alignment error. The threshold is relative to the length of the keyword, meaning that longer words allow more misrecognized phonemes.

4. Experimental Validation

In order to thoroughly asses the performance of our keyword spotter, we randomly selected a number of 184 sentences and 25 unique keywords with a total number of 57 occurrences. The sentences went through the entire processing pipeline: extraction of speech parameters, transcription with the neural network and word spotting. The performance of keyword spotting is two-fold: the precision with which the system detects keywords and the recall of actual keyword occurrences. The application, in which KWS is used, is a decisive factor in choosing a good balance between the two performance measures. Both scores are highly dependent on the value of the error threshold. As such, we computed the recall (R), precision (P) and F-score (F) for various threshold values (see table 4.1)

We ran KWS using four error threshold values: 0.1, 0.2, 0.3 and 0.4. We stopped at 0.4 because the recall value was almost at 100%, which means that by increasing the threshold we would only decrease precision. As shown in the table, a threshold value of 0.2 offers balanced results in term of precision and recall and yields the highest F-score of our system, which is 0.71.

Table 4.1. Precision, Recall and F-score for various error thresholds

Threshold	**Precision**	**Recall**	**F-score**
0.1	1	0.23	0.37
0.2	0.97	0.56	0.71
0.3	0.71	0.63	0.67
0.4	0.44	0.98	0.61

5. *Conclusions and Future development*

In this paper we presented our methodology of performing keyword spotting, using recurrent neural networks and a custom dynamic alignment algorithm based on the Levenshtein distance. Our approach is composed of two main steps: frame-wise phoneme classification and keyword localization within phonetic transcriptions. An important advantage of our method consists in the fact that RNNs require less training data than HMM-based approaches. Our methodology falls within the phonetic modelling keyword spotting approaches which means that explicit recordings of the sought words are not mandatory. Future development plans include testing of other neural network architectures (i.e. Sequence-to-Sequence models) for acoustic modelling as well as extension of the current network architecture and keyword spotting algorithm in joint decoding system. Additionally, we want to explore how the amount of training influences the performance of neural inspired models in both KWS and transcription tasks.

References

Boros, T. (2013). A unified lexical processing framework based on the Margin Infused Relaxed Algorithm. A case study on the Romanian Language. In *RANLP*, 91-97.

Dumitrescu, S. D., Boroş, T.,Ion, R. (2014). Crowd-Sourced, Automatic Speech-Corpora Collection–Building the Romanian Anonymous Speech Corpus. In *Proceedings of CCURL 2014: Collaboration and Computing for Under-Resourced Languages in the Linked Open Data Era*, 90-94.

Graves, A. (2013). Rnnlib: A recurrent neural network library for sequence learning problems. *OL* [2015-07-10], http://sourceforge.net/projects/rnnlib.

Graves, A., Schmidhuber, J. (2009). Offline handwriting recognition with multidimensional recurrent neural networks. In *Advances in neural information processing systems*, 545-552.

Graves, A., Mohamed, A. R., Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *Proceedings of 2013 IEEE international conference on acoustics, speech and signal processing*, 6645-6649.

Keshet, J., Grangier, D., Bengio, S. (2009). Discriminative keyword spotting. *Speech Communication*, 51(4), 317-329.

Perez-Ortiz, J. A., Forcada, M. L. (2001). Part-of-speech tagging with recurrent neural networks. Universitat d'Alacant, Spain.

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. *Speech Communication*, 53(3), 442-450.

Wöllmer, M., Eyben, F., Graves, A., Schuller, B., & Rigoll, G. (2009). Improving keyword spotting with a tandem BLSTM-DBN architecture. In *International Conference on Nonlinear Speech Processing*, 68-75. Springer Berlin Heidelberg.

Wöllmer, M., Eyben, F., Graves, A., Schuller, B., Rigoll, G. (2010). Bidirectional LSTM networks for context-sensitive keyword detection in a cognitive virtual agent framework. *Cognitive Computation*, 2(3), 180-190.

Wollmer, M., Eyben, F., Keshet, J., Graves, A., Schuller, B., Rigoll, G. (2009). Robust discriminative keyword spotting for emotionally colored spontaneous speech using bidirectional LSTM networks. In *Proceeings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 3949-3952).

TEXT NORMALIZATION FOR AUTOMATIC SPEECH RECOGNITION SYSTEMS

ALIN-FLORENTIN VASILE, TIBERIU BOROȘ

Center for Artificial Intelligence, Romanian Academy

{alin, tibi}@racai.ro

Abstract

The results of automatic speech recognition (ASR) systems are not directly usable in Natural Language Processing Applications. The main reason is that an ASR system does not output upper/lower word forms (except when the dictionary and language model contain a word explicitly written in its true case) and it does not include any punctuation marks. Though sometimes speech reflects punctuation (speakers do not always embed punctuation in their speech), there are several cases where pauses and pitch discontinuities are randomly added by the speaker. Also it is not straight forward if a pause is added because of a comma, a parenthesis or a full sentence stop. In our experiments we have obtained an F-score of 0.81 for capitalized/uppercase words and an F-score of 0.71 for comma and dot.

Key words — Automatic Speech Recognition (ASR), Natural Language Processing (NLP).

1. Introduction

Automatic speech recognition (ASR) systems usually transcribe audio files as a flat list of recognized words. These words are almost always written in just one casing (upper or lower) and no punctuation is inserted in the recognition results. Embedding punctuation into recognized results is not straight forward in speech recognition. It is obvious that pauses and pitch discontinuities are often used by the speaker to mark punctuation, but there are several cases where this information is not helpful. As such, short pauses may be randomly inserted by the speaker to allow him to catch his breath or whenever he hesitates during his speech. Also, it is not straight forward to detect if a short pause is used to embed a comma, a parenthesis, a dash or a full sentence stop (which translates into period, question mark or exclamation mark).

An un-normalized text is not usable in natural language processing applications, because sentence splitting, word casing and other punctuation marks make low-level text processing tasks (such as part-of-speech tagging, chunking and parsing) extremely difficult. Because of this, normalizing the text is extremely important for automatic machine translation (MT), speech-to-speech translation, information extraction, dialogue systems, etc.

In this paper we describe our method of performing text normalization on ASR output. Our approach is currently text-based only (we don't rely on any information extracted from speech – pauses or pitch) mainly because transcribed speech resources are difficult to obtain and by relying only on text processing we were able to procure more data for training and testing. The system is designed to normalize the sentences by adding commas, full sentence stops and capitalized/uppercase words from the ASR output using an n-gram based approach and a neural network classifier.

This paper is organized as follows. Section 2 introduces previous work in this field. Section 3 presents the entire system used to auto-complete text with commas and capitalized/uppercase words. The future developments are presented in Section 4 and section 5 gives conclusions of the project.

2. Related work

The importance of text normalization has yielded a large number of studies and research papers on this topic. Most of the methods rely on language modelling with n-gram models, but the particular details of implementations vary. As such, Israel *et al*. (2012) use an n-gram model built on words and POS tags and obtain an accuracy of 61.4%. Wang *et al*. (2013) interpolate 3-gram probabilities in order to analyse a window of 5 words, but they apply their method not for ASR output but to social media text normalization, on which they obtain an accuracy of 77.8%. Methods similar to the later mentioned one are also employed for Tweeter text normalization (82.24%) (Sonmez and Ozgur, 2014) and SMS text normalization (80.70%) (Aw *et al*., 2006).

Some authors also employ hybrid approaches based on language-specific rule-based and statistical phrase-based post-editing (Schlippe *et al*., 2010; Dumitrescu and Boroș, 2013).

Our method for text normalization is also a hybrid approach between an n-gram model for truecasing and a DNN classifier trained with unsupervised word embeddings for punctuation restoration.

3. Experimental Setupand Results

Truecasing has been previously done using n-gram models and this methodology is known to provide stable results. Furthermore, when using a wide-coverage training corpus one can make use of heuristics like the fact that unknown words are likely to be proper names or uncommon abbreviations and acronyms which must be either capitalized or uppercased. However, LVSR is usually limited by its dictionary and out-of-vocabulary (OOV) words end up being mapped to similar sounding groups of words. Because of this we limited our approach to only relying on n-gram and we did not use any suffix or prefix analysis of OOV words which could theoretically yield higher accuracy. However, we intend to investigate this approach in future work.

On the other hand, punctuation restoration has known only limited success when n-grams are applied. One observation is that punctuation marks along functional words are very frequent in any language, thus, when applying any type of smoothing over the n-gram probabilities, high frequency unigrams such as comma or period tend to radically increase the probability of n-grams which contain them and disable the possibility of accurately using comparisons between probabilities of sequences with and without punctuation. In fact, one of our early experiments concluded that if we interpolate 3-gram probabilities over a window of 5 tokens and try to estimate comma insertion probabilities based on this score we only get an F-score of 0.56, because the system tends to add as many commas as possible.

On other researches (Gravano *et al.*, 2009) the F-score obtained on a similar approach based on n-gram model was a little bit over 0.5.

Given the above mentioned, our text-normalization methodology is two-fold: first we establish correct word-casing using an n-gram model, and then we use a DNN classifier to determine punctuation insertion points within the text. In what follows we will detail the each of the two steps.

Truecasing refers to the process of determining if a word should be written in lowercase form, with a capitalized letter or in uppercase form. In special cases, the uppercase form must contain periods after every letter. Given a sentence, our truecaser works by processing each word $w_k$inside the sentence and by determining the correct orthographic form.

The analysis process uses a window of 5 tokens centred on the word being processed. Thus for word w_k we take into consideration words $w_{k-2}$to w_{k+2}. In the process we try alternate orthographic forms for the words inside the feature window. Because words w_{k-2} and w_{k-1} have previously been processed we build the Cartesian product of spellings for the wordsw_k, $w_{k+1}$and w_{k+2}. The spellings refer to the 3 cases: lowercase, capitalized and uppercase. Thus, our system tests 27 possible combinations. For every combination we interpolate the probability of seeing that 5 word window using an n-gram model, as a dot product over a sliding window of size 3. This means that we calculate the group probability as a dot product between 3 n-gram probabilities: $P(w_{k-2}, w_{k-1}, w_k)$, $P(w_{k-1}, w_k, w_{k+1})$, $P(w_k, w_{k+1}, w_{k+2})$. The probabilities are computed from the training corpus and, probability smoothing is applied to handlefor unseen n-grams.

To build our n-gram model we used a Wikipedia English corpus composed 125.138.883 sentences, 3.035.591.789 words, 111.247.856 dots and 70.199.700 commas. The corpus was tokenized and we computed unigram, bigram and trigram counts. To test the functionality of our system we kept aside a random set of 100sentences. This subset was stripped of punctuation marks and all words were converted into their lowercase form. This enabled us to evaluate the performance of our system by seeing if it is capable of restoring the text to its original form. Accuracy does not correctly reflect the ability of the system to perform truecasing,

thus, we measured both the success rate of the words that were changed to a different orthographic form, as well as the number of tokens that were correctly changed versus the number of tokens that should have been changed, but were left untouched by the system. Table 1 shows the detailed results on the test set.

Table 3. Truecaser performance on the test set

Words	Precision	Recall	F-score
Capitalized / Uppercase word	0.79	0.83	0.81

Punctuation restoration requires a different approach than that of truecasing. As previously mentioned n-gram models do not offer sufficient support in the decision of adding punctuation marks. Before we describe the approach which yielded the highest accuracy we will introduce an n-gram based experiment which resulted in a very poor F-score of 0.56. Given a sentence of n tokens, similarly to the n-gram truecasing we used tried to determine if a punctuation mark has to be inserted at any position inside the utterance from 2 to n-1 (no probability of insertion was calculated for the beginning and the end of the utterance). The feature window was composed of 4 words centered on the position in which we want to determine the insertion probability. We used overlapping n-grams and computed the non-insertion probability as $P(w_{k-2}, w_{k-1}, w_k)P(w_{k-1}, w_k, w_{k+1})$ and the insertion probability as $P(w_{k-2}, w_{k-1}, PUNCT)P(PUNCT, w_k, w_{k+1})$. Every time we calculated this probability for comma, the insertion probability was magnitudes higher than the non-insertion probability, resulting in the insertion of commas after almost every word in the utterance. Tweaking n-grams and manually adding rules did not yield much improvements in the insertion precision, thus we stopped this experiment and resorted to a different approach. We must note, that a LM build with higher order n-grams and based upon POS tags, rather than wordforms, intuitively should produce better results. However, POS tagging on non-normalized text is not reliable and we preferred to employ a wordform approach.

Neural inspired models have received an increasing interest from the research community. For us, an interesting development was the unsupervised word embedding extraction method introduced by (Mikolov and Dean, 2013). Using large corpora, this method enables the automatic encoding of words into vector space. An important property is that semantically close words have close distanced vectors, and this pre-processing method has produced remarkable results in tasks such as document classification (Xing *et al*., 2014; Kusner *et al*., 2015; Lai *et al*., 2015), sentiment analysis (Zhang *et al*., 2015), machine translation (sequence to sequence models) (Sutskever *et al*., 2014; Cho *et al*., 2014), prosodic modelling (Wang *et al*., 2015; Ding *et al*., 2015; Rallabandi *et al*., 2015; Rendel *et al*., 2016) etc.

Before we trained our classifier we prepared our training data by running word2vec (Mikolov and Dean, 2013) on a large corpus and automatically extracting word embeddings. The vector size for the embeddings was set to 100. For the classification task we used a 3-layer network, with an input layer size of 600, a hidden layer size of 50 and an output layer size of 3. The input layer was fed with

the word embeddings extracted from a window of 6 words. The window was slid from position 1 to position n-1 over the utterance. Sentence start and end were hardcoded as special input vectors which were used whenever the window exceeded the sentence boundaries. Also unknown tokens were encoded using hardcoded vectors. The network was train to output 3 states: (a) non-insertion, (b) comma insertion and (c) full stop. Our testing procedure was performed similarly to true casing. We kept aside 10% of the available data, which was stripped of punctuation marks. After this, we evaluated the system's capacity to reconstruct the original text. Individual performance values are shown in table 2. The system's F-score is 0.71.

Table 4. Performance figures for punctuation restoration

Punctuation mark	Precision	Recall	F-score
Comma	0.92	0.59	0.72
Full stop	0.75	0.64	0.69
Mixed	0.87	0.60	0.71

4. Conclusions and future development

In this paper we introduced our framework for text normalization of automatically transcribed text from audio streams. As previously explained, this task is extremely important in the development of natural language processing applications, such as machine translation, speech to speech translation, dialogue systems, dictation systems etc. The text normalization we performed refers to sentence boundary detection, comma detection and word casing.

In our experiments we explored two approaches to this task: an n-gram based method and a classifier trained on word embeddings approach. As shown, the n-gram based method provides sufficient accuracy for the word casing task, but it does not offer the necessary support for punctuation restoration. This required us to use the classifier approach which, in turn, has performed the punctuation task with a satisfactory accuracy.

Most of the work carried out here for driven by the need to implement a spoken language translation system for pre-recorded TV shows. The main target of our approach is TED Talks for the IWSLT Challenge (Cettolo *et al.*, 2015) and recorded Skype calls for the MSLT Challenge (same reference as before). The text normalization system is only a small module in the entire system. Additional work has been carried out to train a large vocabulary speech recognition system and to develop a new decoder for machine translation. The decoder is a hybrid system which enables the extraction of translation equivalents from **monolingual corpus** using a small seed dictionary and neural word embeddings. The speech translation system is not yet tested as a whole, but when testing and tweaking is finished, all the modules will be presented in future work.

There are several disadvantages of the text-normalization system we described: (a) the limited context which the n-gram approach can take into account; (b) the n-gram approach is dictionary bound; (c) the neural punctuation restoration is able to perform local optimizations only; (d) speech parameters from the audio file are

currently ignored. As such, our future development plans we intend to investigate how we can combine more features and use different classifiers for normalizing the text. These future developments will revolve around the following ideas:

(a) Recurrent neural networks (RNNs) are able to learn how to model and access long-range dependencies. Also, by combining two RNNs in the input layer and feed data from left to right to one layer and from right to left to the other layer one can obtain what is known as a bidirectional RNN. The power of this methodology is that the system makes it decision by harnessing all available data. Long range dependencies are extremely useful for determining the sentence type when intonation and prosody do not provide sufficient clues. For instance, the sentence "cine a facut asta" (en. who did this) is interrogative and the presence of the word "cine" at the start of the sentence is a decisive factor in this sense. It is likely that a bidirectional RNN will be able to learn this dependency, while an n-gram model is likely to fail.

(b) Prosody offers good clues toward punctuation restoration. However, as previously explained, aligned speech data is hard to come by. While a speech corpus may contain up to 50K sentences, monolingual text data is magnitudes larger. 50M sentences are a reachable goal, thus, we preferred to start our research by using only text corpora. It is however possible to limit the guessing of the classifiers to points in speech when prosody extracted from audio files indicates that it would be appropriate. These speech features we want to use refer to short pauses, pitch discontinuities and speaking style (speed).

(c) To reduce the impact of OOV words on the truecasing we want to use grapheme based methods to see if suffixes, prefixes, letter combinations, combined with surrounding word embeddings are able to provide sufficient clues to determine if the OOV word is indeed a proper name or an abbreviation/acronym.

Furthermore, we intend to evaluate our platform on multiple languages and document genres, to see if the data-driven methods we proposed are easily adaptable to other languages. Also, we want to investigate the possibility of employing bidirectional recurrent neural networks for post-editing ASR results. This method has been previously used on limited domain speech recognition systems (Bang *et al*., 2015; Kim *et al*., 2016), but it would be interesting to see how this would apply to LVSR.

References

Aw, A., Zhang, M., Xiao, J., Su, J. (2006). A phrase-based statistical model for SMS text normalization. In *Proceedings of the COLING/ACL on Main conference poster* sessions, 33-40.

Bang, J., Park, S., Lee, G. G. (2015). ASR Independent Hybrid Recurrent Neural Network Based Error Correction for Dialog System Applications. In

Proceeings of the Workshop „Multimodal Analyses enabling Artificial Agents in Human-Machine Interaction”, Singapore, September 14, 2014

Cettolo, M., Niehues, J., Stüker, S., Bentivogli, L., Cattoni, R., Federico, M. (2015). The IWSLT 2015 Evaluation Campaign. In *Proceedings of the twelveth International Workshop*.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., &Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

Ding, C., Xie, L., Yan, J., Zhang, W.,Liu, Y. (2015). Automatic prosody prediction for Chinese speech synthesis using BLSTM-RNN and embedding features. In *Proceedings of the Workshop „Automatic Speech Recognition and Understanding” (ASRU)*. 98-102.

Dumitrescu, S. D., Boros, T. (2013) A unified corpora-based approach to Diacritic Restoration and Word Casing. In *Proceedings of LTC 2013*.

Israel, R., Tetreault, J., Chodorow, M. (2012). Correcting comma errors in learner essays, and restoring commas in newswire text. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologie*, 284-294

Kim, B., Choi, J., Lee, G. G. (2016). ASR Error Management Using RNN Based Syllable Prediction for Spoken Dialog Applications. In *Advances in Parallel and Distributed Computing and Ubiquitous Services*, 99-106. Springer Singapore.

Kusner, M. J., Sun, Y., Kolkin, N. I., Weinberger, K. Q. (2015). From word embeddings to document distances. In *Proceedings of the 32nd International Conference on Machine Learning*, 957-966.

Lai, S., Xu, L., Liu, K., Zhao, J. (2015). Recurrent Convolutional Neural Networks for Text Classification. In *AAAI*, 2267-2273.

Mikolov, T., Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*.

Rallabandi, S. K., Rallabandi, S. S., Bandi, P., Gangashetty, S. V. (2015). Learning continuous representation of text for phone duration modeling in statistical parametric speech synthesis. In *Proceedings of the Workshop „Automatic Speech Recognition and Understanding” (ASRU)*, 111-115.

Rendel, A., Fernandez, R., Hoory, R., Ramabhadran, B. (2016). Using continuous lexical embeddings to improve symbolic-prosody prediction in a text-to-speech front-end. In *Proceedings of 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5655-5659.

Schlippe, T., Zhu, C., Gebhardt, J., Schultz, T. (2010). Text normalization based on statistical machine translation and internet user support. In *INTERSPEECH*, 1816-1819.

Sonmez, C., Ozgur, A. (2014). A Graph-based Approach for Contextual Text Normalization. In *EMNLP*, 313-324.

Sonmez, C., Ozgur, A. (2014). A Graph-based Approach for Contextual Text Normalization. In *EMNLP*, 313-324.

Sutskever, I., Vinyals, O., Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, 3104-3112.

Wang, P., Ng, H. T. (2013*).* A Beam-Search Decoder for Normalization of Social Media Text with Application to Machine Translation. In *HLT-NAACL*, 471-481.

Wang, P., Qian, Y., Soong, F. K., He, L., Zhao, H. (2015). Word embedding for recurrent neural network based tts synthesis. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4879-4883.

Xing, C., Wang, D., Zhang, X., Liu, C. (2014). Document classification with distributions of word vectors. In *Proceedings of Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, 2014 Asia-Pacific, 1-5.

Zhang, D., Xu, H., Su, Z., Xu, Y. (2015). Chinese comments sentiment classification based on word2vec and SVM perf. *Expert Systems with Applications*, 42(4), 1857-1863.

Gravano, A., Jansche, M., Bacchiani, M. (2009). Restoring punctuation and capitalization in transcribed speech. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, 4741-4744.

CHAPTER 4
LEXICAL AND SEMANTIC RESOURCES

ADJECTIVES IN WORDNET: SEMANTIC ISSUES

TSVETANA DIMITROVA, VALENTINA STEFANOVA

Institute for Bulgarian Language, Bulgarian Academy of Sciences

{cvetana, valentine}@dcl.bas.bg

Abstract

The paper presents some preliminary observations on the classification of the adjectives in WordNet for a discussion on the principles applied. The insights support a work-in-progress on the development and introduction of a more detailed classification of the adjectives in the (Bulgarian) WordNet for enriching it with further information about the concepts and relations between them to increase the WordNet effectiveness and applicability.

Key words — adjectives, semantic classification, WordNet

1. Introduction

The discussion in this paper offers a glimpse into the approach to the classification of the adjectives in WordNet with an aim to outline the basic principles and their effectiveness for a task of construction and application of a more detailed semantic classification of adjectives (to be subsequently applied to the Bulgarian WordNet (BulNet)[1]. These are very much preliminary observations that are meant to support our work-in-progress. In Section 2, we discuss the organisation of the adjectives in WordNet – the Princeton WordNet (PWN; applicable to other wordnets – such as the Bulgarian WordNet and the Romanian WordNet, as both transfer the PWN synset structure despite certain specifics reflected in missing or additional information, cf. Mititelu, 2013; Dimitrova *et al.*, 2014; Koeva, 2014), as well as the German WordNet and the Polish WordNet (that employ more detailed classifications of adjectives). Section 3 discusses the approaches to the adjectives in Bulgarian and the structure of the Bulgarian WordNet for an attempt at outlining the cornerstones of a potential approach to introducing new semantic classes of adjectives in WordNet.

2. Adjectives in WordNet

WordNet (WordNet) is a lexical-semantic network of semantic concepts organised as sets of related words ('synonyms') – or synonym sets (synsets) – that are linked by means of lexico-semantic relations. A synset may contain one or more words

[1]The data in the paper is from the Princeton WordNet, the Bulgarian WordNet and the Romanian WordNet as shown at: http://dcl.bas.bg/bulnet/ – a web inerface for wordnet (for detail about the interface and the visualisation of the data, see Rizov *et al.*, 2015).

(literals). The Princeton WordNet for English (Fellbaum, 1999) is the first such network which has been constructed since 1985 at the Laboratory of Cognitive Studies at the Princeton University (the latest edition – PWN 3.1 – is currently available only online and covers over 117,000 synsets, while wordnets for languages of the world are over 50[2]).

2.1. Adjectives in the Princeton WordNet

WordNet concepts are nodes linked to each other via edges expressing conceptual-semantic relations between concepts (such as hypernymy/hyponymy, meronymy, semantic roles, etc.) and lexical relations between lexical items (antonymy, synonymy, similarity, derivativity, etc.). PWN covers only open-class words – verbs, nouns, adjectives, and adverbs – that are described via relations to other synsets. Some relations are specific to certain parts-of-speech, e.g., hyponymy is specific to nouns and verbs, meronymy is applied only to nouns, and verbs and nouns are related to each other via morpho-semantic relations that hold among semantically similar words sharing a stem with the same meaning (Fellbaum *et al.*, 2009). The concepts are described also by relations within the same synset: literal, part-of-speech, identification number of the synsets (ILI (inter-lingual index) that links a PWN synset to synsets in wordnets for other languages (Vossen, 2002), thematic domain, notes (both on synset and on literal level) usage examples, and others. Each synset is classified by a semantic primitive (Miller *et al.*, 1993; Fellbaum *et al.*, 2009). Nouns are organised into 25 semantic classes (*noun.person, noun.animal, noun.substance, noun.event,* etc.), while verbs are classified under 15 primes (*verb.stative, verb.change,* etc.; for semantic relations in Bulgarian, Romanian and English WordNets, cf. Koeva, 2008; Barbu Mititelu *et al.*, 2015).

Adjectives are classified into two larger classes – descriptive adjectives and relational adjectives – and an additional class of adjectival participles. Descriptive adjectives (*adj.all*) are organised into clusters based on similarity of meaning (synonymy) and binary opposition (antonymy). Relational adjectives (*adj.pert*) are (derivationally) linked to a synset containing their source noun. Adjectival participles are marked as *adj.ppl* and are related via *participle* relation to synsets containing the verbs they are derived from.

Adjectives are organised via relations encoding properties of attribution, antonymy, similarity, derivation, fuzzynymy, thematic category, etc.; some are specific for one of the two classes (attribute, similarity, fuzzynymy – for *adj.all*; pertainym – *adj.pert*; participle – *adj.ppl*), while others are found with more classes (though showing preference) – see Table 1.

Table 1. Relations distributed according to semantic primes of adjectives in PWN

Relation	***adj.all***	***adj.pert***	***adj.ppl***
attribute	602	-	-
antonym	3,738 (1,869)	94 (47)	14 (7)

[2] See http://globalwordnet.org/wordnets-in-the-world/.

similar_to	13,205	-	-
eng_derivative	5,454	2,396	1
has_derived	2,264	270	1
has_pertainym	-	3,617	-
is_participle_of	-	-	229
also_see	1,333	-	-
category_domain	834	243	2
usage_domain	213	7	-

2.2. Relations with adjective synsets in PWN

Attribution is an asymmetric relation (*has_attribute / has_value*) that links a descriptive adjective expressing an attribute with the noun for the value of the attribute. For example, the antonym synsets {accessible:1} and {inaccessible:1, unaccessible:1} are linked to the noun synset for the attribute {n: handiness:1, accessibility,...}, as shown in Ex. 1.

Example 1:

{a: accessible:1}

antonym: {a: inaccessible:1, unaccessible:1}

has_attribute: **{n: handiness:1, accessibility:1, availability:1}**

Antonymy is a symmetric relation that links both descriptive (Ex. 1) and relational adjectives (Ex. 2), though the latter are much fewer (as shown in Table 1).

Example 2:

{a: cathodic:1} *'of or at or pertaining to a cathode'*

antonym: **{a: anodic:1, anodal:1}**

Similarity is a symmetric relation (marked by *similar_to*) that links both descriptive and relational adjectives with other adjectives. It is claimed that adjectives linked via a similarity relation are indirect antonyms to the direct antonym of their antonymic adjective (Fellbaum *et al.*, 1993). This means that in Ex. 3 {a: smart:3} which is a direct antonym of {a: stupid:2}, would be an indirect antonym of all the adjectives that are similar to it, i.e., {a: anserine:1, dopy:1,...}, {a: cloddish:1, doltish:1}, etc.

Example 3:

{a: stupid:2} *'lacking or marked by lack of intellectual acuity'*

has_derived: {b: stupidly:1; doltishly:1}

eng_derivative: {n: stupidity:2}

eng_derivative: {n: stupid:4; stupid person:1;stupe:1; dullard:2; dolt:1…}

antonym: {a: smart:3}

similar_to: {a: anserine:1; dopy:1; dopey:1; foolish:1;goosey:1...}

similar_to: {a: cloddish:1; doltish:1}

similar_to: {a: dense:1; dim:7; dull:6; dumb:4;obtuse:1; slow:5}

also_see: {a: unintelligent:1; stupid:3}

WordNet features also derivational relations between synsets with derivationally related words (literals) (the relation as discussed here is between synsets (Koeva, 2008), but it can also link literals (Fellbaum *et al.*, 2009) for English, (Koeva, 2008; Dimitrova *et al.*, 2014) for Bulgarian; (Mititelu, 2012) for Romanian). In Ex. 3, derivationally related to {a: stupid: 3} are nouns {n: stupidity:2} and {n: stupid:4, stupid person:1, stupe:1}. Adverbs – such as {b: stupidly:1} in Ex. 3 – are linked to adjectives via the asymmetric derivational relation *derived/has_derived*.

Adjectives linked to nouns they are pertaining to are only relational ones (*adj.pert*). They rarely have antonyms and contain fewer literals – see Ex. 4 for PWN, RoWN, BulNet.

Example 4:

{a: planetary:4, terrestrial:4} *'of or relating to or characteristic of the planet Earth or its inhabitants'*

pertainym: **{n: Earth:1, earth:6, world:7, globe:2}**

{a: planetar:1}

pertainym: **{n: glob:4, lume:4, pământ:4}**

{a: земен:4}

pertainym: **{n: Земя:1, земно кълбо:1, свят:9}**

Another group of adjectival elements that are (derivationally) related to a source word are participial adjectives (*adj.ppl*) that are linked to verbs via the asymmetric *participle* relation.

Example 5:

{a: punishing:2} *'resulting in punishment'*

has_derived: {b: punishingly:1}

is_participle_of: **{v: punish:1; penalize:1; penalise:1}**

The fuzzynymy relation (*also_see*) links descriptive adjectives without specifying the exact nature of the semantic relation, as illustrated in Ex. 6.

Example 6:

{a: inadvisable:1, unadvisable:1} *'not prudent or wise; not recommended'*

also_see: **{a: imprudent:2}**

also_see: **{a: foolish:2}**

{a: abundent:1, mare:13, îndestulător:1}

also see: **{a: amplu:1}**

also see: **{a: bogat:12}**

Some adjectives are linked to noun synsets for thematic domain (via *category_domain* relation) – shown in Ex. 7 – or usage domain (via *usage_domain* relation – Ex. 8) to limit their use to a certain domain or style.

Example 7:

{a: acid:1}

category_domain: **{n: chemistry:1, chemical science:1}**

Example 8:

{a: brun:3, brunet:6, negricios:2, oacheș:2}

usage_domain: **{n: arhaism:1}**

Although the classification of the adjectives in PWN and the division into two major types may account for the majority of the adjectives in English, the authors of PWN do not claim complete coverage (Fellbaum *et al.*, 1993). There is information that is missing (on derivation, semantics, usage, etc.) and that would be highly beneficial for raising the effectivity and enriching the connectivity of WordNet. One of the steps towards this aim is to apply a more detailed classification (the low number of some relations in Table 1 can be read as a sign that there is much more behind the PWN three-way – or even two-way division – of adjectives), and this direction has been followed by other wordnets two of which are presented briefly in 2.3 (see also (Mendes, 2006) for WordNet of European Portuguese; (Azarova and Sinopalnikova, 2004) for the Russian WordNet).

2.3. Classification of adjectives in other wordnets

2.3.1. Adjectives in GermaNet

Semantic classification of adjectives in the WordNet for German (GermaNet) is based on a classification by Hundsnurscher and Splett (1982) which employs the modification property of the adjective – a (modifying) adjective is (semantically) linked with a certain (modified) noun to form a separate semantic entity. The classification of Hundsnurscher and Splett (1982) is hierarchically organised into 13 semantic fields that are divided by several semantic sub-features (resulting into 70 classes). Thus, GermaNet draws further distinction by introducing additional semantic features.

The approach here is a hierarchical rather that a cluster-based one (as in PWN) and is comparable to the structures for nouns and verbs in WordNet. There are benefits: elimination of indirect antonyms; limited reliance on antonymy; the division of relational and descriptive adjectives is somehow abandoned. The resulting hierarchy is based on a rich set of semantic classes covering perceptional, temporality-related, material-related, etc. adjectives (each category is divided into further subcategories, e.g., perceptional adjectives may express perception of: lightness, colour, sound, taste, smell, surface; social-related adjectives may express social categories of: stratum, institution/politics, religion, state, race, region; etc.) (Hamp, 1997).

2.3.2. Adjectives in Polish WordNet

The choice of semantic relations for the Polish WordNet (plWordNet 1.0) was guided by WordNet tradition, theory of lexical semantics and lexicographic practice (Derwojedowa *et al.*, 2008) with many relations being taken over from PWN and EuroWordNet (incl. antonymy, hypernymy, meronymy, conversion, relatedness and pertainymy, fuzzynymy (Piasecki *et al.*, 2009). The plWordNet 2.0 offers a much richer set of lexico-semantic relations among adjectives that has been constructed following three criteria: solutions in other wordnets (mainly PWN, EuroWordNet, sometimes GermaNet), lexico-semantic properties of the Polish language, and lexicographic tradition (Maziarz *et al.,* 2012).

The following relations among adjectives are introduced: gradable antonymy and complementary antonymy; inter-register synonymy; hyponymy; value of the attribute; cross-categorial synonymy; state; cause; process; three types of derivation (plus derivational role); fuzzynymy. On the level of synset, there are: hyponymy/hypernymy; value of attribute; gradation; distributional properties; and inter-register synonymy. Relations on the level of the lexical unit are: gradable antonymy, comparative antonymy, converseness, derivational relations, fuzzynymy.

3. Adjectives in Bulgarian

3.1. Classification of adjectives in Bulgarian

Traditionally, the adjectives in Bulgarian are divided into two larger classes – qualitative and relational adjectives. The quality expressed by a qualitative adjective is considered implicit and fairly constant property of an object or an event which does not depend on its relation to another object or event (the adjectives that are synchronically derived from a lexical base, e.g., *golyam* 'big', *zelen* 'green', etc., are often classified as qualitative); they can form comparative and superlative forms. Relational adjectives are derived (mostly) from nouns using a set of suffixes such as *-ov/-in*, *-ski, -en*. This classification, however, is far from clear, with many borderline cases (Radeva 1991; Barbolova 1997; Radeva 2011).

Firstly, Bulgarian is a morphologically rich language, which heavily employs derivation as word formation mechanism, and even adjectives with opaque derivational structure are not unequivocally classified as relational. For instance, the adjectives formed with the suffixes *-en* and *-(ich)en* are often considered qualitative, while those with *-ski* and *-(ich)eski* are classified as relational (Radeva, 2011). Qualitative adjectives can be also formed from nouns (as with denominal adjectives such as *strahl**iv*** 'fearful', *gnev**en*** 'angry', *brad**at*** 'bearded', etc., and deverbal such as *chup**liv*** 'breakable', etc.), i.e., they are derivationally relational but express quality/property of the modified object or event that can be revealed by its relation to another object or event (Radeva, 1991).

Secondly, the adjective is often analysed as a dependent lexical class whose semantic and syntactic properties are fully realised only in its modifying function,

e.g., in relation to a modified noun (Radeva, 1991). Thus, even when an adjective expresses a property of being related to an object or an event (as with relational adjectives), it expresses a relational property of another object or event that manifests in a certain way, to a certain degree or in relation to a certain internal property of the modified object or event (material – *srebar**en** prasten* 'silver ring'; location – *gor**en** etazh* 'above floor'; purpose – *stroitel**en** kran* 'construction crane'; etc.). Transposition in derivational models is highly employed by denominal adjectives – the relation between the denominal adjective and the noun it modifies may express condition (temporal or local), possession, purpose, meronymy, source or material (Radeva, 1991).

Thirdly, relational adjectives are often metaphorically used (Barbolova, 1997) to express a resemblance or metaphorical transfer of the properties of the source noun of the modifying adjective to the properties of the noun modified by this adjective. In such cases, relational adjectives are said to cover some properties of qualitative adjectives. Barbolova (1997) gives an example with the adjective *zlat**en*** 'gold, golden' as in *zlat**en** chovek* 'a very good person', zlat*na* dusha 'very good soul', *zlat**en** glas* 'a very clear and good voice'.

To sum up, the two-way division of adjectives (that is traditional in literature on Bulgarian) is not entirely adequate when taking into account many additional properties and usage of the adjectives, especially in the context of their use as modifying elements to modified nouns (expressing a modified object or event).

3.2. The Task: Adjectives in the Bulgarian WordNet

The organisation of the adjectives in the Bulgarian WordNet follows, in principle, the structure of the Princeton WordNet. Descriptive and relational adjectives are linked to other synsets via different sets of relations and are classified under two semantic classes into separate non-intersecting structures (Koeva, 2014). Since Bulgarian language employs much more derivational means for word production than English including with relational adjectives, e.g., *kon**ski*** 'horse's' as in *kon**ska** opashka* 'horsetail', the Bulgarian WordNet contains some language specific synsets for adjectives.

Our task is to semantically classify the adjectives in the WordNet. As outlined above, the division between descriptive and relational adjectives (followed by PWN) does not hold for adjectives in Bulgarian, and, probably, in other languages either. If the hypothesis about the universality of the WordNet concepts and relations between them holds true, we can transfer the information about the semantic class of adjectives from other wordnets (such as GermaNet) while keeping the PWN two-fold classification (for compatibility with PWN). However, the semantic classes can be further edited through extraction of additional information from the synsets, incl. lexico-semantic relations, definition, usage examples, and the synsets linked via lexico-semantic relations.

To illustrate what information can be extracted from the available data, we give an example with the adjective *zlat**en*** 'gold, golden'. There are seven synsets in the current version of BulNet that contain a literal *zlat**en*** 'gold, golden'[3].

Only two of the synsets are classified as pertainyms (or relational adjectives derived from a noun) – given in Ex. 9. They are marked by a semantic prime *adj.pert* and a relation *pertainym* to a relevant (source) noun synset. In both English synsets, there is no literal that is derivationally related to its pertainym 'gold' but both Bulgarian synsets contain the word/literal *zlaten* 'golden'. The first – {златен:6} – is derivationally related to its pertainym {злато:4}. The other one – {златен:1}– has no relation to any of the literals in the related synset. It is an adjective that can be metaphorically used in certain contexts as a stylistical synonym of the other literals, and the synset contains information that can be used for a more detailed semantic classification (for example, marking a temporal property).

Example 9:

{a: **златен:6**} / {a: aurous:1, auric:1} *'of or relating to or containing or derived from gold'*

has_pertainym: {n: **злато:4**, Au:1} / {n: gold:7, Au:2, atomic number:79}

{a: августовски:1, класически:3, **златен:1**} / {a: Augustan:1} *'relating to or characteristic of the times of the Roman Emperor Augustus'*

has_pertainym: {n: Август:1, Гай Юлий Цезар Октавиан:1} / {n: Augustus:1, Gaius Octavianus:1, Gaius Julius Caesar Octavianus:1, Octavian:1}.

The other five adjectives are classified as *adj.all* but all synsets contain information and relations that can be employed for their further semantic specification. All of them contain a literal *zlaten* 'gold, golden'. As illustrated in Ex. 10, {a: златен:2} can be classified as material- or source-related. It is the only one of the five that is linked to {a: злато:4} / {a: gold:7}, thus posing a challenge for further distinction between {a: златен:6} / {a: aurous:1, auric:1} and {a: златен:2, позлатен:1} / {a: gold:2, golden:6, gilded:2}.

Example 10:

{a: **златен:2,** позлатен:1} / {a: **gold:2**, golden:6, gilded:2} *'**made from** or covered with **gold**'*

eng_derivative: {n: **gold:6**}

eng_derivative: {n: **gold:7;** Au:2; atomic number 79:1}

similar_to: {a: metallic:1; metal:1}

The adjective {a: златен:3; златист:1} / {a: aureate:1; gilded:1; gilt:1; gold:1; **golden:3**} is linked to color-related concepts (both noun and adjective synsets), with mentioning of color in its definition.

[3]Examples from the Bulgarian WordNet are given in Cyrillic (as visualised at: http://dcl.bas.bg/bulnet/).

Example 11:

{a: **златен:3;** златист:1} / {a: aureate:1; gilded:1; gilt:1; gold:1; **golden:3**} *'having the deep slightly brownish **color of gold**';*

eng_derivative: {n: gilt:2; gilding:1}

eng_derivative: {n: amber:2; **gold:3**}

similar_to: {a: chromatic:1}

Both {a: златен:4} / {a: golden:4} and {a: златен:5} / {a: golden:5} do not have any relation or mentioning (even in the definition) of 'gold'. It is fair to assume that they are metaphorically used but their further classification needs additional information that can be found in the hierarchy (in the classification of the related synsets, etc.).

Example 12:

{a: облагодетелстван:1, **златен:4**} / {a: fortunate:2, **golden:4**} *'supremely favored'*

similar_to: {a: blessed:2; blest:1}

{a: **златен:5,** процъфтяващ:2, цъфтящ:3} / {a: **golden:5**, halcyon:1, prosperous:2} *'marked by peace and prosperity'*

similar_to: {a: happy:3}

The information from the related synset can certainly be used for {a: златен:8} / {a: golden:2}. It can be classified as voice/sound-related adjective as it is linked via similarity relation to {a: euphonious:1; euphonous:1}.

Example 13:

{**златен:8**} / {**golden:2**} *'suggestive of gold'*

similar_to: {a: euphonious:1, euphonous:1}

The examples given here attest for the richness of the information available in WordNet. Though our work-in-progress will be focused on Bulgarian, the discussion of the illustrative examples is not specifically focused on Bulgarian adjectives (with the exception of the synsets that contain derivationally related literals as opposed to English). However, if the WordNet is claimed to be a lexical-semantic network of universal concepts, then the steps outlined here would be applicable to any semantic network with the same (or similar structure) and constituting principles.

3. Conclusions

The paper presented some arguments for a more detailed classification of the adjectives in WordNet, including the sparsity and discrepancy of the existing classification of the adjectives in the PWN (as opposed to nouns and verbs), its inadequacy with respect to semantics (again, the classifications of nouns and verbs are much more semantically based) and to the existing semantic classifications of

adjectives for other language. In line with the preliminary character of the observations here as stated in the beginning, these arguments are not exhaustive.

Acknowledgements

The work reported in this paper is carried out within the project *Semantic Classification of the Adjectives in the Bulgarian WordNet* of the Institute for Bulgarian Language which is supported under the the *Program for Career Development of Young Scientists* at the Bulgarian Academy of Sciences.

References

Azarova, I., Sinopalnikova, A. (2004). Adjectives in RussNet. In *Proceedings of the Global Wordnet Conference 2004*, Brno, 251–258.

Barbolova, Z. (1997). *Contemporary Bulgarian Language. Morphology. [Съвременен български език. Морфология]*, Sofia.

Barbu Mititelu, V. (2012). Adding Morpho-semantic Relations to the Romanian Wordnet. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012)*, 2596–2601.

Barbu Mititelu, V., Rizov, B., Tarpomanova, E., Leseva, S., Dimitrova, T. (2015). Noun-Verb Derivation in the Bulgarian and the Romanian WordNet – A Comparative Approach. In *Proceedings of the 11th Internațional Conference Linguistic Resources and Tools for Processing the Romanian Language*, 53 – 62.

Derwojedowa, M., Szpakowicz, S., Zawisławska, M., Piasecki, M. (2008). Lexical Units as the Centrepiece of a Wordnet. In *Proceedings of the 16th International Conference Intelligent Information Systems*, 351–358.

Dimitrova, T., Tarpomanova, E., Rizov, B. (2014). Coping with Derivation in the Bulgarian Wordnet. In *Proceedings of the Seventh Global Wordnet Conference*, Tartu, Estonia, 109-117.

Fellbaum, C. D. (1999). *Wordnet: An Electronic Lexical Database.* Cambridge, MA: MIT Press.

Fellbaum, C., Gross, D., Miller, K. (1993). Adjectives in WordNet. – In *Introduction to WordNet: an On-line Lexical Database. Five Papers on WordNet.* <http://wordnetcode.princeton.edu/5papers.pdf> [22/09/2016]

Fellbaum, C., Osherson, A., Clark, P. (2009). Putting Semantics into WordNet's "Morphosemantic" Links. In *Proceedings of the Third Language and Technology Conference, Poznan, Poland.* [Repr. in: Responding to Information Society Challenges: New Advances in Human Language Technologies. Springer Lecture Notes in Informatics], vol. 5603, 350–358.

Hamp, B. H. (1997). GermaNet – a Lexical-Semantic Net for German, <https://aclweb.org/anthology/W/W97/W97-0802.pdf> [22/09/2016]

Hundsnurscher, F, Splett, J. (1982). *Semantik der Adjektive im Deutschen: Analyse der semantischen Relationen.* Wiesbaden: Westdeutsches Verlag.

Koeva, S. (2008). Derivational and Morphosemantic Relations in Bulgarian Wordnet. *Intelligent Information Systems*, 359–368.

Koeva, S. (2014). WordNet and BulNet [WordNet and БулНет]. In *Language Resouces and Technologies for Bulgarian Language [Езикови ресурси и технологии за български език],* Sofia.

Leseva, S., Todorova, M., Dimitrova, T., Rizov, B., Stoyanova, I., Koeva, S. (2015) Automatic Classification of WordNet Morphosemantic Relations. In *Procedings of BSNLP 2015*, 59.

Mendes, S. (2006). Adjectives in wordnet. pt. *Proceedings of the GWA*.

Maziarz, M., Szpakowicz, S., Piasecki, M. (2012). Semantic Relations among Adjectives in Polish WordNet 2.0: A New Relation Set, Discussion and Evaluation. *Cognitive Studies,* 12, 2012.

Miller G., Beckwith R., Fellbaum C., Gross D., Miller K. (1993). *Introduction to WordNet: an On-line Lexical Database. Five Papers on WordNet.* <http://wordnetcode.princeton.edu/5papers.pdf> [22/09/2016]

Mititelu, V. B. (2013). Increasing the Effectiveness of the Romanian Wordnet in NLP Applications. *The Computer Science Journal of Moldova*, *21*(3), 320–331.

Piasecki, M., Szpakowicz, S., Broda, B. (2009). *A Wordnet from the Ground Up.* Wydawnictwo Politechniki Wrocławskiej, Wrocław.

Radeva, B. (2011). *Relational Adjectives in Contemporary Bulgarian Language [Относителните прилагателни в съвременния български език].* 2011.

Radeva, V. (1991). *Word Formation in Bulgarian Literary Language [Словообразуването в българския книжовен език].* Sofia.

Rizov, B., Dimitrova, D., Barbu Mititelu, V. (2015). Hydra for Web: A Multilingual Wordnet Viewer. In *Proceedings of the 11th Internațional Conference Linguistic Resources and Tools for Processing the Romanian Language,* 19 – 30.

Vossen, P. (2002). WordNet, EuroWordNet and Global WordNet. *Revue française de linguistique appliquée,* 1/2002 (Vol. VII), 27-38.

TERMINOLOGY APPROACH IN ROMANIAN LANGUAGE DICTIONARIES.THEORETIC AND PRACTICAL ASPECTS. STUDY OF DLR AND DEX

MIHAELA MARIN

„Iorgu Iordan – Al. Rosetti“Institute of Linguistics, Romanian Academy

marin.rmihaela@gmail.com

Abstract

Issues followed in this paper: 1. a history of Romanian terminology; 2. some difficulties in lexicographer's work (DEX, DLR); 3. criteria of DLR and DEX terms selection; 4. lexical content enrich; 5. what fields the terms belong to; 6. how is a dictionary entry created; 7. the importance of a Romanian terminological corpus.

Key words - corpus, dictionary, etymology, language, meaning, neologism, influence.

1. Introduction

Most authors talk about specialized terms from linguistics and terminology as an independent science point of view.

The Romanian language and the society updated starting with 1830-1860. There was a long and complex process which required a great effort of writers, scientists, scholars in different Western European countries. Since then just an elementary vocabulary of different sciences had been created, but the basis of Romanian modern terminology was set out in the above-mentioned period.

Cultural staff of so-called “Şcoala Ardeleană” played a big role in Transylvania. Romanian intellectuals, who were also specialists in modern Greek, Russian, French, German, Italian cultures and languages, had an intense activity in Wallachia and Moldavia.

N. A. Ursu studied the forming of Romanian modern terminology based on a great number of old manuscripts, handbooks, journals, magazines, books, etc. edited between the 18-th century and 1860. His works highlighted the way our intellectuals created the scientific and technical language and their contribution to the Romanian literary language development. Byck (1954) and Ivănescu (1955) identified such new words in documents, calendars, religious texts, the 17th century chronicles and of course Dimitrie Cantemir's writings. The latter used a lot of Latin words as an important step of language evolution and suggested his contemporaries to follow his

example. Even if Cantemir's scientific and philosophic work contained a basic terminology of these fields, only a few people read it.

Due to our press and instruction development, lots of words have been borrowed through modern Greek and Russian between the end of 17th century and the first part of the 19th century.

Romanian scientific terminology knew three periods during its modernization process, taking into consideration Ursu's classification (Ursu, 1962). According to this classification, the Romanian scientific terminology went through three periods during its updating process.

1780-1830 Romanian people assimilate modern science and become familiar with scientific trends; 1830-1860, Romanian scientific terminology constitution; 1860-1870, original specific works have appeared and their authors used a rich vocabulary and a clear and precise style.

2. The origin of Romanian scientific terms

Romanian terminology includes an important quantity of loans from different languages and less Romanian achievements.

In the early 1800 our specialized vocabulary numbered some loans from modern Greek and Russian (in Moldavia and Walachia) and some caught up from Latin, German, Hungarian and Italian (in Transylvania). Scientific terms come from translations and national books. We can talk about a parallel influence of French and modern Greek after 1830, but after this period, Greek lexemes were totally replaced by the French ones. The Latin influence in Romanian provinces is a cultural moment justified in different ways. In Moldavia and Wallachia Latinism had not a big impact, it was only an isolated tendency and tried to stop loans from entering the Romanian language. Political and social contexts in Transylvania explained the importance of the Latin influence. Intellectuals of this province fought for independence and this was the main reason of their trying to prove the Latin origin of the Romanian people.

The Russian influence is considerable from 1800 to the second part of 19th century and it was present in the military, technical and administration vocabulary. The Romanian language got some international words through Russian too. The German influence was a new process in our provinces around 1830 and that was possible due to handbooks.

Romanian loans have an interesting characteristic if their form is studied. The same item has two or more phonetic and morphological aspects because it was borrowed from different idioms (e. g. *acid* was taken from Italian, but it has also got the form *aţid*, due to German and Russian provenience). Romanian linguists call this fact multiple etymology to explain each of these forms.

3. DLR vs. DEX

3.1. Lexical inventory building

DLR and DEX lexical inventories include various specialized terms from all fields of human knowledge (science, culture, arts, sport, etc.). A word is a dictionary entry if it is used at least in two different styles of language. Most of the dictionary entries fulfill this criterion, but it is not absolutely compulsory. A word which is frequently used has to appear in a dictionary even if it is attested only in a work. Its use requires its presence in a dictionary word list. DLR is more restricted than DEX from this point of view because it has quotations. DEX lexicographers do not introduce quotations so they can take into consideration more new words than DLR's lexicographers.

3.2. Lexemes list

Both mentioned dictionaries need to be enriched from the lexical inventory point of view. IT, medicine, chemistry, physics, biology, I mean, sciences and technology generally speaking, develop in a high pace so more words are created, but neither DEX nor DLR did care about them. No dictionary is exhaustive because it would be impossible, but it should be as comprehensive as possible.

We are a DLR and a DEX team of lexicographers whose current task is to work on the letter A. DLR bibliography is richer than the previous one and contains more terms from different fields of activity. At the moment there are not enough testimonials for specialized terms. We got old terms proofs in written texts from the 19th century. *Acid's* first proof appeared in 1840 in NEGULICI's lexicon. After that date lots of proofs have been found in books, handbooks, courses, dictionaries, glossaries, lexicons, encyclopedia, journal and magazines, articles, etc. These works are important because they were done by specialists who provided concise and clear definitions of words and idiomatic expressions, too.

3.3. Selection of explanatory contexts in DLR

DLR is a thesaurus so it is an explanatory, historical and etymologic dictionary which offers an image of a word evolution only due to contexts added to definition. In the beginning linguists wanted to collect quotations only from Romanian authors' writings so translations have been let out because they were not original works. Translators used neologisms because either they did not want to look for their Romanian equivalents or they were not experienced users of our language. That is why Sextil Puşcariu, 1910, left out dictionaries and translations. The authors of the first works were created words which did not exist and invented meanings. Such errors could be found in foreign language dictionaries, so Puşcariu was very restrictive with the loan words in Romanian.

Was this criterion applied by our lexicographers? Is this solution available nowadays? Of course, not. Lexicographers have taken into consideration translations and dictionaries, too because they needed attestations for lexical loans. Dictionaries have been used and they are still used as attestation sources because some words appear only there. That is why the data from LB[1], Tiktin[2], HEM[3], DAMÉ[4], CADE[5], SCRIBAN[6] and other monolingual and bilingual dictionaries are still very precious.

3.4. Words meanings in DLR

Specialized terms are usually mono-semantic and very seldom poly-semantic. Lexicographers need more than one word to explain such terms. If an author knows what a lexical unit means this can provide a clear and synthetic definition. On the contrary, a lexicographer who is not familiar at all with such kind of lexeme has to study it very carefully and to check all the information found before explaining its senses and selecting the most suitable excerpts.

3.5. Relation between the meaning and the first proof

DLR is a thesaurus, so there are historical, etymological, grammatical and lexical information. Every DLR entry starts with the oldest meaning of the head-word (the etymon's one). This rule is not absolutely compulsory in the case of specialized terms. DEX entries contain definitions, taking into consideration words frequency, grammar and etymology sections. DEX's lexical inventory has permanently been enriched from edition to edition.

4. Lexicographic definition

Lexicographic definition is another problem in Romanian dictionaries. DLR lexicographers design each lexical entry meaning scheme and add to it related contexts. Some specialized terms have evolved and they belong to slang area, other have got connotations due to users' invention capacity. Editors must make short, clear, concise, correct definitions and this might be sometimes difficult. Some users' complaint referred to DEX and DLR definitions dimensions and sophistication compared to the English ones. This statement is only partially correct. Those claimed English dictionaries were made for foreigners who learn this language, so they must be very clear and concise. English and Romanian languages are completely different. A notion needs only a word or a short sentence to be explained in English. The same thing needs a long sentence and some synonyms to be expressed in our language. Romanian lexicographers try to update and recompose some definitions for more

[1] *The Lexicon of Buda* (1825).
[2] Tiktin (1911-1913), *The Romanian-German Dictionary*.
[3]Hasdeu (1886-1889), *Etymologicum Magnum Romaniae.*
[4] Damé (1900), *The New Romanian-French Dictionary*.
[5] Candrea, Adamescu (1926-1931), *Encyclopaedic Illustrated Dictionary.*
[6] Scriban (1939), *The Dictionary of the Romanian Language.*

clarity and a modern image to their dictionaries. For this purpose they frequently use foreign dictionaries, English and French.

5. Head-word and its forms

These aspects are seriously discussed by linguists in general and especially by lexicographers. It is very difficult to decide the best title-entry of a loan. Loans of the 19th century have more forms justified by their different origin language. Latin forms have been preferred for a great number of neologisms. For instance, a loan attested for the first time in a text from Transylvania (a translation from German) has also got a Latin form. Lexicographers have considered it was the standard form and the others as its *old* variants. There is a relationship between the head word and its first attestation. It is quite impossible to determine the first attestation of the most recent loans in Romanian. After 1990 lots of neologisms of English or other origins were caught up in Romanian language. Speakers are very familiar with some words and their idiomatic expressions even if they have not been attested in any written text mentioned by DLR bibliography. International words are a good example of this statement.

6. Why is a lexicographic corpus useful?

Romanian academic dictionaries need permanently updated lexical inventories because science and society have developed and vocabulary has been enriched with some notions which are expressed in specific terms. Even common words created new meanings, idiomatic expression or become part of different types of terminology.

On the other hand old manuscripts and books are a real treasure for linguistics and terminology field even if it was not sufficiently studied. Here are some benefits of old texts research.

1. Terminology inventory can be enriched with new items.

DEX and DLR editors should compare their word lists in order to have more items. Ex. *acid* appears in DEX and DLR, but they have only, a short number of collocations for adjectival use, namely the following: DEX has registered only the following phrases:*acid acetic, acid acetil-salicilic, acid adipic, acid arsenic, acid arsenios, acid ascorbic, acid azotic, acid azotos, acid azotos, acid barbituric, acid benzoic, acid bibazic, acid boric, acid butyric, acid carbamic, acid carbolic, acid carbonic, acid cianhidric, acid citric, acid clorhidric, acid fenic, acid floric, acid formic, acid fosforic, acid galic, acid gras, acid iodhidric,*etc.

DEX and DLR's authors have taken into consideration some constructions: *acid lactic, acid monobazic, acid naftenic, acid nitric, acid nucleic, acid oleic, acid oxalic, acid palmitic, acid picric, acid pirogalic, acid pirolignos, acid prusic, acid racemic,*

acid ribonucleic, acid salicilic, acid silicic, acid stearic, acid sulfihidric, acid tanic, acid tartric, acid tribazic, acid uric, acid valearinic.

Expressions like *acid margaric, acid muriartic, acid nativ, acid organic, acid salitric* were found in DLR only. The following structures are missing from both mentioned dictionaries: *acid albastru, acid-aldehidă, acid ambric, acid aminocapronic, acid animal (mineral sau vegetal), acid azothidric, acid boracic, acid biliar, acid camforic, acid cerenic, acid de cameră, acid de fierbere, acid de oţet sau oţetos, acid de pucioasă, acid de silitră, acid de turn, acid diallybarbituric, acid dietilbarbituric, acid d-lisergic, acid fiericianhidric, acid folic, acid glutamic, acid Haller, acid hialuronic, acid hidrosulfuros, acid hipoazotic, acid homovanilic, acid humic, acid italic, acid mineral, acid mort, acid paraaminobenzoic, acid paravinic, acid pirosulfuric, acid propionic, acid racemic, acid salis, acid sulfionic, acid uvic, acid vegetal.*

Even if neither DLR nor DEX is a Chemistry term lexicon, the phrases we have already mentioned should appear in a corpus.

2. First proofs identification and excerpts number enlargement in the case loans which are attested in dictionaries only.

3. A corpus provides new meanings and phrasal structures of dictionary entries. Some phrasal expressions are just mentioned in the first lexical element entry, but defined in the second one. There are also exceptions from this rule in DLR and DEX.

4. Contexts in a corpus help linguists understand morphological, syntactical and semantic evolution of words.

Last examples made us conclude that *acid* is a term with a multiple origin in our language because more variants of this word were taken from different languages (German, Latin, French, and Italian). Each of its variants has got a specific form and there are more feminine and masculine singular and plural forms. *Acid* appears in phrasal structures. Some of them are still in use. Those with a hilarious form were rejected by the speakers.

5. English terms' phonetic and morphological adaptation to Romanian language particularities.

Linguists agreed that Romanian speakers had to use English words form and pronunciation as the best solution. It would be too difficult to adapt these words to our language rules. In addition, Romanian speakers are very familiar with English words form.

6. Specialized terms status in our language must be analyzed in order to see which one is still used, which one was replaced by an international word. A linguistic approach could reveal “senior neologisms“ semantic enlargement or restriction, their new idiomatic expressions and of course their presence in slang, or other language styles.

7. Correct accessible definition of terms.

7. Beneficiaries of a lexicographic corpus and its structure

These are not only linguists but translators, glossary and lexicon authors and those who are interested in Romanian language study. They will need a very comprehensive corpus containing literary and non-literary texts collected from all Romanian Geographic areas. Old texts must be digitalized to help researchers find that information they need. At the moment there are some electronic form texts, but those written in Cyrillic and transition character (before 1860) are typed and annotated by hand only. An electronic corpus should contain documents from 16th-21st centuries (books, papers, journals and magazines). The majority of neologisms were taken from French two centuries ago and their users were using them very often. Unfortunately, DLR bibliography contains just a small number of such texts which interest linguists.

Scientific and technological works (like magazines, papers presented by researchers at conferences, colloquia, summer schools) in electronic format or published on the internet sites should not be neglected. Dictionaries are good reference source in the case of the latest '90 and early 2000 neologisms. They are almost the only works attesting such new lexical items. Users look for them in dictionaries because they want to find out their meanings, correct forms, origin, plural forms, verb tenses form, phrasal structures and domain.

A corpus creation involves texts scan, characters recognition, typing Cyrillic texts and correcting the mistakes in all cases. Such an important project cannot be done without a specific software, professional scanners and of course a real collaboration between scientists and linguists.

8. Conclusions

This paper focused on the following aspects: 1) a history of modern Romanian terminology; 2) a comparison between DLR and DEX; 3) some reasons and ways to build a lexicographic corpus.

1) Romanian terminology contains an important quantity of loans from different languages and less Romanian terms. In the early 1800s our scientific vocabulary had only some loans from modern Greek and Russian (in Moldavia and Walachia) and others caught up from Latin, German, Hungarian and Italian (in Transylvania). Scientific terms appear and spread in translations and national books. We can talk about a parallel influence of French and modern Greek after 1830, but after this period, Greek lexemes were totally replaced by their French equivalents.

Latin influence in Romanian provinces was a cultural process justified by different reasons. In Moldavia and Wallachia Latin purism was only an isolated situation and tried to stop loans enter the Romanian language. Political and social context in Transylvania explained the importance of the Latin influence. Intellectuals of this

province fought for independence and this was the main reason why they tried to prove our people's Latin origin.

Russian influence was stronger from 1800 to the second part of 19th century and obvious in the military, technical and administration vocabulary. Romanian language got some international words through Russian idiom too. German influence was a new process in our provinces around 1830 and that was possible due to handbooks. Romanian loans have an interesting characteristic if their form is studied. The same head-word has two or more phonetic and morphological forms because of its indirect provenience (e. g. *acid* was taken from Italian, but it has also got the form *aţid*, due to German, Russian provenience). Romanian linguists called this fact multiple etymology to explain each of these forms.

2) The comparison between DLR and DEX revealed their similarities (a lexical inventory, semantic, grammatical, etymological information, use information, variants, spelling and pronunciation information, etc. in DLR and DEX) and differences (some contexts from the 16th to 20th centuries works in DLR). We have also made some suggestions which are very useful and necessary during the reprint process of these dictionaries.

3) Why is a corpus useful to lexicography?

1. Romanian academic dictionaries need a lexical inventory update and enlargement as a consequence of scientific and technological terms development. Users are familiar with these words, but they are still missing from our dictionaries (DEX and DLR). Old manuscripts and books might offer linguists valuable information about terminology, too.

2. First proofs identification and citations number enlargement in the case of neologisms attested only in dictionaries.

3. Identification of new meanings and more idiomatic expressions in order to be included in our dictionary entries.

4. Contexts in a corpus help linguists understand morphological, syntactical, semantic evolution of the words.

5. A modern etymological approach on those neologisms collected by our language from different idioms (in the 18th and 19th centuries). For instance the word *acid* is an indirect loan from German, Latin, French and Italian languages. This fact justifies the presence of more variants of the same word and these variants contain particularities in the language of origin. *Acid* graphematic variants are: *acid* and *aţid*, and its plural forms are *acizi, acide, aţide, aţiduri*. *Acid* appears in phrasal structures; some of them are still used, but others disappeared.

6. English terms are used in Romanian dictionaries with their genuine form. To adapt an English word to Romanian grammar and phonology rules is a difficult and unsuccessful process.

7. Old specialized terms in our language should be studied in order to see either they are still used or they have been replaced by international terms taking into

consideration vocabulary evolution, meaning change, appearance and disappearing. Some scientific terms have achieved figurative meanings or even turned to slang ones.

8. A corpus helps lexicographers to make lexicographic definitions more modern and clearer due to information provided.

9. A corpus beneficiaries are not only linguists but translators, glossary and lexicon authors or everyone interested in Romanian language study. They will need a very comprehensive corpus of texts written in all the Romanian areas during the past and present moments. Researchers need digitalized texts. At the moment there some electronic form texts in Latin alphabet, but those in Cyrillic and transition character ones until 1860 must be typed and annotated by hand only. Such a corpus should contain documents from 16th-21st centuries, journals and magazines, scientific and technological works (like magazines, papers presented by researchers at conferences, colloquia, summer schools) in electronic format or published on the internet sites. Dictionaries offer sometimes the first attestation of a recent neologism in our language. Users look for words meanings, plural and verb tenses form, phrasal structures, etymology and domain. A corpus building process means texts scan, characters recognition, typing Cyrillic character texts for the moment, correcting the mistakes in all the cases. Such an important project cannot be done without an important budget for professional tools (software, computers, scanners, OCR programs) and a real collaboration between scientists and linguists.

References

Byck, J. (1954). *Vocabularul ştiinţific şi tehnic în limba română din secolul al XVIII-lea*, în *Studii şi cercetări lingvistice*, V: fasc. 1-2, p. 31.

Candrea, A. I., Adamescu, Gh. (1926–1931). *Dicţionarul enciclopedic ilustrat*. Partea I–II. Bucureşt: Editura „Cartea Românească.

Damé, Fr. (1900). *Nouveau dictionnaire roumain-français. V^ème volume, conprenant le lexique roumain-français et français-roumain de la terminologie paysanne.* Bucarest: Librairie Socecu & C^ie.

Dicţionarul explicativ al limbii române (2016). Ediţie revăzută şi adăugită. [Tiraj nou].Bucureşti: Editura Univers Enciclopedic. Academia Română. Institutul de Lingvistică „Iorgu Iordan – Al. Rosetti”.

Dicţionarul limbii române (2010). Ediţie anastatică după *Dicţionarul limbii române* (DA) şi *Dicţionarul limbii române* (DLR). Tom I–XIX. Bucureşti: Editura Academiei Române.

Hasdeu, B. P. (1887, 1893). *Etymologicum Magnum Romaniae. Dicţionarul limbe istorice şi poporane a românilor.* I–III. Ediţie îngrijită şi studiu introductiv de Grigore Brâncuş, 1972-1976, Bucureşti: Editura Minerva.

Ivănescu, G. (1955). *Terminologia filosofică românească până la anul 1830*, comunicare ţinută la Institutului de Istorie şi Filologie „Al. Philippide“ din Iaşi, în 1955.

LB (1825). *Lesicon românesc-latinesc-unguresc-nemţesc, care de mai mulţi autori, în cursul a trideci şi mai multor ani s-au lucrat* seu: *Lexicon valachico-latino-hungarico-germanicum quod a pluribus auctoribus decursu triginta et amplius annorum elaboratum est.* Budae: Typis et Sumtibus Typographiae Regiae Universitatis Hungaricae.

Scriban, A. (1939). *Dicţionaru limbii româneşti*. (Etimologii, înţelesuri, exemple, citaţiuni, arhaizme, neologizme, provincializme). Ediţiunea întâia. Iaşi: Institutu de Arte Grafice „Presa Bună“.

Tiktin, H. (1911-1913). *Rumänisch-Deutsches Wörterbuch*, 2., überarbeitete und ergänzte Auflage von Paul Miron. [Band I–III: 1986-1989]. Wiesbaden: Otto Harrassowitz.

Ursu, N. A. (1965). *Formarea terminologiei ştiinţifice româneşti*. Bucureşti.

A BOOSTRAPPING SYSTEM FOR DICTIONARY MANAGEMENT AND PARSING

MIHAI ALEX MORUZ[1,2], DAN CRISTEA[1,2]

[1] *Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iasi,*
[2] *Institute of Computer Science, Romanian Academy, Iasi Branch*

{mmoruz, dcristea}@info.uaic.ro

Abstract

Electronic lexical resources are the base of various NLP tools (POS taggers, chunkers, NERs, etc.). Out of the largest Romanian language dictionary, "Dicționarul Limbii Române" (DLR), a first XML version was obtained during the eDTLR project. Dictionary Entry Parsing is the process through which structured formats are obtained out of complex dictionary entries presented in input in electronic text formats. At the moment, the electronic version of the DLR, the eDTLR, is undergoing a complete reparsing, based on new input data, which will enhance the quality and remove possible missing parts. This paper describes an approach based on a Content Management System for managing parsed entries, both during the reparsing process and after, for public exposure. The system will manage the current status of all recognized entries of a dictionary (unparsed, partially parsed, successfully parsed and parsed with errors), will be capable of calling successive steps of the parsing process (in order to process new volumes or to reparse entries) and will offer a secure way in which researchers can search for entries, create complex queries and view custom statistics.

Key words — Dictionary Entry Parsing, Dictionary Management System, Content Management System

1. Introduction

During recent years, the number and complexity of Natural Language Processing tools, as well as the need for such tools, have steadily increased. Because of this, the requirement for linguistic resources has increased at a similar rate, bringing to the forefront the issue of digitizing existing resources, in order to make them compatible with the tools and processes currently in use.

One such very large linguistic resource is made up of printed dictionaries, built over the course of many years, by large teams of researchers, and generally edited in an unstructured format. The largest such dictionary for the Romanian language is the "Romanian Language Thesaurus Dictionary" (*Dicționarul Limbii Române* – DLR), made up of more than 175.000 words and variants, written over a span of more than one hundred years, by hundreds of lexicographers. The usefulness of this dictionary

for the field of NLP is unquestionable, as it is the most detailed dictionary for the Romanian language to date, describing not only word senses (which are represented in a tree-like logical structure), but also assigning a large number of examples for each sense (sentences containing the word sense in question extracted from a large set of bibliographical references, covering most of the known uses of that sense over the years). Since the elaboration and publishing of the DLR started more than 100 year ago, most of the volumes published are only available in printed format; in more recent years, some volumes have been edited electronically, but the electronic format was more concerned with the surface form rather than on strictly keeping with norms out of which a structured form can be deciphered, although a tree-like structure of sense entries was always present.

2. eDTLR

During the eDLTR project (Cristea *et al.*, 2007), a team of researchers has attempted to transform DLR from a mostly printed form dictionary to a structured, electronic form dictionary, by doing Dictionary Entry Parsing (DEP). DEP is the process by which, given an unstructured dictionary entry or volume, the sense tree of each entry is automatically obtained on the basis of the various typographical markers which are used to code lexicographic information. For the eDTLR project, a new type of DEP was proposed (Curteanu *et al.*, 2008), which was eventually refined into the DSSD algorithm (Dictionary Sense Segmentation and Dependency – (Curteanu *et al.*, 2010), (Curteanu *et al.*, 2012), (Curteanu *et al.*, 2013)). This algorithm is based on the observation that many thesauri can be parsed by following the same three basic steps:

- Segmentation, either at the entry level or the sense level. This assumes that the input text, although not marked explicitly for structure, is formatted (e.g. bold, italic, superscript, etc.) and that there is a consistent manner of typographically marking senses and entries.
- Dependency assignment, which is to say that the senses identified above can be organized in a tree structure for each entry, by exploiting typographical markers.
- Parsing of atomic definitions. Each sense item (a sense item is a node in an entry sense tree) is generally made up of the marker identifying the sense and the associated gloss for that sense; it can also contain some other information such as examples, citations, bibliographical references, morphological information, etc. All of these items need to be identified so that the information available in the dictionary is fully utilized.

The DSSD algorithm is robust, in the sense that, whatever the entry and whatever errors there may be (in terms of marker usage, scanning quality, etc.), the parsing procedure always returns a result, even if it leads to only a partial structure. For example, given a malformed entry with a correctly represented sense tree but with incorrectly formatted atomic definitions, the DSSD will correctly perform sense segmentation and will also correctly create the associated sense tree; in those cases

where the atomic definitions are correctly represented, they will be parsed without error, and the algorithm will only leave as unparsed those areas which are problematic. This is in contrast to previous DEP algorithm, which usually employed an all or nothing approach, which discarded many of correct parses due to small errors (Curteanu *et al.*, 2008), (Curteanu *et al.*, 2010).

Although the DSSD algorithm is robust and highly accurate (Curteanu *et al.*, 2008), its performance is dependent on the quality of the input data. This means that, in the case of incorrectly formatted entries (e.g. text that should be bold is actually formatted in some other way, incorrectly recognized letters or paragraphs, etc) or an error induced by the OCR process, it is possible that the entry and sense markers will not be recognized, which leads to errors in parsing. This can be solved through the correction of the input data, either by using a better OCR or by applying heuristics able to deal with incorrect input, in order to remove systematic errors. During the eDTLR project, the DLR was completely parsed using a DSSD type algorithm, but, for reasons explained, the parsed version still contains errors such as incorrectly parsed entries, entries that are incomplete or even missing, and entries that have been mistakenly recognized as such (Cristea *et al.*, 2011). Many of the issues mentioned above can be solved by improving the quality of the input data by various means and then reparsing that data.

In order to guarantee that the correction and reparsing process is complete and does not miss any entry, it has to be properly managed so that completeness is guaranteed, and correctness adheres to the minimal standards imposed by the authors of the dictionary. Also, once completeness and correctness are satisfied (as described above), the electronic DLR needs to be made available to the public by means of an online platform.

This paper will describe a bootstrapping system for the management of the parsing process of a dictionary (the dictionary in question is the DLR, but the system can be used for the parsing of any dictionary; it is a bootstrapping approach because it allows for incremental improvements as the parser improves) and its subsequent distribution via an online portal.

3. Using Content Management Systems for Managing DEP

3.1. XWiki

One of the important collaborative web based systems is the Wiki, which allows for the storing and structuring of knowledge by using Web technologies (Dumitriu *et al.*, 2007). Generally, a Wiki system offers a web based interface for collaborative editing where the „main facilities are the simplified syntax, the rollback mechanism, the (possibly) unrestricted access, several search functions, the support for uploading content." (Dumitriu *et al.*, 2007). Currently, wikis are used in various online applications such as encyclopaedias, content management, software development, collaborative wiring, etc (Schaffert *et al.*, 2005).

Since, for an electronic dictionary, it is more relevant to describe relations between entities rather than marked up text, a wiki type platform fits well the purpose. Also, the collaborative feature facilitates users' access to the parsed data, allow experts to report various issues with respect to the parsed form of certain elements and even operate changes. Moreover, since such a platform allows extensive scripting, the parsing algorithm can be imported in the Wiki application and then called from within the platform, thus permitting a bootstrapping approach to dictionary entry parsing.

According to (Dumitriu *et al.*, 2007), the main advantages of using a wiki type system for the parsing and management of an electronic dictionary are based on the following:

- Each of the internal templates of the system is defined as a class, which has an attached semantic meaning. The properties of the class (in practice, class attributes) have attached semantic values, which can be either used locally, on the platform, or exported as semantic information (e.g. as an RDF/OWL format);
- Information is reused and not replicated. In a wiki, information can be referenced from any point of the system (provided that the user has the rights to access that specific type of data) by means of a simple syntax;
- Because the information is rigorously structured and stored, it is easy to retrieve dictionary entries. Moreover, the query system allows for complex searches, as one can aggregate, extract and compare information (e.g. complex searches such as "all nouns attested between 1500 and 1700, which are imported from Slavonic, and have at least two synonyms for each sense")
- Information can be collaboratively maintained, allowing the lexicographers to use the platform for extending, updating and correcting the dictionary entries. Also, because a wiki allows for multiple publishing formats, the same entry can be shown in a user friendly manner or exported in RDF for use in another tool or application.

The wiki version we have chosen for the described system is XWiki (www.xwiki.org). XWiki is a second generation wiki (Structured and Applications Wikis) and can be used to create collaborative web applications, collaborative editing, content management, access control and layered access, semantic modelling of concepts, etc. It requires a container server (Apache Tomcat, Glassfish, etc.) and a relational database connection; it is written in Java as an open source project, and, thus, can be easily extended and modified for any required task. It offers support for a large number of scripting languages such as JavaScript, Groovy, Velocity and others.

3.2. Managing DEP

In this model, all dictionary entries are instances of a given XWiki class, which contains references to the various stages of parsing, state the current step of the parsing process the entry is at, and specifies access rights and the physical volume from which the entry has been extracted.

Since the parsing process is performed in stages, each entry contains a reference to an object representing each stage: unparsed base text of the entry, sense separation and dependency tree, senses with parsed atomic definitions (fully parsed entries). Figure 1 below shows the structure of the class which manages an entry, from the raw input text to the final parse.

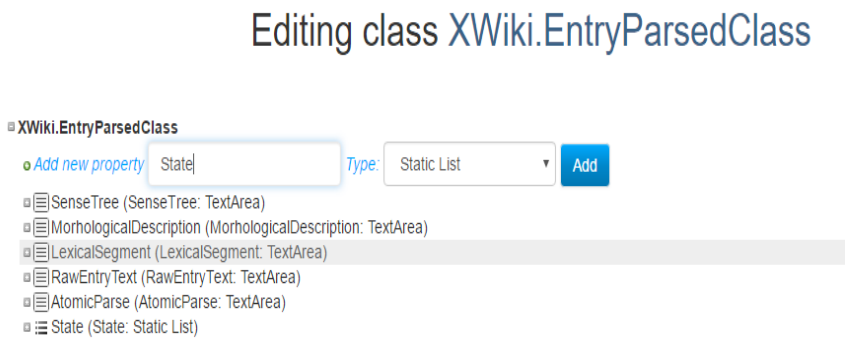


Figure 1. Creating classes in XWiki

Given the class structure described above, each identified entry is an instance of that class. As the entry is further parsed (sense tree identification, atomic definition identification, extraction of morphological and lexical information), the corresponding attributes of the object are filled in, and the **State** attribute is changed from `Unparsed` to `senseParsed` (partially parsed) and finally to `parsed`. The attribute fields contain XML code, firstly because we pass the entire attribute as input to the parsing process, and, secondly, we can attach XSD code to each attribute in order to display or export the information.

Figure 2 below shows a fully completed parse for the dictionary entry “DOMNÍ”. The entry is given in the edit mode, so that all attributes can be seen on the same screen.

Figure 2. Xwiki object for the entry "DOMNÍ"

As new entries are added (and new objects of the type described above are created), the system automatically updates the list of all recognized entries; also, as the parsing system passes over each entry and advances the state of the object, the list is duly modified. This management of the stored entries is necessary firstly because we need a quick way of determining the state of parsing for the entire dictionary, and also because, to our knowledge, no comprehensive list of all entries in DLR is yet available.

The parsing steps (entry separation, sense segmentation, atomic definition identification) have each been attached to a page in the Wiki, and, by using a script, can be called at any moment during entry parsing. Given that each entry is instanced with, at the very least, the raw input data, the procedure calls do not require any argument other than the Xwiki page containing the object in question.

4. Managing Access to the Parsed Data

Although DLR is the largest and most detailed dictionary for the Romanian language, access to it is extremely difficult because of its extremely large size and the fact that, at the moment, it is only available in printed format. The XWiki platform offers a medium in which we can make available the parsed entries of the dictionary in a format that allows complex interrogations, customizable viewing of the parsed data, collaborative editing for experts, layered access rights, etc.

4.1 Access Rights and Collaborative Editing

The Xwiki platform allows for the creation of user groups, each with different access rights and privileges; also, a specific user can be part of more than one group, allowing for further customization of access. Finally, each user also has individual access rights, which override any other access settings. This allows for the creation of groups on the basis of access desired: a normal user who only needs to view entries will be placed in one group, while authors of the dictionary and other experts will be placed in another group, which will allow editing, for example. Furthermore, in the case of partially parsed entries, access of normal users is not desired, but the access of experts is compulsory, especially for those entries which are yet incorrectly parsed.

Each page can be edited by any of the users with appropriate rights, but only one user can edit the page at any given time, as opening the page in edit mode locks that option from all other users in order to prevent inconsistencies. At all times, a complete version history is kept, allowing for version comparisons or reverting to previous versions.

4.2. Customizing Views

Since the data stored in the class attributes is in XML format, XSD transformations can be applied to it in order to customize its appearance. Using the XWiki class sheet associated to the entry class, we can define a default view for the fields of the entry, which can be changed by modifying the associated transformation files. Each object can also be exported as a PDF or HTML file, but, because of the manner in which query results are returned in XWiki, only one page can be exported at a time (each object is an XWiki page).

4.3. Complex Interrogations

XWiki allows for complex queries over the object of the application, using an internal query language similar to SQL, called XWiki Query Language (XWQL). This language is currently supported without programming rights, which means that any user can create a personal query and runt it by her/himself. Also, a general interrogation form can be created in order to offer a visual interface for XWQL.

XWQL allows for the interrogation of any document or document field, but the returned results will only contain those pages which are accessible to the user. In this manner, those entries which need to be hidden from normal users, for whatever reason, are indeed kept hidden even during complex searches.

5. Conclusions and Future Work

In this paper we have described a new manner for managing dictionary entry parsing on the basis of a content management system. For previous attempts of dictionary parsing (the eDLTR project, for example, such a dedicated management system was not used). The system allows for customizable classes, semantically linkable, for exportation of information in rich text format or RDF/OWL syntax, and for uses of a wide array of scripts. User access is itself customizable, with multiple access layers on the groups and individual users. Also, the system has a built-in query engine, similar in power to the SQL query language, which allows for personalized queries from users.

The system is highly flexible and extensive, and, as such, can be used for the management of any parsed dictionary. It can also be used as the basis for new dictionaries, since the semantically encoded information associated with an entry can be transformed or simplified at will. Experts can use the platform collaboratively, in order to improve and extend the stored dictionary (in this case, eDTLR) and can also add new semantic information (such as extra dictionaries or lexicographical resources) as new knowledge bases for improving DLR. These knowledge bases can be accessed using various scripts and subroutines; for example, in the case of a DLR entry, a script can recover a list of potential quotations, ordered chronologically, from the oldest to the most recent ones, or that fit a specific interval of time.

Acknowledgements

The research described in this paper was funded by the "Alexandru Ioan Cuza" University of Iasi, through the grant GI-2015-14, no. 17/03.12.2015

References

Cristea, D., Răschip, M., Forăscu, C., Haja, G., Florescu, C., Aldea, B., Dănilă, E. (2007). The Digital Form of the Thesaurus Dictionary of the Romanian Language. In *Proceedings of the 4th International IEEE Conference SpeD 2007*.

Cristea, D., Haja, G., Răschip, M., Moruz, A., Pătrașcu, M. (2011). Partial statistics on eDTLR-Thesaurus dictionary of the Romanian Language in electronic form (Statistici parțiale la încheierea proiectului eDTLR – Dicționarul Tezaur al Limbii Române în format electronic). In *Acta of the Conference "Romanian Language: hypostasis of linguistic variation, 3-4 December 2010"* (*„Limba română: ipostaze ale variației lingvistice, 3−4 decembrie 2010*"), Romanian Language Department, University of Bucharest.

Curteanu, N., Moruz, A., Trandabăţ, D. (2008). Extracting Sense Trees from the Romanian Thesaurus by Sense Segmentation and Dependency Parsing. In

Proceedings of the Workshop "CogAlex-I", COLING 2008, Manchester, United Kingdom, 55-63, ISBN 978-1-905593-56-9.

Curteanu, N., Trandabăţ, D., Moruz, A. (2010). An Optimal and Portable Parsing Method for Romanian, French, and German Large Dictionaries. In *Proceedings of the Workshop "COGALEX-II"*, COLING-2010, Beijing, China, August 2010, 38-47.

Curteanu, N., Moruz, A. (2012). Toward the Soundness of Sense Structure Definitions in Thesaurus-Dictionaries. Parsing Problems and Solutions. In *Journal of Computer Science of Moldova*, Academy of Science of Moldova, vol. 20, no. 3(60), 275 – 303.

Curteanu, N., Moruz, A., Cojocaru, S. (2013). Formalization of a General SCD-based Parser for Dictionaries Using Parametrized Grammars. In *Proceedings of the International Conference on Intelligent Information Systems IIS 2013*, August 20-23, 2013, Chisinau, Moldova.

Dumitriu, S., Gîrdea, M., Buraga, S. (2007). Knowledge Management in a Wiki Platform via Microformats. In *Proceedings of FLAIRS 2007 Conference* , AAAI Press, 2007.

Schaffert, S.; Gruber, A.; and Westenthaler, R. (2005). A Semantic Wiki for Collaborative Knowledge Formation. In *Proceedings of Semantics* 2005.

AUTOMATIC MERGING OF MARKED UP TEXTS FOR DICTIONARY ENTRY PARSING

MIHAI ALEX MORUZ[1,2]

[1] *"Alexandru Ioan Cuza" University of Iasi, Faculty of Computer Science*

[2] *Institute of Computer Science, Romanian Academy, Iasi Branch*

mmoruz@info.uaic.ro

Abstract

Dictionary Entry Parsing is the process by which dictionaries, given in an unstructured rich text format are transformed into structured information. This transformation attempts to extract such information as title word, gloss, examples, bibliographical information, etc. Such structured information is a vital step towards the indexing of dictionary content, which would allow for complex interrogation and usage. Several large dictionaries have already been transformed into such a format – DEX (Explicative Dictionary for Romanian), TLFI (Tresor de la Langue Francaise), DWB (Deutsches Woerterbuch), etc., but most of this work has been done manually, at great expense. One of the more successful automatic approaches has been that employed for the transformation of the DLR (Thesaurus Dictionary for Romanian) implemented in the eDTLR project (Cristea *et al*., 2007). The approach proposed, described in detail in (Curteanu *et al*., 2008), was successfully employed for the Romanian, French, German and Russsian Thesauri, proving that given a properly formatted and represented input text, the precision of the parser exceeds 90%. However, this is entirely dependent on an electronic format that is correct in both form and content. The precision for the parsing of DLR was low because the input RTF text was obtained by means of OCR, which generated significant errors in formatting and content. While many of the content errors were manually corrected, most of the formatting errors remained, which greatly hindered the parsing process. More recently, the scanning and OCR process for the DLR has been undertaken again, with much better results in terms of formatting but without any manual corrections. In this paper we propose an automatic method of merging the two versions of the marked up text representing DLR entries by keeping most of the manually corrected text (where it exists) and also keeping the formatting information obtained after the more detailed scanning and OCR, while also guaranteeing the correctness of the XML format thus obtained.

Key words — Dictionary Entry Parsing, Markup Language, XML merging

1. Introduction

The Romanian Thesaurus Dictionary, created by the Romanian Academy, has been published in two series: the Academy Dictionary (DA), 1913-1949 (A-C, D-De, F-K, L-Lojniță) and the Dictionary of the Romanian Language (DLR), published from 1965, which contains the rest of the letters. As a general dictionary, it covers most of the written word forms which have been found in popular, artistic or scientific writings. Within the eDTLR project (Cristea *et al*., 2007), this dictionary has been automatically transformed to a structured electronic format (Curteanu *et al*., 2008), (Curteanu *et al*., 2010), (Curteanu *et al*., 2012), (Curteanu *et al*., 2013). The parser described in the referenced papers relies, for the most part, on the recognition and classification of typographical markers (bold, italic, symbol sequences, etc.).

The input data for the parser comes from two distinct sources: electronically edited volumes and volumes that have been scanned, OCR-ed and then manually validated. The electronic volumes are available as DOC format documents; they contain, apart from the necessary typographical information, a set of annotations that are useless for the purposes of automatic parsing (text language information, formatting of spaces, graphical elements, uppercase letters that have been transformed to lowercase, etc.). The scanned volumes are not available in an electronic format because they were edited and printed before the widespread use of personal computers. Because of this it was necessary to scan them in order to have an electronic format. The result of the scan and subsequent automatic recognition of the text are prone to errors, so human intervention is necessary for correcting mistaken letters, adding new paragraphs, transcription of non-Latin letters, etc. The human annotators made use of a web interface which introduces subtle changes to the HTML code of the documents, while keeping the surface form within the bounds set by the annotation guide. As the HTML code is the basis of the parser input, errors at this level are magnified as the processing becomes more refined.

At the end of the eDTLR research grant, all of the volumes of the thesaurus were parsed, but manual evaluation showed many parsing errors (Cristea *et al*., 2011). Further analysis showed that the main sources of errors are: 1) incorrect formatting for the input files due to human error while manually correcting the texts and 2) unnecessary information regarding typographical markers specific to electronic volumes.

This paper discusses a work in progress that intends to automatically solve the various errors that have been identified in the input files by means of cleaning formatting redundancies using regular expressions, heuristics and expert systems (an Expert system is an automatic system that simulates the decision making ability of a human expert), in order to improve the quality of the parsing of DLR entries. We will also use new scans and OCR of all the DLR volumes, which are of higher quality, and thus have better formatting information (even though the text body contains errors). The higher quality formatting information will be weaved with the manually corrected text mentioned above, and the resulting files will be reparsed

and included in the electronic format of the DLR. The paper is structured as follows: section 2 discusses the improvement of dictionary entry parsing by improving the input files, section 3 is concerned with the merging of the information from the manually corrected input and the fresh OCR and section 4 gives some conclusions.

2. Improving Dictionary Entry Parsing

For robust dictionary entry parsing (Curteanu *et al*., 2008) introduces a new strategy purposely built for large dictionaries (thesauri), called Dictionary Sense Segmentation and Dependency (DSSD). The purpose of this strategy is to obtain the sense tree, i.e. the sense hierarchy for a given dictionary entry. The effectiveness of the method largely resides in the fact that it differentiates between two essential steps in dictionary entry parsing: building the sense tree and parsing of the definitions. For the building of the sense tree it uses sense marker classes, their hierarchy and the method of identifying and classifying the sense markers. When a dictionary entry is parsed, the purpose is to create a lexical-semantic tree of the senses which define that entry. The DSSD algorithm was used within the eDTLR research project (Cristea *et al*., 2007) for processing of and obtaining the electronic version of DLR (Curteanu *et al*., 2008).

The quality of the DSSD parsing is largely dependent on the quality of the input files, i.e. the correctness of the formatting of the text, for example bold and italic markings, which must reflect the typographical markers of the scanned original or the rich text formatting of the electronically edited text.

After correcting the scanned text using the online application which was designed for the purpose (Cristea *et al*., 2009), the result was stored in HTML format (and not XHTML). Although this result does indeed contain all of the formatting data required, it also contains a large number of superfluous or unnecessary formatting, which is often contradictory:

```
<p  class="bold"><br/>          RĂNÚNCHI<sup>1</sup>  <span
class="normal">s.                m.                v.</span>
rărunchi<sup>1</sup><b>.</b></p>
```

Figure 1. Superfluos and contradictory formatting (contradictory formatting – light grey, superflous marking – dark grey, formatted punctuation – final marking)

The current paper describes a work in progress which attempts to eliminate these types of formatting errors by automatically cleaning and transforming the HTML code by means of regular expressions and heuristics, thus avoiding the large cost in time and manpower attached to their manual correction. This transformation will be followed by another full parse of the DLR, and the new results will be reevaluated. The final goal of this endeavor is to improve on the current form of the Electronic DLR, in order to increase its availability.

In the case of the electronically edited volumes, the difficulty comes from the fact that multiple formatting solutions have been used to reflect the same surface

structure (spacing of letters in a word by either changing the font properties or by inserting a space character in between all the letters of a given word) . Regardless of the manner in which a surface representation was obtained, the parser needs to take into account the general meaning of that representation and not its direct encoding (e.g. words with spaces between their letters represent partial synonyms to the current sense). Although some of these issues have been identified in during the eDTLR project, recent investigation has shown that many more still remain, and these issues are reflected in parsing errors.

This paper will describe some solutions to the most prevalent of these issues, which would significantly increase the quality of the parsing for the electronically edited volumes of the DLR.

The improving of the parsing needs to take into account the special characteristics of the two types of input available; because of this, the solving of the parsing errors will be treated separately for the scanned volumes and the electronic ones.

2.1. Improving the Transformation of the HTML format of the Scanned Volumes into XML, while Keeping Relevant Formatting

After the scanning and character recognition was performed, the electronic format obtained contains a significant number of errors. These errors can lexical (incorrectly recognized letters, missing letters, etc.) or of a formatting kind (incorrect fonts, text that is not bold, text that is not italic, etc.). In order to remove these errors, a set of volunteers, followed by a set of experts (lexicographers) have manually corrected the data to bring it as close as possible to the printed version. Because this was done using a web application, accessible via any browser, the encoding of the formatting can vary significantly. Moreover, the personal editing style of each expert changes the manner in which the same type of information is encoded.

After it is corrected, the scanned text is stored in HTML format, which, usually, is not also XHTML (which means it cannot be used as input data for the DSSD parser without modifications). The required input for the parser is an XML file which encodes markup denoting text formatting: paragraphs, bold, italic, superscript, subscript, letter spacing, all caps and certain information regarding font or language (for non-Latin characters). As such, the HTML format must be cleaned and all the superfluous information removed; in such cases where a tag is opened but not closed (where two pages meet, for example), this must be automatically closed. Another frequent error is the inclusion of punctuation in bold or italic markup; this is not visible in the editing interface, but must be corrected automatically as it decreases parse precision.

Since the scanned texts have been manually corrected, some of the atomic elements describing a dictionary entry (e.g. sigles for quotations) have been marked by the experts. These types of annotations generally increase the parsing speed (as it is not necessary to run all recognition subroutines). Stylistic differences between annotators need smoothing in this case as well (some annotators highlight the

volume sigle, others include the page number or some other associated element). Because of this, an automatic procedure for the unification of these annotations is necessary.

A similar problem occurs in the case of spaced text (space after each letter). The preferred solution was the highlighting of such cases in the web interface, but some correctors chose to manually introduce spaces in between the letters. An automatic unification method must be used in this case as well.

A set of similar problems has already been solved during the eDTLR project, in the case of sense markers:"During the parsing process, sense markers are vitally important. In order to recognize a sense marker within regular text, that marker should have some defining features, which, ideally, are unambiguous. Usually, the authors of the dictionary write sense markers with bold fonts, and, in the case of primary senses, at the beginning of a new paragraph (e.g. the Arabic numeral sense markers are always written in bold font and, in most cases, are given as the firs element in a new paragraph). In the case of scanned text corrected by experts, some of the sense markers have not been marked as bold text, which renders their recognition impossible during parsing. " (Final activity report for the eDTLR project, 2010, in Romanian)

2.2. Improving the Transformation of the Volumes in DOC format into XML, while Keeping Relevant Formatting

During the eDTLR project, a large number of parsed entries extracted from a volume available in electronic format (i.e. which was not scanned and OCRed, Discord – Doznic, containing more than 1000 entries of various sizes) have been manually verified by lexicographers. As a result of this error analysis, we have identified three main sources:

- Problems with the formatting or content of the input data (using different encodings for white spaces or mixing of typographical markers);
- Some parsing errors have been due to unforeseen issues in the entry encoding, which were not errors with regards to dictionary content (examples without sigles or dates given in different formats);
- Postprocessing errors, due to the manner in which the parsing results were saved in XML files (mixing of XML tags).

During the eDTLR project we have focused on the issues described in the first two categories given above, which were due to the implementation of the parser and which were largely solved at the time. While analyzing the remaining errors, we have discovered that a large set occurs for reasons regarding the quality and formatting of the input data. Some of these problems can be solved easily, using regular expressions (e.g. moving punctuation at the end of bold sequences outside the sequence), but others require complex heuristics which cover high degrees of variance (paragraph recognition, extracting special formatting such as small caps /

all caps, identifying spaced words). An example of an HTML format adapted to the parser entry format is given in Figure 2 below:

```
<p><b>EBENÍST</b>, <b>-Ă</b> s. m. şi f. (Rar la
f.) <b>1.</b> Persoană care confecţionează sau
vinde obiecte din lemn de abanos. <i>Ebenistul
(care lucrează abanosul) şi tâmplariul poleiesc
suprafeţele mobilelor. </i>CONV. GEOM. 2/11.
<i>Diferenţele cari există nu sânt decât ceea ce
există într-o serie mai restrânsă, precum ar fi
între cherestegiu, dulgher, tâmplar, strungar şi
ebenist. </i>GHICA, S. 235, cf. LM, ENC. ROM. II,
248, BARCIANU, ALEXI, W., RESMERIŢĂ, D., ŞĂINEANU,
D. U., SCRIBAN, D. <i>Glumiţi, un mare artist, un
ebenist ca Boulle Exaltă, pune vervă, culoare şi
contur. </i>CĂLINESCU, O. IX, 276. <i>Unul din
aceştia, ebenistul Laffargue,... îşi ucisese amanta
necredincioasă. </i>VIANU, L. U. 457, cf. DL, DM,
```

Figure 2. Types of problems in input

The text sequences highlighted in grey are examples of types of problems in the input data which reduce the precision of the parsing process.

3. Weaving Manually Corrected Text with an Improved OCR Text

Although the procedures described above greatly improve the quality of the parsing, some issues remain which cannot be solved heuristically. Firstly, in the case of the electronically edited volumes, the versions used in the eDTLR project were not the final print (which was unavailable in electronic format), but rather a nearly final version. This is because some modifications were made after the volume was delivered to the publishing house. Secondly, because of the manner in which the scanned versions were manually corrected (because of copyright reasons only part of a page was corrected at one time, and the page parts were delivered randomly to the correctors), some of the page fragments remained unassigned, and thus uncorrected. The uncorrected page parts were not included in the output of the correcting tool, and thus, some gaps appear in the text of the corrected input files. Such gaps in the electronic format or inconsistencies regarding its content as compared to the printed format are unacceptable and render the entire electronic version practically unusable.

Since the end of the eDLTR project, the DLR has been rescanned in full (all the letters from A to Z), and an improved OCR system has been applied to it. Also, within the CLRE project (Corpus Lexicografic Românesc Esenţial – Essential Romanian Lexicographic Corpus)[8], the dictionary entries have been separated manually, and this information completely removes one error source. Although the formatting of the newly recognized text is much improved compared to the version available during the eDTLR project, it has some errors in recognizing letters

(particularly for non Romanian character sets such as Greek or Cyrillic); these letters have been, for the most part, manually corrected but poorly formatted, so the best solution would be the merging of the new formatting to the text of the old formatting. This also has the advantage of filling in the gaps left in the input files as described above.

3.1. New OCR and Separated Dictionary Entries

The new OCR of the DLR is based on a new scan of the entire dictionary using a much better scanning platform, and thus the quality of the base material for the OCR process allows for the better recognition of such information as font weight, distance between letters, etc. However, since some of the prints are over 100 years old, the letters are not always correctly classified, especially in the case of special glyphs.

The information is given in two separate XML files: one which contains the recognized glyphs, with their offset and formatting, and a second file which gives title words, with their string value and offsets from the first file. The title word and entry separation has been carried out manually, by both volunteers and experts, and, as such, is used without verification.

3.2. Identifying Aligned Text Sections

Since we have access to two versions of the same text, both of which have both advantages and drawbacks, it becomes important to merge the information in the two variants such that the manual corrections carried out during the eDTLR project and the higher quality formatting and entry separation of the CLRE scans are both used as input for the parser.

The first step in this process is to identify the passages of text in both versions that are equivalent. Since the older scans were manually corrected, there are situations where there is no correspondence to a given passage in the new scans. Also, there are situations where equivalent texts are not equal because of a misrecognized glyph in the new scan. Solving these issues requires the use of various text distance metrics, which are then combined in order to choose the best possible matches of parallel texts. Currently, we are using four different distance metrics: Hamming distance, Levenshtein distance, Jaro-Winkler distance and the Jaccard similarity coefficient. The input for these metrics is made up of lists of paragraphs from each of the two versions, and the output is the similarity measure of each pair of paragraphs for each of the metrics used. The values are then normalized and aggregated (at the moment we are adding the normalized scores), and the highest scoring match for each paragraph is then chosen for alignment.

3.3. Merging Aligned Texts

Given the list of aligned paragraphs obtained above, we need to combine the information in such a way as to ensure that the useful parts of both sides are kept. Through analysis of the data, we have determined that the quality of the raw text is best in the old scans, and the quality of the formatting is best in the new scans.

Because of this, we have decided to keep all formatting data (i.e. from both versions of the scans) as long as they are not conflicting (an example of conflicting formats is a text that is both bold and italic, something which is impossible in the DLR). In case of conflict we prefer the markup given in the new scans; in the case of overlapping tags (i.e.), we will always choose to end the tag from the old version before (or start it after) the tag from the new scan version. For the transfer of text, we choose to replace all of the text in the new scan with the manually corrected text from the old scan; in those situations where a paragraph from one version was not aligned with a paragraph from the other, the text is kept as is. The text thus obtained then parsed with an improved version of the parser used for the original processing of the DLR.

4. Conclusions

In this paper we have presented a work in progress that attempts to improve the results of the parsing of the DLR by improving on the quality of the input data. The improvements concern the application of heuristics in order to automatically correct classes of errors and in the merging of two versions of scanned and OCRed texts. We have applied these methods to a large part of the dictionary (letters A, B, C, D) and the results show a marked improvement in parsing quality (e.g. the number of valid parses increased by approximately 5%). Analysis has shown that many of the remaining errors are due to some issues with the OCR (such as instances of italic text recognized as bold in certain circumstances) or incorrect entry separation. By the end of the year 2016, we intend to apply these methods to the entire dictionary.

Acknowledgements

The research described in this paper was funded by the "Alexandru Ioan Cuza" University of Iasi, through project GI-2015-14, no. 17/03.12.2015.

References

Clim, M.-R., Tamba, E., Catană-Spenchiu, A.-V., Patrașcu, M. (2013). Corpus lexicographique roumain essentiel. *100 dictionnaires de la langue roumaine alignés au niveau de l'entrée et, partiellement, au niveau du sens, XXVII International Congress of Romance Linguistics and Phylology*, Nancy, 15 – 20 iulie 2013

Cristea, D., Răschip, M., Forăscu, C., Haja, G., Florescu, C., Aldea, B., Dănilă, E. (2007). The Digital Form of the Thesaurus Dictionary of the Romanian Language. In *Proceedings of the 4th International IEEE Conference SpeD*, 2007.

Cristea, D., Haja, G., Răschip, M., Moruz, A., Pătrașcu, M. (2011). Partial statistics on eDTLR-Thesaurus dictionary of the Romanian Language in electronic form (Statistici parțiale la încheierea proiectului eDTLR – Dicționarul Tezaur al Limbii Române în format electronic). In *Acta of Conference Romanian Language: hypostasis of linguistic variation, 3-4 December 2010* („Limba

română: ipostaze ale variației lingvistice, 3−4 decembrie 2010"), Romanian Department, University of Bucharest.

Cristea, D., Raschip, M., Moruz, A. (2009). Steps in Building the Electronic Version of the Thesaurus Dictionary of the Romanian Language. In *Proceedings of the IVth National Conference The Academic Days of the Academy of Technical Science of Romania, ASTR* - the Iasi branch and "Gheorghe Asachi" Tehnical University Iasi, Agir Publishing House, ISSN 2006-6586.

Curteanu, N., Moruz, A., Trandabăţ, D. (2008). Extracting Sense Trees from the Romanian Thesaurus by Sense Segmentation & Dependency Parsing. In *Proceedings of the Workshop "CogAlex-I"*, COLING 2008, Manchester, United Kingdom, 55-63, ISBN 978-1-905593-56-9.

Curteanu, N., Trandabăţ, D., Moruz, A. (2010). An Optimal and Portable Parsing Method for Romanian, French, and German Large Dictionaries. In *Proceedings of the Workshop "COGALEX-II"*, COLING-2010, Beijing, China, August 2010, 38-47.

Curteanu, N., Moruz, A. (2012). Toward the Soundness of Sense Structure Definitions in Thesaurus-Dictionaries. Parsing Problems and Solutions. In *Journal of Computer Science of Moldova*, Academy of Science of Moldova, vol. 20, no. 3(60), 275 – 303.

Curteanu, N., Moruz, A., Cojocaru, S. (2013). Formalization of a General SCD-based Parser for Dictionaries Using Parametrized Grammars. In *Proceedings of the International Conference on Intelligent Information Systems IIS 2013*, August 20-23, 2013, Chisinau, Moldova.

A COLLOCATIONAL APPROACH TO ROMANIAN STRONG NEGATIVE POLARITY ITEMS

MONICA-MIHAELA RIZEA[1], GIANINA N. IORDĂCHIOAIA[2], FRANK RICHTER[3]

[1]*Solomon Marcus Center for Computational Linguistics, University of Bucharest*
monicamihaelarizea@gmail.com
[2]*University of Stuttgart*
gianina@ifla.uni-stuttgart.de
[3]*Goethe University, Frankfurt A.M*
f.richter@em.uni-frankfurt.de

Abstract

This paper proposes an analysis of a special class of Romanian Negative Polarity Items, strong NPIs, following a collocational approach to NPI licensing. From this perspective, NPIs are understood as collocationally-restricted lexical items with idiosyncratic distributional patterns. The collocational approach allows us to model the distributional profiles of individual NPIs and to isolate the co-occurrence requirements specific to strong NPIs. Corpus-linguistics methods are applied in order to determine individual statistical profiles in terms of licenser-NPI collocations and to classify these items according to the occurrence constraints on their licensing environments. Given the fact that some categories of licensers only occur with strong NPIs under specific readings, we also take into account reading-dependent licensing cases. The study is correlated with the practical task of updating the collection of Romanian Negative Polarity Items (CODII-NPI.ro), which is part of a (comparable) multilingual electronic resource in XML format.

Key words — collocational approach to NPI licensing, corpus-based analysis, electronic resource, enrichment of a multilingual resource, strong NPIs.

1. Introduction

This paper is dedicated to the analysis of a special class of Romanian Negative Polarity Items, strong NPIs, following the collocational approach to NPI licensing proposed by Richter and Soehn, 2006; Sailer, 2009b.

Our purpose is to model the distributional profiles of individual NPIs and to isolate the co-occurrence requirements specific to strong NPIs. Corpus-linguistics methods are applied in order to determine individual statistical profiles in terms of licenser-NPI collocations and to classify these items according to the occurrence constraints on their licensing environments. Given the fact that some categories of licensers

only occur with strong NPIs under specific readings (Sailer, 2009a and b), we also take into account reading-dependent licensing cases.

We illustrate the analysis of strong NPIs with the Romanian negative polarity multiword expression (NPMWE) *licenser + țipenie de om (lit. shout.suffix of)*. Additionally, we compare its distributional pattern with a (quasi-)synonymous structure, *picior de om (lit.leg of person)* and with one English equivalent, *licenser + a living soul.* Examining the individual realizations from corpora in relation to a set of diagnostic contexts allows us to distinguish strong from weak NPIs; moreover, the comparison of individual quantitative profiles is relevant since it shows the degree of distributional variation within the strong NPI class.

The study is correlated with the practical task of updating the sub-collection of Romanian Negative Polarity Items (CODII-NPI.ro), which is part of a (comparable) multilingual electronic resource in XML format (www.english-linguistics.de).

The paper is structured as follows: In section 2 we give a definition of NPIs from a collocational perspective and delineate the distributional pattern of strong NPIs as presented in the literature. We then describe our main practical objectives, the methodology, tools, and corpora used (section 3). In section 4 we focus on the analysis of a Romanian NPMWE as currently represented in the CODII-NPI.ro database (in terms of corpus investigation and criteria applied for its classification) and on a comparison of its distributional profile with other two profiles: that of a (quasi)-synonymous Romanian NPMWE and also with the occurrence profile of its English equivalent. The last section is dedicated to concluding remarks and future work.

2. Negative Polarity Items from a collocational perspective

Theoretical studies generally consider NPIs to be distributionally restricted to certain licensing contexts, prototypically negative or negative-like environments (such as interrogatives, antecedents of conditionals, complement clauses of adversative predicates, the restrictor of a universal quantifier, etc.), even if they do not, themselves, express negation. NPI examples are individual lexical items such as *vreo / vreun* (*any*), but also multiword expressions such as *țipenie de / picior de om* (idiomatic meaning: *anyone at all*). They display non-referential, idiomatic readings that are specific to the negative environments in which they appear and are only possible as long as their distributional restrictions are respected:

Example 1:

***N**-am văzut **picior de om** pe stradă.*
not=have seen leg of person in street
I haven't seen a living soul in the street
*[= I haven't seen **anyone at all** in the street.]*
Example 2:
*#Am văzut **picior de om** pe stradă.*
(strange in the literal, positive meaning)
have seen leg of person in street

*I have seen **a living soul** in the street.*

In this approach, NPIs are understood as collocationally-restricted lexical items (i.e. items that have a collocate-collocator relation with their licensing contexts) that also display idiosyncratic distributional patterns. This mainly implies that the licensing patterns characterizing individual items are not predictable and that they have to be determined by applying diagnostic tests such as the ones described below.

The collocational approach allows us to model the occurrence profiles of individual NPIs (the distributional dependence on the licensing contexts is documented with frequency data and real-use examples from large Romanian corpora) and to isolate the co-occurrence requirements specific to strong NPIs.

Distributional patterns of strong NPIs

In our analysis, we start with Sailer's definition of the main occurrence patterns of strong and weak NPIs (Sailer, 2009b: 32-35). We represent the distribution of strong NPIs in contrast with the occurrence patterns of weak NPIs in order to obtain a clear picture of *reading-dependent licensing*[1]. The occurrence patterns are illustrated with English examples as presented in the literature; in section 4, we will apply these patterns to Romanian NPIs.

Table 1. Diagnostic contexts of Strong vs. Weak NPIs

Licensing environment[2]	**S. NPIs**	**W. NPIs**	**Examples**[3]
1. CMN			
clausemate sentential negation expressed on the verb	ok	ok	**S.** As a result, they don't pay *a red cent.*
			W. Peter did not read *any* books.
2. NW			
scope of a clausemate n-word	ok	ok	**S.** We filter it because we don't want it and will never send *a red cent* to them.
			W. Nobody *ever* read this book.
restrictor of a clausemate n-word	ok	ok	**S.** [No one with *a red cent* in his pocket] would support this artist.
			W. [No student who has *ever* studied syntax] could forget this example.
3. WITHOUT			
	ok	ok	**S.** Alfred came to the party without *lifting a finger* to help with the preparations.
			W. Peter left without eating *any* chocolate.
4. DENT			

[1] There are groups of licensers that have some individual realizations or readings that exclude strong NPIs and only license weak NPIs.

[2] There are also other environments that license NPIs, but we only focus on this set of environments for the purpose of this study.

[3] Most of the examples are provided by Sailer, 2009b: 32-35.

scope of downward-entailing *few, not many, at most*	*	ok	**S.** * Not many authors earn *a red cent* with their first novels.
			W. At most 10 users have *ever* borrowed this book.
5. UNIV			
restrictor of a strong quantifier in a 'law-like'[4] sentence	ok	ok	**S.** [Every kid with *a red cent* in his pocket] would buy this candy bar.
			W. [At most 10 users who *ever* borrowed this book] read it completely.
restrictor of a strong quantifier in an 'episodic' sentence	*	ok	**S. *** [Every kid with *a red cent* in his pocket] bought this candy bar.
			W. [Most students who've *ever* read of Hegel] seem to wear hats.
6. IF			
If-clause in threats and 'law-like' sentences	ok	ok	**S.** If I had *a red cent* for every variation of a tulip Ethernet NIC that was ever made for the alpha, I'd have enough for a decent snack at subway.
			W. If you *ever* say that again, you will regret it, got it?
If-clause in promises and 'episodic' sentences	*	ok	**S. *** If Pat earned *a red cent* last night, he had his first lucky day in weeks.
			W. If Pat *ever* reads Syntactic Structures, she'll enjoy it.
7. nCMN			
complement clause to a negated neg. raising predicate	ok	ok	**S.** Personally, I don't think Paul should *pay a red cent* for you.
			W. John doesn't think that Dan *ever* ate any chocolate.
complement clause to negated matrix predicates others than neg. raising predicates	*	ok	**S.** * Pat didn't claim that Chris gave *a red cent* to charity.
			W. John doesn't claim that Dan *ever* ate any chocolate.
8. NV			
complement clause to a non-factive adversative predicate (*deny, doubt*)	ok	ok	**S.** I really doubt they are spending *a red cent* more on gifted kids than on regular kids.
			W. Government officials have reportedly denied that Vreeland *ever* served in the Navy.
complement clause to a factive adversative predicate (*be surprised, regret*)	*	ok	**S.** * Pat is surprised that Chris gave *a red cent* to charity.
			W. Sandy is surprised that Robin *ever* ate kale.
9. QUE			
negatively biased rhetorical	ok	ok	**S.** Did Mary contribute *a red cent* for this

[4] For a definition of law-like vs. episodic readings, see Sailer, 2009a: 456.

questions			cause?
non-rhetorical questions	*	ok	**W.** Do you have *any* tomatoes?

Table 1 shows that strong NPIs cannot occur in the scope of downward entailing operators such as *few*, and that they are excluded from syntactic constructions with an 'episodic' interpretation such as the restrictor of a universal quantifier; similarly, they cannot occur in if-clauses with promise or 'episodic' readings. Additionally, strong NPIs are not licensed by factive adversative predicates such as *regret* and *be surprised* or by negated predicates such as *claim* (i.e. predicates that are not neg. raising).

3. Practical objectives, methodology, and tools

The analysis presented in this paper results from the common initiative of updating the Romanian Collection of Negative Polarity Items (CODII-NPI.ro), which is part of a (comparable) multilingual electronic resource (CODII) in XML format, hosting German, English, and Romanian collections of distributionally idiosyncratic items. We intend both to improve the initial entries, and to enrich the database with negative polarity multiword expressions. In the present study, we only focus on the distributional patterns and corpus investigation of strong NPIs, which we illustrate with the analysis of one CODII entry.

Typically, each entry in the CODII-NPI.ro database is designed to provide General Information (such as usage notes, English glosses and translations), Syntactic Information, a list of Licensing Contexts, information about the NPI Class and Examples[5]. The current design phase (work in progress) brings a number of modifications to the initial collection of Romanian NPIs: 1. more detailed usage notes (including the correspondent negative-polarity expressions in English, when they exist, and semantico-pragmatic characterization, especially in the case of minimizer / maximizer NPIs), 2. syntactic information to include the characterization of the individual parts of the expression and of the entry as a whole[6], 3. sentence examples (from large Romanian corpora - such as roWaC or OPUS2 Romanian) for every valid licenser-NPI pair, 4. statistical profiles for every licenser-NPI collocation 5. information about 'competing MWEs' (including cases of polysemy when the expressions might also exhibit non-NPI senses), and 6. a classification of NPIs that also accounts for reading-dependent licensing cases.

The methodology that we apply in this paper implies a paradigmatic and syntagmatic analysis. We start from the definitions provided in the most important Romanian general dictionaries[7] (mainly DEX 2012 - *The Explanatory Dictionary of*

[5] A detailed description of the conceptual design and technical realization of the entire CODII database is provided, for example, in Trawiński et al. 2008: 1447.

[6] The POS tags used for Romanian are specified under universaldependencies.org/ro/pos/index.html.

[7] The dictionary definitions sometimes provide usage information such as "in negative constructions / sentences"- only 48 times in DEX 2012 - with ≈ 67 000 entries; additionally, some entries are registered with an element such as "nu" ("not") or "nici" ("not even"). For

the Romanian Language) and in DELS 2010 - *The Dictionary of Romanian Expressions, Syntagms, and Phrases*. The items analysed are then checked against large Romanian corpora such as the Romanian Web Corpus (roWaC[8] - nº of words = 44,729,032) via the Sketch Engine online tool[9], by analysing their individual realizations in context, in relation to a predefined set of licensers[10]. Corpus-linguistics methods are used in order to determine individual statistical profiles in terms of licenser-NPI collocations and to classify the NPIs (into *superstrong / strong / weak*[11] – see van der Wouden, 1997) according to the occurrence constraints on their licensing environments. We further refined van der Wouden's criteria by considering reading-dependent licensing in the case of strong NPIs (Sailer, 2009 a & b). Generally, we provide real-use corpus examples (accompanied by English translations) in order to document the compatibility with each category of licenser. However, when there are not enough examples in the corpus (especially when we intend to represent different readings of the same category of licensers), we rely on examples created by Romanian linguists.

4. Analysis of Romanian NPIs

In this section we focus on the analysis of the Romanian negative polarity multiword expression *țipenie de om* as it is currently represented in the CODII-NPI.ro database (in terms of corpus investigation and criteria applied for its classification) and on a comparison of its distributional profile with that of a (quasi)-synonymous Romanian NPMWE *picior de om* and also with the occurrence profile of one of their English equivalents, *a living soul*.

4.1. CODII representation and corpus profile of the NPMWE țipenie de...

Țipenie de ... (shout.suffix of - lit. no one to shout, no living creature) is part of a complex nominal phrase of the type *N1 + DE + N2*, i.e. *țipenie + de + N2* (in the context of a licenser), which, as a whole, functions as an emphatic negator (Dindelegan 2013:128). N2 has limited lexical variation, usually reduced to *om (human / person).* In roWaC, there are also other realizations of N2, such as *vietate (creature)* and *terorist (terrorist)*. An English correspondent can be found in the

example, from 11430 expressions listed in DELS, 518 (i.e. 4.5%) contain "nu" and / or "nici" (but these expressions are not necessarily all NPIs).

[8] This corpus was gathered by Monica Macoveiciuc, Alexandru Ioan Cuza University, Iași.

[9] the.sketchengine.co.uk

[10] These licensers are listed in the CODII.NPI.ro Licensing Contexts section and acquire binary ("yes" / "no") values according to the specific distributional patterns of each NPI entry. See Fig. 1 for details.

[11] Briefly, these criteria are defined as follows:

- **Superstrong** NPIs are licensed only by antimorphic contexts (overt negation).
- **Strong** NPIs are licensed by antimorphic and anti-additive (comprising n-words and without) contexts.
- **Weak** NPIs are licensed by antimorphic, anti-additive, and downward-entailing contexts (plus the remaining ones).

minimizer construction *a (living) soul*. Similarly to the English expression, *țipenie de* has the idiomatic meaning *anyone at all / absolutely anybody* in negative contexts. *Țipenie* usually occurs as a bare noun when preceded by the scalar negator *nici (not even)*; however, it can also be preceded by the negative determiner *nicio (no.fem)*. There are also contexts where *țipenie de* can appear with clausemate negation and no other negative element. This expression is only felicitous in negative contexts. It is part of a special class of Romanian minimizers such as *urmă / umbră / suflare / suflet de* (lit. *trace / shadow / breath / soul of)* that combine with non-gradable entities, usually [+animate], and that can be considered as the faintest manifestations of N2 on a scale of perception. This is a valid mechanism for obtaining emphatic NPI minimizers: negating the minimum imaginable evidence of the existence of an entity N2 rendered by something that is not even a part of N2, not even a material attribute of the entity it stands for. Just like the other minimizers, they evoke the least likely alternative to the entity in focus, which is, actually, N2, the semantic head of the structure. Since N2 has a very limited lexical variation, *țipenie* is many times used alone and it incorporates the meaning of N2:*"Ziua nu întâlneau ţipenie."* (*lit. "During daytime, they didn't meet living.soul"*), meaning *"they wouldn't meet anyone at all."* For example, in roWaC, from 114 occurrences of the word *țipenie*, 30 occurrences (i.e. 26%) represent cases when *țipenie* is used without N2. Below, we provide a fragment of the XML representation of the CODII.NPI.ro entry *țipenie de*, with a focus on the Licensing Contexts section. The structure is checked against a predefined list of licensers that are tagged as "yes" whenever a certain licenser – NPI collocation is possible[12]. Each category of contexts is illustrated with examples. The evaluation of the approved combinations with the licensing contexts allows us to include the expression in one of the three classes defined in section 2.

[12] As mentioned in section 3, we either find the relevant licenser-NPI collocations in corpora, or rely on linguist intuition when the examples from corpora are insufficient.

```
<dii-entry id="tipeniede">
    <dii>
        <ol>țipenie de</ol>
        <en>anyone at all</en>
    </dii>
    <dii-classification>
        <dii-class category="pi" subcategory="npi" type="A5" class="strong"
            original-class="no">
            <bibliography bib-item=""/>
        </dii-class>
    </dii-classification>
    <dii-syntax hits="tipeniede1 tipeniede4 tipeniede5 tipeniede12" cat="NOUNP">
        <dii-expression-syntax>NOUN ADP</dii-expression-syntax>
    </dii-syntax>
    <licensers>
        <cmn given="yes" hits="tipeniede1 tipeniede2"/>
        <ncmn given="yes" hits="tipeniede3"/>
        <nw given="yes" hits="tipeniede4"/>
        <nici given="yes" hits="tipeniede5 tipeniede6 tipeniede7"/>
        <dent given="no"/>
        <nv given="yes" hits="tipeniede8"/>
        <que given="yes" hits="tipeniede9"/>
        <if given="yes" hits="tipeniede10 tipeniede11"/>
        <without given="yes" hits="tipeniede12"/>
        <only given="no"/>
        <univ given="yes" hits="tipeniede13"/>
        <comp given="no"/>
        <sup given="no"/>
    </licensers>
    <dii-queries>
    </dii-queries>
</dii-entry>
```

Figure 1. CODII.NPI.ro - XML representation of the Licensing Contexts section

Corpus examples (source: roWaC)

- **CMN (sentential negation - NM *nu* "not")**

  Nu se zărea țipenie de om, locul părea pustiu.
  Not a living soul in sight, the place seemed deserted

- **NW (n-word)**

  Pe drum, nicio țipenie de om.
  On the road, no living soul.

- **nici (scalar negator *nici* "not.even")**

  Poate de aceea nu e nici țipenie de om în jur.
  Maybe that's why there's no living soul around.

- **Whithout ("fără")**

  Am trecut prin pădure, spre calea ferată, fără să întâlnim țipenie de om.
  We passed through the forest, to the railway, without meeting a living soul.

Reading-dependent licensing (source: linguist)

- **DENT (downward-entailing operator *puțini / puține* "few")**

  #Puțini călători au întâlnit țipenie de om în deșert.
  Few travelers met a living soul in the desert.

- **nCMN (negated verbs – *pretinde* "claim" / *crede* "think")**

#Nu pretind că am văzut ţipenie de om în noaptea aceea.
I don't claim I've seen a living soul that night.
Nu cred c-am văzut ţipenie de om în noaptea aceea.
I don't think I've seen a living soul that night.

- **NV (inherently negative matrix verbs such as *a fi surprins(ă)* "be surprised" or *a se îndoi* "doubt")**

#Mă surprinde că văd ţipenie de om în deşert.
I'm surprised that I see a living soul in the desert.
Mă îndoiesc că voi vedea ţipenie de om în deşert.
I doubt that I'm going to see a living soul in the desert.

- **QUE (in negatively biased rhetorical questions)**

Speri să întâlneşti ţipenie de om pe drum la ora asta?
Do you hope to meet a living soul on the road at this hour?

- **IF (in conditional threats, episodic statements, conditional promises)**

Dacă văd ţipenie de om în această rezervaţie naturală, îmi voi ieşi din minţi!
(threat reading)
If I see a living soul in this protected nature area, I will go mad!
#Dacă întâlnesc ţipenie de om *în deşert, îl salut.*
(episodic reading)
If I meet a living soul in the desert, I say hello.
(#)Dacă întâlnesc ţipenie de om în noaptea asta în bar, plătesc toată băutura.
(promise reading)
If I meet a living soul at the bar tonight, I'll pay for all the drinks.

- **UNIV (in the restrictor of a universal quantifier)**

#Oricine întâlneşte ţipenie de om *în deşert, îl salută. (episodic reading)*
Everyone who sees a living soul in the desert, says hello.
Oricine vede ţipenie de om în bezna asta, se poate considera binecuvântat.
("I strongly doubt" reading)
Whoever sees a living soul in this darkness can consider him / herself blessed.

The analysis of the examples above shows that *ţipenie de* ... behaves like a *strong NPI* (see the distributional patterns of strong NPIs in Table 1) since it is not licensed by downward-entailing determiners such as *puţini / puţine* ("few"), it is felicitous in the context of negative raising predicates such as *nu cred* ("I don't think"), but it is strange in the complement clause of *nu pretind* ("I don't claim"); it is licensed by negative predicates such as *a se îndoi* ("doubt"), but not by *a fi surprins(ă)* ("be surprised"); *ţipenie de* ... is licensed in rhetorical questions and in the antecedent of conditional threats, but it is not licensed in episodic readings; it is not felicitous with conditional promises, unless they receive a sarcastic interpretation - see Horn, 2016: 289-291 - the condition for a promise is interpreted as "I strongly doubt that..."; similarly, it is licensed in the restrictor of a universal quantifier only if the context receives the "I strongly doubt that..." interpretation.

4.2. Comparison of individual distribution profiles

Comparing the quantitative profiles of the two synonymous Romanian expressions, we notice that there is variation in terms of their preference for certain licensers.

Table 2. Distribution profiles of two quasi-synonymous Romanian NPMWEs[13]

Licensing environment / **Corpus**	**Romanian Web Corpus – no of words = 44,729,032**	
	Query: *țipenie de om* **N = 81**	**Query:** *picior de om* **N = 50**
negative(-like) (A+B+C+D+E+F)	N= 76 (94%)	N= 43 (86%)
A:nu (n-/ne-) 'not' NM alone[14]	N= 34 (42%)	N= 38 (76%)
B:NW[15]	N= 2 (2%)	N= 1 (2%)
C:nici 'not even'	N= 37 (46%)	N= 4 (8%)
D:fără 'without'	N= 3 (4%)	N= 0 (0%)[16]
E: NV	N= 0 (0%)[17]	N= 0 (0%)
F: dacă 'if'	N= 0 (0%)	N= 0 (0%)
Other	N= 5 (6%)	N= 7 (14%)

The results show that *nici* (*not.even*) has the highest frequency in terms of occurrence with *țipenie de om,* i.e. 46 %, while only 8% frequency with *picior de om.* In the case of *picior de om*, the highest preference is for the antimorphic operator, *nu*, i.e. 76%. The two NPMWEs also differ slightly in terms of their affinity for the anti-additive preposition *fără* (without). The degree of occurrence of these two NPMWEs in negative contexts is very high. The 14% positive contexts in the case of *picior de om* also include the literal meaning of a body part, so they do

[13] All the abbreviations in the table have been defined under the Corpus Examples.

[14] For this licensing environment we take into consideration the occurrence of the negative marker (NM) without any other n-word.

[15] Romanian is a strict Negative Concord language; therefore, it is expected that n-words and sentential negation co-occur. For this category of contexts we consider the n-word as licenser.

[16] After the evaluation of the corpus data along with the linguist intuition about the allowed licenser-NPI collocations, the Romanian entry *picior de*...is marked as *yes* for the valid combinations that do now show at corpus queries).

[17] If we compare the corpus data with the analysis of the examples in section 4.1., we can see that 0 realizations in corpus does not signify here the fact that *țipenie de om* does not collocate with a certain licenser, but that there are no corpus representations.

not affect the **NPI** *picior de om* that has an idiomatic meaning in negative environments.

These distribution profiles are important since they offer a perspective on the degree of preference of an individual NPI for a certian licenser (in comparison with the other licensing environments) as it is registered in an approximately 45 million word corpus - representative for Romanian contemporary language.

Table 3. Distribution profile of the English expression a living soul

Licensing environment / **Corpus**	**OPUS2 English – no of words = 1,139,515,048**
	Query: a living soul N = 134
negative(-like) (A+B+C+D+E)	N= 103 (77%)
A: not (n't)	N= 69 (51%)
B: NW	N= 34 (25%)
C: without	N= 1 (1%)
D: NV	N= 0 (0%)
E: if	N= 0 (0%)
Other	N= 31 (23%)

A similar strong affinity for negative contexts can be noticed in the case of the English correspondent expression, *a living soul*, whose percentage of positive contexts is largely due to occurrences with the literal, *"walking dead"* meaning (Table 3). Just as in case of the Romanian expression *picior de om*, the strongest preference is for the antimorphic operator *not***.** In fact, all three expressions manifest a high degree of frequency with respect to this licenser: *a living soul* and *picior de om* collocate mostly with *the antimorphic licenser* (76% and 69%, respectively), while *țipenie de* shows a 42% affinity for the NM, which is only 4% lower than its highest frequency (i.e. 46% preference for *nici*).

According to the results, this is only one example that confirms the claims formulated by similar studies on English and Dutch NPIs: namely that (1) *even synonymous NPIs occurring in roughly the same environments may still display some variation with respect to the strength of their affinity to individual contexts)* – see Sailer, 2009 b: 59, 254 and Hoeksema, 1997 **–** and that (2) *comparable items in different languages tend to have comparable distributions* (Hoeksema, 1997).

5. Conclusions and future work

In this paper we presented some results of the project of updating the Romanian Collection of Negative Polarity Items (CoDII-NPI.ro), which is part of a

multilingual resource, CoDII (The Collection of Distributionally Idiosyncratic Items).

To summarize, we focused on defining the distributional paradigm of strong NPIs from the perspective of a collocational theory of NPI licensing. We illustrated the analysis of strong NPIs with one Romanian NPMWE as currently represented in the Romanian NPI database. We also compared individual quantitative profiles of quasi-synonymous NPMWEs and their English correspondent, which pointed to similar results reported for other languages. However, a more extensive study, based on distributional profiles of a larger number of Romanian NPMWEs, is necessary in order to compare our results with those reported in the literature in terms of idiosyncratic variation in NPI licensing across languages and within one language.

Using the existing lexicographic resources (mainly DEX 2012 and DELS 2010), we have extracted 100 NPMWE candidates for further analysis.

Documenting NPMWEs for multiple languages does not only facilitate comparative linguistic studies, but it could also represent a useful resource for translators that search for paraphrases of idioms sensitive to negative polarity, as well as for second language learners who can find real-use examples from corpora (with English glosses and translations) for every licenser-NPMWE pair.

Acknowledgements

This paper is the result of a joint reasearch activity that was initiated during a Short-term Scientific Mission at Goethe University, Frankfurt am Main. This was supported by the European network PARSEME (Parsing and Multiword Expressions), COST Action IC1207. We would like to thank Manfred Sailer for theoretical support and for making this collaboration possible.

References

DELS. 2010. Dicționar de expresii, locuțiuni și sintagme ale limbii române ("The Dictionary of Romanian Expressions, Syntagms and Phrases"), Cătălina Mărănduc. Bucharest: Corint.

Dindelegan, G. P. (2013) The Grammar of Romanian, Oxford University Press.

Horn, L.R. (2016) Licensing NPIs: Some Negative (and Positive) Results. In P. Larrivée and C. Lee (Eds.). Negation and polarity: Experimental perspectives, Cham: Springer, pp. 281-305.

Hoeksema, J (1997) Corpus Study of Negative Polarity Items. Jornades de corpus linguistics 1996-1997, Universitat Pompeu Fabra, Barcelona.

Iordăchioaia, G. (2007) A Case of Negative Polarity in Romanian, Revue Roumaine de Linguistique, LII, 1-2, Bucharest, 195-209.

Iordăchioaia, G., Richter F. (2015) Negative Concord with polyadic quantifiers. The case of Romanian, Natural Language and Linguistic Theory, 33 (2), 607-658.

Macoveiciuc, M. and Kilgarriff, A. (2010). The RoWaC Corpus and Romanian Word Sketches In: Multilinguality and Interoperability in Language Processing with Emphasis on Romanian. Edited by Dan Tufiș and Corina Forăscu, Romanian Academy Publishing House, Bucharest.

Richter, F., Soehn. J. Ph. (2006) Braucht niemanden zu scheren: A Survey of NPI Licensing in German. The Proceedings of the 13th International Conference on Head-Driven Phrase Structure Grammar, Stanford: CSLI Publications.

Sailer, M. (2009a) On reading-dependent licensing of strong NPIs. In A. Riester and T. Solstad (Eds.). Proceedings of Sinn und Bedeutung 13, Stuttgart, 455-468.

Sailer, M. (2009b) A representational theory of negative polarity item licensing. Habilitation thesis, Universität Göttingen.

Soehn, J. Ph., Mingya L., Trawiński, B., Iordăchioaia G. (2010) Nicht sonderlich oder doch sattsam bekannt? Positive und Negative Polaritätselemente als lexikalische Einheiten mit Distributionsidiosynkrasien EUROPHRAS 2008. Helsinki, 273-281.

Trawiński, B., Soehn. J. Ph., Sailer, M., and Richter F. (2008). A Multilingual Electronic Database of Distributionally Idiosyncratic Lexical Items. Proceedings of Euralex 2008.

van der Wouden, Ton (1997) Negative contexts. Collocation, polarity and multiple negation. London and New York: Routledge.

CHAPTER 5
SHORT PAPERS

A DEEPER PERSPECTIVE OF ONLINE TOURISM REVIEWS ANALYSIS USING NATURAL LANGUAGE PROCESSING AND COMPLEX NETWORKS TECHNIQUES

ALEX BECHERU, COSTIN BĂDICĂ

Faculty of Automation, Computers & Electronics, University of Craiova

becheru@gmail.com, costin.badica@software.ucv.ro

Abstract

This paper proposes a deeper analysis of tourist opinions extracted from online reviews. Starting with a graph representation of question-answering user interactions extracted from amfostacolo.ro, we analyze each community's touristic interests. Community detection is done with the help of complex network algorithms. We extract the topics of interest of each community using part of speech tagging and frequent words filtering. Besides nouns, our analysis also targets verbs and adjectives. Thus, causality and special aspects expressed in the reviews are gathered. Furthermore, we investigate the differences, if any, in the topics of interest of the users depending on their activity intensity on the Web site.

Key words — tourist opinion, online comments, algorithms, topis.

1. Motivation and Previous Work

Through this paper we present the results obtained during a deeper analysis of online tourism reviews. The motivation of this paper is the continuation of the authors quest in better understanding online tourism. Previously, our research on online tourism was focused on polarity shifting for sentiment classification (Colhon *et al.*, 2016), discovering and analysing social phenomena that arise between online reviewers (Becheru *et al.*, 2015a). Also, we have proposed and proven useful the use of complex networks for developing tourist review analytics (Becheru *et al.*, 2015b). The current paper builds on our previous mentioned research together with natural language processing (NLP) methods and complex networks analysis (CNA) techniques to further unveil online tourism facts. The main scope of this paper is to get a better perspective on the user communities from *Amfostacolo.ro* (http://amfostacolo.ro/) web site.

2. Experiments

In order to obtain communities of users from *Amfostacolo.ro* we applied the modularity algorithm (Hristea and Popescu) to a previous obtained graph of user

interaction. A vertex in this graph is an abstraction of a user, while arcs represent "question-answer" relations between users, for more details see paper (Becheru *et al.*, 2015a). Thus, we were able to detect communities based on the users' interactions on the web site. We took in consideration only the communities (Blondel *et al.*, 2008) that have 10 or more users, thus avoiding to introduce noise in the results.

We applied POS tagging for Romanian texts (Hristea and Popescu, 2003) extracted from *Amfostacolo.ro* and created a set of tokens used by each user, while preserving their frequency. We aggregated the sets of tokens for each community. Next, we filtered for the top 10 most frequent used tokens by community for the following parts of speech: nouns, verbs and adjectives.

Further, we aggregated various user metadata: sex, age, average score given per touristic entity, number of sentences written, user type as determined by the web site rules. Thus, we obtained community averages for the above mentioned metadata.

3. Results

Currently we are in a ongoing process of comparing the results between communities and interpreting the data, we can argue the following. Community creation does not seem to be influenced by the user age or user type. The age varies between 30 and 36 years while the user type varies between 3.8 to 5.4 (from a possible range of 0 to 12). Currently we are targeting possible correlations between the average score, average number of sentences and user sex. Hopefully, we can explain the creation of some communities through these possible correlations.

Regarding the tokens used by each community we discovered that there is a large variance between communities. The majority of nouns depict touristic entities or objects of interest like: parking, table, restaurant, etc. Regarding verbs, we can only say that *eating* is of great interest for all communities. The majority of adjectives regard monetary value, this finding is consistent among the majority of communities.

References

Becheru, A., Bădică, C., Antonie, M. (2015a). Towards Social Data Analytics for Smart Tourism: A Network Science Perspective. In *Workshop on Social Media and the Web of Linked Data,* 35-48. Springer International Publishing.

Becheru, A., Buşe, F., Colhon, M., Bădică, C. (2015b). Tourist review analytics using complex networks. In *Proceedings of the 7th Balkan Conference on Informatics Conference*, 25.

Blondel, V. D., Guillaume, J. L., Lambiotte, R., Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, vol. 10, P10008.

Colhon, M., Cerban, M., Becheru, A., Teodorescu, M. (2016). Polarity shifting for Romanian sentiment classification. In *INnovations in Intelligent SysTems and Applications (INISTA), 2016 International Symposium,* 1-6.

Hristea, F., Popescu, M. (2003). A dependency grammar approach to syntactic analysis with special reference to Romanian. *Building Awareness in Language Technology.* University of Bucharest Publishing House, 2003.

A ROMANIAN CORPUS ANNOTATED WITH VERBAL MULTIWORD EXPRESSIONS

VERGINICA BARBU MITITELU[1], MONICA-MIHAELA RIZEA[2], MIHAELA IONESCU[3],
MIHAELA ONOFREI[4], ELENA IRIMIA[1]

[1] *"Mihai Drăgănescu" Research Institute for Artificial Intelligence, Romanian Academy*
{vergi, elena}@racai.ro
[2] *Solomon Marcus Center for Computational Linguistics, University of Bucharest*
monicamihaelarizea@gmail.com
[3] *Faculty of Letters, University of Bucharest*
mihaella.ionescu@yahoo.com
[4] *Institute of Computer Science, Romanian Academy – Iași Branch*
mihaela_onofrei21@yahoo.com

Abstract

The practical session is dedicated to informing the participants, as well as the whole community of specialists, about the creation of a new language resource for Romanian, namely a newspaper corpus annotated with verbal multiword expressions, and about the possibility of developing a system for participating in a competition, i.e. a shared task, of automatic identification of verbal multiword expressions in (multilingual) corpora.

Key words — newspaper corpus, multiword expressions, verbs, Romanian.

1. Outline of the Study

Within the PARSEME cost action (http://typo.uni-konstanz.de/parseme/) there is an initiative of organizing a shared task (as a satellite of the EACL 2017 conference) on the automatic detection of verbal multiword expressions (MWE) in corpora. For this, a corpus annotated with MWEs is necessary, to be split in two parts, one for training and another one for testing the systems participating in the competition. The whole corpus needs to have about 3500-4000 annotated MWEs.

Representatives of 21 languages (Romanian being among them) manifested interest in creating such a linguistic resource. The work had 2 pilot stages in which 400 (tokenised) sentences were annotated with verbal MWEs. The aims of these stages were: (i) to test the initial guidelines against all participating languages and refine them according to the feedback received from the annotators for all languages; please note the great effort of creating guidelines that are not English-centred; (ii) to come up with a set of verbal MWEs specific to each language; (iii) to define each

type of verbal MWEs in such a clear way that the annotators should very easily distinguish among the types and have no hesitation when assigning them to the MWEs found in the corpus; (iv) to create decision trees useful in the annotation process.

For Romanian, the following types of verbal MWEs are used in the annotation:

- universal types:
  - light verb constructions (LVC) – with the structure verb+(preposition+)noun; the verb is (almost) semantically empty and only the noun contributes semantically to the meaning of the whole unit; ex.: a pune o întrebare *to put a question* "to ask";
  - idioms (ID) – with the structure verb+any part of speech; their characteristic is the noncompositionality; ex.: a avea fluturi în stomac *to have butterflies in stomach* "to be in love";
- quasi-universal types:
  - inherently reflexive verbs (IReflV) – ex.: a se gândi "to think";
- other (OTH) – any expression that does not fit the types above: ex.: seamănă, dar nu răsare *resembles, but not springs* "do not resemble".

All these are documented and exemplified (even for Romanian) within the guidelines (http://parsemefr.lif.univ-mrs.fr/guidelines-hypertext).

The corpora will be made available to the whole community. They will be used in this shared task, but they will afterwards be available for anyone interested in the study of verbal multiword expressions or in using the resource for other tasks in computational linguistics.

A tool for the annotation was also necessary and a lot of time was also invested in testing various existing ones, but also in creating a new one. In the end, the FoLiA Linguistic Annotation Tool (FLAT, http://proycon.github.io/folia/) was chosen.

We have chosen Agenda as the Romanian corpus: a journalistic one, IPR-cleared, containing 11,763,223 tokens. Its annotation starts late October 2016 and must end by the end of December 2016.

2. Short Biographies

Verginica Barbu Mititelu is a linguist, working as a senior researcher III degree at the "Mihai Drăgănescu" Research Institute for Artificial Intelligence of the Romanian Academy.

Monica-Mihaela Rizea is a linguist, member of the Solomon Marcus Center for Computational Linguistics, at the University of Bucharest.

Mihaela Ionescu is a linguist, working as a teaching assistant at the Faculty of Letters, University of Bucharest.

Mihaela Onofrei is a linguist, working as a research assistant at the Institute of Computer Science of the Romanian Academy – Iași branch.

Elena Irimia is a computer scientist, working as a senior researcher III degree at the "Mihai Drăgănescu" Research Institute for Artificial Intelligence of the Romanian Academy.

A PERSPECTIVE ON THE EVALUATION OF A SYSTEM OFFERING ENHANCED E-BOOK INTERACTION

IONUȚ PISTOL, DANIELA GÎFU

Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași, Romania

{ipistol, daniela.gifu}@info.uaic.ro

Abstract

This paper describes the envisioned difficulties and the solutions considered in the task of evaluating a newly developed NLP enhanced e-book reader. MappingBooks is a current research project whose final aim is to enhance the way e-books are used by adding linguistic and geographical annotations and references to a text. As a final product of this effort a mobile app is produced which allows users to interact with a geography textbook, visualizing internal and external links in text, seeing relevant maps and web data. Some proposed evaluation scenarios are described in this paper, as well as possible quantitative evaluation criteria and their downsides.

Key words — interactive book, linguistic annotations, geography textbook, web data, evaluation scenarios.

1. Outline of the Project

The practical session presented, in one tempo, issues related to the interoperability of text annotation and software exploiting it. *MappingBooks* (Cook and Reichardt, 1979) is an on-going development project aiming to enhance the way a user interacts with e-books using mobile devices. The initial effort is focused on books with rich geographical references (textbooks and travel guides). Mentions of locations and other relevant entities found in text are automatically identified and associated with external geographical data (maps, attributes, geographical references to other entities) and the web. Links can also be found and marked within the text itself, serving as an alternative way to navigate the e-book. For a detailed description of the considered semantic links please see (Gîfu *et al.*, 2015). The developed application is an Android app designed for tablets, due to the higher screen size, and is aimed specifically at high-school pupils, employing a Geography textbooks as a sample e-book.

Part of the effort carried out in this project is the evaluation of the developed system. Since the considered use-cases involve particular types of users (generally consumers of texts rich in geographical references, such as textbooks and travel guides), evaluating such a system has to consider a qualitative approach measuring user satisfaction. Considering a quantitative evaluation (Cook and Reichardt, 1979) would raise the issues of selecting appropriate static criteria in a largely interactive and dynamic system. Only the base text enriched with annotations can be considered as such, but evaluating it would not offer a relevant measure of the integrated system and its discussed separately in (Cristea *et al.*, 2015).

Considering interactive applications and social environments, evaluation has to be largely qualitative (Grinnell *et al.*, 200), by measuring user satisfaction while interacting with the developed system in various meaningful ways. Considering the envisioned users profiles (highschool pupils), various use-case scenarios are considered, such as performing a lesson assisted by our app versus in a standard way (interacting only with the tutor and the paper textbook). Comprehension of the subjects and retention of important data can later bea measured in a standard graded test. A further test considered is addressed to consumers of travel guides aiming to build and support social communities around travel destinations and events. In this case, direct user satisfaction can be measured in a simple questionnaire and by determining user retention.

2. Short Biographies

Ionuț Pistol is a lecturer at the "Alexandru Ioan Cuza" University of Iași (UAIC), Faculty of Computer Science (FII). He has a PhD since 2011, with a thesis named "The Automated process of Natural Language Discourse" under the supervision of prof. dr. Dan Cristea.

Daniela Gîfu is scientific researcher at the Faculty of Computer Science, and she has got a temporary position of Associate Professor at the UAIC, since 2011. Se has got a PhD in Computer Science and has a PhD in Philosophy (2010). Her current research interests include Natural Language Processing tasks, most of them in correlation with discourse analysis.

Acknowledgements

This survey was published with the support of the PN-II-PT-PCCA-2013-4-1878 Partnership PCCA 2013 grant, having as partners „Alexandru Ioan Cuza" University of Iași, SIVECO Romania, and „Ștefan Cel Mare" University of Suceava and of the grant of the Romanian National Authority for Scientific Research and Innovation, CNCS/CCCDI – UEFISCDI, project number PN-III-P2-2.1-BG-2016-0390, within PNCDI III.

References

Gîfu, D., Pistol, I., Cristea, D. (2015). Corpus of Entities and Semantic Relations with Application in Geographical Domains, at the *11th International Conference Linguistic Resources and Tools for Processing The Romanian Language, ConsILR-2015*, 26-27 Nov. 2015, Iași, Romania, pp. 67-78.

Cristea, D., Pistol, I., Gîfu, D., Anechitei, D. (2016). Networking Readers: Using Semantic and Geographical Links to Enhance e-Books Reading Experience, at the 2nd Workshop on Social Media and the Web of Linked Data, RUMOUR 2016, at the 8th *International Conference on Computational Collective Intelligence Technologies and Applications, ICCCI 2016*, September 28-30, 2016, Halkidiki, Greece.

Cristea, D., Gîfu, D., Niculiță, M., Pistol, I., Sfirnaciuc, D. (2015). A Mixed Approach in Recognising Geographical Entities in Texts. In *Linguistic Linked Open Data*. *12th EUROLAN 2015 Summer School and RUMOUR 2015 Workshop*, Sibiu, Romania, July 13-25, 2015, Revised Selected Papers, © Springer-Verlag Berlin Heidelberg, pp. 49-64.

Cook, T.D., Reichardt, C.S. (eds.). (1979). Qualitative and quantitative methods in evaluation research. Vol. 1. Beverly Hills, CA: Sage publications.

Grinnell, Jr, Richard, M., Unrau, Y. (2005). Social work research and evaluation: Quantitative and qualitative approaches. Cengage Learning.

THE E-CULTFOOD PROJECT

DIANA TRANDABĂȚ[1], PETRONELA SAVIN[2], DANIELA GÎFU[1], ANDREEA MACOVEI[1]

[1]Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iaşi
{dtrandabat, daniela.gifu, andreea.macovei}@info.uaic.ro
[2]"Vasile Alecsandri" University of Bacău
savin.petronela@ub.ro

Abstract

The eCULTFOOD project has as main objective the creation of an "Ethnolinguistic audio-visual atlas of the cultural food heritage of Bacau County" as a comprehensive database containing the results of field research and scientific documentation on cultural food traditions in the region. The project's area of intervention is the intangible cultural heritage of traditional food, fulfilling the function of protection (research, promotion), dissemination (diffusion, including via new models developed in the on-line environment) and supporting, first and foremost, education as cultural intervention. Its main aim is to preserve, in cartographic and computerized form, a representative corpus of audio-visual documents recording the traditional food cultural heritage based on surveys involving the older generation from the rural county of Bacău.

Key words — etnolinguistic atlas, culinary partrimony.

1. Outline of the Project

Digitization of the traditional food heritage responds to actual EU policies aspirations considering cultural resources a key factor in improving accessibility and undivided information flow in an economy of knowledge.

The project transfers the latest achievements in the field of geolinguistics researching intangible heritage (Sprachatlas und des Dolomitenladinischen angrenzender Dialekte[1], Atlas linguistique audiovisuel du Valais Romand[2], Atlas linguistic audiovisual de Bucovina[3]). Project team members have experience with the digitization of the Romanian Language Tesaurus (eDTLR) and the New Atlas of Romanian Language - Moldova and Bucovina, as well as with collecting a complex collection of cultural and linguistic heritage of the region Moldova.

[1] https://www.sbg.ac.at/rom/people/proj/ald/allgemwillkomm.htm
[2] http: // www2. unine.ch/islc/page-35066.html
[3] http://www.philippide.ro/alab/

Audiovisual documents will record real communication situations taking the form of directed conversations from 20 towns in the county of Bacau, based on a questionnaire on traditional diets with 50 topics: knowledge and practices about traditional diets, preparation techniques and traditional recipes, social practices, food-related rituals, traditions and verbal expressions relating to eating, considering language the main vector of cultural, traditional expression. For each investigated area, each topic will be accompagnied by a video having a dialectal phonetic transcription, a literary transcription, an English translation and a set of metadata: cultural, ethnographic, linguistic and thematic references.

eCULTFOOD Atlas will take the form of a web platform developed considering the interoperability of operating systems and internet browsers: the platform will be browseble from any device, desktop or mobile, via a web browser. Each entry of the atlas will be composed of four interconnected sections, providing multiple ways to access the database.

Section 1 (questionnaire) will provide access to the raw questionnaires, properly anonymised;

Section 2 (topic): a webpage dedicated to the vizualization of considered topis, including a brief comment and a representative image resulted from the field survey or documentary research;

Section 3 (map): a Google Map marking the GPS coordinates of the cities where questionnaires took place.

Section 4 (video): allowing visualization of individual videos, with dialectal transcription and translation in English.

Thus, the web interface will be an interactive one, allowing users to have access to comparable ethnolinguistic materials for the entire network of locations in Bacau area. The most important novelty of this database is that the collected material is genuine text, which will transform the audiovisual ethnolinguistic atlas of the cultural heritage in a virtual museum, likely to facilitate innovative approaches in education, economy and ethnographic and linguistic research.

In 2009, the EU Council established a strategic framework for cooperation in education and training, ET 2020, where culture is seen as an essential bridge between younger and experienced generations, by developing creativity and innovation on all age groups.

Acknowledgements

This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS/CCCDI – UEFISCDI, project number PN-III-P2-2.1-BG-2016-0390, within PNCDI III.

References

Savin, P. (2015). Experiential Education by Digitalizing the Live Food Patrimony. In *the International Conference SMART 2014*, Social Media in Academia: Research and Teaching, Editura Medimond, 2015, 319-324.

Savin, P. (2014). Romanian traditional food heritage in the context of urban development. In *Globalization and intercultural dialogue: multidisciplinary perspectives*, Editura Arhipelag XXI, Târgu-Mureş, 920-923.

ON THE PHONEMIC STATUS OF THE ROMANIAN VOWELS Ă [ʌ] AND Â [ɨ]: EVIDENCE FROM LARGE SCALE ACOUSTIC ANALYSIS AND AUTOMATIC SPEECH RECOGNITION

IOANA VASILESCU[1], MARGARET E.L. RENWICK[2], BIANCA VIERU[3], LORI LAMEL[1]

[1] *LIMSI, CNRS, Université Paris-Saclay, France,*
{ioana,lamel}@limsi.fr
[2] *University of Georgia, USA*
mrenwick@uga.edu
[3] *Vocapia Research, France*
vieru@vocapia.com

Abstract

This presentation is dedicated to recent approaches aimed to get insight in spoken Romanian specificities trough effective realization and phonemic status of the vocalic inventory. In particular, I will discuss the phonological properties of **ă** [ʌ][1] and **â** [ɨ] and describe different acoustic and ASR analyses conducted to circumscribe their position in the system.

Key words — automatic speech recognition, acoustic analysis, duration, formants, marginal contrast, phonology, Romanian vowels

1. Outline of the Study

We are interested in the question of the phonological status of the Romanian central vowels **ă** [ʌ] and â [ɨ]. [ʌ] and [ɨ] belong to the vocalic inventory of the Romanian phonetic system along with [a], [e], [i], [o], [u] and for some authors the phonemic diphtongs [e̯a] and [o̯a] (Chitoran, 2002). The vowels [ʌ] and [ɨ] are historical allophones, occur in a nearly complementary distribution and are contextually predictable. They are classically considered full phonemes, being opposed in minimal pairs (e.g. [vʌr] vs [vɨr], [rʌu] vs [rɨu] etc.). According to Renwick (2014), their near complementary distribution and occurrence only in a very reduced list of minimal pairs support the hypothesis of a *marginal contrast* as described by Goldsmith (1995) among others. According to Goldsmith (1995) the phonemic contrast is gradient (instead of categorical) and two sounds may be phonemic at different degrees (ranging from fully contrastive segments to "just barely contrastive sounds" as [ʌ] and [ɨ]).

[1] We adopt here the IPA encoding [ʌ] for orthographic ă (Renwick, 2014). The classical transcription is [ə] (schwa), which may suggest that the vowel is a reduced one. The Romanian mid central vowel being a full vowel, the transcription [ʌ] is aimed to avoid any confusion with a schwa.

We conducted several analyses and experiments to assess the level of contrastiveness and phonological status of [ʌ] and [ɨ]. We took advantage of a corpus of broadcast data gathered from various Romanian radio and television shows (7 hours, 141 speakers), from read speech and more spontaneous interactions such as debates portraying the standard Romanian (Vasilescu *et al.*, 2014). The corpus has been automatically aligned at the phone and word level.

We considered the following aspects : acoustic specificity of [ʌ] vs [ɨ], frequency and lexical distribution of [ʌ] vs [ɨ] in continuous speech, functional load (Hall, 2013) of [ʌ]/[ɨ] pair compared to the segmental inventory of the language, salience of [ʌ]/[ɨ] phonemic distinction for automatic speech recognition of Romanian (Renwick, 2014; Renwick *et al.*, 2016*a*, Vasilescu *et al.*, 2016; Renwick *et al.*, 2016*b*).

During the presentation, I will provide a review of these different actions. I will focus on [ʌ] and [ɨ] acoustic characteristics in continuous speech and on the consequences of [ʌ] and [ɨ] distinction as full phonemes *vs* merger for ASR performance.

I will finally underline the relevance for linguistic studies of automatic speech transcription experiments as a new technique which may highlight how crucial a contrast is for a language.

2. Short Biographies

Ioana Vasilescu, PhD is Researcher Scientist (CR1) in Linguistics at LIMSI-CNRS, France. **Margaret E.L. Renwick**, PhD is Assistant Professor in Linguistics at University of Georgia, Athens, USA. **Bianca Vieru**, PhD is Language Processing Specialist at Vocapia Research, France. **Lori Lamel**, PhD is Senior Research Scientist (DR1) in Computer Science at LIMSI-CNRS, France.

Acknowledgements

This work was partially funded by the ANR VERA project (ANR 12 BS02 006 04).

References

Chitoran, I. (2002). *The phonology of Romanian*: *a Constraint-Based Approach. Studies in Generative Grammar*. Berlin ;New York : de Gruyter Mouton.

Goldsmith, J. (1995). Phonological theory. In *J. A. G OLDSMITH* , Ed., The Handbook of Phonological Theory, p. 1–23. Cambridge, MA: Blackwell Publishers.

Renwick, M.E.L. (2014). The Phonetics and Phonology of Contrast: The Case of the Romanian Vowel System. Berlin, Boston: De Gruyter Mouton.

Vasilescu, I., Vieru, B. and Lamel, L. (2014). Exploring pronunciation variants for romanian speech-to-text transcription. In *Proceedings of SLTU-2014*, 161-168.

Hall, K. C. (2013). A typology of intermediate phonological relationships. *The Linguistic Review*, 30, 215–275.

Renwick, M.E.L., Vasilescu, I., Dutrey, C., Lamel, L. and Vieru, B. (2016). Marginal contrast among Romanian vowels : evidence from ASR and functional load. In *Proceedings of Interspeech*, San Francisco, USA.

Renwick, M.E.L., Vasilescu, I., Dutrey, C., Lamel, L. and Vieru, B. (2016). A phonologically weak contrast caninduce phonetic overlap. In *LABPHON15 - Laboratory Phonology Conference*, Ithaca, USA.

Vasilescu, I., Renwick, M., Dutrey, C., Lamel, L. and Vieru, B., (2016). Réalisation phonétique et contraste phonologique marginal, une étude automatique des voyelles du roumain. In *Actes des Journées d'Etude sur la Parole*, Paris, France.

INDEX OF AUTHORS