PROCEEDINGS
OF THE 8TH INTERNATIONAL CONFERENCE "LINGUISTIC RESOURCES AND TOOLS FOR PROCESSING OF THE ROMANIAN LANGUAGE"
8-9 DECEMBER 2011
26-27 APRIL 2012

Editors

Mihai Alex Moruz

Dan Cristea

Dan Tufiș

Adrian Iftene

Horia-Nicolai Teodorescu

Organizers

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

Research Institute for Artificial Intelligence
Romanian Academy, Bucharest

Institute for Computer Science
Romanian Academy, Iași

National Museum of Romanian Literature
Bucharest

Intelligentics
Cluj-Napoca

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

ISSN 1843-911X

PROGRAM COMMITTEE

Vasile Apopei, Institute for Computer Science, Romanian Academy, Iași branch
Verginica Barbu Mititelu, Research Institute for Artificial Intelligence, Romanian Academy, Bucharest
Alexandru Ceauşu, Research Institute for Artificial Intelligence, Romanian Academy, Bucharest and Dublin City University
Lucian Chișu, National Museum of Romanian Literature, Bucharest
Dan Cristea, Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iaşi and Institute for Computer Science, Romanian Academy, Iași branch
Nicolae Curteanu, Institute for Computer Science, Romanian Academy, Iași branch
Corina Forăscu, Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași and Research Institute for Artificial Intelligence, Romanian Academy, Bucharest
Alexandru Gînscă, Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași
Adrian Iftene, Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași
Eugen Ignat, Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași
Radu Ion, Research Institute for Artificial Intelligence, Romanian Academy, Bucharest
Elena Irimia, Research Institute for Artificial Intelligence, Romanian Academy, Bucharest
Doina Jitcă, Institute for Computer Science, Romanian Academy, Iași branch
Elena Mitocariu, Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași
Alex Moruz, Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași and Institute for Computer Science, Romanian Academy, Iași branch
Maria Moruz, Center for Biblical-Philological Studies "Monumenta Linguae Dacoromanorum", "Alexandru Ioan Cuza" University of Iași
Mircea Petic, Institute of Mathematics and Computer Science, Academy of Sciences of Moldova, Chișinău
Ionuț Pistol, Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași
Laura Pistol, Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași
Radu Simionescu, Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași
Dan Ștefănescu, Research Institute for Artificial Intelligence, Romanian Academy, Bucharest
Diana Trandabăț, Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași and Institute for Computer Science, Romanian Academy, Iași branch
Horia-Nicolai Teodorescu, Institute for Computer Science, Romanian Academy, Iași branch and "Gheorghe Asachi" Technical University of Iași
Dan Tufiș, Research Institute for Artificial Intelligence, Romanian Academy, Bucharest

ORGANIZING COMMITTEE

Lucian Chișu, National Museum of Romanian Literature, Bucharest
Marius Corîci, Intelligentics
Dan Cristea, Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași and Institute for Computer Science, Romanian Academy of Iași
Ionuț Dănilă, Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași
Lucian Gâdioi, Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași
Daniela Gîfu, Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași
Adrian Iftene, Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași
Alex Moruz, Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași and Institute for Computer Science, Romanian Academy, Iași branch
Petru-Adrian Istrimschi, "Alexandru Ioan Cuza" University of Iași
Ionuț Pistol, Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași
Iordan Rață, Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași
Horia-Nicolai Teodorescu, Institute for Computer Science, Romanian Academy, Iași branch and "Gr. Asachi" Technical University, Iași
Dan Tufiș, Research Institute for Artificial Intelligence, Romanian Academy, Bucharest
Eugenia Țarălungă, National Museum of Romanian Literature, Bucharest

TABLE OF CONTENTS

FOREWORD

The Consortium for the Informatization of the Romanian Language (ConsILR) has organized the eighth conference Linguistic Resources and Tools for Processing the Romanian Language in the series, in two sessions, 8 9 December 2011 and 26 27 April 2012. As in the previous edition, the organizers of this event have been: the Faculty for Computer Science of the "Alexandru Ioan Cuza" University of Iaşi, the Research Institute for Artificial Intelligence of the Romanian Academy, Bucharest, the Institute for Computer Science of the Romanian Academy, Iași branch, the National Museum of Romanian Literature, Bucharest and the Intelligentics company from Cluj Napoca.

In order to further extend the visibility of the research dedicated to the Romanian language towards an audience wider than that of strict speakers of Romanian, the editors decided to publish the papers of the Conference in English. Moreover, we believe that many of the papers describing resources and tools applied to the Romanian language do incorporate a sufficiently general approach to be of interest to a wider audience, therefore to attract people engaged in research on other languages than Romanian. We also continue to consider this series of conferences and its volumes as springboards for young researchers, whom we wish to educate in the spirit of exigency, rigor and quality.

There are more and more signs that the scientific community working in natural language processing and computational linguistics is preoccupied to place its scientific knowledge and linguistic data into large repositories that could be accessed by everyone. The huge interest issued by the META-NET consortium of projects (www.meta-net.eu) and the META-SHARE initiative (www.metashare.eu) is representative in this sense. On another hand, international scientific consortia are being born almost every year, in which researchers from different countries design and develop methods and technologies for multilingual applications. Many of these approaches are language independent and are particularised for different languages with the help of adequate resources. The design and the acquisition of language resources (sound records, annotated corpora, electronic dictionaries, language models, treebanks, etc.) usually involve huge and extremely qualified human efforts. Even more, parts of these type of linguistic data are also volatile, due to the evolution of language. As such, they are expensive and should be continuously renewed. The scientific community have only recently begun the think to all these aspects and there are not yet agreed strategies that best commit to the requirements of a computational approach to language studies on long term.

We believe that the papers included in this volume reflect rather adequate the researches pursued recently in Romania and the Republic of Moldavia in the direction of development of resources and tools for Romanian language. In line with the other volumes of the series, this fifth volume includes again chapters dedicated to speech processing, to resources supporting Romanian language processing and to applications.

To stress the significant amount of work dedicated to the creation of resources and applications in lexicography, a special chapter was assigned to this domain.

The editors are grateful to all authors and reviewers who contributed to this volume, as well as to the Faculty of Computer Science of the "Alexandru Ioan Cuza" University, which supported the publishing of this volume.

We wish to our readers a pleasant reading and we invite them to visit the Conference site at: http://consilr.info.uaic.ro/consilr2010/.

Iași, București, Aprilie 2012

The editors

CHAPTER 1
SPEECH PROCESSING

THE RO-TOBI ANNOTATION SYSTEM AND THE FUNCTIONAL ANALYSIS PERSPECTIVE OF THE ROMANIAN INTONATION

DOINA JITCĂ, VASILE APOPEI, OTILIA PĂDURARU

Institute of Computer Science of the Romanian Academy

Iași branch

jdoina@iit.tuiasi.ro

Abstract

The paper presents the RO-ToBI annotation system, used to annotate the Romanian intonation, both as a stand-alone system and as a part of a functional annotation system, by means of which the prosodic contours can be partitioned into a hierarchy of functional units. In the latter case, the RO-ToBI labels are used to annotate the local tonal events of the prosodic units which make up the partitions. This double perspective (hierarchy of functional units and sequence of tonal events) on the intonational contours leads to a more accurate understanding of the Romanian intonational contours.

1. Introduction

Defining an inventory of F0 contour events to describe the Romanian intonation can be justified both by theoretical linguistic needs and practical reasons in speech technology. The popularity of the English ToBI annotation system has increased the trust that, by redefining this system for a different language, the intonation of that language can be better presented in a standard format. As a consequence, various annotation systems have been developed: ToDI for Dutch (Gussenhoven et al., 2003), GToBI for German (Grice et al., 2002), SP-ToBI for Spanish (GrEP_SP, 2009), CAT-ToBI for Catalan (GrEP_CAT, 2009), etc.

The development of the RO-ToBI system labels for the Romanian language is included in the trend of making accessible the understanding of the specificity of the Romanian melodic contours and allowing cross-linguistic comparisons at the intonation level.

Concerning the contour events labeled in RO-ToBI, it has to be underlined that enlarging the standard ToBI system by new labels and by a new prefix, has aimed to highlighting some F0 contour patterns which generate focus events (nuclear accents) within the Romanian intonational contours. Therefore, the intonational events of a contour must not be connected only to the significant pitch movements on the accented syllables or boundary tones. After identifying an event, it has to establish its role within the prosodic group to which it belongs, to identify the position of the focus event.

In section 2, this paper presents the functional perspective of the prosodic group partitioning. The RO-ToBI pitch event and boundary tone inventories are presented in section 3. The way the functional annotations are connected to the tone-based RO-ToBI annotations, to fully describe an intonational contour, is exemplified in section 4.

2. A functional perspective on the prosodic groups

To functionally describe an intonational contour, we correlated the elementary patterns of the F0 contour at the accentual unit level with a set of functions from the communicative act level. We defined a set of functions at the communicative act level and a corresponding set of functional labels, that can be assigned to the prosodic units within an utterance (Jitcă and Apopei, 2007), (Jitcă and Apopei, 2009). Thus, a non-elementary prosodic unit (prosodic group) contains an accentual unit with a high target tone and another one with a relatively low target tone, that have their target tones in a tonal contrast. These accentual units were named PUSH and POP units respectively to suggest their sintagmatic relation. Within the descending contours, the PUSH accentual units mark the beginning of a prosodic group by relatively high tones, while the POP accentual units mark the end of the prosodic group by a return to low tones. In addition to the PUSH and POP accentual units, a prosodic group can also contain a distinct elementary segment, corresponding to a FOCUS type event. However, this one can coincide with the PUSH or POP segment. When the FOCUS event overlaps on a PUSH segment, a PUSH+FOCUS type segment results. When the FOCUS event overlaps on a POP segment, a POP+FOCUS type segment results. Therefore, the functional analysis of an intonational contour generates various types of partitions, defined by sequences of functional accentual units: *PUSH - POP* (for "broad focus" intonations), *PUSH - FOCUS - POP*, (*PUSH + FOCUS) – POP*, *PUSH – (POP+FOCUS)*, etc. The functional perspective of our model on the Romanian intonational contours, leading to partitions of the type presented above, is more general than Ladd's model perspective (Ladd, 1996), which defines "weak-strong" partitions at the prosodic group level.

The "weak-strong" partitioning can be regarded as a "nonfocused-focused" one, that corresponds to our (*PUSH + FOCUS) – POP* or *PUSH – (POP+FOCUS)* functional sequences. The Romanian intonational contour analysis gives rise to other partitioning types that cannot be considered as "weak-strong". The *PUSH - FOCUS – POP* partitioning type is one that occurs more frequently in "broad focus" statement intonational contours. It corresponds to a tonal contrast between the tone targets of *PUSH* and *POP* events. They are both "weak" parts. In this case, the FOCUS event generation does not involve a tonal contrast at the prosodic group level; instead, it is based on a particular F0 pattern characteristic and on the maximal level reached by the target tone within the focused accentual unit.

It is important to understand that the pitch accents with significant pitch movements do not always lead to focus generation. They can have only demarcation functions (PUSH/POP functions). An example illustrating this case is the final pitch accent of a yes-no question intonation with the emphasis in a non-final position that has a significant pitch range but does not carry the phrase focus.

After phrase partitioning, the tonal event annotation with a compatible ToBI system completes the intonational phrase description. The RO-ToBI label inventory is presented in section 3.

3. The RO-ToBI label set

Most of the pitch accent and boundary tone labels presented below are also included in the standard ToBI label set. However, when analyzing a Romanian speech corpus composed of various types of sentences (statements, wh-questions, imperative sentences, vocative sentences, yes-no questions, etc.), we found necessary to introduce new labels and a new prefix ("~") for some types of pitch accent labels.

The labels from the RO-ToBI set can be assigned to contour patterns containing a local event, which can be either a pitch accent or a boundary tone.

The significance of all these labels and how they are used to annotate various F0 contours, corresponding to specific contexts, are also presented in an online guide, at:

http://www.etc.tuiasi.ro/sibm/romanian_spoken_language/RoToBi/RoToBi_System.html

3.1. The pitch accent labels

- H* - is a pitch accent with a high target tone pitch movement. It is often phonetically realized by a large or small rising tonal movement or by a jump to a relatively high target tone during the accented syllable. The H* pitch accent can also have a tendency of keeping the pitch movement close to a high tonal level during a prosodic word. Usually, it is involved in generating a PUSH event at a phrase level. It can generate an emphasis when its target tone reaches the highest tonal level during an accented syllable, within the descending melodic contours.
- L* - is a pitch accent with a low target tone pitch movement. It is often phonetically realized by a large or small decreasing tonal movement or by a jump to a relatively low target tone during the accented syllable. The L* pitch accent can also have a tendency of keeping the pitch movement close to a low tonal level during a prosodic word. Usually, it is involved in generating a POP event in statements, in imperative sentences or in wh-questions. It can generate focuses within the ascending melodic contours (yes-no questions, echo wh-questions) when its target tone reaches the lowest tonal level within a prosodic unit.
- L+H* - is assigned to a contour that begins by a slowly rising movement from a relatively low tone and reaches a high target tone during the following steeper rising movement, at the end of the accented syllable. It generates local or global focuses. If the melodic contours are descending, the global focus is generated during the L+H* pitch accent. If the melodic contours are ascending, the global focus is generated during the L+H* pitch accent when the L tone reaches the lowest level of the tonal space.
- L+>H* - is assigned to a contour that displays a small rising movement from a relatively low tone during the accented syllable and reaches a higher target tone by continuing the rising movement during the next unaccented syllable. The L+>H* pitch accents often rise the pitch contours to high levels within the descending melodic contours (giving rise to prominent PUSH events), without generating focuses.

- H*+>L - is assigned to a contour displaying a high pitch accent followed by a very steep decreasing pitch movement during the next unaccented syllable of the same word or of the next word. It generates a marked tonal contrast between the high target tone during the accented syllable and the low tone reached on the next syllable. Therefore, it emphasizes the word related to the accented syllable. For example, it can be found in wh-questions, emphasizing the wh-word.
- H*+L - is a pitch accent phonetically realized as a prominent high tone with a rising pitch movement at the beginning of the accented syllable, a pitch movement at the highest tonal level and a falling pitch movement on the accented syllable. Then, the F0 contour follows a descending trend on the next non-accented syllable(s). In non-neutral statements, this type of accent generates focuses on the words lying on the descending part of a melodic contour. During this pitch accent, a high target tone (close to the top of the tonal space) is reached.
- H+L* - is a pitch accent phonetically realized as a fall from a relatively high level (reached during the pretonic syllable) to a low target tone, during the accented syllable. In neutral statements, it can generate more prominent POP or POP+FOCUS events.
- L*+H - is a pitch accent phonetically realized as a constant pitch movement at a relatively low level or as a small decreasing pitch movement to a low target tone, during the whole accented syllable or only during its first part. The low pitch movement is followed by a prominent rising movement during the post-accented syllable or only during its last part.

The "~" sign can precede an H* label when the corresponding prosodic word has the same levels for its beginning and ending tones. This event can occur within various F0 contour contexts: (a) when it is placed between two high pitch events, it indicates either a flat F0 contour of the corresponding prosodic word or a small peak pattern; (b) when it is placed between two low tonal events, its F0 contour displays a rising, followed by a falling pitch movement, with prominent peaks generating focuses of various strengths.
Similarly, a low level plateau pitch accent can be annotated by a ~L* label when the prosodic word has the same level at its beginning and its end. For the ascending contours, the ~L* labels can be assigned to the word in the focus position when the tone reaches the lowest level of the tonal space.

3.2. The boundary tone labels

In Romanian, the boundary tones can be either monotonal (low-L%, high-H%, medium-M%) or bitonal (LH%, HL%) as follows:

- L% - The Low boundary tone can be found at the end of the descending melodic contours of statements, wh-questions, imperative wh-questions and imperative sentences;
- H% - The High boundary tone can be found in confirmation-seeking yes-no questions, imperative echo wh-questions, conter-expectational echo yes-no questions and imperative yes-no questions;
- M% - The Medium boundary tone can be found in some vocative sentences;

- LH% - The Low-High boundary tone sequence can be found in some information-seeking yes-no questions with the emphasis in the final position, imperative yes-no questions and neutral echo yes-no questions;
- HL% - The High-Low boundary tone sequence can be found in information-seeking yes-no questions with the emphasis in a non-final position and in certain vocative sentences.

4. Case studies in intonation description, from the combined perspective of the functional model and the RO-ToBI system

In what follows, we shall analyze the intonational contours corresponding to several types of Romanian enounces, to illustrate how the RO-ToBI labels and the functional labels complete each other to describe these contours. The contours are extracted from a speech corpus built for the Romance ToBI Workshop (Jitcă , et al., 2011) and based on a questionnaire containing 31 sentence types. Each sentence type was uttered by three speakers.

The contours in Figs. 1 and 2 correspond to two utterances of the affirmative statement "Maria mănâncă mandarine" ("Maria is eating tangerines."), with a subject-verb-object syntactical structure. The contours are extracted from the utterances of two speakers. Their differences are related to the particular melodies that the speakers currently use in uttering the "broad focus" statements.

The intonational contours are composed of a sequence of three AUs (prosodic words). For the first contour, the RO-ToBI sequence of pitch accents starts with an L+>H* type one (an accent that does not generate focuses) and ends in an L* type accent, which does not generate focuses in descending intonational contours. Their target tones define the tonal contrast of the intonational phrase. Their corresponding prosodic words are the PUSH (PH) and POP (PO) events. The pitch accent in the medial position, corresponding to the verb "mănâncă" ("is eating") generates a focus by a L+H* pitch accent, within a prosodic word having the same level for its beginning and ending tones (~L+H*).

The functional description of this contour, given by a PUSH – FOCUS – POP (PH/F/PO) sequence, highlights the medial position of the focused word. The result of combining the two annotations is given by (1), where the RO-ToBI labels are indices for the corresponding functional labels, giving rise to a specific PH/F/PO partition.

$$PH_{L+>H^*} /F_{\sim L+H^*}/PO_{L^*} \quad (1)$$

The second contour differs from the one previously presented by a more pronounced PUSH unit (without changing the focus position), leading to a less focused verb. The stronger prominence of the PUSH unit is the result of changing the L+>H* pitch accent into an L*+H accent. The decrease in F0 frequency variation during the second prosodic word has led to a reduced focus on the verb. From a functional point of view, this contour is described by the same general sequence (PH/F/PO), accompanied by RO-ToBI labels as in (2).

$$PH_{L^*+H} /F_{\sim H^*}/PO_{L^*} \quad (2)$$

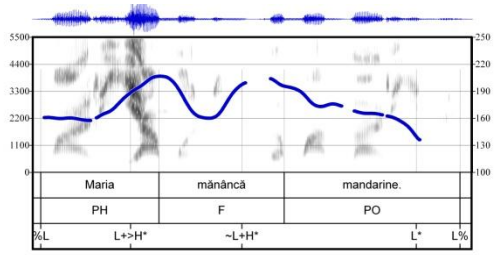

Figure 1: Waveform and F0 contour of the utterance of the affirmative statement "*Maria mănâncă mandarine*" ("Maria is eating tangerines."), produced by speaker A, with an intonational contour described by (1).

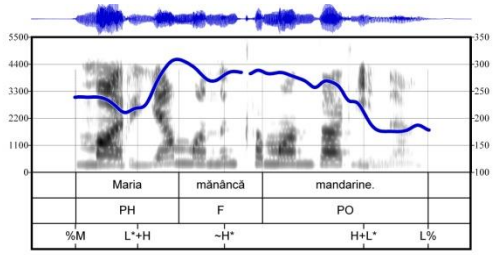

Figure 2: Waveform and F0 contour of the utterance of the affirmative statement "*Maria mănâncă mandarine*" ("Maria is eating tangerines."), produced by speaker B, with an intonational contour described by (2).

In a second example, we shall analyze two intonational contours, corresponding to the confirmation-seeking yes-no question "*Vei veni să mănânci, nu?*" ("You will come for dinner, won't you?"), illustrated in Figs. 3 and 4. The contours have been extracted from the utterances of the two speakers. Their differences can be related to the particular melodies that the speakers currently use in this dialogue context. Within a real dialogue scene, one speaker can be dominated by the "to come" action and the other one, by the "eating" action (expressed by the word "dinner").

The intonational analysis has led to the following two sequences, given by (3) and (4) respectively.

$$(PH+f_{H*}/PO_{L*})_{PH}/PO+F_{L*+H} \quad (3)$$

$$(PH_{H*}/PO+f_{\wedge H+!H*})_{PH}/PO+F_{L+H*} \quad (4)$$

The PH+F functional label corresponds to a PUSH + FOCUS event (including the global focus), at the intonational phrase level. The PH+f functional label also corresponds to a PUSH + FOCUS event (including a local focus), of the lower level prosodic group.

These descriptions show explicitly that the two contours are structured by two levels. The group on the inferior level works as a PUSH event for the intonational phrase. The global focus of this interrogative intonational phrase is placed on the final position and it is generated within the PO+F event corresponding to the negation word "nu". The F0 pattern of the PO+F unit generates the ascending phrase-final contour specific to the yes-no questions.

The inferior group has a statement type intonation, corresponding to a verbal group. Since we deal with a statement, the word reaching the tonal maximum on the accented syllable carries the focus – in our case, the local focus at the inferior group level.

For the first utterance, the intonational segment of the inferior group emphasizes the first verb (an active verb) of the verbal group, while for the second utterance, it emphasizes the subjunctive verb.

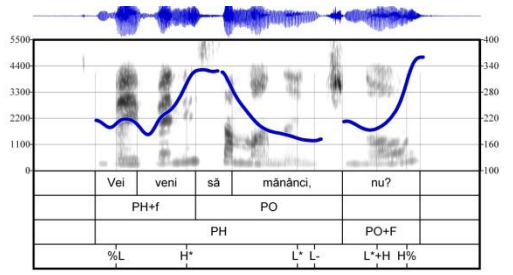

Figure 3: Waveform and F0 contour of the yes-no question "*Vei veni să mănânci, nu?*" ("You will come for dinner, won't you?") with an intonational contour described by (3).

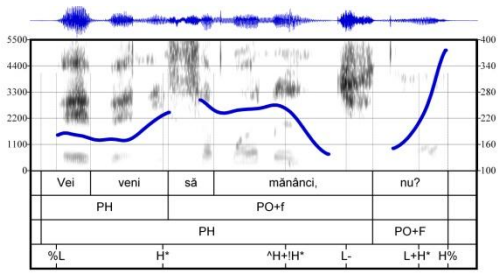

Figure 4: Waveform and F0 contour of the yes-no question *"Vei veni să mănânci, nu?"* ("You will come for dinner, won't you?") with an intonational contour described by (4).

In the first contour, the H* type target tone reaches the maximum level of the tonal space (the second pitch accent is an L* pitch accent). In the second contour, the same H* event is followed by an even higher target pitch accent and it lost the emphasis. Consequently, the focus is generated within the POP event resulting a PO+f event.

The third example illustrates a contour corresponding to the imperative sentence "*Vino aici, te rog!" ("Come here, please!")*- Fig. 5. Similarly to the statements, its contour is descending, but the high target tones reach more elevated levels. Its functional description is given by the PH/F/PO general sequence and its particular instance is given by the RO-ToBI labels, as specified in (5).

$$PH_{L+>H^*}/F_{H^*+>L}/PO_{L^*} \quad (5)$$

The functional sequence is the same to the one corresponding to the statement in the first example, except for the fact that now, the focus in the medial position is actually an emphasis generated by the contrast between the high target tone during the word "aici" ("here") and the low tonal level at the beginning of the next word ("te").

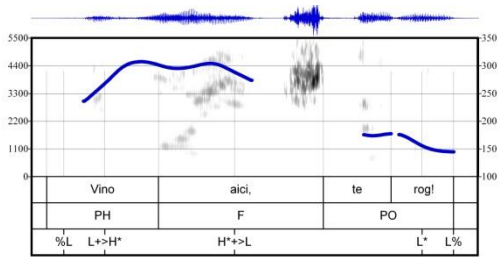

Figure 5: Waveform and F0 contour of the utterance of the imperative sentence "*Vino aici, te rog!"* (*"Come here, please!"*), with an intonational contour described by (5).

The RO-ToBI system can describe more accurately the contour pattern giving rise to the emphasis in this particular case, using an H*+>L type label. Despite the break that occurs before the polite request ("te rog" - "please") it does not divide the intonational phrase. The prosodic word corresponding to the polite request generates the POP event

and facilitates the emphasis generation by the high pitch accent of the word in the medial position.

5. Conclusions

We conclude that using the RO-ToBI labels leads to an intonational description closer to the phonetic characteristics of the Romanian intonational contours. In addition, the functional analysis of the intonational contours, materialized in the partitioning of the prosodic groups, successfully completes the RO-ToBI description. The functional perspective of our model on the Romanian intonational contours, leading to partitions of the type presented in section 2, is more general than Ladd's model perspective which defines "weak-strong" partitions at the prosodic group level. It is important to understand that the pitch accents with significant pitch movements do not always lead to focus generation. They can have only demarcation functions (PUSH/POP functions). The levels of the target tones reached within focus events are close related to the maximum or minimum tonal levels of the tonal space.

Therefore, these descriptions lead to a more accurate understanding of the Romanian intonational contours within comparative cross-linguistic studies and in designing syntactico-prosodic or prosodico-acoustic interfaces for Romanian speech synthesis.

References

Grice, M., Baumann, S. (2002). Deutsche Intonation and GToBI. *Linguistische Bericthte*, 267-298. http://www.coli.uni-saarland.de/publikationen/softcopies/ Grice:2002: DIG.pdf

Gussenhoven, C., Rietveld, T., et al. (2003). Transcription of Dutch Intonation. http://todi.let.kun.nl/ToDI/home.html

GrEP_Sp (Group of Prosodic Studies - Spanish) (2009). http://prosodia.upf.edu/ sp_tobi/en/labeling_system/labeling_system.html

GrEP_Cat (Group of Prosodic Studies - Catalan) (2009). http://prosodia.upf.edu/ cat_tobi/en/labeling_system/labeling_system.html

Jitcă, D., Apopei, V. (2007). Corpus de voce pentru limba română adnotat cu etichete funcționale la nivelul unităților de accentuare. *Lucrările atelierului "Resurse lingvistice și instrumente pentru prelucrarea limbii române"*, Iași, 31-39.

Jitcă, D., Apopei, V., Jitcă, M. (2009). The F0 contour Modeling as Functional Accentual Unit Sequences. *International Journal of Speech Technology*, 12: 2-3, 75-82.

Jitcă, D., Apopei, V., Păduraru, O. (2011). Transcription Of the Romanian Intonation-Ro_TooBI, *Workshop on Romance ToBI (PaPI 2011)*, June, Tarragona, http://prosodia.upf.edu/activitats/wromtobi/home/

Ladd, D. R. (1996). Intonational Phonology, Cambridge: Cambridge University Press, 1996.

BLIND SPEECH SEGMENTATION APPLIED TO THE ROMANIAN LANGUAGE

TIBERIU BOROȘ

Romanian Academy Research Institute for Artificial Intelligence

tibi@racai.ro

Abstract

The creation of large scale speech databases requires speech segmentation and time alignment with text and phonetic transcriptions. For this purpose we created a semiautomatic tool that uses a method called "blind speech segmentation".

1. Introduction

Building large scale speech databases is crucial to any research in the field of speech technology for the Romanian language. By our knowledge, the largest freely available speech database is the Romanian Speech Synthesis (RSS) database (Stan et al., 2011). It consists in about 4 hours of time aligned recordings, text and transcriptions. The creation of spoken corpora starts from pre-recorded speech and implies labeling the boundaries for each allophone found in the recordings. This is not a task that can be done manually and requires some method of automatic speech segmentation. Common such methods use the output of automated-speech-recognition (ASR) software, employing refinement techniques over the raw boundaries (Sethy and Narayanan, 2002; Kim and Conkie, 2002; Jarifi et al., 2008). Due to the lack of freely available resources for the Romanian language and in order to speed up the boot-strapping process for speech recognition, we created a tool (Speech Labeling Tool – SLT) for semiautomatic labeling of speech units. SLT handles the labeling in two steps. The first step uses a technique called *"blind speech segmentation"* (according to Sharma and Mammone, 1996) (Aversano et al., 2001; Cherniz et al., 2007; Almpanidis and Kotropoulos, 2008; Wang et al., 2003) which is useful in the absence of ASR software. Because this method creates non-uniform speech units (diphones, triphones, syllables or even words) a second step is necessary for labeling at phoneme level, using a different technique. This paper covers only the "blind speech segmentation" part of our process.

2. Overview of the blind speech segmentation

Our implementation of blind speech segmentation uses clustering to group similar acoustic frames and dynamic time warping (DTW) to create a time alignment between clusters and text units. We used the Zero Crossing Rate (ZCR), the Energy (E), the Spectral Variation Function (SVF) and the number of peaks detected in the spectrum (PC) for the calculus of the similarity function between clusters.

Some text pre-processing is required in order to extract tokens from the text that can be easily aligned with acoustic frames (see section 2.2). As a remark, this is the only part where this method is not language independent.

The block diagram of the system is presented in figure 1. The text processing block and the clustering block are used by the DTW to create an optimal alignment between text tokens and acoustic frames.

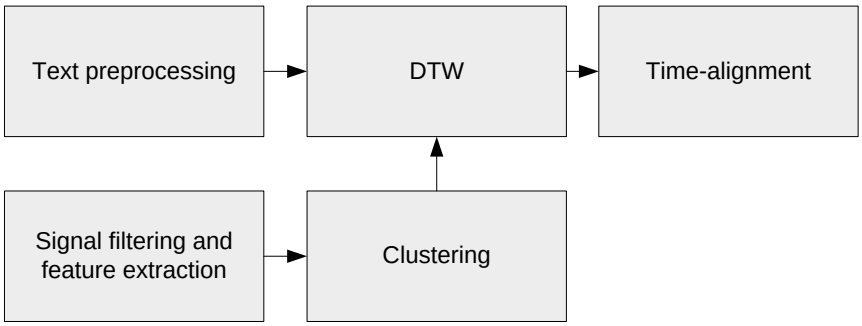

Figure 1: System overview

The basic steps of the method are: signal filtering and feature extraction (section 2.1); text pre-processing (section 2.2); clustering applied to acoustic frames (section 2.3); DTW alignment (section 2.4).

2.1. Signal filtering and feature extraction

A few generic steps specific to digital signal processing (DSP) were taken before performing feature extraction on the acoustic data. First, because in speech most of the energy is located between 0-4Khz, a low-pass filter with the frequency response graphically illustrated in figure 2 was used on the input data. The filter was designed using an online Kaiser-Bessel filter generator written by A.R. Collins[1]. The parameters for the filter were: F_a=0 F_b=8 KHz, attenuation 98dB and filter order 37.

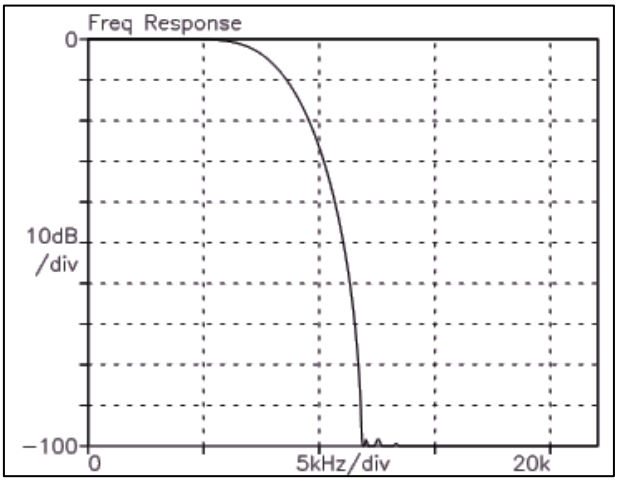

Figure 2: Frequency response

After the filter is applied, the signal is split into 20ms overlapping segments using a Hamming window function (equation 1), to narrow down the spectral leakage effect (Burileanu and Dan, 2000). Each signal window will be treated as an individual cluster later in the process.

[1] http://arc.id.au/FilterDesign.html

$$w(n) = \begin{cases} 0.54 - 0.46\left(1 - \cos\frac{2\pi n}{N-1}\right), for\ 0 \le n \le N-1 \\ 0\ , otherwise \end{cases} \quad (1)$$

The features extracted from the acoustic frames are: the zero crossing rates (equation 2), the mean energy contour (equation 3), the spectral peak count and the spectral variation function (equation 4 and 5).

$$ZCR_m = \frac{1}{2}\sum_n |\mathrm{sgn}\{x_{n-m}\} - sgn\{x_{n-1-m}\}| \quad (2)$$

$$\mathrm{sgn}\{x_n\} = \begin{cases} 1, if\ x_n > 0 \\ -1, if\ x_n < 0 \end{cases}$$

$$\mathrm{E_m} = \frac{1}{\mathrm{N}}\sum_{\mathrm{n}}\left(\mathrm{w_{(n)}x_{n-m}}\right)^2 \quad (3)$$

$$V_k = \sum_n w_{(n)} x_{n-m} \quad (4)$$

$$SFV(Q,P) = \sqrt{(Q_1 - P_1)^2 + (Q_2 - P_2)^2 + \cdots + (Q_n - P_n)^2} \quad (5)$$

2.2. Text pre-processing

Before we can proceed with clustering and DTW we need to do tokenization and encoding on the input text data.

Tokenization has to do with the fact that smooth transitions are present between adjacent phonemes, especially when they are part of the same syllable. This happens even when the two phonemes are part of different words in a sentence (e.g. "această_afacere" English: "this affair"). Also, if a word ends in a certain phoneme and the next word starts with the same phoneme, the two words share this phoneme between them. The tokenization is used to group letters into what we call pseudo-syllables and helps the clustering and DTW blocks in their tasks. This is done using a reduced rule based syllable splitter that keeps adjacent vowels together ignoring the hiatus rule.

Encoding is linked to the fact that DTW requires a way to measure the similarity between the elements of the two sequences that it aligns. Types of sounds (vowels, fricatives, plosives etc.) have similar acoustic behavior, so we treated letters that represent phonetic sounds of the same type in a similar manner when comparing them to acoustic frames.

Our tokenization (extraction of pseudo-syllables) is done as follows:

1. The replacement of the grapheme "x" with "cs";
2. The replacement of "*ch*", "*gh*", "*c*"-i, "*c'*'-e, "*g'*'-i, "*g'*'-e with "~";
3. The elimination of hyphens (which just show that the two words should be spoken together);
4. Applying all the rules except for the hiatus rule for syllable splitting and treating the "~" symbol as a predetermined syllable;
5. After syllable splitting we replaced "~" with the original sequence.

For example, the units obtained from the text "cine este acolo" (English "who's there") are: "ci", "ine", "es", "te", "a", "co", "lo". Our text encoding method that produces the output for the DTW block is described below:

1. As a preliminary step we added white spaces at the beginning and the end because recordings usually start and end with silence;
2. White spaces where encoded with *"S"*;
3. Groups like *"ce", "ci", "ghe", "ghi" "che", "chi", "ge", "gi",* which have a similar pronunciation behavior where encoded with the symbol *"D"*;
4. We used *"V"* for vowels (*"a", "e", "i", "o", "u", "ă", "â", "î"*);
5. Graphemes *"f", "s", "ș", "j", "h", "ț", "r", "v", "z", "m", "n", "l", "r"* where encoded with *"F"*;
6. Graphemes *"p", "b", "t", "d", "c", "g"* where encoded with *"P"*.

Our task was simplified by the fact that the Romanian language has a preponderantly phonemic orthography, therefore, we did not need to include a phonetic transcription module in our tool.

2.3. Clustering

We used agglomerative bottom-up clustering and modified the algorithm to take into consideration only adjacent audio frames, because of the time-domain restriction

An aspect of any clustering implementation is how to know when to stop merging (bottom-up) or dividing (top-down) clusters. It is obvious that the target number of clusters should be at least equal with the number of pseudo-syllables previously obtained. This is why the exit clue from the clustering loop was given by the number pseudo-syllables multiplied by a constant value[2] (k). The similarity between two clusters was measured using a distance function calculated as a weighted sum of the normalized values for ZRC, E, PC and SVF (equations 6 and 7)

$$D_{ij} = \alpha\left|ZRC_i - ZRC_j\right| + \beta\left|E_i - E_j\right| + \gamma SVF_E(V_i, V_j) + \delta\left|PC_i - PC_j\right| \quad (6)$$

$$\alpha = 0.2, \beta = 0.2, \gamma = 0.4, \delta = 0.2 \quad (7)$$

2.4. Dynamic Time Warping alignment

DTW has been used before in speech processing to align sequences based on a similarity between their elements. To express the similarity between text symbols (the encoding for the text) and clusters we based our calculus on two parameters (E and ZCR) mainly because they were already extracted at a previous step. We normalized the values as ER for E and ZCRR for ZCR. To have a reference point we calculated the average values for ZCRR and ER from a manually aligned corpus for each of the 5 symbols used in our encoding (see table 1 for results). These are the expected values for ZCRR (EZCRR) and ER (EER) for a given symbol in the encoding. We expressed the alignment cost between a symbol and a cluster as the Euclidian Distance between points given by the coordinates (EZCRR, EER) (for the symbol) and (ZCRR, ER) (for the cluster) (equation 8). Each time the alignment results are validated by a human expert, SLT updates the average values of the ZCR_R and E_R.

[2] After some experiments with different numbers we came to the conclusion that the minimum value should be 3 (partially because it is the average number of sounds in a syllable)

Table 1: Expected values for ZCR_R and E_R

Symbol	EE_R	$EZCR_R$
D	0.464	0.652
V	0.466	0.112
F	0.118	0.660
S	0.030	0.180
P	0.085	0.065

$$D = \sqrt{(\mathrm{EZCR}_R - \mathrm{ZCR}_R)^2 + (\mathrm{EE}_R - E_R)^2} \quad (8)$$

Some alignment errors are solved using a set of simple heuristics. In cases where more text tokens are aligned with the same acoustic cluster, they are reshaped to form a bigger unit. An example is the case of "vine mama" (English "mother comes"). The resulting text units are: *"vi", "ne", "ma", "ma"*.

- The encoding is FVFVSFVFV

In this particular case, we turned the units "vi" + "ne" to "vine" and "ma" + "ma" to "mama" as shown in figure 4. The test file was downloaded from Technical University of Iași – Sounds of the Romanian Language Corpus (Teodorescu et al., 2011).

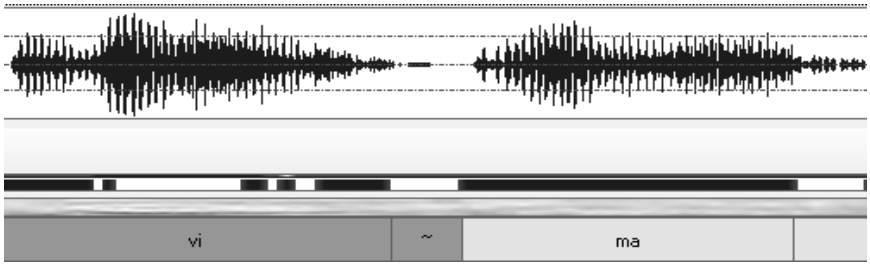

Figure 4: Alignment for *"vine mama"*

3. Testing and conclusions

Because this is a semi-automatic tool, a good estimate of the performance of the system has to reflect the level of human intervention required in the alignment process. To test our tool, we extracted non-uniform speech units from a recorded corpus. Each time a segment was edited, SLT would realign the other units. All the manually aligned segments have their position fixed and there is no need to realign them. The measure of effort (EF) was given by the number of manual edits versus the total number of segments extracted from the recorded data. We used a set of 103 sentences. The total number of tokens obtained from the test set was 1465 with a number of 339 manually edited segments. Using equation 9 the total effort was 23%.

$$EF = \frac{number\ of\ manual\ edits}{number\ of\ tokens} \quad (9)$$

Collecting large corpora of speech units is crucial to all domains of speech processing. This is a laborious task, especially when manual segmentation of phonetic boundaries is

employed. This paper presents the blind speech segmentation technique we used to create a tool for collecting non-uniform speech units. Manual intervention is mandatory for the fine tuning of the segment boundaries, but this is less time consuming then full manual segmentation.

References

Almpanidis, G., Kotropoulos, C. (2008). Phonemic segmentation using the generalized Gamma distribution and small sample Bayesian information criterion. In: *Speech Communication*, 50, 38-55.

Aversano, G., Esposito, A., Esposito, A. and Marinaro, M. (2001). A New Text-Independent Method for Phoneme Segmentation. In: *Proceedings of the Institute of Electrical and Electronic Engineers International Workshop on Circuits and Systems*, Dayton, Ohio, USA.

Burileanu, D., Dan, C. (2000). Fundamentals and Basic Techniques in Digital Signal Processing, *Printech Publication*, Bucharest, ISBN 973-652-127-3.

Cherniz, A. S., Torres, M.E., Rufiner, H.L., Esposito A. (2007). Multiresolution Analysis Applied to Text-Independent Phone Segmentation. In: *Journal of Physics: Conference Series*, 90, 1-7.

Kim, Y.J., Conkie, A. (2002). Automatic segmentation combining an HMM-based approach and spectral boundary correction. In: *Proceedings of the 7th International Conference on Spoken Language Processing*, 145-148.

Jarifi, S., Pastor, D., Rosec, O. (2008). A fusion approach for automatic speech segmentation of large, corpora with application to speech synthesis. In: *Speech Communication*, 67-80.

Sethy, A., Narayanan, S. (2002). Refined speech segmentation for concatenative speech synthesis. In: *Proceedings of the 7th International Conference on Spoken Language Processing*, 149-152.

Sharma, M., Mammone, R. (1996). ”Blind” speech segmentation: automatic segmentation of speech without linguistic knowledge. In: *Proceedings of the 4th International Conference on Spoken Language Processing*, 2, 1237-1240.

Teodorescu, H. N., Pistol, L., Feraru, M., Zbancioc, M., Trandabăț, D. (2010). Sounds of the Romanian Language Corpus. *“Gheorghe Asachi” Technical University of Iași*. Retrieved from: http://www.etc. tuiasi.ro/sibm/romanian_spoken_ language/ index.htm. Access date: April 2, 2011.

Wang, D., Lu, L., Zhang, H. J. (2003). Speech segmentation without speech recognition. In: *Proceedings of the Institute of Electrical and Electronic Engineers International Conference on Acoustics, Speech and Signal Processing*, 468–47.

CHAPTER 2
LANGUAGE PROCESSING RESOURCES

CASUISTRY OF ROMANIAN FUNCTIONAL DEPENDENCY GRAMMAR

CENEL AUGUSTO PEREZ

"Al. I. Cuza" University, Faculty of Letters, Iași – Romania

"Al. I. Cuza" University, Faculty of Computer Science, Iași – Romania;

augusto.perez@info.uaic.ro

Abstract

In this paper we discuss a series of problems met during the process of syntactic annotation of approximately 4000 Romanian sentences. The intention is to put in evidence controversial situations of syntactic annotation and to discuss solutions. The annotation process is part of a large project that has as purpose the achievement of a Romanian Treebank.

1. Introduction

One of the most challenging issues in the filed of Natural Language Processing, both for linguists and for computer scientists, is the lack of linguistic resources in electronic form. In order to collect them, linguists and specialists in computer science need to find ways to collaborate and to find a common understanding of the problems connected to the complexity and difficulty of the language.

A treebank is a corpus of texts where each sentence is annotated for syntactic structure. This syntactic structure is usually represented as a tree structure (hence the name of "treebank"). Our purpose was to create a Romanian Treebank, as a collection of texts selected from a wide range of registers of the language, that could be used to train, test and evaluate a syntactic parser for the Romanian language. This type of parser is, presently, under development in a collaborative research at the Institute for Computer Science of the Iași branch of the Romanian Academy and the Faculty of Computer Science of the "Alexandru Ioan Cuza" University of Iași.

There are known treebanks for languages like: Chinese, French, German, Italian, Japanese and others. The most famous ones are: Penn Treebank – built at the Pennsylvania University, Philadelphia (Marcus et al., 1993) and containing over 4.5 million words of American English, with more than half of it syntactically annotated, and Prague Dependency Treebank – built at the Charles University of Prague (Hajič et al., 2001) for Czech.

There is no standard treebank for the Romanian language so far, but we can mention some researches that had as purpose the development of such a data base. For example, Călăcean and Nivre (2008) report the development of a treebank, (4,042 sentences including 36,150 tokens[1]). There is a limitation in the complexity of the syntactic structure, as there are no subordinate clauses. Although the texts were said to be chosen in such a way that they would offer a representative sample of the modern written

[1]Token – a word or any other element from a line of characters.

standard language, their collection includes only texts from newspaper articles (specially, political and administrative topics). Each sentence has an average of 8.94 tokens.

A very important study for the syntax of the Romanian language was developed at the University of Geneva. The authors of the research (Sereţan et al., 2009) built a syntactic parser, for the Romanian language, which does not have a Treebank as a data base (the resources of the parser are a lexicon and a set of grammatical rules). However, the product, obtained after the automatic processing of some texts can be considered a set of parsed trees.

Finally, an approach of building a Romanian Treebank is described also by Florentina Hristea and Marius Popescu (2003), from which we have adopted the main part of the relation names. In this paper we are mainly interested to present examples of syntactic analyses that usually are taken as problematic or controversial.

2. Syntactic structures

The type of syntactic annotation we are referring in this paper is the one recommended by Tesniére (1959).

The annotation process is begun by elaborating a list indicating possible relations that link subordinate words to their heads as well as part of the surrounding context. To this list many other entries have been added during the syntactic annotation, as new cases have been discovered. For example, in the table below (first row), the head is a noun. Its significance is that the corpus contains at least one sentence that includes a noun followed by a preposition, which itself is followed by an adverb. In the dependency tree, on the arrow between the noun (*plimbarea*) and the preposition (*de*), the adverbial attribute (*a.adv.*) will appear, like in the example: *Plimbarea de azi*, because *de azi* is an adverbial attribute for the noun *plimbarea*.

	HEAD	SUBORDONATE WORD	FOLLOWED BY	RELATION	ABREVIATION	EXAMPLE
1.	Substantiv	prepoziţie	adverb	atribut adverbial	a. adv.	Plimbarea de azi
2.	Substantiv	prepoziţie	verb nefinit	atribut verbal	a. vb.	Maşina de spălat
3.	Substantiv	prepoziţie	nominal	atribut substantival	a. subst.	Praf de puşcă
4.	Substantiv	numeral	-	atribut adjectival	a. adj.	Doi oameni
5.	Substantiv	articol	-	determinare	det.	Un om
6.	Substantiv	adjectiv	-	atribut adjectival	a. adj.	Fată tânără
7.	Substantiv	pronume	-	atribut adjectival	a. adj.	Casa lui

Figure 1: Part of the list with the dependency relations

The method for describing the syntactic structure of the natural language sentences is that of the D-trees (trees resulted from using a dependency grammar in the syntactic analysis).

Between a word and its neighbours the mind perceives connections. The sum of those connections forms the structure of the sentence. The structural connections establish dependency relations between words, and each connection, basically, unifies a *superior*

term with an *inferior* one. The superior term is usually called *head* and the inferior term is the *subordinate*.

In most of the cases (see in the example of Figure 2), the root of the tree should be the predicative verb of the sentence. If the sentence contains more than one clause which are in subordination relation then the predicative verb of the main clause will be the root. As we will see in the following section, in case more clauses are in a coordination relation at the most superior level, then a conjunction will be chosen as the root of the tree. Inside a clause, once the main element, the predicative verb, is placed in the root position, the next step is to look for the subject and the objects, which will become immediate descendents of the respective verb.

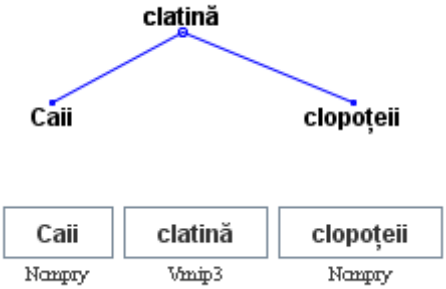

Figure 2: Predicative verb – the central element in the tree (the ROOT node)
(*The horses shake the bells*)

Figure 2 represents a simple example of determining a dependency structure of a sentence. In the case of more complex sentences and that of more clauses, the building of the tree is done top-down. The classical style of pursuing a syntactic analysis, i.e. starting from the main element and following with the descendents, is guiding the annotation process.

Next we will present a bunch of examples, pointing out elements which, in a FDG type of analysis, are different then in a classical syntactic analysis.

3. Identifying the structure

3.1 Coordination

A coordination relation is established between two or more sentences or between parts of one sentence. In our treebank we have observed the following convention: the coordinative element (punctuation – comma, colon, and semicolon, or a coordinative conjunction) is taken as the head of the two coordinated elements and will therefore be placed in the root position with respective to the two elements. When more than two elements are in a coordination relation, then this rule is applied recursively at each level. Since the coordinative element (punctuation, conjunction) generally coordinates two elements and since the resulted tree should be totally connected, we decided that the first coordinative element should be the head of the coordinated element from its left and of the coordinative element from its right. Then this last substructure is repeated as many times as needed, resulting in N coordinative elements that coordinate N+1 elements.

The example in Figure 3 shows such a structure in which three commas coordinate four elements. We can notice that the first comma is the head of the first coordinated element (*la curățat – to clean*) and of the next comma, that this one, on its turn, is the head of the following coordinated element (*la spălat – to wash*) and the next coordinative comma, and that this one coordinates the last two elements (*la făcut focul – to make fire* and *la cărat peștele – to carry the fish*).

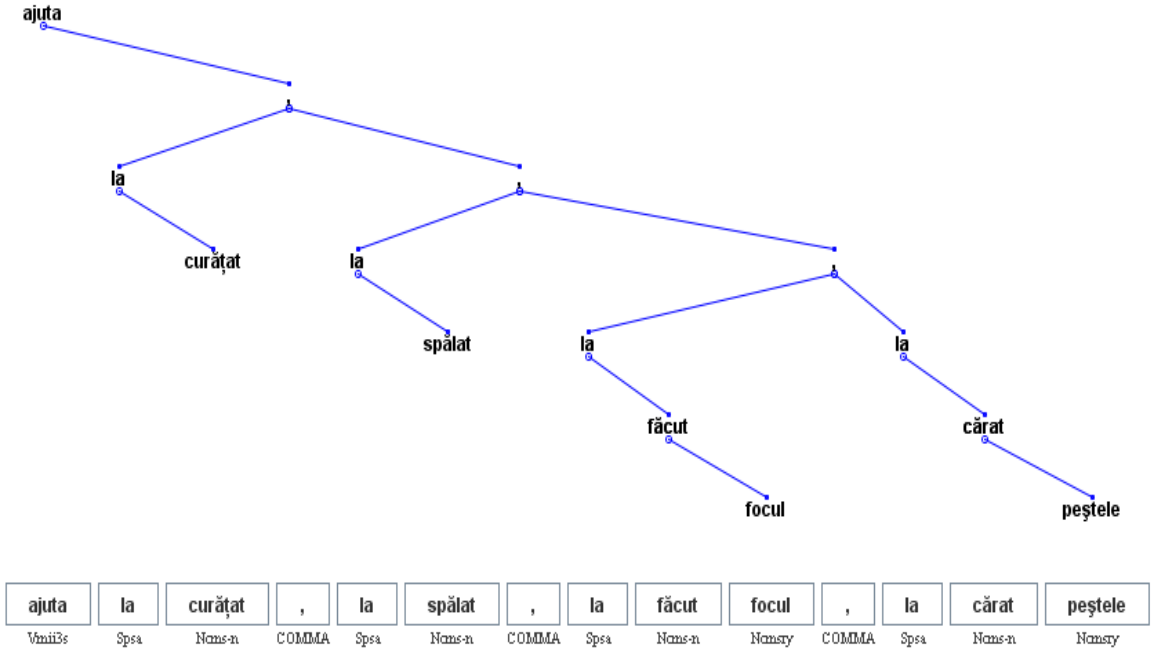

Figure 3: Coordination of more than two elements
(*It helps to clean, to wash, to make fire and to carry the fish.*)

3.2 Punctuation

Same as the exclamation mark and the question mark, the full stop is a direct decendent of the central element (the root node) of the sentence. In similar dependency relations we place the converted commas (graphical signs used when we want to render a text exactly the same way as it was said by someone) at the beginning and ending of a clause/phrase, meaning that they should be descendants of the same head as the respective clause/phrase. The example in Figure 5 presents such a separation function of commas, where two commas graphically divides some parts of the sentence (or of a clause) from the rest of its constituents.

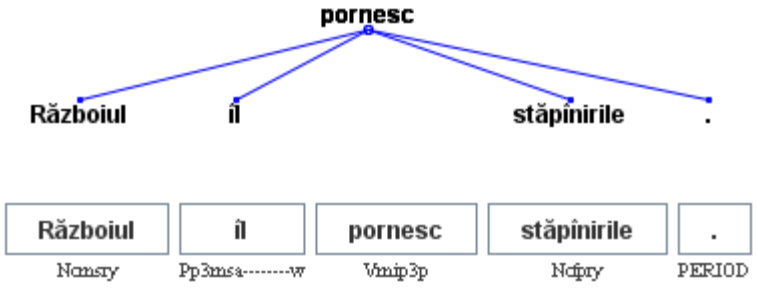

Figure 4: The position of the full stop at the end of the sentence
(*The sovereignties start the war*).

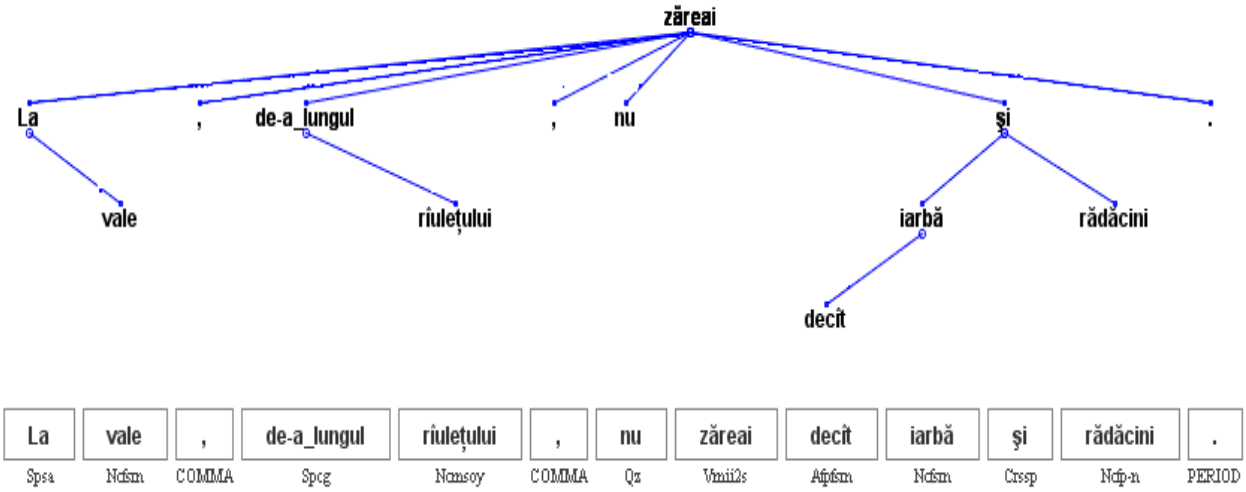

Figure 5: Commas used to separate a syntactic group from other parts of the sentence (*In the valley, along the brook, there could be seen only grass and roots.*)

According to the Grammar of the Academy[2], a comma graphically marks certain short pauses made during the rendering of a sentence or a phrase. These kinds of pauses are used on purpose in two situations:

a) to group certain words, which form meaning units, in one place (this way, separating them from the rest of the phrase or the sentence);

b) to focus attention on certain words by separating them from the rest of the phrase.

3.3 Prepositions

In the dependency tradition the prepositions are considered heads for the noun phrases following them in the surface string. This rule applies equally well in English as in Romanian. In Romanian, according to the Grammar of the Academy, the preposition, as a connector, is placed in a strict ternary structure, the presence of it being conditioned by the co-occurrence with two lexical autonomous terms, which exist in a dependency relation (*zi de iarnă – day of winter* [*winter day*], *fuge la mama – runs to mama*). Thus the preposition becomes head for the term on its right. This strict relation is conditioned also by the fix order of the two components: the preposition is always placed in front of the noun phrase (*tablă de şah – table for chess* [*chess table*], not *tablă şah de*). In the dependency trees, according to this rule, the preposition is placed in an intermediate position between the two nouns (the determinant and the determiner).

[2] Academia Română (2005). *Gramatica limbii române*, Ed. Academiei Române, Bucureşti

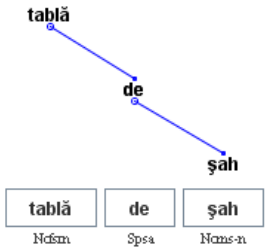

Figure 6: Subtree with a preposition linking two nouns
(*chessboard*)

Figure 6 shows how the preposition *de* connects the noun *tablă* to the noun *șah*, which is seen as an attribute.

3.4 Numeral

In most cases, the numeral is placed in the tree under a head which is the noun that it counts, like in Figure 7.

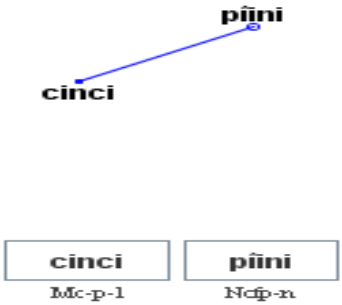

Figure 7: Example of a simple case with numeral
(*five loaves of breads*)

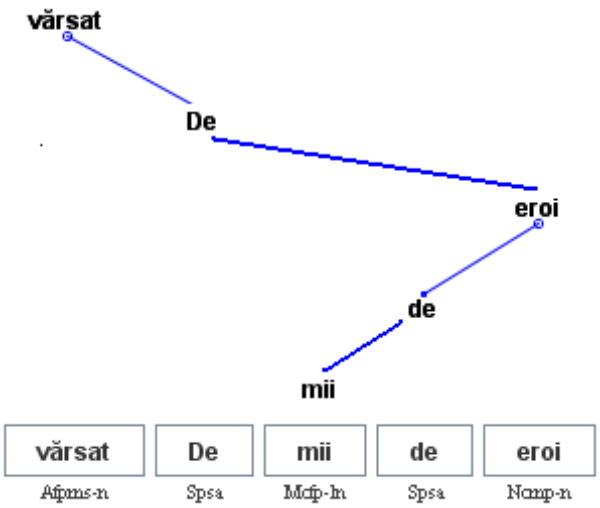

Figure 8: Numeral with preposition

In other cases, if between the numeral and the determined word appears a preposition, we have to take into consideration also the previous rule that apply to prepositions. In

the example *E sângele vărsat de mii de eroi…* (*It is the blood spilled by thousand of heroes*), the numeral *mii* (*thousand*) is a subordinate of the noun *eroi* (*heroes*), but between these two words the preposition *de* is present, which cannot be a head for *eroi* because the other preposition *de* is the head of this word. We keep the rule of positioning the preposition as head, but this time the subordinated word is the numeral (see Figure 8).

4. Establishing the dependency relations

In the process of annotation, the process of building the dependency structure is followed by the one of giving names to the dependency relations, on each arrow linking words.

To establish the types of dependencies between words, the grammatical functions of those words are thought, in relation with each others. Thus we followed a series of general rules noticed to apply in most of the cases:

- the syntactic function (given by the classic syntactic analysis) of the dependent word is taken into consideration;
- the morphological characteristics of the dependent word are considered (e.g. the *auxiliary relation*, when the dependent word is an auxiliary verb);
- some dependency relations are dependent of the head word (the *prepositional relation* when the head is a preposition, or a *coordinative relation* when the head is a coordinative element).

In the following we present some difficult cases of establishing the type of dependency.

4.1 Coordinative relation

Ellipses incur difficult decisions for establishing both dependencies and relations. Figure 9 shows a case of a conjunction combined with an ellipses. When the main verb of a clause is missing, the way we treat the implication results from the way we treat the simple coordination. In this example, the missing element is a verb (*mănâncă* - *eats*, in the second clause), and the coordinative conjunction *și* (*and*) will collect all its subjections. When a coordinator represents all the missing elements of a clause, it also inherits all the properties of the missing verbal elements.

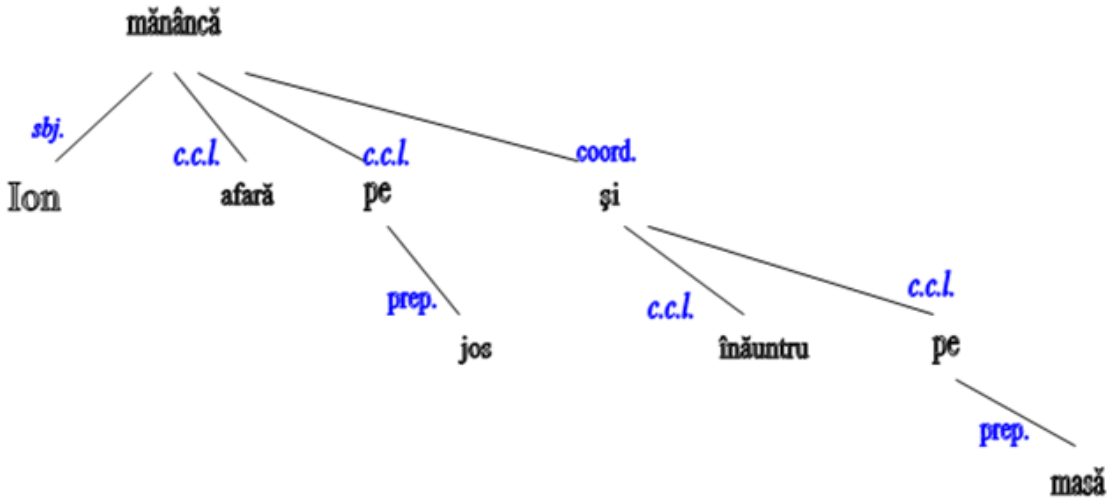

Figure 9: Sentence with elliptical subject and predicate
(*John eats outside on the ground and inside on the table.*)

This solution is also computationally efficient because there is no need to create special node for the missing words. From a descriptive point of view, it is no problem if a coordinative element takes over the syntactic properties from the elements it connects.

4.2 Comparative relation

In a comparative relation three elements are connected. The adjective becomes the head in this case and the comparative relation is attributed to those elements that help the formation of the comparative degrees or is attributed to the word that helps building the superlative-relative degree. Figure 10 shows an example:

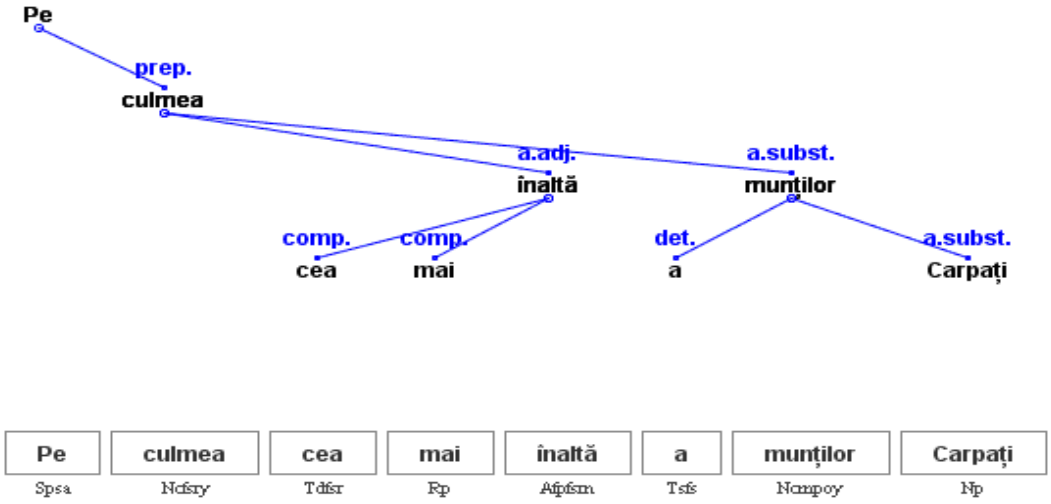

Figure 10: The comparative relation
(*On the highest peak of the Carpathian mountains.*)

4.3 Narrative relation

When a coordinative conjunction appears at the beginning of a sentence, it marks the connection to a previous statement. In this case, the conjunction, as well of other markings, all have a discursive role, signalling rhetorical relations. They announce a new topic of discussion, or an argumentation related to the previous one. This markings are treated in a special way in our structures. In case of conjunctions they don't have their usual coordinative meaning, but suggest the succession of an unbreakable chain of actions. Such occurrences are typical in the narrative style. Given their role of discourse markers, we decided to place them at the head of the whole construction following it. Consequently, the relation linking a discourse with the head of the clause names the rhetorical relation.

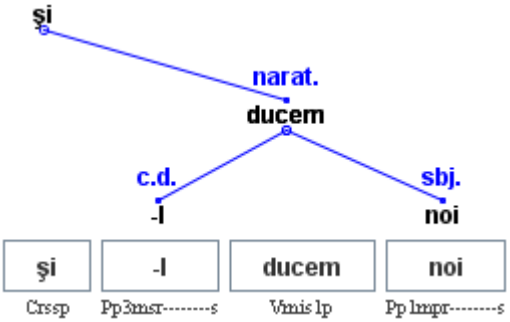

Figure 11: A narrative rhetorical relation
(*And we will bring it.*)

5. Conclusions

The rules established here had as starting point the norms of the Grammar of the Academy, but we had to impose our own conventions in order to accommodate the idiosyncrasies of the dependency grammar formalism.

The corpus includes at this moment about 4000 Romanian sentences manually annotated in the FDG formalism. The purpose of this corpus is to be placed at the basis of the elaboration of a syntactic parser for the Romanian language. We are aware that its size should be increased in order to describe the whole diversity of the Romanian syntax, and to assure the redundancy that will make accurate a learning process.

As the elaboration of a significant corpus is very costly and time consuming, any strategy able to boost this process should be put to work. We see at this moment, a number of possible ways to follow:

- try to merge all initiatives to build Romanian treebanks in a coherent unified corpus of annotated trees. But such a tentative will certainly be hindered by the diversity of conventions used by different authors and by differences in names of relations. As such, a hierarchy of relations should first be established and, where possible, merging strategies should be imagined;

- try to engage in the annotation process a large number of contributors. Certainly this is not easy, because annotation should follow strict rules in order to be accurate and this means the use of linguist experts. However, to accommodate the need for expert

knowledge with the need of large number of contributors, techniques of human computing could be imagined.

References

Academia Română (2005). *Gramatica limbii române,* Ed. Academiei Române, București.

Academia Română, Institutul de Lingvistică "Iorgu Iordan" (1996). *Îndreptar ortografic, ortoepic și de punctuație*, Ediția a V-a Univers Enciclopedic, București.

Călăcean, M., Nivre, J. (2008). *Data-driven Dependency Parsing for Romanian*, Uppsala University.

Hajič, J., Pajas, P., Hladká, B. V. (2001). The Prague Dependency Treebank: Annotation Structure and Support, in *Proceedings of the IRCS Workshop on Linguistic Databases*, University of Pennsylvania, Philadelphia, USA.

Hristea, F., Popescu, M., (2003). *Building Awareness in Language Technology*, București, Editura Universității din București.

Marcus, M., Santorini, B., Marcinkiewicz, M.A. (1993). Building a large annotated corpus of English: the Penn Treebank, in *Computational Linguistics*, 19.

Seretan, V., Wehrli, E., Nerima, L., Soare, G. (2010). FipsRomanian: Towards a Romanian Version of the Fips Syntactic Parser, in *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, European Language Resources Association (ELRA), May.

Tesniére, L. (1959). Elements of structural syntax, *Editions Klincksieck*.

LEXICAL DERIVATION APPROACHES FOR FUNCTIONAL EXTENTION OF COMPUTATIONAL LINGUISTIC RESOURCES

MIRCEA PETIC

Institute of Mathematics and Computer Science, Academy of Sciences of Moldova, Chişinău, Republic of Moldova

mirsha@math.md

Abstract

This article represents a description and a realization of a new methodology of studying the issues in computational derivational morphology, related to the algorithmization of certain linguistic mechanisms, e. g., affix substitution, derivatives projection, derivational constraints and formal derivational rules. The established mechanisms, which permitted the elaboration of algorithms and corresponding programs, led to generation of a significant number of derivatives with different affixes.

1. Introduction

The linguistic resources represent the fundamental support for automatic tools development in the processing of linguistic information. The need of the lexical resources enrichment is satisfied not only by borrowings of words from other languages, but also by the use of some exclusively internal processes.

The particularities of the derivational morphology mechanisms help in lexical resources extension without any semantic information. Moreover, there are processing mechanisms similar for different languages spoken in Europe, namely English, French, Spanish, Russian, Romanian. The approaches and mechanisms presented in the paper have been studied on the examples from Romanian language, but in majority of cases can be applied to different languages.

From the above we conclude that lexicons completion can be achieved by automatic means taking into account the productive properties of derivational processes. Thus the basis for generating new derivatives is an existing lexicon. The lexicon should contain not only graphical representation of the words, but also their parts of speech.

2. Romanian computational linguistic resources

Automatic derivation process requires preliminary experiments, which would allow the deduction of the mechanisms relating to the behavior of Romanian language affixes. In our case we will work with 3 Romanian computational resources, the most reliable to our scope: *DMLR* (Morphological Dictionary of the Romanian language in the electronic version), *RRTLN* (Reusable Resources of Natural Language Technology) and eDCD (Dictionary of derivative words in electronic version, adapted to the needs of studying mechanisms and elaboration of algorithms for automatic generation of derived words).

DMLR is a significant resource for Romanian language and represents a morphological dictionary (Lombard and Gâdei, 1981). This dictionary contains about 30. 000 words belonging to the various parts of speech (nouns, adjectives and verbs), divided into classes depending on the inflection of their training. An example of an entry in the DMLR is:

```
echilibra V201
```

where (a) `echilibra` is the word base, and V201 denotes inflection class, namely: the verb group 201 (Cojocaru, 1997).

RRTLN[1] - contains a database of linguistic information at the level of words and a set of programs to manage (Boian et al., 2005). Thus, the thesaurus contains not just parts of the speech, but also information about the categories and the possible morphological analyses of syntactic functions. RRTLN has about 100.000 word lemmas and about 1.000.000 flexes. It should be mentioned that a word can have several entries for different parts of speech, so having a different semantics, e. g., the adjective bun (eng. good), bun (eng. approving) as an adverb and bun (eng. property) as a noun [11].

eDCD - contains only the list of derivatives and constituent morphemes without having information about the part of speech of the derivatives and their morphemes, although the vast majority is nouns, verbs and adjectives. eDCD was obtained after the paper version was scanned, OCR-ized and corrected using the original entries. eDCD allows detection of derivatives morphemes with the appropriate type (prefix, root and suffix) (Petic, 2009). For easier processing of the lexicon entries, a regular expression was developed, which represents the following derivative structure:

```
derivat = (+morpheme)*.morpheme(-morpheme)*
```

where `+morpheme` represents a prefix, `.morpheme` is a stem, and `-morpheme` is a suffix. An example of an entry in the lexicon is:

```
antistatal=+anti.stat-al
reprogramabil=+re.programa-bil
```

In the brief description above we see that the information in each of these three computational linguistic resources is different. Therefore, this article will present several studies that will use several resources simultaneously.

3. Collection of Romanian affixes

Any morpheme that is outside the root of the word is called affix. In the name of global affixes we include prefixes and suffixes. After the position it occupies to the root the affixes are divided into two categories, namely:

- placed before the root (*prefixes*);
- attached at the end of the root (*suffixes*).

[1] Lexicon can be found on the site http://imi201.math.md/elrr/

The word that is formed by adding a prefix or suffix is called derivative (Carstairs-McCarthy, 2010).

3.1. Collection of prefixes

From the point of view of the origin of the prefix in (Stoichițoi, 1994) were established the following classes:

- Latin – 12 prefixes (e. g. *des-*, *în-*, *stră-*, etc.);
- Slavic – 13 prefixes (e. g. *ne-*, *răs-*, etc.);
- Greek – 18 prefixes (e. g. *anti-*, *arhi-*, *hiper-*, *hipo-*, etc.);
- Multiple origins – 29 prefixes (e. g. *ante-*, *circum-*, *co(n)-*, *contra-*, *ex-*, *extra-*, *non-*, *post-*, *re-*, *ultra-*, etc.).

From the point of view of semantic information, it contains, we can highlight the following categories of prefixes:

- negative meaning (*a-*, *in-*, *non-*, *ne-*, *i-*);
- that indicates a repetition or an inversion (*re-* in *re*citi, eng. *re*read, *de-* in *de*colonizare, eng. *de*colonization);
- that indicates the time, space, relation level (*inter-* in *inter*planetar eng. *inter*planetary, *hiper-* in *hiper*tensiune eng. hypertension, *ex-* in ex-student, *supra-* in *supra*răcire eng. *super*cooling etc.).

The most numerous derivatives of the following prefixes (in descending order of frequency of occurrence) are: *ne-*, *re-*, *în-*, *des-*, *pre-*, *anti-*, *auto-*, *sub-*, *dez-*, *supra-*, *de-* and *îm-*. These 12 prefixes of 42 form 88.2% of all derivatives with prefixes, recorded in eDCD (Petic, 2010).

3.2. Collection of suffixes

Most often, new words created by suffixation give a certain amount of semantic and morphological value, which allows to perform the classification of derivatives in several major categories, as follows:

- agent name – e. g., munci*tor* (Eng. work*er*);
- instrument name – e. g., ascuți*toare* (Eng. sharpen*er*);
- derivatives with collective meaning – e. g., țărănime (Eng. peasantry);
- abstract derivatives – e. g., răutate (eng. badness);
- derivatives that indicate the origin – e. g., românesc (Eng. Romanian);
- augmentative derivatives – e. g., băiețoi (Eng. big boy);
- diminutive derivative – e. g., căluț (Eng. small horse).

Morphological classes or parts of speech to which they belong, derivatives formed by suffixes can be classified into the following categories (Boian et al., 2005):

- noun: *-tor* (e. g., citi*tor*, eng. reader), *-an* (e. g., americ*an*), etc.;
- adjective: *-ic* (e. g., acrobatic), *-al* (e. g., doctor*al*), *-esc* (e. g., moldoven*esc*, eng. moldovan), etc.;
- verb: *-iza* (e. g., mineraliza, eng. mineralize), etc.;
- numeral: *-ime* (e. g., optime, eng. eighth).

The most numerous derivatives of the following suffixes (in descending order of frequency of occurrence) are: *-re, -tor, -toare, -eală, -ie, -ătoare, -iza, -oasă, -ar, -ător, -ească, -os, -aș, -esc, -tură, -iță, -ist, -uță, -el, -i, -ui, -ătură, -ește, -ism, -a, -ărie, -ică, -ime, -itate, -ioară, -işor, -işoară, -ic, -uleț, -că, -ean, -iș, -easă, -bil, -uț, -at, -oaică, -ușor, -an, -oi, -uliț, -iu, -enie, -istă, -al,* and *-ea*. 51 from 433 suffixes recorded in eDCD form 87.7% of all derivatives with suffixes. Other suffixes have an insignificant number of derivatives (Petic, 2010).

4. Procedural completion of the lexicon with derivatives

This study aims to exploit existing resources in such a way that it is possible to generate lexical derivational families of Romanian language. Comparing the intentions of this work to the Italian model (Carota, 2006) and its derivational morphology reversal of priorities is observed, as in the case of the thesaurus is organized so that it is possible to draw derivative families present in the resource.

The subject of research is the procedural method, for which it is necessary to establish rules so that the derivatives can be obtained in an algorithmic way from root/themes (Boian et al., 1994).

Taking into account the productive process properties of derivation, the lexicon completion can be performed using automated means (Boian et al., 2011). Schematically this process represents a cycle (Figure 1). This cycle can be applied several times (Cojocaru et al., 2009). To the end, after a finite number of cycles it is possible that the cycles can no longer produce new words, finally obtaining a completely "saturated" lexicon in terms of derivation.

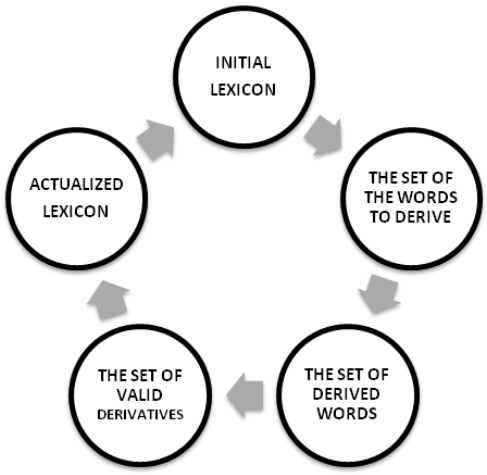

Figure 1: Schema of derivation cycle

5. Multilingual approaches of derivatives generation

5.1. Establishing candidate words for derivation

To develop algorithms for automatic generation of derivatives it is necessary to determine whether a word is a candidate to be derived. At this stage we verify whether a sequence of characters represents a correct word in Romanian language and if from this word we could generate other derivatives. A common feature of systems built for different languages is the use of computational linguistic resources, from which it is started the process of automatic generation of words (Carota, 2006). However, in the case of automatic derivation algorithm, computational linguistic resources function is not used in derived words extraction, but families likely to generate derivatives. Resources also contribute to the process of validating the derived words generated automatically. In this way the initial sequence of characters can be verified initially in RRTLN. If the sequence of characters is not found in the mentioned resource, it will be verified using Internet resources (Petic et al., 2011). The diagram in Figure 2 illustrates the procedure applied.

Figure 2: Establishing candidate word for derivation.

After the set was fixed for derivation, the application of models of derivation follows. A distinction of the presented approaches to those of other languages is the lack of semantic information in computational sources, with whom it operated. The most important patterns of derivation that does not involve the use of semantic information are the following: affix substitution, derivation projection, formal models of derivatives derivation, derivational constraints.

5.2. Generated derivatives validation

Automatic derivation represents an over generating mechanism. That is why validation of generated words is needed. One of the methods of new word validation consists in manual verification of every new generated derivative as to correspond to semantic and morphologic rules. In the case of the proceeding is performed by a specialist in domain, the specific disadvantages of a manual work appear: considerable resources of time and the possibility to make mistakes. So, this method of validation becomes inefficient (Cojocaru et al., 2009).

Another method of validation consists of the verification of the derivatives in the existent electronic documents. There are different types of electronic documents.

The first idea that appears - to validate words using existent corpora that represent verified documents - seems to be the best solution. The condition for being the panacea in the new word validation is a representative corpus, with a big number of words from different domains.

On the other hand there are documents on Internet, that are not verified, that are why they are not credible. In order to make it more precise, the searching on the Internet, using *Google.com* search engine, should be made for the documents typed only in a specified language. Besides this, it is necessary that the following be assured: the possibility to exclude word segmentation; the part of speech of the derivatives.

This validation tool divides the generated derivatives in three categories. The first one contains words that are not found by Google.com searching engine. The second consists of the derivatives that appear less than a frequency limit of *n*, in our case $n = 1000$. Derivatives that are more frequent that limit *n*, are registered in the third group. This classification pretends that the words, that are listed more than frequency limit of *n*, are surely valid. Those, which are from the second group, can be valid but should be verified by specialists in linguistics. The derivatives, that are not present, could not be valid (Petic et al., 2011). The idea of classification pretends to be a mixed method of validation, because needs only the manual verification for the words from the second category (Figure 3).

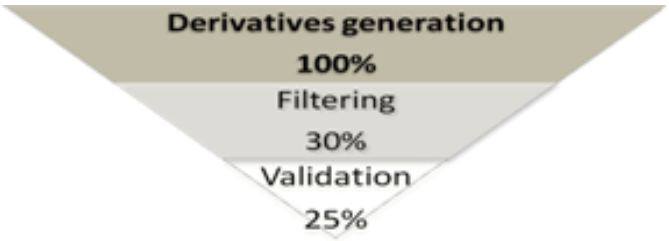

Figure 3: Process of derivatives validation.

5.3. Affixes substitution

The idea is inspired from Serbian derivational morphology (Duško and Krstev, 2005), where the generated derivatives have predictable meanings, namely the gender modification in the case of suffix substitution, e. g., munci*tor* ↔ munci*toare* (eng. worker), and in the case of prefix substitution there is meaning change, e. g., *ante*belic ↔ *post*belic (eng. pre-war – after-war).

Affixes substitution is not specific only for Romanian and Serbian derivational morphology, but also for other European languages, e. g., Spanish (e. g., amortizar-amortizable, eng. to amortize-redeemable), French (e. g., revoir-prevoir, eng. revise-foresee), Russian (e. g., прочитать-дочитать, eng. read – read till the end) etc.

In general case for suffix substitution, let be x_1 a word of the form $x_1=\omega\alpha_1$ with the suffix α_1. After the substitution $\alpha_1 \rightarrow \alpha_2$ we obtain the word $x_2=\omega\alpha_2$, e. g., corig*ență*-corig*ent*. In the case of prefix substitution, let be x_1 a word of the form $x_1=\alpha_1\omega$, where

α_1 is a prefix. After the substitution $\alpha_1 \rightarrow \alpha_2$ we obtain the word $x_2=\alpha_2\omega$, where x_2 is the obtained derivative, e. g., *în*chide-*des*chide (Petic, 2011).

From the information above a new and original algorithm was developed which consists in examining the words in the lexicon and substitution of the affixes in those cases that correspond to the categories established by the above-mentioned rules.

5.4. Formal models

Formal models of derivation rules, represent the basis of which it can generate derivative words with a high degree of accuracy. A similar approach in derivational morphology is met in French language (Fiammetta and Dal, 2000). But when French system works with only 3 suffixes (-able,-ite,-is (er)) for which rules have been found, in the case of Romanian derivational morphology this study consist of 3 prefixes (ne-, re-, in-/im-) and 2 suffixes (-re,-iza).

- Rules for prefixes:
  - ✓ re- $[\omega]_{inf} \rightarrow [\text{re}\ [\omega]_{inf}]_{inf}$
  - ✓ ne- $[\omega'\beta]_{adj} \rightarrow [\text{ne}\ [\omega'\beta]_{adj}]_{adj}$

    $\beta \in$ {-tor, -bil, -os, -at, -it, -ut,-ind, -înd }
  - ✓ in-/im-=γ $[\omega'\beta]_{adj} \rightarrow [\gamma\ [\omega'\beta]_{adj}]_{adj}$
  - ✓ $\in$ {-bil, -ent, -ant}
- Rules for suffixes:
  - ✓ -re $[\omega]_{inf} \rightarrow [[\omega]_{inf}\ \text{re}]_{subst}$
  - ✓ -iza $[\omega'\beta\alpha]_{adj} \rightarrow [[\omega'\beta]_{adj}\ \text{iza}]_{inf}$

5.5. Derivatives projection

The projection of derivatives represents a method of word formation of the prefixed words from the suffixed words of the same root. According to Spanish researchers, the Spanish verb *amortizar* can be derived with the prefix *des*- obtaining *desamortizar*. Also, *amortizar* can be derived with suffixes *–cion* and *–able*. So, the derivative with prefix *des*- can derive with the suffixes *–cion* and *–able*. The hypothesis is that derivatives can inherit/project the derivatives with suffixes of the stem whose the prefixation was realized (Santana, 2004). This method is not exclusively Spanish, but it can be applied to other languages; e. g., in English from the root *read* one can form derivatives readable and unread, therefore, it is possible to form the derivative unreadable.

Generalizing the above noted, we conclude that it is possible to present in a formal way the mechanism for Romanian derivational morphology. Let us consider a Romanian word ω, α - its prefix and β - its suffix. Then, the following relation is valuable (Petic, 2011):

$$(\omega \rightarrow \alpha\omega) \wedge (\omega \rightarrow \omega\beta) \Rightarrow (\omega \rightarrow \alpha\omega\beta),$$

for example, (a lucra → a *pre*lucra) ∧ (a lucra → lucr(a)*ător*) ⇒ (a lucra → *pre*lucr(a)*ător*);

$$(\omega \rightarrow \alpha\omega) \wedge (\omega \rightarrow \alpha\omega\beta) \Rightarrow (\omega \rightarrow \omega\beta),$$

for example, (a capitula→ recapitula) ∧ (a capitula →recapitulație) ⇒ (a capitula→capitulație)

$$(\omega \rightarrow \alpha\omega\beta) \wedge (\omega \rightarrow \omega\beta) \Rightarrow (\omega \rightarrow \alpha\omega),$$

for example, (a centraliza → *des*centraliz*ator*) ∧ (a centraliza → centraliz*ator*) ⇒ (a centraliza → *des*centraliza);

Examining the words in the lexicon and verifying them in correspondence with relations above, a new and original algorithm has been developed that generates derivatives by affixes projection.

5.6. Derivational constraints

Where there is no clear model, according to which it would be possible to generate derivatives, some preconditions will appear, called derivational constraints. The most common derivational constraints: parts of speech, inflection classes, affixes, changes that take place in the case of derivation, the letters preceding/succeeding prefixes/suffixes. So, derivational constraints represent some schemes with several parameters that reduce the class roots and affixes in order to form derivatives. E. g. functions of the form:

```
f: {wrd, pos, mod, sla, fgw, mvca} → derivative
```

where *wrd* is a word to derivate, *pos* - part of speech of *wrd*, *mod* - model of derivation, *sla* - the set of letters to which the affix is attached, *fgw* - flection group of *wrd*, *mvca* - modifications and vocalic or consonant alternations (Petic, 2011).

Examining the words in the lexicon and verifying them in correspondence with relations above, has been developed a new and original algorithm that generates derivatives by derivatives constraints.

As examples of generating derivatives by the derivation constraints can serve as automatic derivation of words with the prefix *des*- and suffixes *-bil* and *-ime*.

```
f: {a spinteca, verb, des<verb>, ...s..., V14, evitarea dublării
                    consoanei} → de(s)spinteca.

  f: {a programa, verb, <verb>bil-itate, ...a..., V201, ...} →
                          programabilitate

     f: {crud, adjectiv, des<adjectiv>, ..., A3, alternanța
                 consonantică d - z} → cru(d)zime.
```

Therefore, derivational constraints necessary for the automatic generation process, do not depend on just the affix type, but also the value of the prefix or suffix, moreover, each language has its own peculiarities in the derivation of words.

6. Conclusions

Studies on derivation process allow us to conclude that we cannot propose an effective algorithm for automatic derivation in general, but we can highlight some models of derivation, for which construction of such algorithms is possible.

The new derivatives validation is one of the steps in automatic derivation that raises many questions. In the case it is difficult to set up the criterion for words validation by means of Internet, it is important to use the digital variant of the derivatives dictionary, which will permit the establishing of the morphemes of the derivatives with its type (prefix, root and suffix).

Acknowledgements. This article is carried out as part of the project ref. nr. 12.819.18.09A supported by Supreme Council for Science and Technological Development from Republic of Moldova.

References

Boian, E., Ciubotaru, C., Cojocaru, S., Colesnicov, A., Demidova, V., Malahova, L. (2005). Technologization of Romanian: linguistic resources, applications, tools. In: *Proceedings of the 4th International Conference on Microelectronics and Computer Science*. II, 519-522.

Boian, E., Ciubotaru, C., Cojocaru, S., Colesnicov, A., Malahov, L., Petic, M. (2011). Creation and Development of the Romanian Lexical Resources. In: G. Anghelova, et.al (eds), International Conference Recent Advances in Natural Language Processing Proceedings. Hissar, Bulgaria, 12-14 September, 678-685.

Boian, E., Danilchenco, A., Topal, L. (1994). Automation of word forming process in the Romanian language. In: *Studies in Informatics and Control*, March Bucharest, Romania, 3:1, 43-52.

Carota, F. (2006). Derivational Morphology of Italian: Principles of Formalization. In: *Literary and Linguistic Computing*, 21: Suppl. Issue, 41-53.

Carstairs-McCarthy, An. (2010). The Evolution of Morphology. Oxford University Press, 254 p.

Cojocaru, S. (1997). Romanian Lexicon: Tools, Implementation, Usage. In: *Dan Tufis, Poul Andersen (eds.). Recent Advances in Romanian Language Technology*. București: Editura Academiei, I, 107-114.

Cojocaru, S., Boian, E., Petic, M. (2009). Derivational morphology mechanisms in automatic lexical information acquisition. In: *Proceedings of the International Conference on Knowledge Engineering, Principles and Techniques*. Cluj-Napoca: Presa Universitară, 49-52.

Duško, V., Krstev, C. (2005). Derivational Morphology in a E-Dictionary of Serbian. In: *Zygmunt Vetulani (ed.), Proceedings of the 2nd Language & Technology Conference*. Poznan, Poland, 139-143.

Fiammetta, N., Dal, G. (2000). GéDériF: Automatic generation and analysis of morphologically constructed lexical resources. *Second International Conference on Language Resources and Evaluation (LREC)*. Athens, Greece, May 31 – June 2, 1447–1454.

Lombard, C., Gâdei, A. (1981), Dictionnaire morphologique de la langue roumaine. București: Editura Academiei, 232 p.

Petic, M. (2009). Automatic derivation as a model of lexical acquisition. In: *Conference Mathematics & Information technologies: research and education (MITRE – 2009)*. Chișinău, Moldova, October 8-9, 2008, Abstracts, 67-68.

Petic, M. (2010). Mecanismele generative ale morfologiei derivaționale. În: *Lucrările Conferinței "Resurse Lingvistice și Instrumente pentru prelucrarea limbii române"*. București: Editura Universității "Alexandru Ioan Cuza", 195-202.

Petic, M. (2011). Generative models in automatic derivational morphology. In: *Conference Mathematics & Information technologies: research and education (MITRE – 2011)*. Chișinău, Moldova, 21 - 23 august, Abstracts, 146-147.

Petic, M., Gîsca, V., Palade, O. (2011). Multilingual mechanisms in computational derivational morphology. *Proceedings of Workshop on "Language Resources and Tools with Industrial Applications"* Cluj-Napoca, Editura UAIC Iasi, 29-39.

Santana, O., Perez, J., Carreras, F., Rodriges, G. (2004). Suffixal and Prefixal Morpholexical Relationships of Spanish, *Lecture Notes in Artificial Intelligence*, Ed. Springer-Verlag, 407-418.

Stoichițoi, A. I. (1994). Prefixarea în româna actuală. În: *Limba și Literatura română*, 2, 6-8.

LAYING THE FOUNDATION FOR THE REPRESENTATIVE CORPUS OF CONTEMPORARY ROMANIAN

VERGINICA BARBU MITITELU, TIBERIU BOROȘ, CORINA FORĂSCU, RADU ION, ELENA IRIMIA, DAN TUFIȘ

Romanian Academy Research Institute for Artificial Intelligence

{vergi, tibi}@racai.ro, corinfor@info.uaic.ro, {radu, elena, tufis}@racai.ro

Abstract

This article provides an overview of one of the projects conducted at the Romanian Academy Research Institute for Artificial Intelligence. The project is in line with the highest priorities of the Romanian Academy, namely the study and preservation of the Romanian language. We briefly describe the core of a representative corpus of contemporary Romanian, its annotation layers and future prospects.

1. Introduction

In 1866, when *Societatea Literară Română* (that later became the Romanian Academy) was founded, the Romanian society was getting through a process of modernization. The intellectual elite felt responsible for establishing the orthographical norms, creating and publishing a dictionary and establishing the grammatical norms. Such desiderata were in line with those in western European academies, thus revealing the modernity in thinking and in the attitude towards society.

All these aims have been achieved so far. The last orthographical norm dates from 2005 and it is reflected in *Dicționarul ortografic, ortoepic și morfologic al limbii române* (known as DOOM2) realized at the Institute of Linguistics "Iorgu Iordan - Al. Rosetti", under Ioana Vintilă Radulescu's supervision.

The Grammar of Romanian, due to Timotei Cipariu, was published in 1968 and awarded by the Romanian Academy. Since then the researchers at the Institute of Linguistics "Iorgu Iordan - Al. Rosetti" have elaborated updated Grammars of Romanian, reflecting language evolution and adopting newer linguistic theories for facts presentation and analysis. The last one was published in 2005 and was edited by Valeria Guțu Romalo.

The elaboration of the dictionary was a century hard work. In 2010 the last of the 37 volumes was published. It was the work of more than 200 lexicographers from the three institutes of linguistics of the Romanian Academy: the Institute of Linguistics "Iorgu Iordan - Al. Rosetti" from Bucharest, the Institute of Linguistics and Literary History "Sextil Pușcariu" from Cluj-Napoca and the Institute of Romanian Philology "Al. Philippide" from Iași.

Lately, many researchers have formulated the need for a corpus of big dimensions on which further research to be based. The trend in linguistics stresses the need for constructing the theory starting from the evidence in language, not from examples fabricated by the mind of theoreticians. Furthermore, the recurrence of certain patterns

and their frequency are more important than a single instantiation of a structure. These can be studied only when a large collection of texts is available, alongside with meta-information for them.

In an era in which the visibility and even the survival of a language are helped by its existence in electronic form it is vital for the most important cultural and scientific forum of a nation to adopt as one of its priorities the development of a representative corpus for its language, to ease data collection, to ensure the infrastructure for texts annotation, to make the results accessible for those interested.

The Romanian Academy Research Institute for Artificial Intelligence has acquired extensive experience in working with corpora, in preprocessing and processing them and also has the necessary infrastructure for storing large quantities of data. This year the institute proposed a strategic research program, led by Acad. Dan Tufiș, for building a large reference corpus for contemporary Romanian language. The program has been approved by the Section for Science and Technology of Information and the General Assembly of the Romanian Academy. We have taken up the responsibilities of developing a prescriptive methodological framework for the computational study of Romanian, according to the international practice and recommendations and of developing core applications of Romanian language processing. Starting from purely engineering aspects such as characters encoding, morpho-lexical and syntactic descriptions and ending with modeling linguistic competence and performance, this project focused on Romanian will develop research and implementation methodologies for various linguistic levels (lexicon, syntax, semantics, pragmatics), with an eye towards multilingual contexts. The language resources envisaged are: corpora, lexical indices of frequency, morpho-lexical dictionaries (based on the occurrence frequency in corpora). The research program includes the creation of a unified management system for maintaining and exploiting the linguistic data.

2. Why a representative corpus?

A computational corpus is an electronic collection of textual or multimedia representations of some fragments considered illustrative for the real use of a language. There are several motivations justifying the interest for such linguistic resources: due to the naturalness of the contained texts, a corpus may or should be used as an indispensable working base for a linguist aiming to describe various aspects of the language; parallel or comparable corpora for more (related or not) languages offer material for a comparative study of those languages; for lexicographers the corpus offers valuable material to work on when editing dictionaries (for general or special use); for language engineers corpora offer the training, learning and testing material for the tasks they implement. For the process of language learning, a corpus provides specific examples of possible contexts for words, of relations they establish with other words, etc. Even for the Romanian classes in school, a corpus can be a useful means for teaching and evaluation of students.

At an international level, there are an increasing number of corpora available, of great dimensions, for more and more languages: English, Russian, Bulgarian, German, Croatian, Polish, Spanish, and others.

Developing a representative corpus presupposes: defining its structure, its linguistic coverage, collecting texts according to the established structure, solving problems of copyright, processing text with linguistic technologies (segmentation, lemmatization, tagging, etc.), text indexing according to various criteria useful in exploitation, extracting statistical data, developing an exploitation platform, as friendly and flexible as possible, establishing secured access methods in order to prevent vandalism or misuse. In the context of public access, the hardware architecture has to be adequate to the simultaneous access of more users.

A representative corpus of a language reflects its structure and functions. Thus, it has to display several characteristics:

- large dimensions;
- proportional representation of registers and styles;
- pre-processing, for lexical units identification (i.e. language structure);
- annotation, which distinguishes a corpus from a collection of texts and highlights the way a language functions;
- utility in language study.

3. Representative corpora in the world

English (www.natcorp.ox.ac.uk), Czech (ucnk.ff.cuni.cz), Russian (ruscorpora.ru), Bulgarian (http://ibl.bas.bg/en/BGNC_classific_en.htm), Arabic (www.bibalex.org/unl/Frontend/Project.aspx?id=9), Croatian (www.hnk.ffzg.hr/default_en.htm) are languages for which representative corpora have been created. They are either the work result of an institution or of a consortium. Their size is up to hundreds of millions of tokens. The oral and the written styles are represented in the corpora, the former in a much smaller percent than the latter. Various domains tend to be covered, so that as many words and word meanings as possible should be encountered in the corpora. Annotation is made at a morphological level, syntactic, even pragmatic and semantic. Usually, only a sample of these corpora is available for free, although online searching is possible through the entire corpus.

4. The foundation

The Romanian Academy Research Institute for Artificial Intelligence already has a corpus of 34,000,000, tokens which was called ROMBAC, short for ROManian Balanced Annotated Corpus (Ion et al., 2012). It displays five genres: journalistic (news and editorials), pharmaceutical and medical short texts, legalese, biographies of the major Romanian writers and critical reviews of their works, and fiction (both original and translated novels and poetry). The texts are tokenized, morpho-syntactically tagged, lemmatized, shallowly parsed (chunked) and XCES-compliant encoded.

The journalistic sub-corpus of ROMBAC consists of the issues of the ***Agenda*** newspaper[1] published daily between 2003 and 2006. The **Agenda** sub-corpus is a middle-sized journalistic corpus, having 7 millions tokens. It evolved from a very large collection of journalistic articles, initially available in various formats (doc, rtf and pdf). They were converted into ASCII format, with diacritical characters encoded initially as SGML entities and recently in UTF8.

The second sub-corpus of ROMBAC has been extracted from the **EMEA** corpus. EMEA is a parallel corpus made out of PDF documents from the European Medicines Agency, compiled by Jörg Tiedemann. All files are automatically converted from PDF to plain text. For more details about the corpus and the conversion strategy, see (Tiedemann, 2009). The Romanian-English part of the corpus was downloaded from http://opus.lingfil.uu.se/EMEA.php. From the Romanian part, a number of 800 documents (most of the texts are drug leaflets) containing around 7,000,000 words were randomly selected to be part of the Romanian Balanced Corpus.

The juridical sub-corpus has been extracted from the **JRC-Acquis corpus,** a collection of legislative texts representing the total body of European Union (EU) law applicable to the EU Member States. It is a parallel corpus available in 22 languages: all the official languages in European Union minus Irish, for which translations are not currently available (Steinberger et al., 2006). This is a big collection of documents, containing laws published from 1958 until 2006. The Romanian files available in the corpus were initially in Microsoft Word format and they had to be converted in the text format. The conversion requested some intermediary processing steps for removing the translators' comments, deleting the footnotes and headers, normalizing the diacritics usage (each of the characters "ş" and "ţ" were represented by two different codes). For our purposes, we retained only the documents published between 2003 and 2006, summing around 7,000,000 words.

The fourth sub-corpus of ROMBAC is based on the content of the **Romanian Literature General Dictionary** (DGLR, 2009), a 7 volumes critical anthology which contains biographies of Romanian writers, poets, essayists as well as commentaries about their work, information about publications, literary concepts, literary trends, anonymous writings, literary institutions, translations from/into Romanian, etc. This impressive dictionary, created by the Institute for Literary History and Theory "George Călinescu" (http://www.institutulcalinescu.ro/) of the Romanian Academy, has been provided in UTF8 text format by the authors, as part of their commitments to the METANET4U project. The text contains 5,189,909 words.

The fifth part of the ROMBAC corpus is a collection of novels and poems authored by 28 classical Romanian writers from the end of the 19th and beginning of the 20th centuries. This corpus was in part written with the old Romanian orthography. The orthography was updated to the current norms and the codes for the diacritical characters were unified.

[1] http://www.agenda.ro/. We acknowledge here the permission to use this data and the openness of the Chief Editor towards supporting corpus-based research.

There is also the Romanian version of the TimeBank corpus that was translated based on a minimal set of translation recommendations. The sentence alignment of the corpus was obtained as a direct output of the translation. In the 4,715 sentences of the current version of the Romanian corpus there are 65,375 lexical tokens, including punctuation marks, representing 12,640 lexical types.

5. Annotation of currently available corpora

The texts in the corpora were normalized at the orthographic level, cleaned of footnotes, headers and page numbers and the punctuation was separated from the words. After this preliminary phase, the corpora were subject to an annotation process using the TTL text processing platform developed at RACAI (Ion, 2007, Tufiș et al., 2008). TTL is entirely written in Perl and performs named entity recognition, sentence splitting, tokenization, POS tagging and chunking. We have exposed it as a SOAP compliant web service with the WSDL file available at http://ws.racai.ro/ttlws.wsdl and also as a REST web-service for the WebLicht platform (Henrich et al., 2010).

The TTL tokenizer is language aware and recognizes Romanian multiword functional expressions, clitics and contractions. Then, the tokens were annotated at the morpho-lexical level (MSD annotation), using TTL's HMM tiered tagger. The tagset used in the ROMBAC is a large one: 614 MSD tags fully compatible with the MULTEXT-East morpho-lexical specifications (http://nl.ijs.si/ME/V3/msd/html/msd.html) plus 20 named entity tags (Tufiș & Ion, 2007). The reduced (hidden) tagset used for tiered tagging (Tufiș, 1999; Tufiș & Dragomirescu, 2004) contains 93 tags for words and 10 tags for punctuation.

The corpora were further lemmatized through a look-up procedure in a large word-form lexicon whose entries have the form: <word-form> <lemma> <tag>. In Romanian, as in many other languages, most of the time a word-form and its tag uniquely identify the lemma. When this is not the case, the lemmatizer selects the most frequent lemma out of the competing ones. For the tokens not in the word-form lexicon (and which are not tagged as proper names), the lemma is provided by a HMM-based guesser, trained on the word-form lexicon. It scans the ending of the unknown word, right to left, detects all the known endings and selects the most probable one. The selected ending is stripped off and the lemma is generated according to the morpho-lexical properties encoded into the attached tag (more often than not, the stripped off word-form is the lemma itself). The next processing step is the text chunking. It is guided by a set of regular expression rules, defined over the MSDs and it deals with recognizing adjectival, adverbial, nominal, verbal and prepositional phrases. With respect to the verbal phrases, the chunker recognizes only the analytical forms of the verbs (compound tenses and passive constructions).

The output of TTL is an XML file encoding sentences (with paragraph information codified in the attribute 'id' of the sentence <s> element) and tokens, each token being classified either as a word (marked with the <w> element) or as a punctuation (marked with the <c> element). Each word has several attributes that will specify its lemma, its POS label (the 'ana' attribute), its membership to a chunk and its orthographic form given as the content of the <w> element.

XML format is useful for a large number of NLP applications since it conveniently delimits the units of text along with their annotations but, when clarity and standards compliance are in question, a better, more explicit and metadata aware representation is expected. Since the Romanian Balanced Corpus is going to be released as a METANET deliverable (http://www.meta-net.eu/), we chose to automatically convert our XML notation to the standard XCES Schema notation, revision 0.4 which is available for parsing and download from http://www.xces.org/schema/2003/.

The XCES Schema has support for a wide range of annotations (including different types of alignments and the possibility to reference annotations from external files) and also for inclusion of metadata in the header of each document. This schema supports annotations on multiple layers in different files but, for our purposes we will use the types defined in the 'xcesDoc.xsd' schema.

Using the TTL module, the texts in RoTimeBank were tokenized, POS-tagged, lemmatized, and chunked.

Following the TimeML development, the Romanian corpus annotation was adapted to the ISO version of the standard and, meanwhile, we proceeded with the improvements (Forăscu, 2009, 2011) needed for the portability to Romanian of the ISO-Time standard (2009). We ground the Romanian specific rules and/or adaptations on the Romanian Academy grammar (GA, 2006). We also took into account the rules applied to other Romance languages: Italian (Caselli, 2010), French (Bittar et al., 2011). For all the tags in ISO-TimeML, we can apply almost the same rules from English. The main improvements concern the EVENT tag.

In order to reflect the Romanian tense system, with four tenses denoting the past, we propose to use two more values for the "tense" attribute of the EVENT tag, SIM_PAST for the "simple perfect" of the indicative (*perfect simplu* in Romanian) and PLUS_PAST for the "more than perfect" tense of the indicative (*mai mult ca perfect* in Romanian). For the "imperfect" tense (*imperfect* in Romanian), as well as for the "composed past" (*perfect compus* in Romanian) we use the value PAST; the distinction between these two tenses is realized through the value of the "aspect" attribute.

For the category of aspect, we stick to the Romanian grammar and we include in the Romanian TimeML guidelines only the distinction between PERFECTIVE and IMPERFECTIVE verbs, manifested on the "imperfect" and "simple future" Romanian tenses on one side, and all the other tenses of the indicative mood, on the other side.

Trying to keep compatibility between the ISO-Time standard, the Romanian grammar, as well as the other Romance ISO-TimeML standards, we include for the "mood" attribute of the EVENT tag: CONDITIONAL/ IMPERATIVE/ SUBJUNCTIVE respectively for the conditional/ imperative/ subjunctive mood of the Romanian verbs. By default, the verbs in the indicative mood will have the NONE for the "mood" attribute.

The "vform" attribute has four values in Romanian, corresponding to the non-personal moods, namely verbs in the INFINITIVE, GERUND, PARTICIPLE.

Based on these considerations and on the ISO-Time standard, in a final processing step, we corrected the annotations in the Ro-TimeBank in order to have the annotations compliant with the ISO version of the standard.

6. Prospective work and conclusions

Starting from the material that we have, our aim is to cover important domains in our corpus, by collecting corpus of appropriate dimensions for each of them. Several other types of texts and registries are to be covered (poetry, technical writing, transcribed speech etc.). The representative corpus of contemporary Romanian will be equipped with a management corpora system, allowing a user to multicriterially search for linguistic information. For increasing the number of annotation levels, we envisage the development of a syntactic parser for Romanian. A public web-interface to the corpus will ensure the free access to a wealth of linguistic tools and data on Romanian language. A key point here will be contacting the right institutions for deciding on the limitations of our work (in terms of time coverage, canonical literature, copyright, corpus accessibility).

Moreover, as there is interest for the speech component of language, we envisage collecting a corpus in audio format, processing it and making it available to those interested. To our knowledge, the largest freely available corpus is the Romanian Speech Synthesis (RSS) database (Stan et al., 2011). The RSS Database has a total of 4 hours of recordings with time aligned text transcriptions. A set of 3,500 sentences (3.5 hours) contains 1,500 randomly chosen sentences from news sources, 1,000 phonetically rich sentences from news sources, 1,000 sentences randomly chosen from works by Ion Creangă. A set of 500 sentences (0.5 hours) contains 200 randomly chosen sentences from news sources, 100 randomly chosen sentences from novels and short stories, 200 semantically unpredictable sentences. In order to promote research in speech technology similar resources need to be created.

Acknowledgements. The initial work has been supported by the MetaNet4U PSP European project under the grant no. #270893. The future work will be carried on within the Romanian Academy's research program

References

Bittar, A., Amsili, P., Denis, P., Danlos, L. (2011). French TimeBank: An ISO-TimeML Annotated Reference Corpus. *Proc. of the 49th Annual Meeting of ACL*, Portland, Oregon, 130–134.

Caselli, T. (2010). It-TimeML: TimeML Annotation Scheme for Italian - Version 1.3.1. *Technical Report,* ILC CNR Pisa, Italy.

DGLR (2009). Dicționarul General al Literaturii Române. *București: Univers Enciclopedic*, Vol I-VII, 1993-2009.

Forăscu, C. (2009). A Romanian Corpus of Temporal Information – a Basis for Standardisation. *Proc. of KEPT 2009*, Cluj-Napoca, Romania, 77-80.

Forăscu, C. (2011). Contribuții la prelucrarea limbii române folosind metode de analiză a discursului. *PhD thesis* (in Romanian), Romanian Academy, Bucharest, 210 p.

Forăscu, C., Tufiș, D. (2012). Romanian TimeBank: An Annotated Parallel Corpus for Temporal Information. *Proceedings of LREC*, Istanbul, Turkey.

(GA, 2006) *The Grammar of the Romanian Language* (in Romanian Gramatica limbii române. Vol. I Cuvântul, vol. II Enunțul). București: Romanian Academy Publishing House.

Henrich, V., Hinrichs, E., Hinrichs, M., Zastrow, T. (2010). Service-Oriented Architectures: From Desktop Tools to Web Services and Web Applications. *Multilinguality and Interoperability in Language Processing with Emphasis on Romanian*. București: Editura Academiei (Dan Tufiș & Corina Forăscu, eds.), 69-92.

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

Ion, R., Irimia, E., Ștefănescu, D., Tufiș, D. (2012). ROMBAC – The ROManian Balanced Annotated Corpus. *Proceedings of LREC*, Istanbul Turkey.

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.

Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufiș, D., Varga, D. (2006). The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'2006)*. Genoa, Italy.

Tiedemann, J. (2009). News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces. *Recent Advances in Natural Language Processing* (vol V). Amsterdam/Philadelphia: John Benjamins (N. Nicolov and K. Bontcheva and G. Angelova and R. Mitkov, eds.), 237-248.

Tufiș, D., Ion, R. (2007). New Tagset Specifications. *Research Report*, RACAI, Bucharest, June (in Romanian)

Tufiș, D., Dragomirescu, L. (2004). Tiered Tagging Revisited. *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004)*. Lisbon, Portugal, 39-42.

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

A POOL OF BASIC RESOURCES FOR PROCESSING THE ROMANIAN LANGUAGE

DAN TUFIȘ

Research Institute for Artificial Intelligence, Romanian Academy

tufis@racai.ro

Abstract

METANET4U (http://metanet4u.eu/) is part of a cluster of projects aiming at fostering the technological foundations of a multilingual European information society. These projects follow specifications and recommendations issued by the META-NET Network of Excellence (http://www.meta-net.eu) and commonly use META-SHARE (developed within META-NET) a sustainable network of repositories of language data, tools and related web services documented with high-quality matadata, aggregated in central inventories allowing for uniform search and access to resources (http://www.meta-net.eu/meta-share). As a partner in METANET4U, RACAI delivered through META-SHARE several mono- and multi-lingual textual resources which will be briefly described in this article.

1. Introduction

A few years ago, in an invited talk at the Austrian Academy of Science, we made the following statement: "*In the quest for fast deploying of NL-based applications it seems that the concern on the major problems of language resources is losing momentum and there is an overestimation of what machine learning can do in avoiding the highly expensive manual involvement in the process of building adequate language resources. A well known slogan of the data intensive approaches to language processing (attributed to Bob Mercer) is «Better Data is More Data». The motivation behind this credo is that, due to natural redundancy in language, the main linguistic regularities would be revealed by statistical computing over huge amounts of raw data. While this continues to be true, it needs amendments: «Better Data is More Accurately Pre-Processed/Annotated Data». With the intentional ambiguity embedded into this new slogan, the idea is that exploiting the existing state-of-the-art linguistic pre-processing technologies (language identification, tokenization, tagging, lemmatization, chunking, dependency linking, text categorization, etc.), available for most of the languages, the data sparseness threat is tremendously reduced and intelligent workflows architectures for automatic acquisition, annotation and indexing of linguistic data, with humans involved in the process, can lower the data hunger and increase the quality of the targeted linguistic services*"[1]. Exactly in this spirit, META-NET, started in 2010, is a Network of Excellence dedicated to fostering the technological foundations of a multilingual European Information Society (http://www.meta-net.eu/). The idea to

[1] Dan Tufiș: "Going for a hunt? Don't forget the bullets!", FLaReNet , The 1st European Language Resources and Technologies Forum: *Shaping the Future of the Multilingual Digital Europe, Austrian Academy of Sciences, 12-13 February 2009*

collect large linguistic resources (data and tools), to ensure their maintenance for long term, to improve the data quality and promote interoperability as well as to create appropriate wide distribution channels was put into practice at the beginning of 2011 when three daughter projects (METANET4U, METANORD and CESAR) launched in parallel to synergically implement the largest pan-European linguistic infrastructure ever planned. The three projects, which include representatives of all EU member states, are strategically coordinated by the META-NET so that the same good practices and standards (where they exist) be used for high quality resources creation, documentation and distribution for all European languages. Romania is represented in Meta-NET by the Romanian Academy (Research Institute for Artificial Intelligence, Bucharest) and University "A.I. Cuza" of Iași (Faculty of Informatics) and actively participates with the two representatives to the strategy implementation within the METANET4U project (http://metanet4u.eu/).

In accordance with the META-NET strategy and guidelines, there are three main goals of the METANET4U project (as well as for the other two sister projects):

a) to collect, organize and disseminate information that gives an updated insight into the current status and the potential of language related activities, for each of the national and/or language communities represented in the project. This includes organizing and providing a description of: language usage and its economic dimensions; language technologies and resources, products and services; main actors in different areas, including research, industry, government and society in general; public policies and programs; prevailing standards and practices; current level of development, main drivers and roadblocks; etc;

b) to assemble and prepare language resources for distribution. This includes collecting languages resources; documenting them and upgrading them to agreed standards and guidelines; linking and cross-lingual aligning them where appropriate.

c) to distribute the assembled language resources through exchange facilities that can be used by language researchers, developers and professionals. This includes collaborating with other projects and, where useful, with other relevant multi-national forums or activities. This includes also help in building and operating broad inter-connected repositories and exchange facilities;

d) to mobilize national and regional actors, public bodies and funding agencies by raising awareness with respect to the activities and results of the project, in particular, and of the whole area of language resources and technology, in general.

These projects commonly use META-SHARE (developed within META-NET) a sustainable network of repositories of language data, tools and related web services documented with high-quality metadata, aggregated in central inventories allowing for uniform search and access to resources (http://www.meta-net.eu/meta-share). The timetable for the METANET4U project specifies three milestones deliveries, generically called *batches*: batch 1 at the end of November 2011, batch 2 at the end of August 2012 and batch 3 at the end of January 2013. As a partner in METANET4U,

RACAI already delivered through META-SHARE, several mono- and multi-lingual textual resources briefly described in the rest of this article.

2. RACAI Batch 1 resources

The resources delivered by RACAI for the batch 1 were developed in their initial form during several years of hard work. To answer the METANET requirements, they had to be documented according to the prescribed metadata format, validated, extended and updated for new formats compliance. There were 8 ”heavy weight” resources which are now accessible via METASHARE v1.0 under the individually specified licenses: RO-WordNet 3.0, WEB-DEX, RO-TblWordform, Multilingual News Corpus, RO-JRC-ACQUIS, Romanian-English SemCor corpus, Romanian-English TimeBank and Romanian Balanced Corpus. Two exogenous additional resources were cleaned, adapted and documented at RACAI: RO-SAM speech corpus and its textual transcription (part of EUROM multilingual speech corpus) and a set of sentences, manually annotated for subjectivity (POS, NEG, NEU). Due to space restrictions we will provide in the following, very brief descriptions of each of the endogenous resources uploaded on RACAI’s METASHARE platform (http://ws.racai.ro:9191/), a snapshot of which is shown in Figure 1. They can be downloaded according to conditions specified in the associated licenses (most of them, free for research purposes).

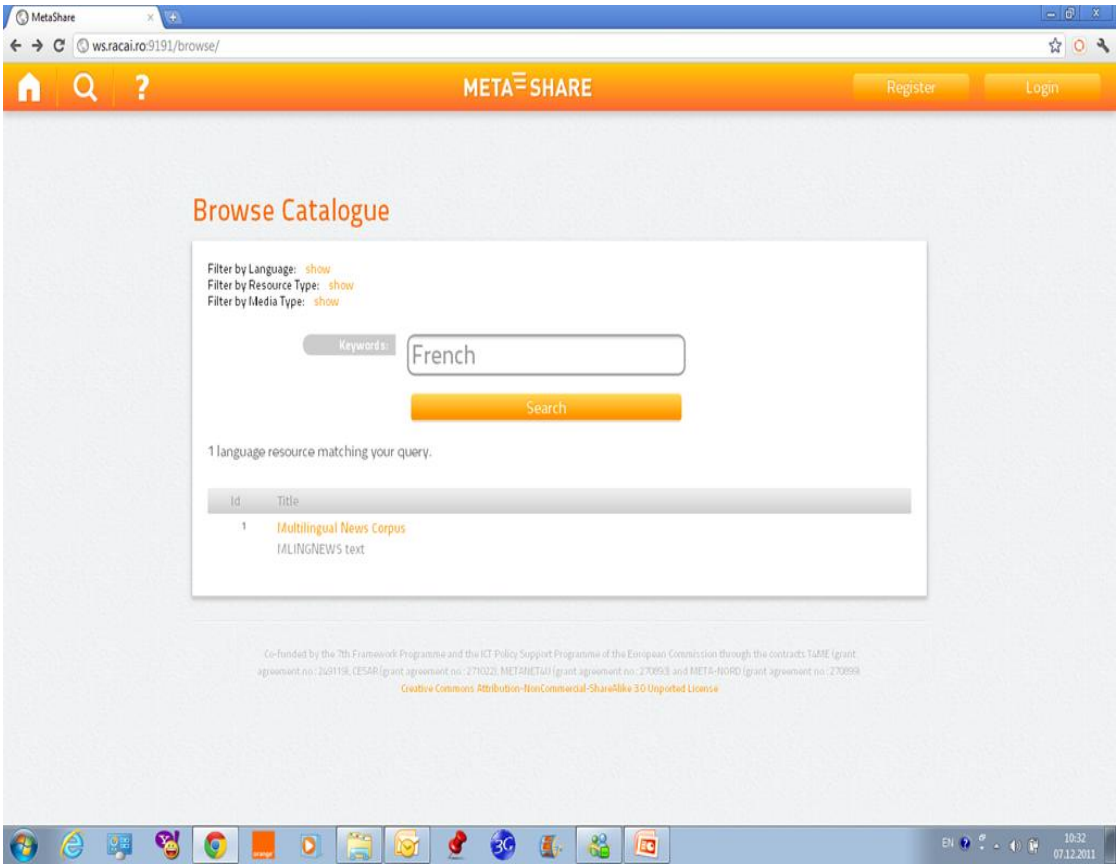

Figure 1: Snapshot of the RACAI MetaShare platform

2.1. RO-WordNet3.0

Ro-WordNet (RWN) is a lexical ontology following the Princeton WordNet (PWN http://wordnet.princeton.edu/wordnet/download/) organizational principles (Fellbaum, 1998). The synsets in RWN are aligned with PWN3.0 and, additionally, they are associated with SUMO/MILO concepts and labeled with DOMAINS3.0 categories. RWN is distributed as an XML file, observing the encoding of BalkaNet wordnets. The characters have been encoded in UTF8, multiple typing errors have been corrected, several semantic conflicts (same sense occurring in two or more synsets) have been removed and alignment was computed to the Princeton WordNet 3.0. Version 3.0 of the Princeton WordNet achieved a major restructuring of the lexical ontology and the same restructuring has been observed in the Romanian WordNet. Due to the new lexical ontology architecture, some previous synsets disappeared, some others were split and others were partially merged. A typical entry (synset) of the lexical ontology has the structure exemplified in Figure 2.

```
<SYNSET><ID>ENG30-xxxxxxxx-C</ID><POS>cat</POS>
<SYNONYM>[<LITERAL>literal<SENSE>k</SENSE></LITERAL>]+</SYNONYM>
           <DEF> a definition </DEF>[<BCS>n</BCS>]
          [<ILR>synset-ID<TYPE>name-of-relation</TYPE></ILR>]+
          [<DOMAIN>a domain</DOMAIN>]+
          [<SUMO>a sumo-concept<TYPE> a type of mapping
</TYPE></SUMO>]
<\SYNSET>
```

Figure 2: A typical entry structure in the RO-WordNet3.0

The value of the <ID> tag is a unique identifier for the aligned synset in PWN3.0 (the numerical value is the offset of the respective synset in the PWN database). The trailing character C in the ID value is one of N, V, R, A. The value of the <POS> is one of the N, V, R, A (identical to the character C) identifying the part of speech of the literals in the current synset. One should notice that in the Romanian wordnet the adjectival satellites (marked with the category S in PWN) are included into the A category.

Under the tag <SYNONYM> there are one or more <LITERAL>s each being immediately followed by a sense number. Unlike in PWN, here the numbering is not related to the frequency of the respective sense of the literal, but it follows the numbering conventions from the Romanian Explanatory Dictionary (DEX), the reference dictionary by the Romanian Academy. The tag <DEF> marks up the definition from DEX. In some cases (namely when the respective sense was not documented in DEX, the definition is a professional translation of the corresponding PWN definition). The <BCS> tag is optional and marks up the so called base concept synsets. The value of the tag is 1, 2 or 3, according to what was called in BalkaNet BCS1, BCS2 and BCS3 synsets (see Tufiş et al, 2004a, b, Tufiş et al., 2008b).

A synset entry contains one or more relations towards other synsets. This information is encoded by means of the <ILR> tag (Internal Language Relation) which uniquely identifies the target synset of the relation specified by the tag <TYPE>. The relations (except for the language specific ones) are transferred from PWN3.0.

The tag <DOMAIN> is one of the labels specified by the DOMAINS-3 taxonomy (Bentivogli et al., 2004). The tag <SUMO> marks up the SUMO/MILO concept corresponding to the synset in PWN3.0 that is the translation equivalent for the

Romanian synset. The tag <TYPE> embedded into the content of <SUMO> tag describes the type of mapping: "=" defining exact mapping and "+" defining an approximate mapping (the SUMO concept is more general than the meaning of the current entry).

The current (validated)[2] version contains 30,006 synsets, with the following distribution: 21158 Noun synsets, 7163 Verb synsets, 851 Adjective synsets and 834 Adverb synsets.

2.2. WEB-DEX

WEB-DEX is an explanatory dictionary based on the 1996 edition of the standard explanatory dictionary of Romanian published by the Romanian Academy. The lexical stock covers the basic general language vocabulary of Romanian. It contains 54.222 entries XML encoded, according to the Concede schema (http://www.itri.brighton.ac.uk/projects/concede/DR2.1/concede.dtd). The structure of an entry in WEB-DEX is exemplified below:

```
<entry id="JACHETĂ">
   <hw>JACHETĂ</hw><stress>JACH`ETĂ</stress>
    <alt><brack><gram>nominativ_feminin_singular_indefinit</gram>
                <orth>jachetă</orth></brack>
          <brack><gram>nominativ_feminin_plural_indefinit</gram>
                 <orth>jachete</orth></brack></alt>
   <pos>substantiv</pos><gen>feminin</gen>
    <struc>
      <def>Haină ( tricotată) femeiască încheiată în față, care acoperă
partea de sus a corpului și care se poartă peste bluză sau peste
rochie </def>
     <struc type="Sec">
       <def>Haină bărbătească de ceremonii, croită pe talie, lungă
până aproape de genunchi.</def> </struc>
   </struc>
  <etym>Din limba  <lang>fr.</lang>jaquette</etym>
</entry>
```

Figure 3: An example of an entry in WEB-DEX

A multi-criterial search engine has been implemented (not delivered yet) in JavaScript. Till the end of the METANET4U an updated version (much faster and more intuitive to use) will be implemented.

2.3. RO-TblWordForm

This resource is a wordform lexicon containing statistical information extracted from a large collection of texts (more than 41,000,000 tokens). The lexicon is a flat file, one entry per line, fields being tab separated, all the characters being UTF8 encoded. There are 111462 entries and each entry is a four-field line, tab separated:

`<wordform><tab>lemma<tab><MSD><tab><frequency>` where:

- <wordform> is the occurrence form in the underlying corpus,

[2] A larger version (not entirely validated) of Ro-WordNet can be browsed at the web address www.racai.ro/wnbrowser.

- <lemma> is the lemma of the wordform or "=", if the wordform is the lemma form,
- <MSD> is a morpho-syntactic tag compliant with the Multext-East specifications[3],
- <frequency> is the number of occurrences of the wordform in the underlying corpus.

For reliable statistical use of the lexicon, only the word forms that occur at least 5 times in the corpus have been retained. Each of the 14 grammatical types defined by the updated Multext-East specifications (Tufiş and Ion, 2007) are represented in the wordform lexicon.

The MSD encoding is a linear attribute value representation with fixed positions for each part of speech. Each position corresponds to a specific attribute and it is filled in by one character code. If the respective attribute is not relevant for the combination of the other attribute-values its position is filled in with the special character "-". For instance, a singular (s) masculine (m) common (c) noun (N) definite form (y) and in an oblique case – genitive or dative (o) will be encoded as **Ncmsoy**; the code **Vmip2s** describes a main (m) verb (V) indicative mood (i), present tense (p) second person (2) singular (s). The MSDs have been manually assigned by trained linguists.

2.4. Multilingual News Corpus

This is a collection of 5541 strongly comparable documents (UTF8 character encoding) in three languages: Romanian, English and French. The text types contained by the corpus are: journalistic language as used in the daily newspapers and official language as used in legal documents.

The tri-lingual corpus is represented in XCES format (http://www.xces.org/) and is provided as 5 sets of data grouped in separate folders ("ec.europa.eu", "euronews", "europarl1", "europarl2", "europarl3"[4]). Each folder has 3 subfolders named "en-xces", "ro-xces" and "fr-xces" for English, Romanian and French documents (in xces format). The filenames for comparable entries start with the same unique identifier (either a numeric value or a randomly generated GUID) and end with the character '_' and their language code (e.g. 1_EN.xml). Examples:

euronews\en-xces\1_EN.xml euronews\ro-xces\1_RO.xml euronews\fr-xces\1_FR.xml

europarl1\en-xces\1_EN.xml europarl1\ro-xces\1_RO.xml europarl1\fr-xces\1_FR.xml

The unique identifier is relative to each set (europarl1, europarl2, euronews etc.) meaning that "euronews\en-xces\1_EN.xml" is not the same document as "europarl1\en-xces\1_EN.xml". The quantitative data for the multilingual corpus is summarized below:

- ec.europa.eu (set 1 of files): 137 documents for each language (total 411 documents),
- Euronews (set 2 of files): 506 documents for each language (total 1518 documents),

[3] http://nl.ijs.si/ME/V4/
[4] http://www.statmt.org/europarl/

- europarl1 (set 3 of files): 492 documents for each language (total 1476 documents),
- europarl2 (set 4 of files): 500 documents for each language (total 1500 documents),
- europarl3 (set 5 of files): 212 documents for each language (total 636 documents).

The number of tokens (words) is 1,334,942 for English, 659,031 for Romanian and 1,480,103 for French. All the texts in the three languages are tokenized, tagged, lemmatized and chunked by means of our TTL environment (Ion, 2007, Tufis et al., 2008). TTL is entirely written in Perl and performs named entity recognition, sentence splitting, tokenization, POS tiered tagging Tufiș, 1999, Tufiș and Dragomirescu, 2004) and chunking. We have exposed it as a SOAP compliant web service with the WSDL file available at http://ws.racai.ro/ttlws.wsdl and also as a REST web-service for the WebLicht platform (Henrich et al., 2010).

The example below shows the XML mark-up for two parallel sentences (Romanian and French) from the multilingual corpus:

```
<xces:p id="p7"><xces:s id="s5_RO_7">
<xces:tok base="asistenţă" msd="Ncfsry;Np#1"
type="word">Asistenţa</xces:tok>
<xces:tok base="vrea" msd="Va--3s;Vp#1" type="word">va</xces:tok>
<xces:tok base="fi" msd="Vanp;Vp#1" type="word">fi</xces:tok>
<xces:tok base="furniza" msd="Vmp--sf;Vp#1"
type="word">furnizată</xces:tok>
<xces:tok base="în" msd="Spsa;Pp#1" type="word">în</xces:tok>
<xces:tok base="trei" msd="Mc-p-l;Pp#1,Np#2"
type="word">trei</xces:tok>
<xces:tok base="sau" msd="Ccssp;Pp#1,Np#2" type="word">sau</xces:tok>
<xces:tok base="patru" msd="Mc-p-l;Pp#1,Np#2"
type="word">patru</xces:tok>
<xces:tok base="rată" msd="Ncfp-n;Pp#1,Np#2"
type="word">rate</xces:tok>
<xces:tok base=":" msd="COLON"
type="punctuation">:</xces:tok></xces:s></xces:p>
<xces:p id="p7"><xces:s id="s5_FR_7">
<xces:tok base="il" msd="Pp3mp;Vp#1" type="word">Ils</xces:tok>
<xces:tok base="etre" msd="Vmif3p;Vp#1" type="word">seront</xces:tok>
<xces:tok base="disponible" msd="Af-fp;Ap#1"
type="word">disponibles</xces:tok>
<xces:tok base="en" msd="Sp;Pp#1" type="word">en</xces:tok>
<xces:tok base="trois" msd="M;Pp#1,Np#1" type="word">trois</xces:tok>
<xces:tok base="ou" msd="Cc" type="word">ou</xces:tok>
<xces:tok base="quatre" msd="M;Np#2" type="word">quatre</xces:tok>
<xces:tok base="tranche" msd="Ncfp;Np#2"
type="word">tranches</xces:tok>
<xces:tok base=":" msd="COLON"
type="punctuation">:</xces:tok></xces:s></xces:p>
```

Figure 4: Two parallel sentences from the Multilingual News Corpus

2.5. RO-JRC Acquis

The corpus consists of a subset of the Romanian version of the JRC Acquis (Steinberger et al., 2006), based on the common set of laws of the European Union member states

(Acquis Communautaire). The language of the corpus is standard Romanian, orthography being compliant with the current Romanian Academy norms. The diacritical signs are in place (Tufiș and Ceaușu, 2008). The text type of the corpus is the official language as used in legal documents. There are 10,704 documents which were selected so that their equivalent documents also exist in English and French. The corpus contains 34,234,437, out of which 27,968,652 are words and the rest punctuation marks.

This corpus, as all the other corpora developed at RACAI, is represented in XML Corpus Encoding Standard (XCES) format which is compliant with the XCES Schema revision 0.4 (2003). The RO-JRC Acquis corpus has been carefully cleaned and all its characters are UTF-8 encoded. A special mention is due for the correction of the Romanian letters "ș" and "ț" and their upper case variants "Ș" and "Ț" which were not encoded as in the Latin 2 character set. The corpus is annotated at paragraph, sentence, constituent group and word levels, providing morpho-lexical, syntactic information and sense disambiguation. One should note that the document paragraphs are marked with unique IDs (CELEX codes), same in any language (except for the language code) for the documents which contain the same information. These codes allow for the unambiguous identification of parallel documents in any of the 22 languages covered by JRC-Acquis.

The sense of a content word is specified by a new attribute *ili* of the *xces:tok* tag (see Figure 4). Its value represents the Princeton WordNet sense identifier for the current token. It has been automatically computed based on the WSD methods for parallel corpora, described in Tufiș et al., (2004c) and Ion (2007). The terms (multiword units, glued together by the underscore) and words missing from the Romanian WordNet are not sense disambiguated. As the WSD process is minimal error committed, uncertainty is preferred to wrong decisions. This is why some tokens are labeled with the most probable subset of their possible senses.

2.6. SemCor Corpus

En-Ro-SemCor corpus (Lupu et al., 2005; Ion, 2007) is an English-Romanian parallel corpus which was developed starting from the English SemCor (Mihalcea and Pedersen, 2003), a sense-tagged corpus created at Princeton University by the WordNet Project research team. SemCor is a subpart of the Brown balanced corpus (Kučera and Francis, 1967), containing news articles, literature, scientific and religious texts. In spite of its small dimension, SemCor has been extensively used both as training and testing data in various Word-Sense Disambiguation experiments and competitions, as word-sense annotated resources are scarce.

The Romanian side of the corpus is a partial translation (only 81 out of 352 original files were translated by the NLP group at FII-UAIC), and in the En-Ro-SemCor corpus only the translated files were included. En-Ro-SemCor contains a total of 178,499 words for English and 175,603 words for Romanian (Ion, 2007) and is marked-up conformant to XCES format. The corpus is annotated at paragraph, sentence, constituent group and word levels. The alignment is encoded in the sentence ids. Sentences having the same id are reciprocal translation. Each sentence is segmented into tokens, including punctuation. The diacritics and all special characters are encoded as SGML entities. Each token has a descriptor attribute containing syntactic and semantic information

about its grammatical meta-category[5], lemma, morpho-syntactic descriptor (msd) – tag, syntactic chunk membership (Np – Noun Phrase; Vp – Verb Phrase; Ap – Adjectival Phrase; Pp – Prepositional Phrase), associated Princeton WordNet 3.0 word-sense and *syntactic dependency link(s)* in the current sentence. The chunking annotation has been achieved based on a regular grammar defined over the MSD tags. The word-sense labels in the English part of the corpus have been manually assigned and, via word alignment, transferred to the translation equivalents in the Romanian part.

Besides the tags which were used in other corpora annotations, RO-EN SemCor uses a new value, included into the *ili* attribute, namely a numerical index, immediately following the sense identifier. It represents, on a 0-based positioning in the current sentence, the word *lexically attracted* (a kind of dependency relation, see (Ion and Barbu-Mititelu, 2006) for further details) by the word under consideration. For instance, in the example shown below, the annotation of the word *said*:

*<xces:tok base="say" msd="1+,Vmis;Vp#1;ili:ENG30-01009240-v;**1**" type="word">said</xces:tok>*

specifies that the word at position 1 (*Fulton_County_Grand_Jury*) is the one entering the relation of *lexical attraction* with the word *said*. The information on lexical attraction has been automatically annotated using LexPar (Ion and Barbu-Mititelu, 2006), an application using Lexical Attraction Models (Yuret, 1998) further developed as Meaning Affinity Models (Ion, Tufiș, 2007).

```
<xces:p id="p1">
 <xces:s id="br_a01_1_1_en">
<xces:tok base="the" msd="2+,Dd;Np#1" type="word">The</xces:tok>
<xces:tok base="Fulton_County_Grand_Jury"
msd="8+,Np;Np#1;ili:ENG30-00031264-n;0"
type="word">Fulton_County_Grand_Jury</xces:tok>
<xces:tok base="say" msd="1+,Vmis;Vp#1;ili:ENG30-01009240-v;1"
type="word">said</xces:tok>
<xces:tok base="Friday" msd="1+,Ncns;Np#2;ili:ENG30-15164463-n;2"
type="word">Friday</xces:tok>
<xces:tok base="a" msd="21+,Ti-s;Np#3;5" type="word">an</xces:tok>
<xces:tok base="investigation" msd="1+,Ncns;Np#3;ili:ENG30-05800611-
n;3" type="word">investigation</xces:tok>
<xces:tok base="of" msd="5+,Sp;Pp#1;7" type="word">of</xces:tok>
<xces:tok base="Atlanta" msd="8+,Np;Pp#1,Np#4;ili:ENG30-09076675-n;5"
type="word">Atlanta</xces:tok>
<xces:tok base="&apos;s" msd="21+,St;Pp#1,Np#4;7"
type="word">&apos;s</xces:tok>
<xces:tok base="recent" msd="1+,Afp;Pp#1,Np#4,Ap#1;ili:ENG30-01730444-
s;10" type="word">recent</xces:tok>
<xces:tok base="primary_election" msd="1+,Ncns;Pp#1,Np#4;ili:ENG30-
00182571-n;3" type="word">primary_election</xces:tok>
<xces:tok base="produce" msd="1+,Vmis;Vp#2;ili:ENG30-02141146-v;10"
type="word">produced</xces:tok>
```

[5] The meta-categories are hand-made clusters taking into consideration the empirical evidence of POS translation affinities: if two or more grammar categories are in the same meta-category (e.g. N, V, A), then words from these categories may be translated, under specific circumstances, by words from another category in the same cluster.

```
<xces:tok base="&quot;" msd="DBLQ"
type="punctuation">&quot;</xces:tok>
<xces:tok base="no" msd="22+,Dz3;Np#5;14" type="word">no</xces:tok>
<xces:tok base="evidence" msd="1+,Ncns;Np#5;ili:ENG30-05823932-n;11"
type="word">evidence</xces:tok>
<xces:tok base="&quot;" msd="DBLQ"
type="punctuation">&quot;</xces:tok>
<xces:tok base="that" msd="31+,Cs;19" type="word">that</xces:tok>
<xces:tok base="any" msd="22+,Di3;Np#6;18" type="word">any</xces:tok>
<xces:tok base="irregularity" msd="1+,Ncnp;Np#6;ili:ENG30-00737188-
n;19" type="word">irregularities</xces:tok>
<xces:tok base="take_place" msd="1+,Vmis;Vp#3;ili:ENG30-00339934-v;14"
type="word">took_place</xces:tok>
<xces:tok base="." msd="PERIOD" type="punctuation">.</xces:tok>
</xces:s>
</xces:p>
```

Figure 5: An annotated English sentence from the parallel corpus En-Ro SemCor

2.7. Ro-TimeBank Corpus

Result of a PhD project research (Forăscu, 2011), Ro-TimeBank corpus is another example of semantic annotation transfer, based on word alignment, from a heavily annotated English corpus into the Romanian translated texts. The source corpus was TimeBank corpus version 1.2[6] (Pustejovsky et. al., 2006).

The 183 files in the original TimeBank were carefully translated into Romanian trying to preserve, when possible, the same word order and avoiding paraphrases[7]. Afterwards, both English and Romanian texts were tokenized, tagged, lemmatized and finally word aligned by means of TTL and YAWA aligner (Ion, 2007). The word alignment has been checked and manually corrected and was followed by another hand validation and correction. The final word alignment was the means by which all the TimeML annotations pertaining to an English word were automatically transferred to its Romanian equivalent. In spite of 96.53% of valid transfers, (Forăscu, 2011 p. 115) manual corrections were necessary on the Romanian TimeML mark-up. In (Forăscu, 2011) the major sources for the annotation transfer problems are listed: words not translated, different cross-lingual syntactic properties of some verbs, temporal SIGNALs in English not lexicalized in Romanian etc.

Besides the TimeML annotated Romanian corpus, the usual RACAI encoding of parallel corpora has been provided for the En-Ro bitext. The quantitative data for both parts of the bitext is shown in Table 1, while Figure 6 exemplifies the encoding of a translation unit (a pair of translation equivalent sentences) from the bilingual corpus. The encoding shown in Figure 6, slightly different from the previous examples, is conformant with RACAI's previous XML schema. The differences are minor and a Perl script is available for automatically converting this annotation into fully conformant XCES mark-up.

Table1: Quantitative data about the parallel corpus Ro-En Time Bank

[6] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2006T08

[7] This decision was motivated by the intention of ensuring the best possible word alignment, based on which the TimeML annotations were transferred from English to Romanian.

Unit	RoTimeBank	EnTimeBank
Sentences	4715	4715
lexical units	65375	61042
unique lexical units	12640	10586

```
<tu id="16">
<seg lang="en">
<s id="ABC19980304.1830.1636_en_16">
<w lemma="once" ana="14+,Rmp" chunk="Ap#1">Once</w>
<w lemma="colonel" ana="1+,Ncns" chunk="Np#1">Colonel</w>
<w lemma="Collins" ana="8+,Np" chunk="Np#1">Collins</w>
<w lemma="be" ana="3+,Vais3s" chunk="Vp#1">was</w>
<w lemma="pick" ana="1+,Vmps" chunk="Vp#1,Ap#2">picked</w>
<w lemma="as" ana="5+,Sp" chunk="Pp#1">as</w>
<w lemma="a" ana="21+,Ti-s" chunk="Pp#1,Np#2">a</w>
<w lemma="NASA" ana="8+,Np" chunk="Pp#1,Np#2">NASA</w>
<w lemma="astronaut" ana="1+,Ncns" chunk="Pp#1,Np#2">astronaut</w>
<c>,</c>
<w lemma="she" ana="13+,Pp3fsn" chunk="Vp#2">she</w>
<w lemma="follow" ana="1+,Vmis" chunk="Vp#2">followed</w>
<w lemma="a" ana="21+,Ti-s" chunk="Np#3">a</w>
<w lemma="normal" ana="1+,Afp" chunk="Np#3,Ap#3">normal</w>
<w lemma="progression" ana="1+,Ncns" chunk="Np#3">progression</w>
<w lemma="within" ana="5+,Sp" chunk="Pp#2">within</w>
<w lemma="NASA" ana="8+,Np" chunk="Pp#2,Np#4">NASA</w>
<c>.</c></s></seg>
<seg lang="ro">
<s id="ABC19980304.1830.1636_ro_16">
<w lemma="O" ana="1+,Mc">O</w>
<w lemma="dată" ana="1+,Ncfsrn" chunk="Np#1">dată</w>
<w lemma="ce" ana="4+,Pw3--r" chunk="Np#2">ce</w>
<w lemma="colonel" ana="1+,Ncmsry" chunk="Np#2">colonelul</w>
<w lemma="Collins" ana="8+,Np" chunk="Np#2">Collins</w>
<w lemma="avea" ana="3+,Va--3s" chunk="Vp#1">a</w>
<w lemma="fi" ana="3+,Vap--sm" chunk="Vp#1">fost</w>
<w lemma="alege" ana="1+,Vmp--sf" chunk="Vp#1,Ap#1">aleasă</w>
<w lemma="ca" ana="14+,Rc">ca</w>
<w lemma="astronaut" ana="1+,Ncms-n" chunk="Np#3">astronaut</w>
<w lemma="NASA" ana="8+,Yn" chunk="Np#3">NASA</w>
<c>,</c>
<w lemma="el" ana="13+,Pp3fsr--------s" chunk="Vp#2">ea</w>
<w lemma="avea" ana="3+,Va--3s" chunk="Vp#2">a</w>
<w lemma="urma" ana="1+,Vmp--sm" chunk="Vp#2,Ap#2">urmat</w>
<w lemma="o" ana="1+,Mc">o</w>
<w lemma="ascensiune" ana="1+,Ncfsrn" chunk="Np#4">ascensiune</w>
<w lemma="normal" ana="1+,Afpfsrn" chunk="Np#4,Ap#3">normală</w>
<w lemma="în" ana="5+,Spsa" chunk="Pp#1">în</w>
<w lemma="cadru" ana="1+,Ncmsry" chunk="Pp#1,Np#5">cadrul</w>
<w lemma="NASA" ana="8+,Yn" chunk="Pp#1,Np#5">NASA</w>
<c>.</c></s></seg>
</tu>
```

Figure 6: An example of a translation unit from the En-Ro TimeBank Corpus

In Figure 7 the TimeML annotation for a Romanian sentence is exemplified. The name entities and events appear in bold face:

*Filiala din **SUA** a **Ratners Group PLC** a fost **de acord** să **achiziționeze** vânzătorul de bijuterii **Weisfield's Inc.** pentru **$50** pe acțiune, sau aproximativ **$55 milioane**.*

```
<s>Filiala din<ENAMEX TYPE="LOCATION">SUA</ENAMEX> a <ENAMEX
TYPE="ORGANIZATION">Ratners Group PLC</ENAMEX> a fost <EVENT
aspect="PERFECTIVE" class="I_ACTION" eid="e1" eiid="ei1993"
eventID="e1" polarity="POS" pos="NOUN" tense="PAST" mainevent="YES"
pred="de_acord">de acord</EVENT> să <EVENT aspect="NONE"
class="OCCURRENCE" eid="e3" eiid="ei1994" eventID="e3" polarity="POS"
pos="VERB" tense="PRESENT" mainevent="NO" pred="achiziționa"
mood="SUBJONCTIVE" vform="NONE">achiziționeze</EVENT> vânzătorul de
bijuterii <ENAMEX TYPE="ORGANIZATION">Weisfield's Inc .</ENAMEX>
pentru <NUMEX TYPE="MONEY">$ 50</NUMEX> pe acțiune , sau aproximativ
<NUMEX TYPE="MONEY">$ 55 milioane</NUMEX>.</s>
```

Figure 7: An annotated Romanian sentence from Ro-TimeBank corpus

2.8. Romanian Balanced Corpus (ROMBAC)

The previous corpora contain Romanian translations from other languages (mostly English). Besides their general use for NLP and machine translation, they can be extremely helpful for translational studies. Similarly, the large collection of comparable corpora, collected by the ACCURAT project (www.accurat-project.eu), containing huge quantities of similar documents (most of them translations) in English, Estonian, German, Greek, Latvian, Lithuanian and Romanian, may be appealing to specialists in translation studies, although they were collected for more practical purposes of statistical machine translation.

ROMBAC is the core of the future Reference corpus for Contemporary Romanian. It will consider both original Romanian texts and professional translations, and a large palette of linguistic text types. The metadata that will be associated to the corpus will allow text selection on many criteria, including source language and linguistic text type.

In its present state, ROMBAC consists of equal shares of texts from 5 different text types: journalism, national legislation, fiction, medicine and biographical data for Romanian literary personalities. For each category, texts have been selected containing around 8,000,000 words, so that the entire corpus counts 41,534,961 tokens (660,000 unique words), including punctuation. The corpus is represented in XCES format with all characters UTF8 encoded.

The initial documents for the corpus (PDF, doc, docx) were first converted into text files, normalized at the orthographic level, cleaned of footnotes, headers and page numbers and the punctuation was separated from the words. After this preliminary phase, the corpus was subjected to an annotation process using the TTL text processing platform developed at RACAI (Ion, 2007; Tufiș et al., 2008a).

Because of limited human resources, time constraints and the dimension of the corpus, hand validation of each individual token was out of question. Therefore, the validation stage was implemented as a coherent methodology for automatically identifying as many POS annotation and lemmatization errors as possible. This methodology implies, among other techniques, several iterations of the analysis for the tokens whose biased

annotation is different from the one in the initial annotated corpus. The biased annotation is produced by means of language models constructed from the same data one would like to annotate. Where the initial annotation and the biased annotation differ it is likely to have an error, so that we restricted our hand validation only to these cases. The complete methodology is described in detail in (Tufiș & Irimia, 2006) and, as shown there, the estimated error rate is around 2%.

3. Conclusions

This article presented a few language resources for Romanian Language processing created at the Research Institute for Artificial Intelligence of the Romanian Academy. Several other useful resources were delivered to METANET4U by the Faculty of Computer Science of the "A. I. Cuza" University of Iași. Yet, there are several other Romanian groups that own many valuable resources and language processing tools. Releasing them, under whatever licenses the owners prefer, would be highly beneficial for farther and faster advancement of Language Technology in Romania. The academic content creators, whose work should be appropriately acknowledged and cited, with the intellectual property rights carefully protected, may significantly ease the dissemination of the textual and multi-media data by getting in closer contacts with the NLP researchers and developers. The language industry in Romania is in its infancy and their general complaint refers to the very difficult (even impossible) access to high quality data locked under provisions of the IPR regulations. Well regulated channels of data collection, cleaning and distribution are currently constructed within large European and trans-European language technology infrastructures such as CLARIN-ERIC or META which complement and improve the services of older language resources associations such as ELRA/ELDA in Europe or LDC in USA.

Acknowledgements. This work has been supported by the MetaNet4U PSP European project under the grant no. #270893.

References

Bentivogli, L., Forner, P., Magnini, B., Pianta, E. (2004). Revising WordNet Domains Hierarchy: Semantics, Coverage, and Balancing. *Proceedings of COLING 2004 Workshop on "Multilingual Linguistic Resources"*, 101-108.

Ceaușu, A. (2008). Colectarea și procesarea documentelor românești ale corpusului JRC-Acquis. In Diana Maria Trandabăț, Dan Cristea, Tufiș D. (eds.), *Lucrările atelierului Resurse Lingvistice și Instrumente pentru Prelucrarea Limbii Române*, Editura Universității "Al. I Cuza", Iași.

DGLR (2009). Dicționarul General al Literaturii Române (DGLR), author: Romanian Academy, Univers Enciclopedic Publishing House, Vol I-VII, 1993-2009.

Erjavec, T. (2004). MULTEXT-East Version 3: Multilingual Morphosyntactic Specifications, Lexicons and Corpora. In *Proceedings of the 4th LREC Conference*, LREC'04, Lisbon, 1535–1538

Fellbaum, Ch. (Ed.) (1998). WordNet: An electronic lexical database. *Cambridge, MA: MIT Press*.

Forăscu, C. (2011). Contributions to Romanian language processing through discourse analysis methods. (in Romanian). *PhD thesis*. Romanian Academy, Bucharest, 210 pages.

Henrich, V., Hinrichs, E., Hinrichs, M., Zastow, T. (2010). Service-Oriented Architectures: From Desktop Tools to Web Services and Web Applications. In *Tufiș D., Corina Forăscu (eds.): Multilinguality and Interoperability in Language Processing with Emphasis on Romanian*. Editura Academiei, București, ISBN 978-973-27-1972-5, pp. 69-92.

Ide, N., Suderman, K. (2004). The American National Corpus First Release. *Proceedings of the Fourth Language Resources and Evaluation Conference* (LREC), Lisbon, 1681-84.

Ide, N., Romary, L. (2007). Towards International Standards for Language Resources. In Dybkjaer, L., Hemsen, H., Minker, W. (Eds.), *Evaluation of Text and Speech Systems*, Springer, 263-84.

Ide, N., Baker, C., Fellbaum, C., Fillmore, C., Passonneau, R. (2008). MASC: The Manually Annotated Sub-Corpus of American English. *Proceedings of the Sixth Language Resources and Evaluation Conference (LREC)*, Marrakech, Morocco.

Ion, R., Barbu-Mititelu, V. (2006). Constrained Lexical Attraction Models. *In Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference*, Menlo Park, Calif., USA. AAAI Press, 297-302.

Ion, R. (2007). Word Sense Disambiguation Methods Applied to English and Romanian. (in Romanian). *PhD thesis*. Romanian Academy, Bucharest. 153 pages.

Ion, R., Tufiș, D. (2007). Meaning Affinity Models. In *Proceedings of SEMEVAL Workshop*, ACL 2007, Prague, Czech Republic.

Ion, R., Ștefănescu, D. (2010). RACAI: Unsupervised WSD Experiments @ SemEval-2, Task 17. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval-2, Uppsala, Sweden, July 2010. Association for Computational Linguistics. ISBN: 978-1-932432-70-1, 411-416.

Kučera, H., Francis, N.W. (1967). Computational analysis of present-day American English. *Brown University Press*, Providence, Rhode Island.

Lupu, M., Trandabăț, D., Husarciuc, M. (2005). A Romanian SemCor Aligned to the English and Italian MultiSemCor. *In Proceedings of the Romance FrameNet Workshop and Kick-off Meeting, EuroLAN'05*, Babes-Bolyai University, Cluj-Napoca, Romania, 20–27.

Mihalcea, R., Moldovan, D. (1999). A method for word sense disambiguation of unrestricted text. *In Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics* (ACL 1999), College Park, MA.

Mihalcea, R., Moldovan, D. (2001). *A Highly Accurate Bootstrapping Algorithm for Word Sense Disambiguation*. International Journal on Artificial Intelligence Tools, 10(1–2).

Mihalcea, R., Pedersen, T. (2003). An Evaluation Exercise for Word Alignment. *In Proceedings of the HLT-NAACL 2003 Workshop: Building and Using Parallel Texts Data Driven Machine Translation and Beyond*, Edmonton, Canada, 1-10.

Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D., Miller, K. (1990). Introduction to WordNet: An On-Line Lexical Database. *International Journal of Lexicography*, 3:4, 235-244.

Niles, I., Pease, A. (2001). Towards a Standard Upper Ontology. *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems*, 2-9.

Pustejovsky, J., Verhagen M., Sauri R., Littman J., Gaizauskas R., Katz G., Mani I., Knippen R., Setzer A.. (2006). TimeBank 1.2. Linguistic Data Consortium, Philadelphia, ISBN: 1-58563-386-0.

Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufiș, D., Varga, D. (2006). The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'2006)*. Genoa, Italy.

Tiedemann, J. (2009). News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces. *In N. Nicolov and K. Bontcheva and G. Angelova and R. Mitkov (eds.) Recent Advances in Natural Language Processing* (vol V), John Benjamins, Amsterdam/Philadelphia, 237-248.

Tufiș, D. (1999). Tiered Tagging and Combined Classifiers. *In F. Jelinek, E. Nöth (eds) Text, Speech and Dialogue*, Lecture Notes in Artificial Intelligence 1692, Springer, 28-33.

Tufiș, D. (2002). A Cheap and Fast Way to Build Useful Translation Lexicons. In *Proceedings of the 19th Intl. Conf. On Computational Linguistics*, Taipei, 1030-1036.

Tufiș, D., Dragomirescu, L. (2004). Tiered Tagging Revisited. In *Proceedings of the 4th LREC'04 Conference*, Lisbon, 39-42

Tufiș, D., Barbu, E. (2004). A Methodology and Associated Tools for Building Interlingual Wordnets. In *Proceedings of LREC2004*, Lisbon, Portugal, 1067-1070.

Tufiș, D., Cristea, D., Stamou, S. (2004a). BalkaNet: Aims, Methods, Results and Perspectives. A General Overview. In *Romanian Journal on Information Science and Technology*, Tufiș D. (ed.) Special Issue on BalkaNet, Romanian Academy, ISSN 1453-8245, 7:2-3, 9-34.

Tufiș, D., Barbu, E., Barbu Mititelu, V., Ion, R., Bozianu, L. (2004b). The Romanian Wordnet. *Romanian Journal on Information Science and Technology*, 7: 2-3, 107-124.

Tufiș, D., Ion, R., Ide, N. (2004c). Word sense disambiguation as a wordnets validation method in BalkaNet. In *Proceedings of the 4th LREC Conference*, Lisbon, 741-744.

Tufiș, D., Irimia, E. (2006). RoCo_News - A Hand Validated Journalistic Corpus of Romanian. In *Proceedings of the 5th LREC Conference*, Genoa, 869-872.

Tufiș, D., Ion, R. (2007). Specificații pentru clasa de etichete folosite în adnotarea morfo-lexicală a limbii române. *Raport de cercetare*, iunie, Institutul de Cercetări pentru inteligență artificială, 24 pagini.

Tufiș, D., Ceaușu, A. (2008). DIAC+: A Professional Diacritics Recovering System. In *Proceedings of the 6th LREC Conference*, Marrakech.

Tufiș, D., Ion, R., Ceaușu, A., Ștefănescu, D. (2008a). RACAI's Linguistic Web Services. In *Proceedings of the 6th LREC Conference – LREC'08*, Marrakech.

Tufiș, D., Ion R., Bozianu, L., Ceaușu, A., Ștefănescu, D. (2008b). RO-Wordnet. In *Proceedings of the 4th Global WordNet Association Conference*, January 22-25, Szeged, Hungary.

Vossen, P. (Ed.) (1998). A Multilingual Database with Lexical Semantic Networks. *Dordrecht: Kluwer Academic Publishers*.

Yuret, D. (1998). Discovery of linguistic relations using lexical attraction. *PhD thesis*, Department of Computer Science and Electrical Engineering, MIT.

BUILDING A ROMANIAN CORPUS FOR SENTIMENT ANALYSIS

[1]ALEXANDRU-LUCIAN GÎNSCĂ, [1]ADRIAN IFTENE, [2]MARIUS CORÎCI

[1]*UAIC: Faculty of Computer Science, "Alexandru Ioan Cuza" University, Romania;*
[2] *Intelligentics Cluj-Napoca, Romania.*

{lucian.ginsca, adiftene}@info.uaic.ro, marius@intelligentics.ro

Abstract

In this paper we describe the process of building a Romanian corpus for the task of sentiment analysis, starting with determining the annotation standard, gathering the resources to be included in the corpus and including the methodology that was followed by the annotators. More than that, we give detailed statistics on the results we obtained so far and we present a couple of use-cases in which such a corpus is helpful. We also propose a set of metrics that are relevant for the evaluation of a sentiment analysis system. The purpose of this corpus is to be used in a broad set of scenarios, not just text classification based on the type of sentiment that it transmits. The main majority of corpora for sentiment analysis available in other languages are either obtained by automatic or semi-automatic approaches in which the whole text is labeled positive, negative or neutral and that doesn't have a high confidence level or manually annotated corpora, where the labeling is done at sentence level. Our approach operates on a word annotation level and includes markings for named entities, for sequences of words that have a certain sentiment polarity, excluding the neutral ones and links between named entities and sentiment sequence of words. We intend to publish the first polished results as open source in order for Romanian researchers that are interested in this subject to be able to contribute.

1. Introduction

Sentiment analysis or opinion mining, represents in recent years an important topic in the research community but also in the industrial environment, due to the valuable insight on customer preferences, product trends etc. that sentiment analysis can provide to interested companies. The important growth of sentiment analysis can also be observed by the increasing number of workshops dedicated to this topic that are associated with major computational linguistics conferences and a rising number of sentiment analysis start-ups.

One of the main reasons for having a gold sentiment corpus is to provide a standard testing resource to help the development phase of a sentiment analysis tool, but also to encourage a clear, unified benchmark that can be used to objectively compare the results of different systems. From a machine learning perspective, a gold corpus can be used to train a classifier or can be divided in order to be used in a training/testing split. It can also prove useful for a rule based system, in which case, having such a corpus can give important clues on how adding or removing certain rules affects the overall accuracy of the system. At the moment, as far as we know, there doesn't exist any gold sentiment corpora for Romanian, neither licensed, nor an open source one that could serve as a

starting point for adaptation to different sentiment analysis tasks. Our goal is to create such a corpus and to make it publicly available to anyone interested, especially researchers that conduct work in this domain.

The international interest in sentiment analysis is demonstrated by the significant number of sentiment annotated corpora available either open source or with different type of commercial and academic licenses available in multiple languages, such as English, German, Italian, Chinese and Japanese. Next, we will present the most notable and widely used sentiment corpora.

One of the most important corpora for English is the MPQA Opinion Corpus[1] that contains manually annotated news articles gathered from a variety of sources. The annotations respect the GATE[2] format and are provided not only for opinions, but also for beliefs, emotions, speculations etc. (Wiebe et al., 2005) and (Wilson, 2008). A corpus mostly used in opinion classification tasks is the Movie Reviews Large Dataset[3] (Maas et al., 2011) that provides 2.500 movie reviews for training and another 2.500 for test. A major difference between our proposal and this corpus is that our approach is that we use a fully manual annotation process and we target a high level of granularity for the annotations, as it will be explained in the following section, whereas the previously mentioned dataset is mostly automatically built by scraping user rated reviews and it uses only a binary positive/negative classification. Another important work worth mentioning is SentiWordNet[4] (Baccianella et al., 2010), a WordNet[5] based lexical resource for sentiment bearing words. SentiWordNet provides real values in the 0 to 1 scale symbolizing the degree of positive, negative or objectiveness for multiple meanings of the same word and it has been successfully translated into other languages, some of the most described ones being the Indian languages, as presented in (Das et al., 2010). A description of how to use SentiWordNet in a multilingual context is detailed in (Denecke, 2008). One of the most relevant corpora regarding our approach is the JDPA[6] corpus (Kessler et al., 2010) which consists of blog posts containing opinions about automobiles and digital cameras. All of these posts have been manually annotated for mentions of entities, which can be named, nominal, and pronominal. Entities are marked with the aggregate sentiment expressed toward them in the document. Also, the modifiers are annotated. An important aspect that is also captured in our proposed annotation is how different sentiment bearing segments of text influence the entities found in a sentence. This is present in JDPA by annotating the expressions which convey sentiment toward an entity with the polarity of their prior and contextual sentiments as well the mentions they target. Another important sentiment corpus is the

UMass Amherst[7] corpus which contains user reviews for different products sold online. These reviews can be found in Chinese, English, German and Japanese (Noah et al., 2008) and (Potts, Schwarz, 2008).

[1] MPQA: http://www.cs.pitt.edu/mpqa/index.html
[2] GATE: http://gate.ac.uk/
[3] LMRD: http://ai.stanford.edu/~amaas/data/sentiment/
[4] SentiWordNet: http://sentiwordnet.isti.cnr.it/
[5] WordNet: http://wordnet.princeton.edu/
[6] JDPA: http://verbs.colorado.edu/jdpacorpus/
[7] UMass: http://semanticsarchive.net/Archive/jQ0ZGZiM/readme.html

2. *Annotation process*

In this section, we present how the text that was used for annotation were gathered, we describe the annotation standard that we propose and we describe the annotation workflow.

2.1. Data acquisition

In the pilot phase of building our corpus, we used texts representing online news articles (Media Fax, România Liberă, realitatea.net etc.) and blog posts (chinezu.ro, zoso.ro etc.). In order to provide a common denominator for this initial version of the corpus, we targeted the telecommunications domain and we gathered articles about major companies, like Orange, Vodafone, Cosmote etc. It is important to mention here that these texts are used solely to aid defining the annotation standard and to develop annotation experiments. Due to copyright issues, we have no guarantee that they will be found in the open source release of the corpus.

2.2. Annotation standard

As it can be seen in Figure 1, the main components of our annotation set are the "paragraph" tag (with attribute "id"), "sentimentGroup" tag (with attributes "value" (between -4 and 4) and "id_group") and "entity" tag (with attributes "type", "sentiment", "id_entity" and "id_group").

```
<paragraph id=""></paragraph>
<sentimentGroup value="" id_group=""></sentimentGroup>
<entity type="" sentiment="" id_entity="" id_group=""></entity>
```

Figure 1: Annotation tag set.

We consider the following major categories for an entity's "type" attribute: c*ity*, o*rganization*, *company*, *country*, *person* and additionally we consider categories like *brand*, *product* and *publication* (Iftene et al., 2011). The "id_group" attribute is used to link one or more sentiment groups to an entity.

```
<paragraph id="3">
De altfel, sub semnul clarității a stat întregul eveniment,
show-ul de lumini și muzica marcând lansarea <entity type=
"product" sentiment="2.0" id_entity="1" id_group="1,2,3"> HD
Voice</entity>, <sentimentGroup value="1.0" id_group="1">
serviciu gratuit</sentimentGroup> care <sentimentGroup value=
"1.0" id_group="2">asigură un plus de claritate</sentimentGroup>
și <sentimentGroup value="1.0" id_group="3">calitate
</sentimentGroup> convorbirilor telefonice.
</paragraph>
```

Figure 2: Annotation example.

In Figure 2, we give an annotation example. As it can be observed, there is a single named entity, "HD Voice", which has the "product" type and an associated sentiment value score of "2.0". Also, it is illustrated that the 3 identified sentiment groups influence the entity, this meaning that its "id_group" attribute has the value "1,2,3".

2.3. Annotation workflow

We have experimented with two different workflows. In the first one, we first run the initial plain texts through our sentiment analysis system described in (Gînscă et al., 2011) and outputted files containing annotations for paragraphs, named entities, sentiment groups and links between the last two. The annotators were given the tasks of correcting the annotations generated by the system by removing incorrect annotations, modifying annotations or attributes and adding new annotations for the elements that the system didn't identify. In the second one, we provide to the annotators the texts labeled only with the paragraph tags, leaving them the task to annotate with the remaining tags. After observing the productivity of both methods and obtaining feedback from the annotators, we concluded that the second one provided better results and implied less time spent annotating.

For the actual annotation of texts, we have used the Serna[8] open source WYSIWYG (what you see is what you get) XML editor. The main arguments for using Serna are its flexibility in adapting it for new annotation scenarios and it has an intuitive and easy to use interface. We emphasize on the importance of the last two features due to the fact that in the annotation process, besides computer science students have also been involved two students without any computer science background. Serna allows Python plug-ins, XSLT and XSL-FO configuration files that can be used to define the annotation tag set, attributes and attribute values, but also formatting settings, such as the colour of annotations.

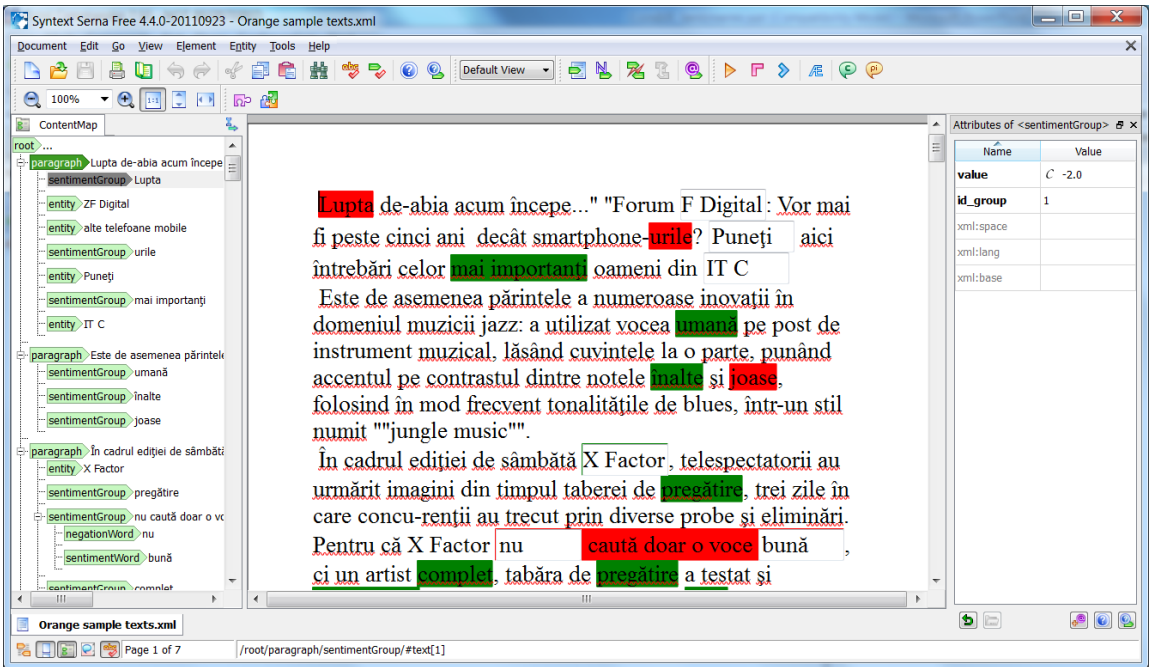

Figure 3: Serna usage example.

In the annotation example presented in Figure 3, the words highlighted with red and green represent positive and negative sentiment groups and in the uncolored rectangles named entities are emphasized. In the tree found in the left side of the image, we can see

[8] Serna: http://www.syntext.com/products/serna/

all the named entities and sentiment groups from the loaded text grouped by the paragraphs in which they are found.

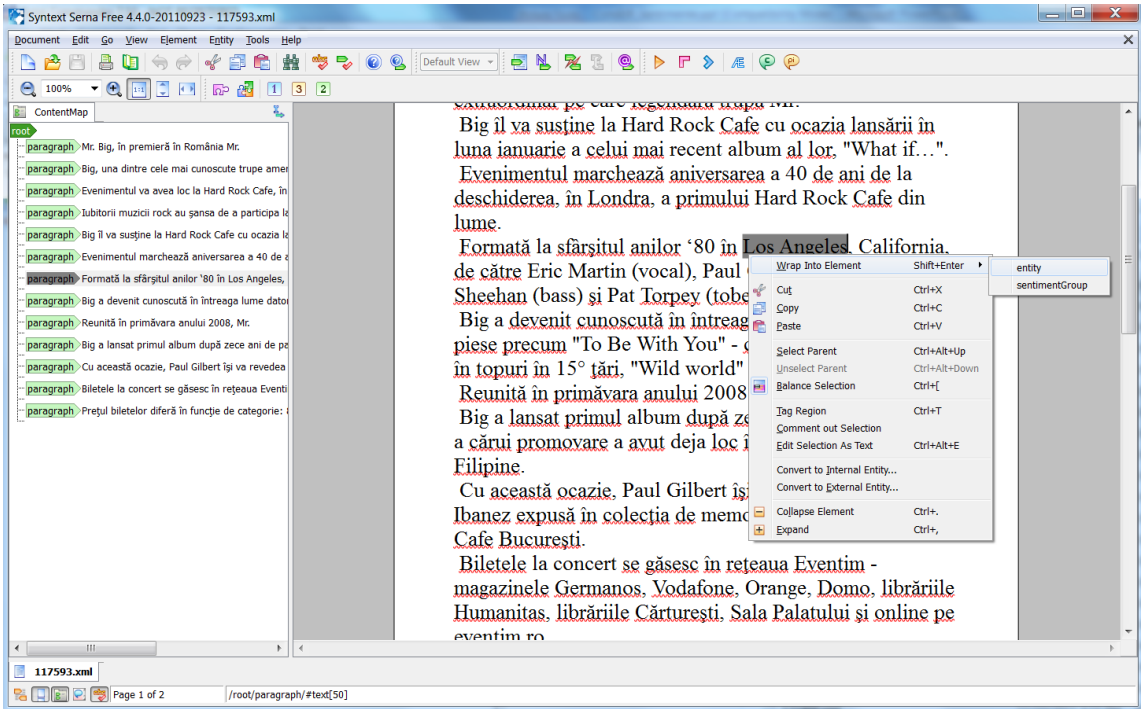

Figure 4: Serna plain text view.

We can also see in Figure 4 how the annotators are shown the text in its original form, although in the background, the text is in the XML format and contains annotations form the annotation tag set presented in Figure 1. In this manner, we use the benefits of Serna that, on one hand, provides the annotators a fast and simple working environment and, on the other, it assures the correctness of the underlying XML document.

3. Usage Scenarios

3.1. Evaluation

This corpus can be used to evaluate different type of applications, for example it can be used for named entity identification, named entity classification, sentiment text segment identification, and relation association between an entity and a sentiment bearing segment. In order to use our corpus in the evaluation process of a sentiment analysis system, we propose a couple of metrics that make use of the detailed annotation structure that we propose.

We define below a sentiment group identification precision that considers a correct instance only an exact match of the segment of text that was annotated as a sentiment group.

$$P_{sentimentGroup} = \frac{\#\ correct\ found\ sentimentGroups}{\#\ total\ found\ sentimentGroups}$$

Another type of precision is defined for associations between the sentiment groups and entities as follows:

$$P_{NEinfluence} = \frac{\#\ NEs\ with\ correctly\ associated\ sentimentGroups}{\#\ total\ NEs}$$

Taking into consideration that even with a high inter annotator agreement, the annotations still carry a mark of subjectivity. For this reason, we define a relaxed sentiment group value precision metric that accepts a small deviation of the sentiment's value, as it can be seen in the following formula:

$$RP_{groupValue} = \frac{\sum_{s_{SG} \in CG} partialMatch(s_{SG})}{|CG|},$$

where

$$partialMatch(s_{SG}) = \begin{cases} 1, |v_F(s_{SG}) - v_G(s_{SG})| < 1 \\ 0, otherwise \end{cases}$$

Where *CG* represents the set of sentiment groups correctly identified, $v_F(s_{SG})$ represents the value of the sentiment group computed by the system that goes under testing and $v_G(s_{SG})$ represent the value of the sentiment group given by the human annotator. As a remark, we mention that this measure is applied only over the correctly identified sentiment groups.

We also suggest the use of an average deviation metric for sentiment groups:

$$D_{groupValue} = \frac{\sum_{s_{SG} \in CG} |v_F(s_{SG}) - v_G(s_{SG})|}{|CG|},$$

Where CG, $v_F(s_{SG})$ and $v_G(s_{SG})$ are defined as above.

3.2. Training a classifier

The use of machine learning techniques for sentiment analysis has become a common practice in recent years. (Tan et al., 2009) and (Pang et al., 2002). The sentiment corpora that we propose is designed to be easily used as a training corpus for different classification tasks, such as entity orientated sentiment classification and named entity type classification. Furthermore, we exploit the in depth level of annotation and propose multiple classification scenarios. By transposing the values of the sentiment group from a continuous space to a set of two labels (positive and negative), we can easily define a binary classification problem for the general opinion expressed in a paragraph or, by applying the same mapping to sentiment values associated with named entities, we obtain a classification scenario for the opinion expressed towards and entity in a paragraph. By keeping real numbers instead of binary labels, we will have a regression problem. Another important aspect captured within the annotations of the corpus is represented by the links between sentiment groups and named entities, a fact that can be exploited by incorporating distance based features, besides lexical ones in the trained model.

4. Results

In this section, we present statistics over the annotated corpus that we obtained after a first series of annotation experiments.

After a work of approximately 2 weeks, 11 annotators have annotated 110 files representing news articles and blog posts, obtaining 1.988 paragraphs, 2.044 sentiment groups, 4.301 named entities and 1.101 links between the last two.

In the annotation process, the value of a sentiment group and a sentiment linked to a named entity can have a minimum of -4 (strong negative) and a maximum of +4 (strong positive, with 0 representing neutral. Below, in Figure 6 we present the distributions of the entity sentiment values and sentiment group values, disregarding the neutral values.

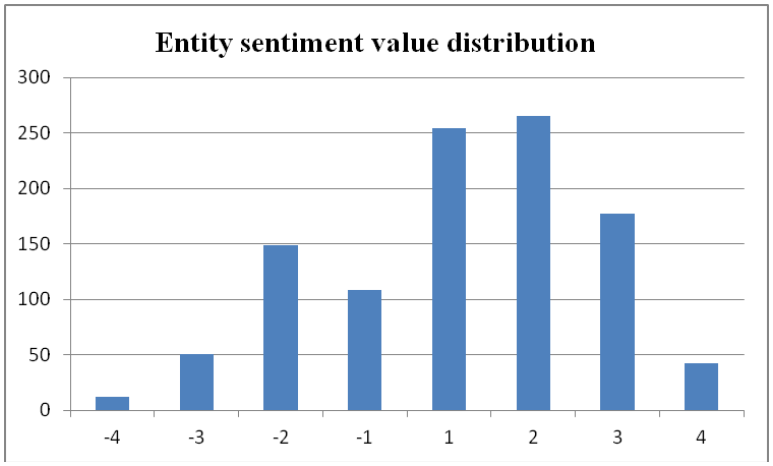

Figure 6: Entity sentiment value distribution.

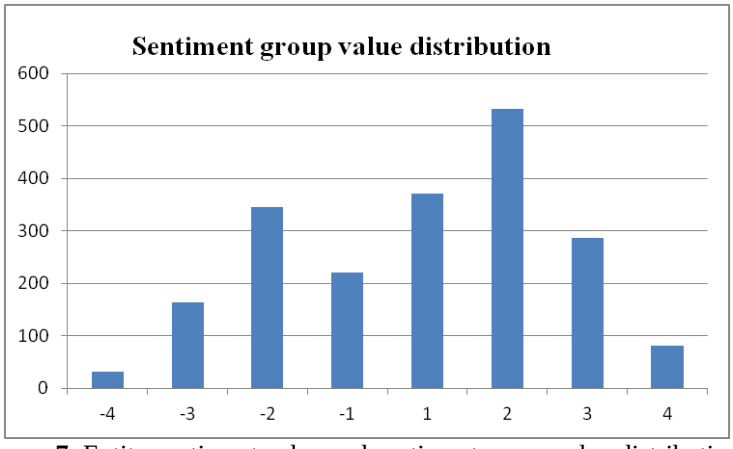

Figure 7: Entity sentiment value and sentiment group value distributions.

As it can be seen in Figure 7, most of the values for sentiment groups correspond to the "+2" value (over 500), "+1" and "-2" (over 300) and the least represented values are those of the two extremities "+4" and "-4" (under 100). With respect with the context in which the sentiment groups are found, the values of the sentiments are associated with the surround entities depending on the distance (viewed as a number of words) and the punctuation marks that separate them. It is also worth mentioning the strong correlation between the distribution of the values of the sentiment groups and those of the sentiments associated with entities.

5. Conclusion and future work

This paper presents a series of arguments for building a gold standard sentiment corpus for Romanian and the work we have done so far towards this goal. From our experiments, we discovered an efficient annotation process, we have observed what can be obtained in a short time span and with limited human resources and we have identified the main difficulties in building a sentiment corpus, such as detecting possible copyright issues concerning the texts that are used for annotation and the need for multiple annotations of the same text.

The main direction in which our work is heading is adapting the annotation process to allow multiple annotators to work on the same file. This is important in obtaining a more objective corpus by preserving only the documents with a high inner annotation agreement. Another approach we want to use is a guided annotation process by experimenting with different active learning techniques. Another aspect of our work will be guided towards obtaining a collection of texts that will not generate any copyrights infringements and that will be used for the open source release of the annotated corpus. For now we use inline annotations but we will investigate the stand-off approach.

Acknowledgements. This work was partly funded by the "Al. I. Cuza" University of Iasi and the Sector Operational Program for Human Resources Development through the project **—** Development of the innovation capacity and increasing of the research impact through post-doctoral programs POSDRU/89/1.5/S/49944. The authors of this paper thank the colleagues from Faculty of Computer Science Iasi, for the help offered in this project.

References

Baccianella, S., Esuli, A., Sebastiani, F. (2010). SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. *Proceedings of the 7th Conference on Language Resources and Evaluation (LREC 2010)*, Valletta, MT, 2010, pp. 2200-2204.

Das, A., Bandyopadhyay, S. (2010). SentiWordNet for Indian Languages. *Asian Federation for Natural Language Processing*. Beijing, China, 21-22 August 2010.

Denecke K. (2008) Using sentiwordnet for multilingual sentiment analysis. In *Proceedings of the IEEE International Conference on Data Engineering (ICDE2008)*, pp 507–512

Gînscă, A. L., Boroș, E., Iftene, A., Trandabăț, D., Toader, M., Corîci, M., Perez, C. A., Cristea, D. 2011. Sentimatrix - Multilingual Sentiment Analysis Service. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (ACL-WASSA2011)*. Portland, Oregon, USA, June 19-24, 2011

Iftene, A., Trandabăț, D., Toader, M., Corîci, M. (2011). Named Entity Recognition for Romanian. In *Proceedings of the 3th Conference on Knowledge Engineering:*

Principles and Techniques Conference (KEPT2011). In Studia Universitatis, Babes Bolyai, Volume 2, Pp. 19-24, Cluj-Napoca, Romania, July 4-6, 2011.

Kessler, J., Eckert, M., Clark, L., Nicolov, N. (2010). The ICWSM 2010 JDPA Sentiment Corpus for the Automotive Domain. In the *4th International AAAI Conference on Weblogs and Social Media Data Challenge Workshop (ICWSM-DCW 2010)*, 2010. Washington, D.C.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., Potts, C. (2011). Learning Word Vectors for Sentiment Analysis. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, June, Portland, Oregon, USA, Association for Computational Linguistics, pp. 142-150.

Noah, C., Davis, C., Potts, C., Schwarz, F. (2008). The Pragmatics of Expressive Content: Evidence from Large Corpora. To appear in *Sprache und Datenverarbeitung*.

Pang, B., Lee, L., Vaithyanathan, S. (2002) Thumbs up?: Sentiment Classification Using Machine Learning Techniques, *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing*, p.79-86, July 06, 2002

Potts, C., Schwarz, F. (2008). Exclamatives and heightened emotion: Extracting pragmatic generalizations from large corpora. Ms., UMass Amherst.

Tan, S., Cheng, X., Wang Y., Xu, H. (2009). Adapting Naive Bayes to Domain Adaptation for Sentiment Analysis, *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, April 06-09, 2009, Toulouse, France.

Wiebe, J., Wilson, T., Cardie, C. (2005). Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, volume 39, issue 2-3, pp. 165-210.

Wilson, T. (2008). Fine-Grained Subjectivity Analysis. *PhD Dissertation, Intelligent Systems Program*, University of Pittsburgh.

CHAPTER 3
APPLICATIONS
IN LANGUAGE PROCESSING

PUBLIC TEXT CATEGORIZATION

DANIELA GÎFU[1], DAN CRISTEA[1, 2]

[1] *"Alexandru Ioan Cuza" University, Faculty of Computer Science, Iași – România*
[2] *Institute for Theoretical Computer Science, Romanian Academy - Iași branch.*

{daniela.gifu, dcristea}@info.uaic.ro

Abstract

To analyze public discourse (media, politic, religious, etc.), means to analyze two dimensions: rational and emotional. This paper introduces an important natural language processing (NLP) problem, text categorization from the perspective of public language. Classification or categorization is the task of assigning words from a text corpus to two or more classes. The goal in text categorization is to classify the theme of a text, but, also, the dominant tonalities in a discourse. Our sets of semantic classes (33 for this version) are the extracts from many dailies, which we monitored in time (especially, in different crisis contexts). Here we present a computational tool, *Discourse Analysis Tool* (DAT), based on natural language processing (NLP) techniques for the interpretation of the public discourse. The idea behind it is that the vocabulary betrays the speaker's orientation (emotional or rational). Practically, the receptor identifies with the transmitter (journalist, politician, priest and so on), who becomes the legitimate voice of common ideals. When the object of study is the public discourse in print media, an investigation on these dimensions could put in evidence features influencing the auditory. Our purpose was to develop a computational platform able to offer to researchers in the humanities or social sciences, to the public at large (interested to consolidate their options before any public confrontation), and, why not, even to public speakers themselves, the possibility to measure different parameters of a written public discourse.

1. Introduction

Public discourse can be characterized from a rhetorical perspective, depending on its specific strategies: orientation to change opinions or to determine action, ratio between rational (*logos*) and emotional (*pathos*), etc. The main directions of research of public language are content analysis, with quantitative investigations of vocabulary (key words, frequent words) and rhetorical-pragmatic analysis of discursive strategies (presence of the person I, preference for vague statements, generics, etc.). In USA, the tradition of quantitative analysis is rather strong, starting from Lasswell (Lasswell, 1936); in Europe the interest grew more for discursive-rhetoric analysis. The situation, already described by Desideri (1984a: 11-13), hasn't changed very much in the meantime. On the other hand, the American analyses are often neutral, technical, comparative, while the European analysis (especially the model CDA[1]) has a critic component and a strong enough ethicist.

[1] "Critical theories, thus also CDA, are afforded special standing as guides for human action. They are aimed at producing «enlightenment and emmancipation». Such theories seek not only to describe and explain, but also to root

The current approaches in analyzing the public language are based on Natural Language Processing (NLP) techniques designed to investigate syntactic, lexical-semantic and pragmatic aspects of the discourse. The domain of NLP includes a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications (Liddy, 2001). To be able to interpret correctly a public phenomenon we must take into account the past events. Each public event "is an action that always tends to alter a pre-existing condition" (Perelman and Tyteca, 1970: 72). We can consider a public speech like an "aggressor", because it promotes and supports the programs and values of a group, capable to answer the auditor's expectations. Receptors demand from speakers the logicians' opened mind, the philosophers' deep meditation, the poets' metaphoric expression, the jurists' bright memory, tragedians' penetrated voice and, I'd say, a famous actor's gestures (Cicero, 1973: 51). The basic assumption in the public analysis is that any text isn't merely a string of signs placed randomly. Any group of signs is hierarchically organized, the signs can define various informational and interaction relations (Fox, 1987). Our analysis is meant to highlight the relevance and to understand different forms of communication, as captured by the print media in different contexts. Print media discourse may mean that the actual object (theme, context of a word, sentence) sometimes appears incoherent, incomplete, etc., as more general rules and principles, which does not mean it cannot be interpreted, at least in part, its purpose being to convince. In fact, the deviation in terms of rules of construction may be, on one hand, deliberate, so as to achieve specific rhetorical or aesthetic purposes, or, on the other hand, may be an expression of social and cognitive characteristics of those who use language such as memory limitations, the strategic aspects of speech production, etc.

In this paper we describe a platform (*Discourse Analysis Tool* – DAT) which integrates a range of language processing tools with the intent to build complex characterizations of the public discourse. A linguistic portrait of an author is drawn by putting together features extracted from the following linguistic layers: lexicon and morphology (richness of the vocabulary, rare co-occurrences, repetitions, use of synonyms, coverage of verbs' grammatical tenses, etc.) and semantic (semantic classes used).

The paper is structured as follows. Section 2 shortly describes the previous work. Section 3 discusses the lexical and semantic features having rhetorical values and section 4 presents the platform for multi-dimensional public discourse analysis. Next, section 5 discusses an example of comparative analysis of discourses very distant in time, elaborated during elections. Finally, Section 6 highlights interpretations anchored in our analysis and presents conclusions.

2. Previous work

The aim of an interdisciplinary approach such as analyzing the language of public speeches is to define and explain different discursive contexts (political, social,

out a particular kind of delusion. Even with differing concepts of ideology, critical theory seeks to create awareness in agents of their own needes and interests" (Wodak, 2006).

economic, etc.), in this case, reflected in the print media. The studies in this direction have mainly concentrated on three tasks: the first had to do with a cognitive side and, often, with an emotional side, of how humans acquire, produce, and understand language. The second aimed at understanding the relationship between the linguistic utterance and the world, and the third – at understanding the linguistic structure of the language as a communication device. Linguistics has usually treated language as an abstract object which can be accounted for without reference to social or political concerns of any kind (Romaine, 1994). Noam Chomsky (1968) and a whole range of scholars following him have given incentive ideas over topics that are placed on the immediate horizon today, their perspective on structural linguistics being at the origin of a whole range of theories in modern linguistics. From a different perspective, another reference model for communication theory was formulated by Habermas[2] (Stevenson, 1995). His thesis is that the public domain, in which we communicate, comes increasingly under the control of private business interests, either through direct and interactive forms, such as phone or Internet, or by means of mass communication, centrally controlled, such as audiovisual and print media.

As we will see, one aspect of the platform that we present touches a lexical-semantic functionality, which has some similarities with the approach used in *Linguistic Inquiry and Word Count* (LIWC), an American product used on the American elections in 2008. There are, however, important differences between the two platforms. LIWC-2007[3] is basically counting words and incrementing counters associated with their declared semantic classes. A previous version of DAT performs part-of-speech (POS) tagging and lemmatization of words. The lexicon contains a collection of lemmas (9500) having the POS categories: verb, noun, adjective and adverb. In the context of the lexical semantic analysis, the pronouns, numerals, prepositions and conjunctions, considered to be semantically empty, have been left out. Our current version includes 33 semantic classes, chosen to fit optimally with the necessities of interpreting the public discourse, five of them being added recently (`failures`, `nationalism`, `moderation`, `firmness`, `spectacular`). The second range of differences between the two platforms regards the user interface. In DAT, the user is served by a friendly interface, offering several services: opening one or more files, displaying the file/s, modifying/editing and saving the text, functions of undo/redo, functions to edit the lexicon, visualization of the mentioning of instances of certain semantic classes in the text, etc. Then, the menus offer a whole range of output visualization functions, from tabular form to graphical representations and to printing services. Finally, another important development for the semantic approach was the inclusion of a collection of formulas which can be used to make comparative studies between different authors. A special section of the lexicon includes expressions. An expression is defined as a sequence: <root-list> => <semlist>, in which <root-list> is a list of roots of words, therefore each optionally followed by the '*' sign. Gîfu and Cristea (2011) report similar approaches of human validation.

[2] Habermas's thesis is applied to the evolution of the British press, which said that industry trade press led to two types of journalism: quality journalism (for a small audience, educated and informed and with great power to attract publicity) and scandal journalism (for a group with low incomes and low power to attract advertisers).
[3] www.liwc.net.

3. Lexical and semantic features with rhetorical values

The use of language in public sphere has a "sanctifying" role (Edelman, 1964/1985) in the tentative to gain the trust of the auditor. The object of language could seem sometimes incoherent, unfinished, deprived of sense, etc., if confronted against general rules or principles of the language, but it can still be deciphered and function adequately. The deviation from the rules of language construction can be intended, in which case it is commanded by some rhetorical or aesthetic goals, expressing thus strategic aspects of the production of discourse, or can represent social or cognitive characteristics of the speakers, as memory limits, lacks in culture, etc. (van Dijk, 1972). The trajectory of rhetoric's (as a theory of discourse persuasion) has been intimately interlinked with the public discourse since Antiquity till our days. The only means to impose yourself in the public life is to convince by spreading your word. Today, the art of rhetorical discourse is understood only in correlation with performance, by combining in a highly elaborate way four ingredients: be rational, have ideas, master the language, and use an adequate style. It is extremely difficult to make an objective evaluation of this magic mélange of methods, but at least some parts of it can be measured. It is what we try to do in this research.

3.1. The context

The public discourse is, especially, a contextual discourse. Therefore, the analysis of public discourse, spoken or written, involves the analysis of the context in which it is transmitted. It becomes a context of speech, the whole reality that surrounds a sign, a verbal act or a discourse, as a "science" of speakers, physical presence and activity. We distinguish three context types (Coșeriu, 2004: 319-324):

1. the *idiomatic* context, created by the language itself, as a background of the speech. In other words, inside of the idiomatic context, each word meaning is defined in a smaller context, which is its field of meanings. Thus, a name of color, such as *portocaliu* (orange), has a meaning in relation to other color names of the same language (e.g. *roșu* (red), *albastru* (blue), etc.).

2. the *verbal* context is the speech itself. For each sign and discourse sequence, it becomes "the verbal context" not only what was said before, (Bally, 1950:43-44), but, also, what will be said in the same discourse. Thus, *Crin's bank account* and *the bank account in Switzerland* include contextual elements, highlighting the significance of the phrase *bank account*.

3. the *extra-verbal* context consists of all non-linguistic circumstances that are directly perceived by the speakers. We distinguish several subtypes: *physical* (things that are in the visual sight of the speakers or to which a sign adheres), *empirical* (objective things, which are known by those who speak in a certain place and moment, although they are not in the sight of the speakers), *natural* (totality of possible empirical contexts), *occasional* (occasional speech), *historical* (historical circumstances known to the speakers), *cultural* (cultural tradition of a community). Given their importance in establishing the semantic classes and, also, the correct interpretation of each entry in these classes, in our analyses we specify, particularly, this extra-verbal context,

especially the last three. Thus, the global economic crisis is an occasional extra-verbal context which gives a strong significance for the public discourse, from 2007 to present.

3.2. The lexical-semantic perspective

The speaker in a public space is determined to collect empathy and to convince the auditor. Yet, placing himself within the general limits of the public goals, very often a skilful speaker studies the public for fixing the type of vocabulary and the message to be delivered. He might exploit connections between more daring ideological categories (as is for instance the class nationalism) and those generally accepted (for instance, belonging to the classes social, achievements). The present day public language puts in value the virtues of the metaphor, its qualities to pass abruptly from complex to simple, from abstract to concrete, imposing a powerful subjective, i.e. emotional dimension to the discourse (the class emotional). Nonetheless, the public metaphor may lose the virtues of poetical metaphor, becoming vulgar (the class injuries).

But often, words have multiple senses. Among the number of senses words are registered within dictionaries we have retained only those considered relevant for the semantic classes selected. As such, each semantic class is mapped against a lexicon of word senses. Thus, the disambiguation task resides in using the context of a word occurrence for making a forced choice among the retained connotations. For sense disambiguation we have used the classical bag-of-words paradigm. The following preliminary steps have been followed to prepare the corpus against which word sense have been disambiguated:

1. A number of semantic classes have been retained, considered relevant for the type of discourses we have concentrated on: the public discourse (see section 4 for a list of these classes).

2. For each of these semantic classes, we have selected a number of words (actually lemmas), to each of them retaining the appropriate, intended, sense for the semantic class at hand.

3. The selected senses have been looked for in the electronic version of the biggest dictionary for Romanian language, eDTLR (Cristea et al., 2007). This dictionary includes for each sense of each word a great number of citations selected from writings of Romanian authors.

4. The citations attributed to the selected senses of the selected words have been copied from eDTLR and processed (by lemmatizing and eliminating the stop words) in order to build the "master" sense vectors to be used in further word sense comparisons.

The interpretation of word senses in our approach follows a perspective in which words of a document are having a narrow semantic spectrum. This means that all occurrences of the same word in the same text are supposed to have the same sense. As such, when a focus word *w* is to be decided its sense in the text, all words belonging to its occurrences (windows of a sentence size around the occurrences of *w*) are collected to assemble a test vector, which is compared against the master vectors of the recorded senses, by using a simplified-Lesk algorithm (Lesk, 1986), (Kilgarriff, Rosenzweig, 2000).

4. The DAT platform

The *Discourse Analysis Tool* (DAT, currently at version 3) considers the public discourse from two perspectives: lexical and semantic. We describe shortly our platform which integrates a range of language processing tools, with the intent to build complex characterizations of the public discourse. The concept behind this method is that the vocabulary used by a speaker opens a window towards the author's sensibility, his/her level of culture, her/his cognitive world, and, of course, the semantic spectrum of the speech, while the syntax may reveal the level of culture, intentional persuasive attitudes towards the public, etc. Some of these means of expression are intentional, aimed to deliver a certain image to the public, while others are unintentional. Figure 1 shows a snapshot of the interface showing a semantic analysis, during a working session. To display the results of the lexical-semantic analysis, the platform incorporates two alternative views: graphical (pie, function, columns and areas) and tabular (Microsoft Excel compatible).

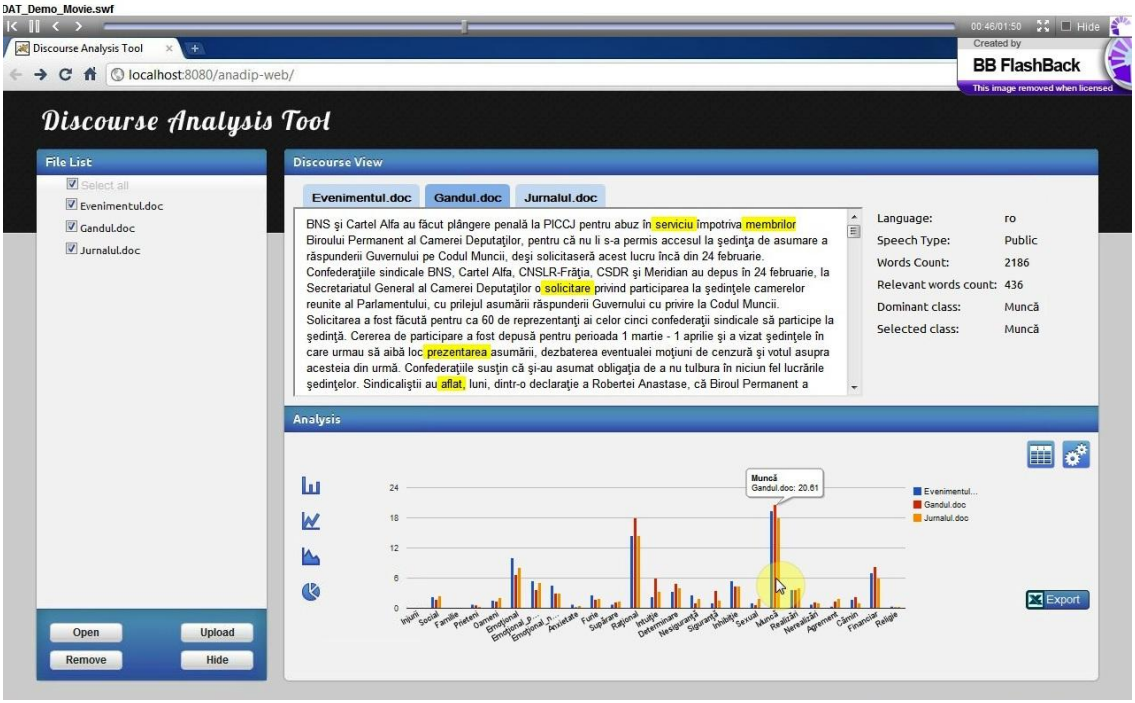

Figure 1: The DAT interface: in the left window appear the selected files, in the middle window – the text from the selected file, and in the right window, information about the text (language, word count, dominant class, etc.). Bellow, a plot chosen from a range of graphical styles is displayed. By selecting a specific class in the middle window, all words assigned to that class are highlighted in the text.

In DAT, the user has an easy-to-interact interface, offering a lot of services: opening of one or more files, displaying the file/s, modifying/editing and saving the text, functions of undo/redo, functions to edit the lexicon, visualizing the mentioning of instances of certain semantic classes in the text, etc. Then, the menus offer a whole range of output visualization functions, from tabular form to graphical representations and to printing services.

The vocabulary of the platform covers 33 semantic classes (`swear`, `social`, `family`, `friends`, `people`, `emotional`, `positive`, `negative`, `anxiety`, `anger`, `sadness`, `rational`, `intuition`, `determine`, `uncertain`, `certain`, `inhibition`, `perceptive`, `see`, `hear`, `feel`, `sexual`, `work`, `achievements`, `failures`, `leisure`, `home`, `financial`, `religion`, `nationalism`, `moderation`, `firmness`, `spectacular`), considered to fulfill optimally the necessity of interpreting the public discourse in different contexts. Some of these categories are placed in a hierarchical relation.

Linguistic processing begins by tokenization, part of speech tagging and lemmatization. Only the words belonging to the lexicon are considered relevant and therefore count in establishing the weights of different semantic classes. Since the lexicon maps senses of words to different semantic classes, depicting a semantic radiography of the text should follow a phase in which words are sense disambiguated. As mentioned already, our hypothesis is that in all the occurrences of a multi-sense word in a text the word displays the same sense. This hypothesis facilitates the disambiguation process, because all contexts of occurrence of a word participate in the disambiguation and that sense is selected which maximizes a bag-of-words-like analysis among all recorded possible choices. In response to the text being sent by the user, the system returns a compendium of data which includes: the language of the document, the number of words, and the type of discourse detected, a unique identifier (usually the file name), and a report of the lexical-semantic analysis.

Our interest went mainly in determining those discursive attitudes able to influence the audience decision. But the system can be parameterized to fit also other conjunctures: the user can define at will her/his semantic classes and the associated lexical, which, as indicated, are partially placed in a hierarchy. As an example, for the lemma *jurnalist* (journalist), the following classes are assigned: 2 = `social` and 5 = `people`. The class `people`, is a subclass of the class `social`. Whenever an occurrence belonging to a lower level class is detected in the input file, all counters in the hierarchy, from that class to the root, are incremented. In other words, the lexicon assigned to superior classes includes all words/lemmas of its subclasses.

5. A comparative study

Given that political discourse is a type of public discourse, we propose below a comparative analysis starting from political texts of two liberal leaders that we have found in print media.

5.1. The corpus

The corpus used for our investigation was configured to allow a comparative study over the discursive characteristics of two political leaders, both embracing liberal convictions, although in quite distant periods. The first one, I. C. Brătianu, is known to have led the basis of the liberal ideology in Romania, one of the most complex personalities of the Romanian history. Patriotic values were very important in influencing the auditory in the 19th century. The main theme of the speech is integrated in the class `nationalism`. The second political actor was chosen based on similar

criteria: Crin Antonescu, a contemporary liberal political leader. Amid a permanent crisis (economic, political, moral, etc.), the Romanian political discourse contains many arguments for improving living standards. The main theme of the speech is integrated in the class `work`. We are, this way, putting on the balance two styles of political discourse that are distant in time by one century and a half, interval which witnessed many changes in the state (the union of the Romanian provinces, wars, economical crises, etc.). For the elaboration of preliminary conclusions over the two Romanian elections processes, conducted in December 1858 (Marinescu and Grecescu, 1938) and November 2009, we collected, stored and parsed manually and automatically, political texts published by four national publications having similar profiles[4]. This corpus includes a collection of 1548 political sentences/phrases (units), each containing one or more clauses.

5.2. The lexical-semantic analysis

We present below a chart with two streams of data, representing the political texts in electoral context between the two liberal leaders mentioned above. Our experience shows that an absolute difference value below the threshold of 0.5% should be considered as irrelevant and, therefore, ignored in the interpretation. Apart from simply computing frequencies, the system can also perform comparative studies. The assessments made are comprehensive over the selected classes because they represent averages on collections of texts, not just a single text. To exemplify, one type of graphics considered for the interpretation was the one-to-one difference, as given by Formula (1), included in the DAT Mathematical Functions Library:

$$Diff_{x,y}^{1-1} = average(x) - average(y) \quad \textbf{(1)}$$

where *x* and *y* are two streams; *average(x)* and *average(y)* are the average frequencies of *x* and *y* over the whole stream, and the difference is computed for each selected class. Since a difference can lead to both positive and negative values, these particular graphs should read as follows: values above the horizontal axis are those prevailing at the candidate Brătianu versus the candidate Antonescu, and those below the horizontal axis show the reverse prominence. A zero value indicates equality.

So, the graphical representation in Figure 2, in which the present day politician is compared against the outstanding politician of the past should be interpreted as follows: Ion C. Brătianu's was interested more on Romanian specific aspects (the `nationalism` class) uttered in an emotional tone (the `positive` class) than Crin Antonescu, whose discourse had an argumentative (the `rational` class) attitude.

[4] National newspapers of general informations, are presented as a tabloid with a circulation of tens of thousands of copies per edition: *Românul* (19th century), *Evenimentul zilei*, *Gândul* and *Ziua* (our days).

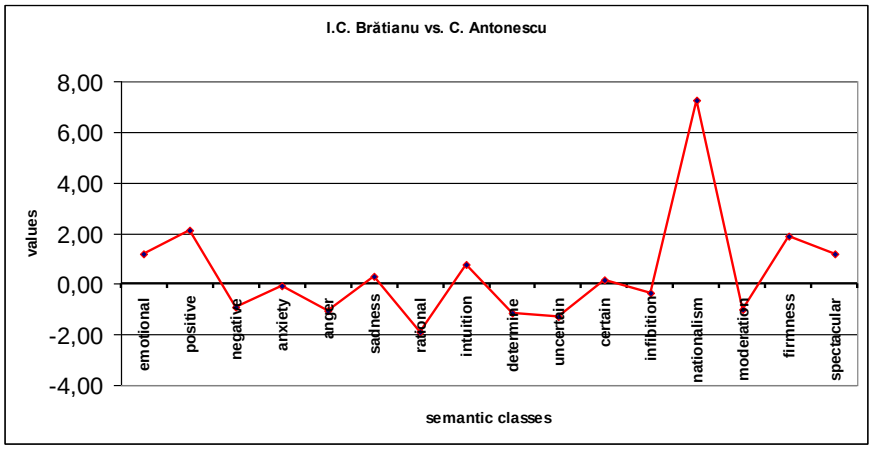

Figure 2: The average differences in the frequencies for each parent class (>0.5%) after processing political discourses, between Ion C. Brătianu and Crin Antonescu.

6. Conclusions

Surely, the problem of characterizing the public text receives no final solution with our approach. We believe, however, that our method sheds an interesting light and opens new perspectives. It is clear that some of the differences at the level of discourse which we have evidenced as differentiating the two political actors should be attributed only partially to idiosyncratic rhetorical styles, because they have also historical explanations. Moreover, speeches of many public actors, especially today, are the product of teams of specialists in communication and, as such, conclusions regarding their cultural universe, for instance, should be uttered with care. We believe that the platform helps to outline distinctive features which bring a new, and sometimes unexpected, vision upon the discursive characteristics of public speakers (politicians, columnists and so on).

In the future, new features will be added to the platform, with a special emphasis on the syntactic and rhetorical level analysis. The new release of DAT should help the user to identify and count relations between different parts of speech and to put in evidence patterns of use at the syntactic and rhetorical level.

The collection of manually annotated texts should also be augmented. Another line to be continued regards the evaluation metrics, which have not received enough attention till now. We are currently studying other statistical metrics able to give a more comprehensive image on different facets of the public discourse.

Acknowledgments. The DAT platform has been developed by Mădălina Spătaru at the Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași. In order to perform this research the first author received financial support from the POSDRU/89/1.5/S/63663 grant. The work of the second author is partially supported from the ICT-PSP project METANET4U.

References

Bally, Ch. (1950). Linguistique générale et linguistique française. Berna, 43-44.

Chomsky, N. (1968). Language and Mind, Harcourt Brace Jovanovich, Inc., chapter III.

Cicero, (1973). Opere alese. *Ed. Univers*, București, II, 51.

Coșeriu, E. (2004). Teoria limbajului și Lingvistică generală, *Ed. Enciclopedică*, București, 319-324.

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 SpeD 2007 *Speech Technology and Human - Computer Dialogue*, Iasi, May 10-12, 2007.

Desideri, P. (1984a). En marge du discours politique. *Degrés*, 12, 37, 1-9.

Edelman, M. (1985). The Symbolic Uses of Politics. *Urbana: University of Illinois Press*. Originally published in 1964.

Fox, B.A. (1987). Discourse structure and anaphora. *Written and conversational English*. Cambridge: Cambridge University Press.

Gîfu, D., Cristea, D. (2011). Computational Techniques in Political Language Processing: AnaDiP-2011. *In J.J. Park, L.T. Yang, and C. Lee (Eds.), FutureTech 2011*, Part II, CCIS 185, 188–195.

Kilgarriff, A., Rosenzweig, J. (2000). English SENSEVAL: Report and Results. In *Proceedings of the 2nd International Conference on Language Resourcesand Evaluation*, LREC, Athens, Greece.

Lasswell, H. D. (1936). Politics: Who Gets What, When, How., McGraw-Hill, New York.

Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *SIGDOC '86: Proceedings of the 5th annual international conference on Systems documentation*, New York, NY, USA. ACM, 24-26.

Liddy, E.D. (2001). Natural Language Processing in *Encyclopedia of Library and Information Science*, 2nd Ed. NY. Marcel Decker, Inc.

Marinescu, G., Grecescu, C. (ed.) (1938). Ion C. Brătianu. Acte și cuvântări, vol. I – part I (june 1848 = decembrie 1859). *Cartea Românească*, Bucharest, 228-237.

Perelman, C., Olbrechts-Tyteca, L. (1972). Traité de l'argumentation. *Éd. de l'Institut de Sociologie de l'Université Libre de Bruxelles*, 72.

Romaine, S. (1994). Language in society. An Introduction to Sociolinguistics. *Oxford University Press Inc.*, New York.

Stevenson, N. (1995). Understanding Media Cultures: Social Theory and Mass Communication, Londra: Sage.

van Dijk, T.A. (1972). Textual Structures of News in the Press. *Working notes*, University of Amsterdam, Department of General Literary Studies, Section of Discourse Studies, 14.

Wodak, R. (2006). Critical Linguistics and Critical Discourse Analysis. *Handbook of Pragmatics*, Benjamins.

INFERRING DIACHRONIC MORPHOLOGY USING THE ROMANIAN THESAURUS DICTIONARY

RADU SIMIONESCU[1], DAN CRISTEA[1], GABRIELA HAJA[2, 3]

[1]*Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași*

[2]*"Alexandru Philippide" Institute of Philology*

[3]*Romanian Academy, the Iași branch*

{radu.simionescu ,dcristea}@info.uaic.ro; gabihaja@gmail.com

Abstract

This paper presents a first step towards constructing the diachronic Romanian morphology. First, the "deformation" of a word is introduced and a classification of such deformations is proposed. The conducted research aims at detecting deformations in the roots of inflectional words (nouns, adjectives and verbs). The algorithm we present uses two important resources: a morphological dictionary of the current Romanian language, which also models the inflectional paradigms of the language, and eDTLR – the digital version of the Romanian Thesaurus Dictionary. In eDTLR each title word has associated a set of citations extracted from the Romanian literature, each having attached the year of publication. The algorithm detects root deformations in words by comparing word forms of the current language with forms extracted from the eDTLR citations. For every root change, the deformed root is deducted and all the diachronic forms are inferred. Also, using the chronology of citations, for each diachronic root a period or a year is established. The research was conducted on 4 volumes of eDTLR. The algorithm successfully detected 2,700 root deformations and inferred a total of 30,000 diachronic inflexions.

1. Morphological sources for Romanian language

eDTLR[1] (Cristea et al., 2007) is the digital version of *Romanian Language Dictionary* (DLR), edited by the Romanian Academy, between 1906 and 2010, and including its sources in digital form and the software to access them. The Dictionary describes, following lexicographical norms, all words registered in written Romanian texts, starting with *Scrisoarea lui Neacșu* (*The Neacșu's letter*), 1521, the first known text in Romanian, until today. It includes the word's etymology and quotations extracted from a large collection of texts, attributed to all social and cultural domains (2,500 titles and approx. 3,000 volumes).

The morphological variation in the evolution of Romanian is mirrored in the rich collection of citations that eDTLR includes (more than 1.3 million). Richly sensed entries could display tenths of pages in the original paper dictionary (100 for a verb like *a veni*/*to come*). Moreover, the citations cover all historical periods in the evolution of

[1] Built between 2007-2010, in a project financed by the Romanian Government and coordinated by UAIC-FII (https://consilr.info.uaic.ro/edtlr/wiki/index.php?title=Digitalizing_the_Thesaurus_Dictionary_of_the_Romanian_Language)

written and spoken Romanian language (Rosetti, et al., 1968; Gheție, 1977; Gheție and Chivu, 2000), which makes them extremely valuable as a source of data in the attempt to reconstruct a diachronic morphology. Each citation includes exactly one occurrence of the title word. Moreover, citations are paired with codes identifying uniquely the source document and the pages from where it has been extracted. An external database, called *the chronology*, has been compiled, as code-year or code-interval pairs, where the year/interval are publishing dates of the source. As such, a certain morphological form of the title word can be precisely located in time.

AnaMorph (Cristea, Forăscu, 2006) is a paradigmatic word flexing instrument for Romanian. It sees a word as a lexical unit made up of two morphemes, a stem (root) and an ending. In its morphological variations, a word can have more stems, as given by the irregularities in declination or conjugation. There are mainly two causes of these irregularities: inheritance of old forms and phonetic alternations. The number of stems, the complete set of endings and the association of different stems with endings in flexing, assembles a paradigm. Usually, a paradigm is shared by a class of words having the same part of speech. In AnaMorph, parts of the paradigms have been defined manually, following a grammar of modern Romanian. The rest of them were generated automatically using a morphologic dictionary provided by DEX (in its online version[2]). The total number of paradigms is now 366, which include 150 sets of endings, completely covering the morphology of contemporary Romanian.

2. Going back in time

If we compare the language spoken or written today with that of the first quarter of the previous century we get fewer differences than between the today Romanian and that of the middle 19th century. The more we go back in the past, the bigger the differences are. But this can be taken also in the sense that we expect to find more common word forms between today's language and the one spoken 75 years ago than between today Romanian and the language spoken 160 years ago. Even more, changes are not abrupt, affecting the whole vocabulary at once, but merely involve the class of words belonging to the same paradigms and sometimes only isolated words. Mainly, at one moment in time or over a certain interval, one paradigm gradually changed. Very rarely, abrupt changes may also occur, in which case they are issued by rules that Academia imposed and which were gradually adopted by the society[3].

The approach presented in this paper consists in analyzing the set of examples contained in eDTLR to infer old forms and associate them with certain periods of time. Then, to use these periods for evidencing phenomena related to the evolution of the Romanian language. Even more, by inferring old forms, a diachronic morphologic dictionary can be built, which could, for instance, be used for POS-tagging old Romanian texts.

Since citations are paired with years, this task seems straightforward. Still, two things complicate the problem: the difficulty to recognize the morphological features of the

[2] www.dexonline.ro

[3] Romania being rather a conservative and stubborn society, sometimes the rules imposed by the Romanian Academy, the only forum that has the right to impose changes in the official orthography, are not completely observed. For instance, the 1993, new orthography regulations have divided the society in two currents: those accepting to use *â* in the inner position and those insisting to keep the old written form *î* (among other details).

occurrence of the title word in a citation, and the fact that more forms could have been in use in the same period.

We have detected four ways in which a word can change its form over time:

- the word suffered changes in one or more of its roots;
- the word migrated to another paradigm;
- the word is a noun and changed its grammatical gender;
- the word suffered a combination of the above deformations.

In this paper we will deal strictly with detecting and inferring the forms which suffered a root change. For our study, we have taken into consideration only the forms which are not present in the morphologic dictionary of the current Romanian language and can be obtained in conjugation or declination from a known lemma (for which a paradigm is known). In the present study we have considered only nouns, adjectives and verbs (the three categories with the richest morphology) in Romanian.

3. The algorithm

In the following, we refer to a word as being "known" if it is present in the morphological dictionary of the current Romanian language. The occurrence of the title word in the citations is detected imposing a one-occurrence-of-title-word-per-citation restriction, and making use of a variation of the Levenshtein distance (Levenshtein, 1966).

Given a known title word (a lemma) *l*, framed under the modern paradigm *p*, and an unknown inflexion form *f* (which is not found in the list of *l*'s inflexion forms), determine if *f* can be framed under *p* and, if so, infer a root and its inflexion forms. Solving such a problem is required when, given a title word and an old inflexion form extracted from one of its citations, we want to establish if this old form is a root change – it is part of the same paradigm as the title word but something differs in the root. We define *s(p)* as the list of suffixes indicated by the paradigm *p*.

To determine if *f* can be framed under the paradigm *p*, for every suffix *s(p)[k]* that matches *f* at the end we assume that *f* might be constructed from a deformed root plus the suffix *s(p)[k]*. By trimming each such matching suffix from the form, we create a set of candidate roots *R(f,p).*

Next, the validation follows. For each candidate root *R(f,p)[i]* generate a list of fictive inflexion forms *F(R(f,p)[i], p)* by attaching the suffixes imposed by *p* to the candidate root *R(f,p)[i]*. Define a score *for R(f,p)[i] as* the number of fictive inflexion forms in *F(R(f,p)[i], p)* which are present in any of the eDTLR citations or in the section dedicated to morphological specifications. If none of the candidates have a score higher then 0, then conclude that *f* cannot be framed under paradigm *p*. Otherwise, conclude for the root having the best score, *R(f,p)[j]*, as being an old deformed root. The forms F*(R(f,p)[j], p)* can now be inferred and morphologically classified due to the data model of the paradigms, which associate a part of speech for each suffix that they contain.

Since the chronology of the citation can be mapped to all words belonging to it, the inferred forms after applying the root changing algorithm, once detected in some citation, become automatically attributed to the time/period of the citation. The examples below will put in evidence other details of the algorithm.

3.1. The verb "a dansa" (to dance)

The paradigm the title word *dansa* is part of accepts the following suffixes: {*a*, *am*, *ai*, *ați*, *au*, *asem*, *aseși*, *ase*, *aserăm*, *aserăți*, *aseră*, *ează*, *ez*, *ezi*, *ează*, *ăm*, *ați*, *eze*, *ași*, *ă*, *arăm*, *arăți*, *ară*, *ând*, *ându*, *at*, *ată*, *ați*, *ate*}. For example *dansa* represents the concatenation of the root *dans* and the suffix *a*, which is associated with infinitive (as well as the homonymous form in past simple third person singular, for this particular paradigm).

The word *dănțată* (past participle) has been found in a citation under the *dansa* title word. In this case, two suffixes match: *ă* and *ată*, so there are two candidate roots: *dănțat* and *dănț*. The validation of the candidates gets on like this:

- the *dănțat* root generates the forms: *dănțata*, *dănțatam*, *dănțat*, *dănțatai*, *dănțataserăm*, *dănțatezi* etc. None of these, all different from the initial unknown form *dănțată*, can be found anywhere in eDTLR – so the score is 0;
- the *dănț* root generates the forms: *dănța*, *dănțam*, *dănțau*, *dănțând*, *dănțaserăm* etc. Leaving out the found unknown form, two of the generated forms are found in eDTLR (*dănțat* and *dănțând*) – so the score is 2.

Since the root *danț* is at the origin of a score higher than 0, it is considered a deformed root and inserted in the diachronic morphologic dictionary under the same paradigm as *dansa*, so all its inflexion forms can be generated. The publication years of the citations from which *dănțată*, *dănțat* and *dănțând* were found provide enough information so that their root can be associated with a period of time.

3.2. The adjective "dator" (indebted)

The previous example has a particularity, in that the verb *dansa* displays only one root for its inflexion. This example illustrates an adjectival paradigm which accepts two roots, each associated with its own suffixes. For the title word *dator* (adjective), the form *deatori* (masculine plural indefinite) has been found in one of the citations. The paradigm of *dator* accepts the following suffixes, grouped by the two different roots: {*VOID*, *ul*, *ului*, *i*, *ii*, *ilor*} {*e*, *ea*, *ei*, *ele*, *elor*}. By the *VOID* suffix we indicate the empty string, which means that if the root is *dator* then it is also a inflexion form.

There are two suffixes which match the form *deatori*: the *VOID* suffix (always matches) and the *i* suffix. So, the two candidate roots are: *deatori* (by trimming the *VOID* suffix) and *deator* (by trimming the *i* suffix). This time, the validation of the candidates is done somehow differently, compared to the previous example. For each root, the forms which are looked up in the dictionary are formed by adding only suffixes belonging to the same group (list of suffixes) as the one to which the matching suffix belonged.

For instance, *deatori-elor* won't be searched for when validating (won't be considered a fictive form). The candidate root *deatori* was found by subtracting the *VOID* suffix. The

fictive forms of *deatori* are generated by attaching only the suffixes belonging to the same group as VOID, which doesn't contain *-elor*.

- *deatori* generates the forms: *deatori*, *deatoriul*, *deatoriului*, *deatorii*, *deatoriii*, *deatoriilor*. Out of these, one form is found in the dictionary: *deatorii*.
- on the other hand, *deator* generates: *deator*, *deatorul*, *deatorului*, *deatori*, *deatorii*, *deatorilor*. This time two different forms are found: *deatorii* and *deator*.

In this case, both candidates produced scores greater than 0, still the one with the best score is considered as the actual root of the old version of this word, and that is *deator* – which is true actually. But what would have happened if the form *deator* wouldn't have been found in the dictionary? This would end in a tie, and in such a case the shorter root chosen. This heuristic, generally, seems to guess the correct forms.

4. Results

The morphologic dictionary of the current Romanian language, which is used for determining if a word is "known" or not, contains a total of 1.15 million forms, corresponding to approx. 145,000 distinct lemmas. The algorithm described above was applied for 41,911 entries (the letters D, P, S, V, of a total of approximately 175,000, as the whole dictionary contains), for which the dictionary includes 205,654 citations. We have found a total of 14,782 unknown inflexion forms which have a known lemma. Out of them we inferred a total of 22,697 new inflexion forms, by using 7,295 forms that were found in the entries as pilot forms (in citations or in the morphological specifications paragraphs). In total, we have classified morphologically 29,870 old, unknown words. The total number of new roots inferred was 2,705 for 1,938 known lemmas.

Since the second example mentioned also a number of heuristics, it means that in rare cases the algorithm can fail. When a root is inferred incorrectly, it triggers a set of incorrect old, deformed inflexion forms. In order to report statistical values about its accuracy, a manual evaluation was performed on all the inferred roots (for nouns and adjectives only). The correctors were 20 master students in Computational Linguistics[4]. Each student received a packet which contained random entries, where an entry is a word with one of its roots being automatically inferred. The other roots are unknown. The correctors' job was to identify the roots which were inferred erroneously and also to type in the unknown roots.

Each entry was randomly distributed to 2 correctors. After the first phase of the correction, the contradictions between packets have been revealed to the students. In the next phase they discussed and negotiated their choices in order to decrease the number of contradictions. In the end, still some contradictions remained. By counting the entries which were, in the end, considered correct by both students, we got a total of 2,064 correctly inferred roots, out of the 2,120 total entries. This represents a percentage of 97.358% for the case of nouns. We chose to leave out the verbs from this correction

[4] At the Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași

project because we considered that manually filling in the unknown roots of old forms of verbs is going to be too difficult and time consuming for the time we had at our disposal. The total number of new roots manually typed in by the students, which didn't conflict among correction packets was 550. The number of roots which did conflict was 181. The small number of roots inserted manually is explained by the fact that only a third (36%) of the nouns and adjectives contained more than one root.

The big ratio of conflicts (32%) is explained by the fact that we were very much constrained by time in the second part, when the negotiations between correctors had to happen. Even more, almost half of the students didn't manage to contribute to this second part at all. The experiment proved that guessing a root of an old word is a tricky process and requires an extensive knowledge about the history of the language, as in the corpus of citations given for correction/completion there are words which have been in use some 400 years ago.

5. Conclusions

Determining the forms the words had over time, anchored in transformations of roots and the paradigmatic morphology, is the first step in inferring the general rules of the evolution of Romanian language. Out of this study, we aim to reconstruct the general trends that governed the evolution of Romanian language.

The next step is to investigate also other cases of variation of word paradigms, mentioned in the first section. After precisely defining the paradigms associated with each title word and the interval of time each paradigm has been in use, we intend to build chronological records of each title word, by arranging their paradigms on the time axis. Then we will correlate these chronological records in search for patterns of variation, with the intent to infer the rules of language evolution. Various resources will be built in the process, which could be used for creating fascinating tools, like a diachronic part of speech tagger, or a tool which would automatically predict the interval in which a text has been written.

Acknowledgements. The research reported in this paper was partially supported by the PSP-ICT project METANET4U.

References

Cristea, D., Forăscu, C. (2006). Linguistic Resources and Technologies for Romanian Language. *In Journal of Computer Science of Moldova, Academy of Science of Moldova, Institute of Mathematics and Computer Science*, 14: 1(40), ISSN 1561-4042, 34-73.

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 SpeD 2007 Speech Technology and Human - Computer Dialogue*, Iasi, May 10-12, 2007.

Gheție, I. (1977). (coord.) Istoria limbii române literare. Epoca veche (1532-1780). *București, Editura Academiei Române*.

Gheție, I., Chivu, G. (2000). (coord.) Contribuții la istoria limbii române literare. Secolul al XVIII-lea (1688-1780), București, *Editura Academiei Române*.

Levenshtein V.I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady* 10: 707–10.

Rosetti, A. et al., (1968 – 1973). Istoria literaturii române. *Bucureşti: Editura Academiei Republicii Socialiste România*.

ROMANIAN PROCESSING CHAINS IN METANET4U

PISTOL IONUȚ CRISTIAN

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

ipistol@info.uaic.ro

Abstract

This paper's main contribution is to describe the completed and planned development of processing resources at UAIC[1] as part of the work done for the METANET4U[2] research project. Significant for the project is the development of processing resources as UIMA[3] components integrated in the U-Compare[4] system, which offers significant advantages in terms of workflow development and evaluation.

1. Introduction

The main goal of the METANET4U project is to collect language resources for seven European languages and distribute them using a platform called METASHARE. The contributed resources can have many forms, from annotated corpora to complex processing systems and from open-source tools to pay-per-use web services.

Complex NLP applications such as information extraction systems comprise several separate tools such as tokenizers, part-of-speech taggers, named entity recognizers, etc. Whilst these tools may be developed for the purposes of a particular application, it is desirable if they can be re-used in other applications. This is because the same basic processing steps are often common for a number of different NLP applications. As part of METANET4U, work has been carried out on the WPS Work package, whose main goal is to show if and how processing tools originating in various sources and usable for various languages can be combined to build complex processing workflows.

Supporting this goal, U-Compare (Kano et. al, 2011) is a workflow management system based on UIMA (Ferrucci and Lally, 2004), a well know NLP meta-system allowing users to contribute processing tools and use them together with other integrated resources to perform various processing tasks.

As part of UAIC's contribution to METANET4U (and the WPS Work package), we selected 18 processing tools developed at UAIC to be contributed to METASHARE (14 of which will be integrated in UIMA and U-Compare). This paper describes part of this work, next section making a short overview of UIMA and U-Compare, as well as the effort required to integrate a new tool. UAIC tools and the current state of the integration is described in section three. Integration issues and future considerations are discussed in the last section of this paper.

[1] http://www.uaic.ro/uaic/bin/view/Main/?language=en
[2] http://metanet4u.eu/
[3] http://uima.apache.org/
[4] http://u-compare.org/

2. UIMA/U-Compare integration

UIMA (Unstructured Information Management Architecture) is the result of an IBM development project (completed in 2002) aiming to develop an "industrial-strength, scalable and extensible platform for creating, integrating and deploying unstructured information management solutions from combinations of semantic analysis and search components." (Ferrucci and Lally, 2004). It was designed for maximum performance and scalability, intended to serve as a linguistic annotation black-box used to add whatever linguistic information was available to any type of data. By "unstructured information" IBM meant all types of available electronic resources, as a whole, without a common structure. UIMA's goal was to process all this data and add linguistic information and structure to it, thus significantly improving classification, advanced search and data transfers.

U-Compare (Kano et al., 2011) has been developed by the University of Tokyo, the National Centre for Text Mining (NaCTeM) at the University of Manchester and the University of Colorado, with the goals to support construction of NLP applications from reusable resources and to allow easy evaluation of applications against gold-standard annotated data. U-Compare is based UIMA and inherits UIMA's description of annotations as Types (basically each annotation is an instance of a particular UIMAType class, offering access to read existing elements and writ new ones observing a specified Type specification). This has the benefit of guaranteeing interoperability between components using the same Types as input/output, but has the significant drawback of requiring users to adapt their tools to access annotated data not directly from an external resource but internally, using access methods available for that particular Type.

Before METANET4U, U-Compare included a set of over 30 integrated processing resources, most of them available for English. For those tools, U-Compare offered means to combine them in various workflows using a graphical interface, which serves as a repository of available resources and allows users to check whether the components added in sequence to a workflow actually match input/output formats (indicated as specific U-Compare Types part of the U-Compare Type System.

Since one of the main developers of U-Compare (University of Manchester) is also part of METANET4U and the leader of WPS, U-Compare has been adapted to the conditions and issues raised so far during the project, particularly in terms of handling multilingual components and workflows created.

3. UAIC WPS current status

The first stage involving UAIC required us to select tools we can contribute to METASHARE. We selected 18, all developed at UAIC (and all available for free, either as open source or web service). Of them, 14 were selected for integration in UIMA/U-Compare. We kept the tools relevant in multilingual contexts, performing tasks relevant for other languages if the required resources are provided. Table 1 below shows the 15 UAIC tools to be integrated, together with the selected U-Compare Type System input and output format.

Table 1: UAIC tools in WPS

Tool name	Input	Output	Observations
Splitter	org.u_compare.shared.document.Text	org.u_compare.shared.document.Segment (new type added by UAIC)	Splits to discourse units
Tokenizer	org.u_compare.shared.document.Text	org.u_compare.shared.syntactic.Token	
Lemmatizer	org.u_compare.shared.syntactic.POSToken (or Text)	org.u_compare.shared.syntactic.RichToken	Two versions with different input type
FDG-parser	org.u_compare.shared.syntactic.POSToken	org.u_compare.shared.syntactic.Dependency	
NP-chunker	org.u_compare.shared.syntactic.POSToken	org.u_compare.shared.syntactic.Chunk	
RARE	org.u_compare.shared.syntactic.Chunk	org.u_compare.shared.semantic.CoreferenceAnnotation	Performs anaphora resolution
Discourse Parser	org.u_compare.shared.semantic.CoreferenceAnnotation	org.u_compare.shared.semantic.DiscourseTree (new type added by UAIC)	
SRL	org.u_compare.shared.syntactic.RichToken	.u_compare.shared.semantic.SemanticClassAnnotation	Performs semantic role labeling
Summarizer	org.u_compare.shared.document.Text	org.u_compare.shared.document.Text	Output is a different UIMA view of the same document
OntologyBuilder	org.u_compare.shared.syntactic.RichToken	org.u_compare.shared.syntactic.OWL (new type added by UAIC)	Builds an ontology from keywords and definitions
QA	org.u_compare.shared.document.Text	org.u_compare.shared.document.Text	Output is the answer to the input questions
TE	org.u_compare.shared.document.Text	org.u_compare.shared.document.Text	Checks entailment between two input fragments
OccurenceFinder	Any	Keeps original format	Finds occurrences of an annotation pattern
Categorizer	org.u_compare.shared.document.Text	org.u_compare.shared.document.Category (new type added by UAIC)	Labels text with general semantic categories

Using the above tools as well as those contributed by other project members, a set of 26 multilingual workflows were designed to be implemented by July 2012 (Branco et al., 2011). 22 of the 26 workflows involve components developed by UAIC. An example of such a workflow can be seen in figure 1.

Of the tools in Table 1, the first 3 are already integrated in UIMA/U-Compare and were use to build and test 4 workflows (two of them using also components developed by RACAI, the other METANET4U partner from Romania). An example of one of these workflows, as it appears in U-Compare's interface, can be seen in figure 2. This particular workflow uses plain text as input and produces tokenized, POS-tagged and lemmatized output.

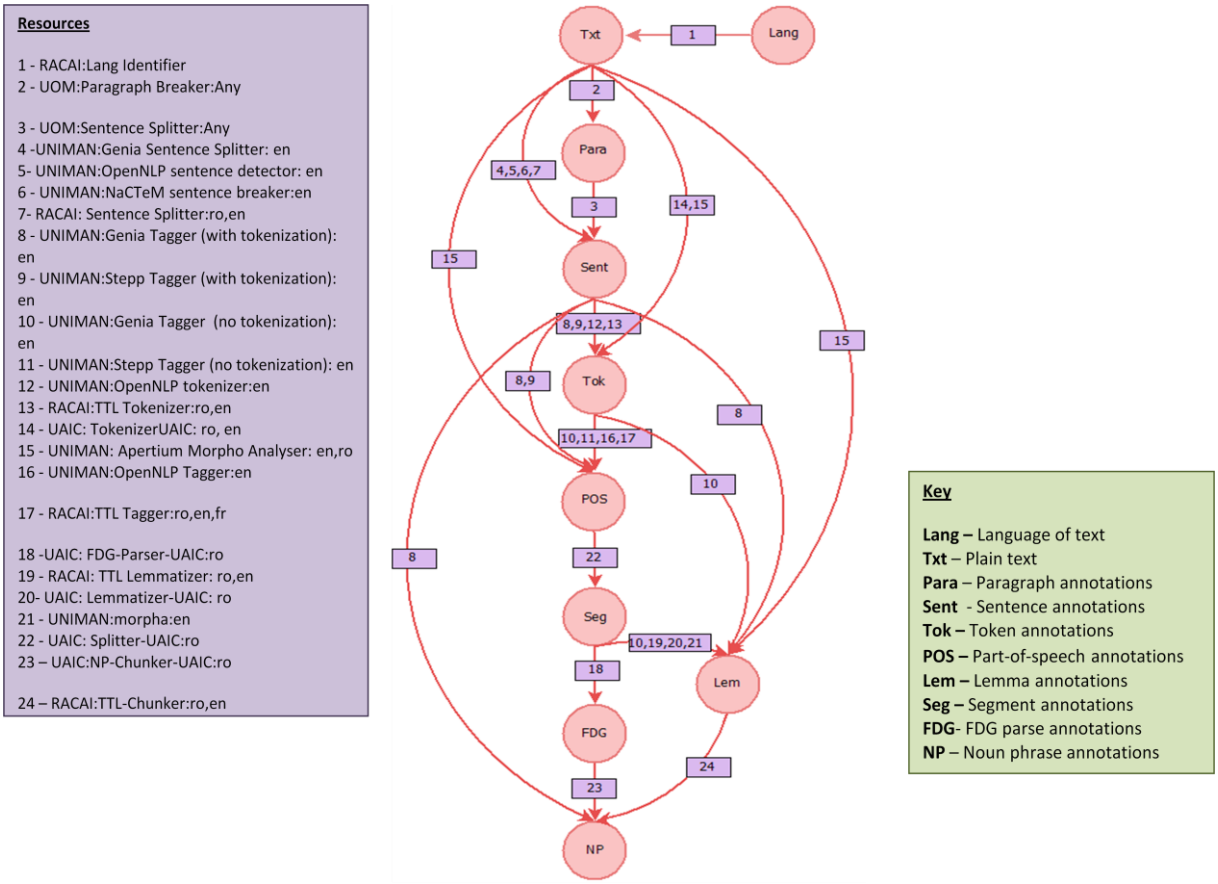

Figure 1: An example of a multilingual WPS workflow (adapted from (Ananiadou et. Al. 2011))

The integration faced some difficulties, requiring adaptation of both U-Compare and individual tools to satisfy requirements apparent only after the integration begun, such as issues with exporting components and workflows within U-Compare and accessing external resources (dictionaries, language models) required by some components. These difficulties were largely overcome, and the integration of the next set of UAIC components is under way.

A significant change required for most UAIC or other partner's tools is to adapt to a common standard way of reading and writing annotations, which usually involves changing the current implementations. This is usually manageable for endogenous tools, where some of the original developers are available to make changes.

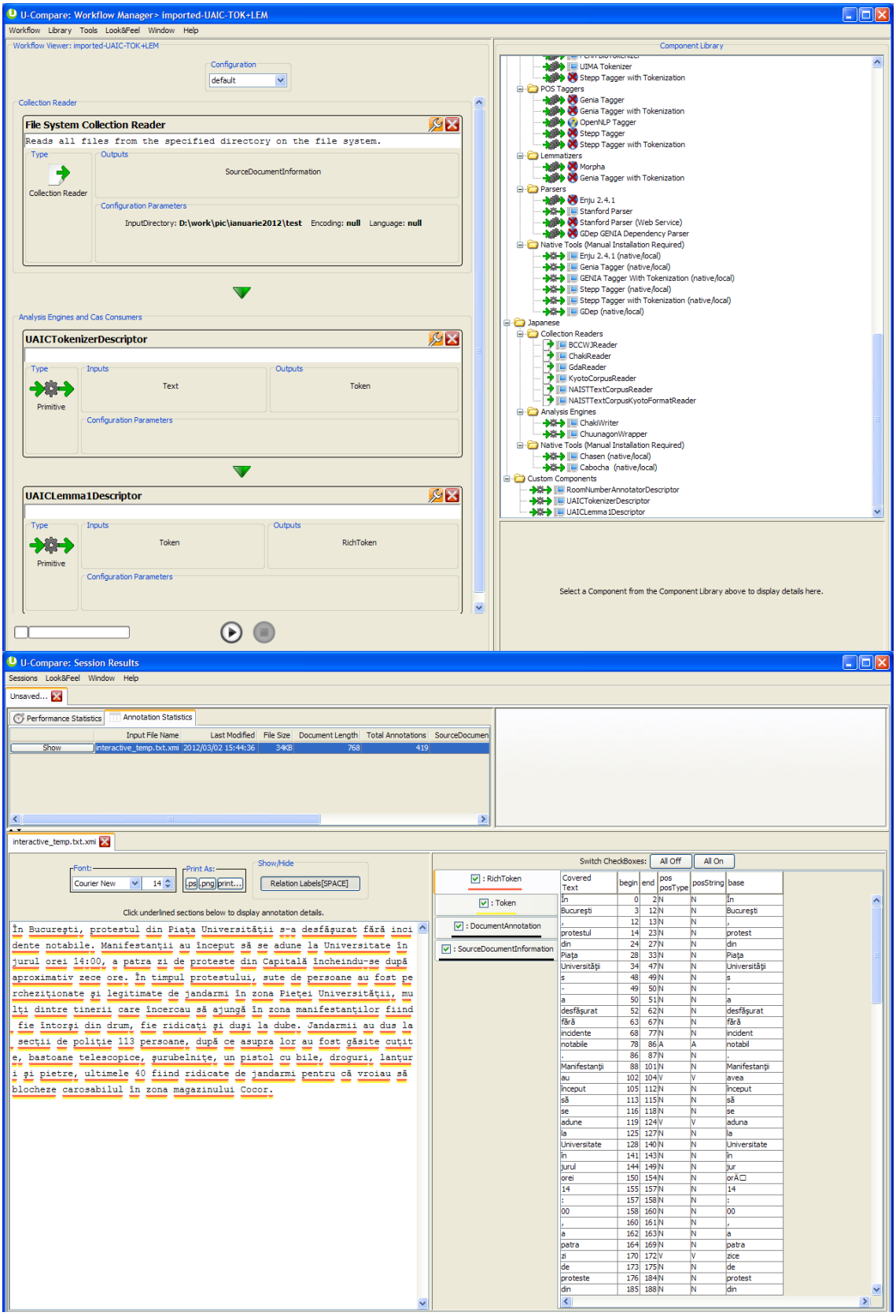

Figure 2: A workflow using UAIC components (above) and the results produced in U-Compare for a short text (below)

4. Conclusions

The benefits of collecting NLP resources from multiple developers and many languages and showing how they can be combined and compared is significant, both for application developers and the uninformed user of NLP techniques. The developer benefits knowledge of other similar tools, independent comparison of the results and guaranteed compatibility with relevant other components. The uninformed user can select available workflows without knowing their internal architecture, and can be assured that the components selected are compatible and working with the efficiency provided by the UIMA integration.

The benefits for the Romanian language are most of all of visibility, the large set of language processing components contributed by the Romanian partners (second largest in METANET4U, after English) proves again that Romanian is on the leading edge of NLP development.

What projects like METANET4U prove is that standardization brings significant advantages only if it involves large sets of developers and allows for some flexibility. The work carried out so far showed that open source components, web services and proprietary software can work together seamlessly if a minor standardization effort is made by motivated partners.

Acknowledgements. The work described in this paper was partially supported by the METANET4U EC CIP project #27089.

References

Branco, A., Trancoso, I., Ananiadou, S., Thompson, P., McNaught, J., Cristea, D., Tufis, D., Rosner ,M., Moreno, A., Bel, N. (2011). Specification of pilot services and applications. *Document METANET4U-2011-D2.2*, EC CIP project #270893, available on http://metanet4u.eu/.

Ferrucci, D., Lally, A. (2004). UIMA: an architectural approach to unstructured information processing in the corporate research environment, *Natural Language Engineering 10*, No. 3-4, 327-348.

Kano, Y., Miwa, M., Cohen, K., Hunter, L., Ananiadou, S., Tsujii, J. (2011). U-Compare: a modular NLP workflow construction and evaluation system. *IBM Journal of Research and Development*, 55: 3, 11:1-11:10.

METANET4U RESOURCE VALIDATION PROCESS

ȘTEFAN DANIEL DUMITRESCU

Research Institute for Artificial Intelligence, Romanian Academy, Bucharest

sdumitrescu@racai.ro

Abstract

The purpose of the article is to present the resource validation process for the European project METANET4U. The project's theme is the creation of a digital library that contains resources and associated tools for improving linguistic area in Europe. One step in the project is to collect the linguistic resources from the partners in the project and to validate these resources with the intention to make them available for METANET4U users. The validation process is described as a 4-steps process: Unicode verification, XML validation, visual inspection and counting the entities. During the performing of those 4 steps two tools have been developed. The validation reports are sent to the partners for a new validation-reporting cycle.

1. Introduction

The paper presents the validation process of resources within the METANET4U European project.

METANET4U is a project designed to help develop linguistic and multilingualism technology in Europe. The project is part of the META-NET excellence network. The main objective of the project is to develop a pan-European digital platform that provides its users with linguistic resources and services, covering data packages and dedicated software applications, in language and voice language processing.

The METANET4U project has 8 participants in 5 countries: Portugal is represented by Lisbon University and INESC (Institute for Systems Engineering and Computers), England by Manchester University, Romania by Alexandru Ioan Cuza University and the Research Institute for Artificial Intelligence within the Romanian Academy (RACAI), Malta by Malta University, and Spain by Technical University from Catalonia and Pompeu Fabra University.

One of the necessary steps in the project is collecting the linguistic resources from the partners in the project, correcting and aligning those resources to a minimum standard. Some of the requirements for this minimum standard are that each resource has a description document and, where applicable, to be encoded in Unicode (UTF8). There are also recommendations: for example, for text linguistic resources to be presented in the XML XCES format. Because of the complexity of converting one corpus from a format to a different format, these are only recommendations and not requirements.

In the project's third work package 3 (WP3 – "Enhancing language resources"), RACAI is responsible for collecting and validating the partners' resources. The received resources have different characteristics. Overall there have been received 83 distinct resources, each of them with a size between 2KB and 6GB. We have received XML

files (with the associated XSD and DTD schemas), text files (with or without annotations), WAV files for voice corpora, OWL files for ontologies and also specific formats like EXB, A, etc.

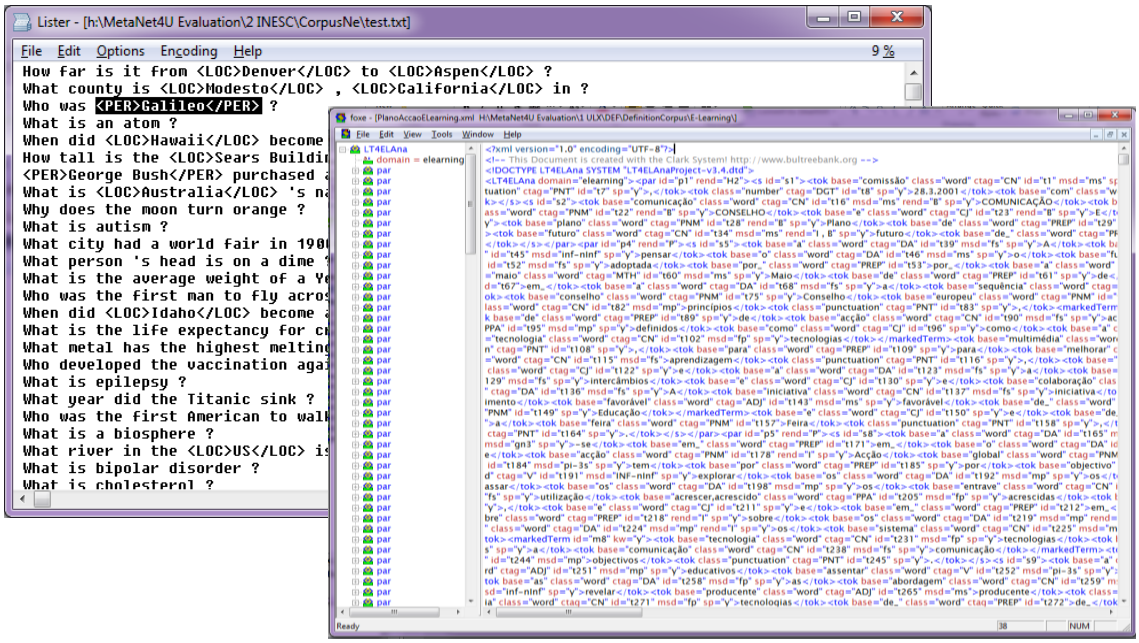

Figure 1: Example of an annotated text file and XML file.

In terms of content, resources can be free texts (in natural language) annotated with part of speech tags, entities, word senses (ex: RO-SemCor (Lupu et al., 2005), RO-WordNet (Tufiș et al., 2004)) etc.; annotated parallel texts for statistical machine translation systems; training corpora for question–answering systems; voice corpora (ex: Portuguese Spoken Corpus (Bacelar, 2001)); dictionaries (ex: WEB-DEX (Tufiș et al., 1999)) different tree/sentence/time banks (ex: RO-TimeBank (Forăscu, 2011)). Table 1 presents a breakdown on generic categories and number of corpora per category.

Table 1: Received corpora distribution

Corpora type	**# of resources**
Voice corpora	6
Lexicons	14
Named Entity / Question Answering / Textual Entailment	8
Parallel Corpora	4
Sense/Semantic Repositories	10
Annotated Text Corpora	25
Tree/Sentence/Time banks	6
Dictionaries	7
Other	3

Each resource that must be validated contains a file in Word format named “Corpus Description File”, hereinafter referred to as the description document.

The validation process consists in checking this file, the resources themselves and correlating the existent information in the description document and the actual resources. This process was developed by the specific needs that have arisen from

studying the received corpora and it is tailored to them, although the methods and applications developed can be used (if applicable) to any other corpora.

Validation steps – for each received corpus, the following 4 steps have been applied sequentially:

- Step 1. Unicode Validation
- Step 2. XML Validation
- Step 3. Entity Counting
- Step 4. Visual Inspection

followed by noting results from each step and sending them to the project partners.

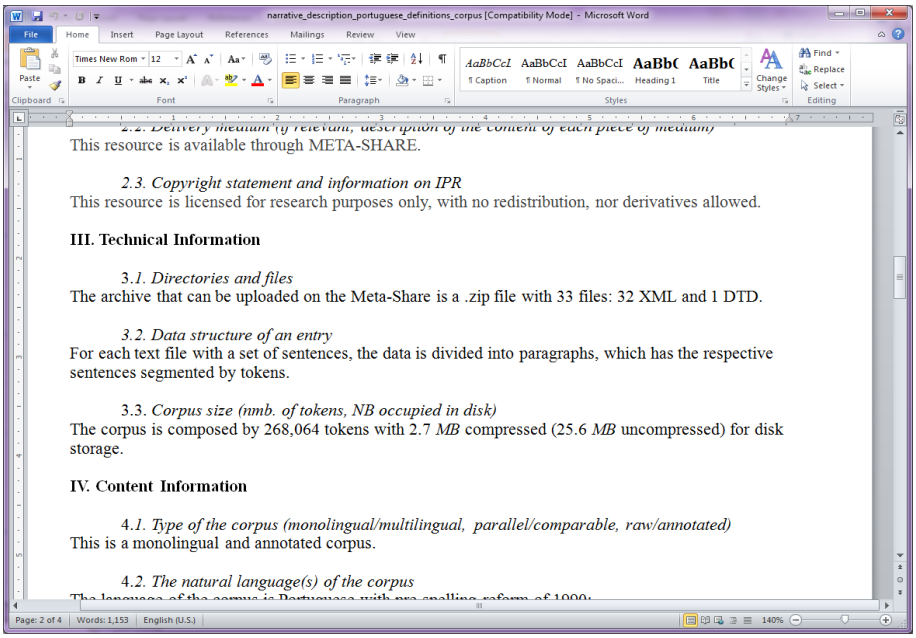

Figure 2: Example of resources description document

Further on, we describe the validation process in detail, along with applications that have been implemented to enable and streamline this process.

2. Resource validation process

2.1. Step 1 – Unicode Validation

The main problem of older corpora is the existence of SGML[1] encoding (Standard Generalized Markup Language) containing characters that cannot be represented in standard ASCII format (English character set).

For instance, this is how a SGML coded sentence looks like in the Romanian language:

*"… o diferen**ţă ş**i mai mare …"*

while in Unicode looks like this:

[1] http://www.w3.org/MarkUp/SGML/

"… o diferență și mai mare …"

This difference in encoding creates certain difficulties in processing the corpora. For a SGML corpus, a programmer should have a mapping table of SGML entities for recognizing them in body (and implicitly for *not* treating the characters that make up that entity as individual characters). This requires an additional step needed to be performed before the actual processing by any user of a corpus obtained from the METANET platform. As the SGML standard contains many possible encodings and has many variants, there can always be entities not present in that mapping table, leading to processing errors. This additional required processing step and these problems do not appear when all the corpora conform to a single standard encoding (Unicode[2]).

The Unicode standard defines characters / symbols representation from the majority of existent or past languages. This is possible by using 4 bytes to represent a symbol, thus being able to define 2^{32} possible symbols (about 4.3 billion different symbols). Encoding is done using one of the 3 representation formats: UTF32 (using 4 bytes), UTF16 (2 bytes) or UTF8 (using only one byte). UTF8/16 encodings are a more compact representation of symbols, where possible in terms of used space, being still able to represent any Unicode character by reverting to the 4 bytes representation if necessary. Currently, the most prevalent Unicode format in Europe and America is UTF8, as almost all of the characters used in these regions can fit in the single byte encoding.

Thus, such a tool is necessary to allow validation of the corpora. An application was created that can identify all SGML entities in a certain corpus. This allows rapid identification of corpora that do not respect Unicode standards.

Also, the next logical step was to expand the table of SGML entities with their Unicode counterparts for automatic conversion. Thus, a table of 2093 correspondents of SGML – UTF8 entities was manually created. This has led to a series of SGML corpora that were automatically converted to UTF8 by RACAI. In addition, a series of heuristic rules were added to the conversion tool. For instance, XML files explicitly require some character encoding such as < or & in *<* or *&*. Thus, for XML files, special SGML entities that were present in the data field of XML entities were not converted. Another heuristic implemented was converting the Romanian ***ş*** and ***ţ*** letters to the new representation. Specifically, in Windows XP, ***ş*** and ***ţ*** letters were represented as ***s*** and ***t*** letters with cedilla below them (eg. in SGML: *ş*) instead of comma (in SGML: *&scomma;*) due to limitations in the available set of characters at the time. The developed application allows automatic character substitution of the cedilla *s* and *t* to the comma *s* and *t*.

This step allowed rapid evaluation of the corpora in order to meet the Unicode standard and, when necessary, to transform SGML corpora to Unicode corpora.

2.2. Step 2 – XML Validation

The second validation step, perhaps even more important than the first, is XML[3] validation. Numerous linguistic corpora are represented in XML. The XML format

[2] http://unicode.org/
[3] http://www.w3.org/XML/

gives files a regular, computer-readable structure. This semi-structure is generally specified in XSD[4] files (XML File Schema) or DTD files (Document Type Definition). Because the XML format is widespread, there are a number of parsers (software applications) that read this format. Nevertheless, although XSD/DTD chosen schemas require a set of rules, it is not guaranteed that XML files referencing these schemas will comply with them. Therefore, an XML file that references an XML or DTD schema but that does not correspond to that schema will generate an error regardless of the parser used. This basically translates into the corpus suddenly becoming unusable, a user having to correct these errors in order to be able to automatically read the corpus. As the METANET platform is designed to be a repository of varied resources, the resources themselves have to be error-free. Therefore, XML validation of corpora assumes a significant importance, considering that most of the text corpora are encoded in XML.

A C# application was developed to automate this task. The tool has the following options:

- Validation of XML files referencing an XSD or DTD external schema;
- Validation of XML files referencing an XSD internal schema;
- Validation of XML files referencing a DTD internal schema;
- Validation of the XML structure for the files that are not referencing any schema.

For example, the command: "**xmlValidate.exe ApertiumBatch1 d**" will validate all XML files from ApertiumBatch1 directory, taking into account only DTD internally referenced schema files:

```
xmlValidator will not use an external DTD, only internal
referenced DTDs.
Source: All 20 xml files in [.]
     File 1 [Apertium-ca-it.ca-it-LMF.xml]:Document is OK.
     File 2 [Apertium-en-ca.en-ca-LMF.xml]:Document is OK.
     File 3 [Apertium-en-es.en-es-LMF.xml]:Document is OK.
     File 4 [Apertium-en-gl.en-gl-LMF.xml]:Document is OK.
     File 5 [Apertium-es-ast.es-ast-LMF.xml]:Document is OK.
     File 6 [Apertium-es-ca.ca-LMF.xml]:Document is OK.
     File 7 [Apertium-es-ca.es-ca-LMF.xml]:Document is OK.
     File 8 [Apertium-es-ca.es-LMF.xml]:Document is OK.
     File 9 [Apertium-es-gl.es-gl-LMF.xml]:Document is OK.
     File 10 [Apertium-es-gl.gl-LMF.xml]:Document is OK.
     File 11 [Apertium-es-pt.es-pt-LMF.xml]:Document is OK.
     File 12 [Apertium-es-ro.es-ro-LMF.xml]:Document is OK.
     File 13 [Apertium-eu-es.eu-es-LMF.xml]:Document is OK.
     File 14 [Apertium-eu-es.eu-LMF.xml]:Document is OK.
     File 15 [Apertium-fr-ca.fr-ca-LMF.xml]:Document is OK.
     File 16 [Apertium-fr-es.fr-es-LMF.xml]:Document is OK.
     File 17 [Apertium-oc-ca.oc-ca-LMF.xml]:Document is OK.
     File 18 [Apertium-oc-es.oc-es-LMF.xml]:Document is OK.
```

[4] http://www.w3.org/XML/Schema

```
     File 19 [Apertium-pt-ca.pt-ca-LMF.xml]:Document is OK.
     File 20 [Apertium-pt-gl.pt-gl-LMF.xml]:Document is OK.

All files are VALID.
DONE.
```

Although there are many free and paid XML validators, it was necessary to implement a new one for several reasons, out of which we present the most relevant:
1. Some XML files reference certain XSD schema files that are online. Sites that host these schema files block access to them for a few seconds (generally, between 5-60 seconds) in order to protect themselves from a wave of online schema requests with each opening of a local XML file that references the online schema. A simple validator, to validate "n" xml files would perform "n" requests for that schema, meaning a validation time "n" times greater than would be required. Thus, the implemented application preloads these schema files in memory and does not perform further online requests.

2. Errors can be very varied, coming from multiple files in any corpora. Because reports must be presented to partners, indexing these errors is necessary. The application does this automatically, creating for each error a list of affected files. In this way, the time required for creating reports is shortened.

It should be noted that much of the verified corpora presented XML validation errors, beginning from format errors, not respecting XML schemas to bad referencing of schema files. This second step was the most time consuming as also error correction was attempted before sending the validation reports back to the corpora owners.

2.3. Step 3 –Entity Counting

The last two resources validation steps assume verifying the information residing existing in the description documents. We can divide this verification in two distinct categories: step 3 that requires already existing tools in the Linux environment and step 4 that requires direct verification, without additional tools.

Thus, step 3 involves checking of the “countable” information in the corpora description documents. For example, an annotated natural language text corpus can contain information such as the number of sentences, of words, of entities, of senses, of verbs / nouns / etc., internal references, so on.

All this information must be verified. Since resources present themselves in various formats, fast verification requires using native Linux tools like **sed, grep, cat, tr, sort, wc.** We have chosen this set of tools because 1. They cover all the needs for this validation step, so there is no need for further tool implementation at this point 2. Are available on any Linux distribution – Linux operation systems are widespread in the academic / research domain, so access to these tools is a non-issue.

For instance, to count the words marked in all the XML files in a directory, the following command is used:

```
“cat *.xml | grep –o “<token” | wc –l”
```

To count how many XML files are in all existing directories and subdirectories in a certain corpus, we use:

```
"find . -name "*.xml" | wc -l"
```

As presented in the previous example, these tools can be chained together: the output of a tool is the input of the following one. This creates the opportunity to perform complex searches/counts very fast.

2.4. Step 4 – Visual Inspection

The last step in the resource validation process involves the visual inspection. In the description documents the list of files that constitute each resource is specified, the format of those files as well as other information such as the corpus size in kilobytes.

Thus, for each claim of this kind in every description document we carry out a visual inspection, making sure that all the files listed in the description document are present in the corpora (and the other way around), that the folder structure corresponds to the corpora, that the existing files really are their stated sizes, etc.

This last step has actually identified missing or additional files, inconsistencies in the folder naming / structure, and other such errors.

2.5. Example of a Validation Report

For exemplification, we choose one of the received resources: The Portuguese Definitions Corpus (LXDEF). This Portuguese lexicon was analyzed and the following notes have been made:

"

- *In the description document accompanying the corpus, is stated that the corpus has **274,000 tokens**. We have found only **223,049 tokens** identified in the 23 xml documents by tag <tok>.*
- *In the same document it is said: "Information Society (92,825 tokens), Information Technology (90,688 tokens), and e-Learning (91,225 tokens).". We have been unable to identify how the 23 files are clustered in the specific sub-domains, or how to identify the sub-domains at all.*
- ***While the files are valid xml files, they do not validate against the provided DTD**. Is it the final version that should accompany these files? We get the errors of the type (example):*

claroline_manual_teacher_portuguese.xml:4: element definingText: validity error: Element definingText content does not follow the DTD, expecting (chunk | tok | markedTerm)+, got (tok markedTerm connector tok markedTerm tok markedTerm tok tok tok tok tok tok tok tok tok markedTerm markedTerm tok tok tok tok tok tok markedTerm tok)

because file LT4ELAnaProject-v33.dtd contains:

<!ELEMENT definingText (chunk | tok | markedTerm)+ >

Even if we extend the definition of definingText by adding " | connector" by hand to the DTD, it will still fail validation because entity connector is not defined."

It can be noticed that errors were identified in the number of the words contained in the directory structure of the corpus and that the corpus presents XML validation errors. Errors were tried to be corrected with simple changes but failed to be removed. Where errors could be quickly removed, the method used was noted and sent to the partners in order to speed up their correction process that would also speed up the next round of validation.

3. Conclusions

The article presented the task of resource validation within the METANET4U project. This is a cyclical task with two phases: 1. Validation, 2. Reporting notes and partners validation. Partners correct the errors following these notes and resubmit them for another round of validation – reporting.

The actual validation process consists in 4 steps: 1. Unicode Validation, 2. XML Validation, 3. Entity Counting and 4. Visual Inspection. For the first two steps two special applications were implemented. The first application verifies if a corpus is in Unicode format (UTF8), with the possibility of conversion from SGML to UTF8. The second application verifies corpora in XML format and creates a report on the parse errors encountered.

This validation process is necessary for METANET4U platform users to access resources according to the standard required by the project – to have updated metadata (information) on these resources and for the resources themselves to be directly usable, without further changes.

Future work includes validating the next batch of resources and integrating them into the unified interface provided by the METASHARE platform. Furthermore, the project will also prepare a set of language processing tools that can be applied on the received corpora. The tools will also have to be validated and then integrated in the digital platform.

Acknowledgements. This work was funded by the project METANET4U by the European Commission under the Grant Agreement No 270893.

References

Bacelar do nascimento, F. (2001). (coord.) Português Falado, Documentos Autênticos. *Gravações áudio com transcrições alinhadas*, em CD-ROM, Lisboa, Centro de Linguística da Universidade de Lisboa e Instituto.

Forăscu, C. (2011). Contributions to Romanian language processing through discourse analysis methods. (in Romanian). *PhD thesis*. Romanian Academy, Bucharest.

Lupu, M., Trandabăț, D., Husarciuc, M. (2005). A Romanian SemCor Aligned to the English and Italian MultiSemCor. *In Proceedings of the Romance FrameNet*

Workshop and Kick-off Meeting, EuroLAN 2005, Babes-Bolyai University, Cluj-Napoca, Romania, 20-27.

Tufiș, D., Barbu, E., Barbu Mititelu, V., Ion, R., Bozianu, L. (2004b). The Romanian Wordnet. *Romanian Journal on Information Science and Technology*, 7: 2-3, 107-124.

Tufiș, D., Rotariu, G.,Barbu, A.M. (1999). Data Sampling, Lemma Selection and a Core Explanatory Dictionary of Romanian. *In Proceedings of the 5th International Workshop on Computational Lexicography* COMPLEX, Pecs, Hungary, 219-228.

GRAPHICAL GRAMMAR STUDIO AS A CONSTRAINT GRAMMAR SOLUTION FOR PART OF SPEECH TAGGING

RADU SIMIONESCU

University Al. I. Cuza Iași, Faculty of Computer Science

radu.simionescu@info.uaic.ro

Abstract

This work presents a hybrid part of speech tagger which successfully combines a statistic model with a rule based system. The rules behave as constraints used to reduce the ambiguity of the tokens. The novelty is in the tool used for building such rules. Graphical Grammar Studio (GGS) is an open source software designed for matching/finding sequences of tokens in a similar manner as the regex language is designed to match sequences of characters. Differences are that GGS can also annotate the matched sequences and that a GGS rule / grammar is a Recursive Transitional Network which can be edited using a user friendly visual tool. To ease the manual building of such rules, a custom made tool was used, which classifies tagging errors and shows the precision yielded by the rules. Finally, the paper presents the results of the part of speech (POS) tagging model for a Romanian.

1. Introduction

When a simple statistic part of speech (POS) tagger generates a type of error systematically, the only solution to fix it is to tweak various parameters. This usually leads to a trial and error approach which is not guaranteed to fix the problem. Modifying some of the parameters might also require retraining of the entire statistic model, which can take a lot of time.

By detecting the linguistic conditions for which the classifier generates a type of error, the hybrid model described in this paper can be configured to fix such errors with little effort.

Most common POS taggers use a POS dictionary for constraining each word to a small set of possible output tags. Then, a statistic model is used to disambiguate among these. The Hybrid POS tagger presented uses a method of reducing the ambiguity even further, by using rules which can take into account any features of the words within the input sentence.

POS taggers sometimes fail to correctly classify cases for which linguists can easily decide the correct part of speech. These types of errors are generated due to noise in the training data but also because a machine learning method cannot yet detect all the linguistic phenomena in the training set. The main goal of this work is to overcome this limitation.

Another goal is to introduce Graphical Grammar Studio, a tool for creating rules in a visual environment, displayed as networks of tokens, which are used for matching and annotating sequences. With this tool, a system for constructing constraint rules has been

implemented. The rules are very easy to edit, and understand due to the visual representation.

2. Graphical Grammar Studio as a constraint rules system

Before applying the rules, each word of the input sentence is associated with a list of possible tags, based on a POS dictionary. If a word is not found in the POS dictionary, it is associated with a predefined list of tags, considered the guesser tagset.

Applying a rule on a word, results in the reduction of the number of the possible POS tags associated to some tokens (usually, for the token for which it is applied, but other tokens can also be affected). All rules are applied on all words, from left to right.

The general architecture of the hybrid part of speech tagger is not new (Simionescu 2011). The novelty lies in the solution for the partial disambiguation rules system. In the previous approach (Simionescu 2011), a constraint rules language was designed, particularly for this task. This approach was similar to Constraint Grammars (Karlsson, et al. 1995) and JAPE (Cunningham, Maynard and Tablan 2000). After testing intensively with these, while building more and more complex rules, limitations of such "text based" languages were reached. It becomes very difficult to manage and understand the behavior of complex rules, written in programming languages, because they end up having a very elaborate complicated and unorganized look.

2.1. Graphical Grammar Studio

Graphical Grammar Studio (GGS) is a tool developed by the author and published as an open source project on SourceForge[1]. GGS offers the tools for creating and applying an extended variation of Recursive Transitional Networks (RTN) on tokenized text. It has been successfully used for creating a rule based deep Noun Phrase Chunker for Romanian.

At its core, GGS is a tool very similar with Nooj (Silberztein, NooJ: an Object-Oriented Approach 2004), but the latter is meant to become a comprehensive development environment for NLP tasks. Nooj is an improved version of the INTEX system (Silberztein, INTEX: a corpus processing system 1994). It has support for dictionaries, morphological grammars, paradigmatic representations etc. and it can read an impressive number of input formats. It also provides means to create chains of grammars.

GGS on the other hand is oriented only towards sequence matching and annotating. It is a tool meant for easy integration in various processing chains. It doesn't have support for dictionaries like Nooj and it doesn't require the presence of certain attributes for its input tokens. All token attributes are treated as key-value pairs which a GGS grammar can refer to, using regular expressions for both keys and values. GGS is the more out-of-the-box type of tool, because it doesn't require any prior configuration/installation whatsoever.

[1] http://sourceforge.net/projects/ggs/

Being a tool specialized on matching and annotating sequences of tokens, GGS contains several features which Nooj lacks in this aspect, like recursive depth and loop limits, or look ahead and look behind assertions. But before anything else, GGS is meant to be an open source project which can be used by anyone. Also, the Java platform might offer some technical advantages for some users, over the .NET platform.

GGS's main component is the GGS Editor. A secondary component is a java library used for integrating the GGS engine in java code.

GGS grammars are composed mainly of token matching/consuming nodes and empty nodes (which are used for visually organizing the aspect of a network of nodes, and for other features of GGS; empty nodes do not consume tokens from the input). The nodes are structured in sub networks (graphs). There are also jump nodes which can transition from one graph to another. Recursive jumps can thus be described, making this manner of defining matching grammars a very convenient one for many NLP rule based tasks. Each GGS grammar has a main graph. The starting and ending nodes of this grammar are the actual start and final states of the machine that runs behind the scenes.

A GGS grammar can be applied on xml input. The name of the tags which represent text units (usually sentences) must be provided. GGS expects to find sequences of token tags as the children of these xml nodes, which will be considered the input stack of tokens. A token tag can have an unlimited number of XML attributes. A token matching node accepts/consume the first unconsumed input token based on a condition which is described as a sequence of key-value pairs which must or must not be present in these attributes map. Moreover, both the keys and the values can be specified in such a condition using regular expressions, providing a great amount of flexibility.

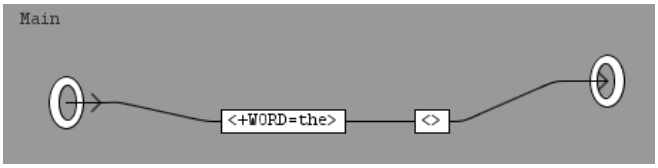

Figure 2: Simple GGS graph example

The main graph in Figure 1 matches all pair of words in which the first one is "the". The "<>" node matches any token because it doesn't impose any conditions. For the first token, the code is interpreted as: the next input token will be consumed if it has an attribute "WORD" equal to "the".

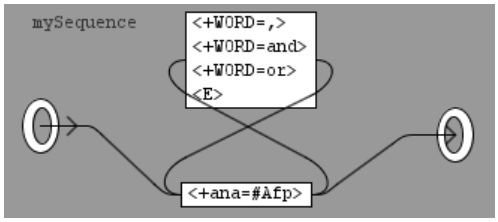

Figure 3: Secondary graph

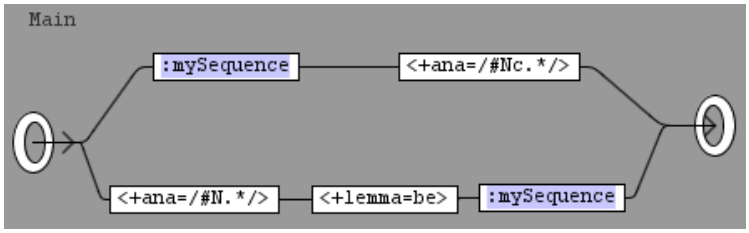

Figure 4: Main graph with jump nodes

Figure 3 and 3 show a secondary graph and how it is transitioned to, by nodes from the main graph. The mySequence contains a loop and will match any sequence of adjectives separated by optional conjunctions or commas (provided that input tokens which are adjectives have the attribute ana="#Afp"). The code <+ana=/#Nc.*/> is interpreted as: the next input token will be consumed if it has the attribute +ana mapped to a value which matches the regex "#Nc.*" (which will match tags which stand for common nouns, in this particular example). Regular expressions can also be used on the attributes names. The code <+/.*/=/\p{Lu}.*/> will match the input token if it has any attribute (regex .*) having a value which matches the regex \p{Lu}.* (which stands for string starting with upper case letters). The syntax of this language allows for specifying multiple.

2.2. Constraint rules system based on GGS

For the task of partial POS disambiguation, constraint rules must be able to define a conditional part, which can relate to the target token's (the token for which the rule is applied / the current token) neighbors and their attributes. Rules must also contain specifications for the actions to be taken in case their conditions are satisfied. These consist in manipulations of the lists of possible POS tags associated to tokens from the input sentence.

GGS is ideal for the conditions part. The system presented uses GGS for this; constraint rules are actual grammars which are used for matching sequences of tokens. This way, one can create such rules using GGS visual editor. GGS can also annotate matched sequences. This mechanism can be used for specifying what actions to be taken for particular tokens (a rule usually affects only its target token, but it can also affect any other token from a sentence).

An xml format for feeding the input to GGS is required. The matching engine can take one sentence at a time. Each token tag must model using XML attributes, among other details, its list of possible POS tags (based on a morphologic dictionary). Below is a sample of the chosen feed format. This is important in the creation process of rules.

```
<s>
  <W in_dict="true" lemma0="un" msd0="Timsr">Un</W>
  <W in_dict="true" lemma0="vrea" lemma1="om" lemma2="om"
msd0="Vaip1p" msd1="Ncmsrn" msd2="Ncmson">om</W>
  <W in_dict="false" lemma0="Laasdf" msd0="Afpmsry"
msd1="Afpfsoy" ... msd96="Vaip2p">Laasdf</W>
  <W in_dict="true" lemma0="eră" lemma1="fi" msd0="Ncfsry"
msd1="Vmii3s">era</W>
  <W in_dict="true" lemma0="de~ajuns" msd0="Rg">de ajuns</W>
```

```
  <W in_dict="true" lemma0="." msd0="PERIOD">.</W>
</s>
```

There is an unknown word in the example. This has 97 different possible POS tags because that is the size of the guesser tagset.

In the workflow of the entire process, a piece of code looks up words in the morphologic dictionary and creates the input to feed GGS, for each sentence. Then, the GGS matching process comes into action and, for each token (from left to right) and for each rule, it creates an output composed from the same input text with some of the tokens annotated with action specifications. Below is an example of such output.

```
<s>
  ...
  <W in_dict="true" lemma0="vrea" lemma1="om" lemma2="om"
msd0="Vaip1p" msd1="Ncmsrn" msd2="Ncmson">om</W>
  <KEEP regex="Np.*">
    <W in_dict="false" lemma0="Laasdf" msd0="Afpmsry"
msd1="Afpfsoy" ... msd96="Vaip2p">Laasdf</W>
  </KEEP>
  ...
</s>
```

The annotated output is interpreted by a piece of code. For the previous example, all the POS tags which match "Np.*" will be kept in the list of possible POS tags of the token identified by the child tag and the rest will be removed. In the same manner, an action for REMOVE can be specified.

By default, GGS was designed for finding matches anywhere in the given input text unit. It does this by actually applying the matching process repeatedly, offsetting the input each time by 1 position to the right. In the case of these constraint rules, this is not required. With the GGS java library, it was possible to explicitly request only one matching attempt starting from a particular offset from the input stream. This is how GGS was tweaked to create one output for each token-rule pair. GGS's flexibility made it very easy to be wrapped in a standalone NLP component.

2.3. Example rules

These are actual rules from the Romanian pos tagger implemented and described in the next section.

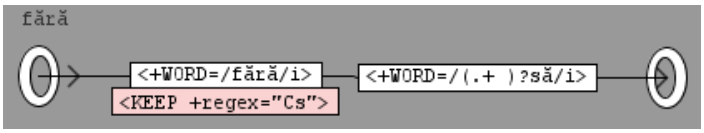

Figure 5: A simple pos reduction rule

Figure 5: A simple pos reduction rule shows a very simple rule, which if a token word matches "fără" (the *i* at the end indicates that the match is case insensitive) and the next word is "să" or is compound and has "să" at the end (e.g. "ca să", "în loc să", "are să" are words tokenized as one), then the first token can only be tagged with Cs (which stands for conjunction). This way, "fără" will not be confused by the statistic model for a preposition, in these cases.

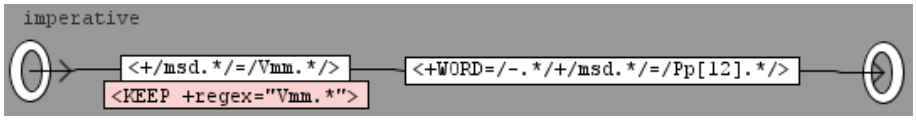

Figure 6: A more complex rule

The rule from Figure 6 deals with imperative verbs. If a token can be an imperative verb (it has an attribute starting with "msd" and equal to a value starting with "Vmm") and it is followed by a word which starts with a hyphen and can be a Personal Pronoun (in the first or second person), then the verb must certainly be an imperative one. This rule deals with cases as in "[culcă][-te]" (translation: "go to sleep") or "[ascultă][-mă]" (translation: "listen to me") or "[ridicați][-le]" (translation: "lift them") (brackets represent token boundaries), in which the verbs morphological features are usually confused by the prediction system.

For the system to behave as desired, it is necessary to make sure that the target token of a rule has a fixed position in the sequence which can be matched by its grammar. This is a problem, only for the rules which are looking to/consuming a variable number of tokens to the left of their target. Given the fact that rules are applied from left to right on each offset, there might be cases where such rules don't behave as desired – some tokens might get skipped by such rules; also, such rules might be applied multiple times, resulting in redundant operations. To overcome this, rules which look at a variable number of tokens to the left of their target should use look behind assertion conditions on the node which is supposed to consume the target token. This way the target token has a fixed position in the matched sequence of the rule, and the tokens to its left are checked by a look behind condition (which is actually a secondary grammar being applied). Not obeying this design rule results in slight changes in the constraint rules' behavior. In addition, assertions conditions in general can be used to define rules which cannot be defined otherwise. For instance: a word can be a conjunctive verb, only if there is a words "să" at a maximum distance of 5 tokens to its left, with the condition that there is no other potential conjunctive verb in between.

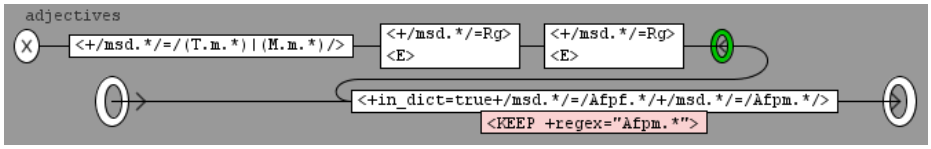

Figure 7: Example rule using positive look behind assertion

The rule from Figure 7 deals with a gender confusion for adjectives which have the same form in both masculine and feminine declination (e.g. "mare", "tare"). The disambiguation is done based on the presence of a masculine article or numeral to the left, with the possibility of two adverbs in between ("cea mai tare" - feminine vs "cel mai tare" - masculine).

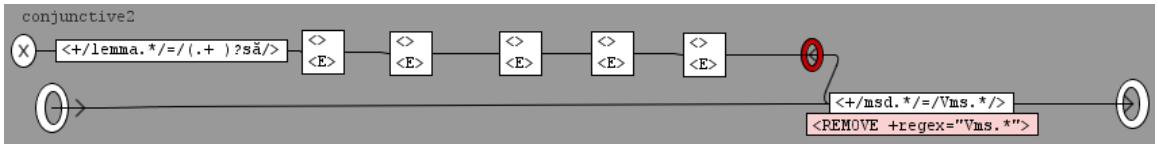

Figure 8: Example rule using a negative look behind assertion

Figure 8 shows a rule which uses a negative look behind assertion. If a word can be a conjunctive verb but there is no word "să" to its left at a maximum distance of five tokens then restrict the possibility of the word to be tagged as conjunctive verb.

Assertions are actually secondary grammars which are applied on input text. Look behind assertions particularly are consuming the tokens from right to left, and that is also the same order in which their nodes are parsed. The positive or negative attribute of assertions state whether the grammar must or must not match respectively, on the input tokens.

3. A rule editing tool

Writing rules to enhance the precision of the POS tagger can be difficult without any information about the errors generated at evaluation. For this reason a custom made application was developed which, while evaluating the model on a test corpus, shows various statistics regarding the fail cases detected. The software also provides the functionality to evaluate only a certain rule (or set of rules) and see the increase or decrease of precision yielded by it. This tool is completely language independent.

The rules are separated into final and test rules. The evaluation of the model when using only the final rules is a reference point for determining the differences yielded by the test rules. Evaluating the test rules results in actually evaluating the final rules plus the test rules together. Only the fail cases which are tagged differently in the full evaluation are considered test rules errors. For both final rules errors and test rules errors, the interface provides classifications by the following criteria:

- Most failed output tags
- Most failed expected tags
- Most frequent confusions(most frequent output-expected tags pair)
- Most problematic words

For each class of errors the user can visualize the sequences of words from the test corpus for which the POS tagger has failed (**Figure 9**).

By analyzing the final rules' fail cases, a linguist can detect contextual features for solving frequent errors, and write a new rule. This can then be easily refined because the user can see the exact fail cases and statistics for the new rule (by making it the single test rule in the system).

The application doesn't require multiple initializations of the POS dictionary and the statistic model, and that is why the work flow is smooth. Evaluating usually takes less than a few seconds. Nevertheless, the interface offers the possibility to use only a fraction of the testing corpus, to speed things up, if necessary.

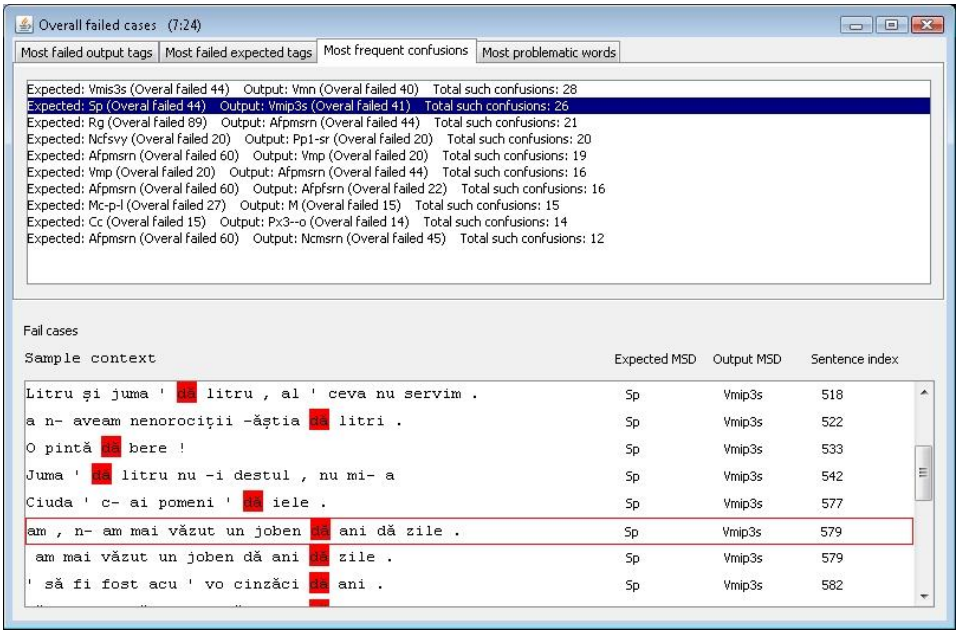

Figure 9: Window showing most frequent confusion

4. Romanian Hybrid POS tagger

A Romanian POS tagger has been developed by the author using the system described in the previous sections. The resources (morphologic dictionary, training and test corpora, tagset) and statistical model used were the same from the previous work (Simionescu 2011).

The morphologic dictionary was built with the help of the DexOnline.ro[2] database and Wikipedia[3] proper nouns collection, and it contains 1.25 million words associated to 230 000 distinct lemmas. The tagset used (406 tags) is based on the MSD classification (Erjavec 2004) used in MULTEXT-East[4], and it is a reduced version of the one used by the Research Institute for Artificial Intelligence, Romanian Academy[5] (Tufiș 2000) (about 600 tags).

The training corpus used is composed of NAACL 2003[6] plus 28.000 sentences extracted from JRC-ACQUIS[7] and tagged with the RACAI POS-tagger – 67000 sentences in total. The test corpus is the Multext-East "1984" corpus having around 6000 sentences.

The statistical model and its configuration are also the ones used in the previous reference research (Simionescu 2011). The statistic model used is the maximum entropy model. In his work (1998) Adwait Ratnaparkhi describes the use of maximum entropy for POS tagging. The maximum entropy model is used by the state of the art POS tagger

[2] http://www.dexonline.ro is the digitalization of some prestigious Romanian dictionaries. Part of the database used by DexOnline is available under the GNU license
[3] http://ro.wikipedia.org
[4] http://nl.ijs.si/ME
[5] http://www.racai.ro referred in the NLP community as "RACAI"1)
[6] A parallel corpus for Romanian-English created at the HLT/NAACL 2003 workshop, titled "Building and Using Parallel Texts: Data Driven Machine Translation and Beyond"
[7] JRC-ACQUIS is the largest parallel corpus. It is composed of lows for the EU Member States, since 1958 till present, translated and aligned for 23 languages

for English –Stanford Tagger– having precision of 97.32% (Toutanova, et al. 2003). Only recently has this score been overtaken (97.50%) (Søgaard 2011).

An online version of this POS tagger is available, both as a web application and a web service, at http://nlptools.infoiasi.ro/WebPosTagger/.

Table 1 shows the precisions obtained with and without applying the rules for the implemented Romanian Hybrid POS tagger.

Table 1: precision without and with rules for all words and unknown words

Precision	**Without rules**	**With rules**
For unknown words	88.88%	93.31%
For all words	95.12%	**97.03%**

The slight increase of precision from the previous version (96.66) is due to the fact that there were a few more rules added.

5. Conclusions

GGS is a tool which can be wrapped very nicely into a constraint grammar tool. With the implemented mechanism, one can easily create very complex rules which can be organized so that they become simple in aspect and structure. The new method of creating rules, using a visual tool opens up possibilities for less experienced users to experiment with what is one of the first bricks in computational linguistics – POS tagging.

The author intends to release the hybrid POS tagging model presented, as an open source project in the near future.

References

Ceaușu, A. (2006). Maximum Entropy Tiered Tagging. *Proceedings of the Eleventh ESSLLI Student Session* Janneke Huitink & Sophia Katrenko.

Cunningham, H., Maynard, D., Tablan, V. (2000). JAPE: a Java Annotation Patterns Engine (Second Edition). *Technical report CS--00--10*, University of Sheffield, Department of Computer Science.

Erjavec, T. (2004). MULTEXT-East Version 3: Multilingual Morphosyntactic Specifications, Lexicons and Corpora. *In Proc. of the Fourth Intl. Conf. on Language Resources and Evaluation*, LREC'2004, ELRA.

Karlsson, F., Voutilainen, A., Heikkilä, J., Anttila, A. (1995). Constraint Grammar: A Language-Independent System for Parsing Running Text. *Natural Language Processing*, 4. Mouton de Gruyter, Berlin and New York. ISBN 3-11-014179-5.

Ratnaparkhi, A. (1998). A Maximum Entropy Model for Part-Of-Speech Tagging. *Philadelphia: University of Pennsylvania Dept. of Computer and Information Science*.

Silberztein, M. (1994). INTEX: a corpus processing system. *Kyoto, Japan: COLING 94 Proceedings*.

Silberztein, M. (2004). NooJ: an Object-Oriented Approach. INTEX pour la Linguistique et le Traitement Automatique des Langues, C. Muller, J. Royauté M. Silberztein Eds, Cahiers de la MSH Ledoux. *Presses Universitaires de Franche-Comté*, 359-369.

Simionescu, R. (2011). Hybrid POS Tagger. *Cluj: Proceedings of "Language Resources and Tools with Industrial Applications" Workshop* (Eurolan 2011 summerschool).

Søgaard, A. (2011). Semi-supervised condensed nearest neighbor for part-of-speech tagging. *Portland, Oregon: The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (ACL-HLT).

Toutanova, K., Klein, D., Manning, C.D., Singer, Y. (2003). Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. *In Proceedings of HLT-NAACL 2003*, 252-259.

Tufiş, D. (2000). Using a Large Set of EAGLES-compliant Morpho-Syntactic Descriptors as a Tagset for Probabilistic Tagging. *International Conference on Language Resources and Evaluation LREC'2000*, Athens, 1105-1112.

SEMI-AUTOMATIC ALIGNMENT OF OLD ROMANIAN WORDS USING LEXICONS

MARIA MORUZ[1], ADRIAN IFTENE[2], ALEX MORUZ[2, 3], DAN CRISTEA[2, 3]

[1] *Centre of Biblical-Philological Studies "Al. I. Cuza" University, Iasi*
[2] *Faculty of Computer Science, "Al. I. Cuza" University, Iasi*
[3] *Institute for Computer Science, Romanian Academy, Iasi Branch*

mhusarciuc@gmail.com
{adiftene,mmoruz,dcristea}@info.uaic.ro

Abstract

This paper discusses an approach for the semi-automatic alignment of old Romanian words taken from three 17th century translations of the Bible. The alignment is first carried out at the verse level, then by means of lexical matching using Levenshtein distance, and then further refined by using a series of heuristics for matching the remaining words; to compensate for synonymy and high lexical variance, we have employed a lexicon. The biblical variants used are the 1688 Bible (in Romanian *Biblia de la București*), Manuscript 45 and Manuscript 4389, and the modern translation of the Bible given in the *Monumenta Linguae Dacoromanorum* series.

1. Introduction

Until recently, natural language processing has been primarily concerned with the modeling and analysis of the modern version of languages, in a synchronic manner. However, there has been a recent increase in the interest for digitizing and analyzing older versions of languages, as shown in (Roselli del Turco, 2010).

In the case of the old Romanian language, this interest has recently been sparked by the availability of digitized old Romanian texts, which were made available in the electronic version of the Romanian Thesaurus Dictionary (Cristea et al. 2009) and within the "Monument Linguae Dacoromanorum" project (Haja and Munteanu, 2010). As a result, a number of applications that make use of these resources have been proposed. One such application, for example, envisages the creation of a diachronical morphology for the Romanian language of 17th century by exploiting a large number of citations from eDTLR (Simionescu et al. 2012).

This paper is concerned with the creation of a lexical similarity equivalence database for the old Romanian language used in three 17th century translations of the Bible. The database is mainly concerned with semantic similarity at the word level, but we also intend to extend this similarity to expressions and idioms. The database is obtained by means of lexical alignment of parallel versions of texts, given that the three translations available are already aligned at the chapter and verse level.

The structure of the paper is the following: section 2 describes the source for the digitized version of the Bible translations, the "Monumenta Linguae Dacoromanorum" project, and the format of the digital version; section 3 presents the methods we have

employed for aligning the translation versions; section 4 describes the results obtained during the alignment process and section 5 presents conclusions and discusses future work.

2. The "Monumenta Linguae Dacoromanorum" project

The "Monumenta Linguae Dacoromanurum – 1688 Bible" project (Haja and Munteanu, 2010) is an international philological project started in 1988 by professor Paul Miron of the Albert Lüdwigs University of Freiburg im Breisgau, Germany, in partnership with the "Al. I. Cuza" University of Iasi, Romania. The purpose of this project is the creation of a philological edition, together with studies and linguistic commentaries, facsimiles and indices for words and variants, of the three contemporary and complete Romanian translations of the Bible from the 17th century: the 1688 Bible (also known as the Șerban Cantacuzino Bible or the Bucharest Bible) printed in Cyrillic characters, Manuscript 45 from the Romanian Academy library in Cluj, which contains the "revised Milescu version" from the 17th century, and the Romanian Academy library Manuscript 4389, which contains the "Daniil Panoneanul" version, also from the 17th century.

Every volume in the series has the following components:

1. The texts of the translation variants, arranged on 5 columns as follows:
    a. The facsimile of the original printed in 1688 (first column)
    b. The phonetic and interpretative transcript of the 1688 Bible (second column)
    c. The phonetic and interpretative transcript of Manuscript 45 (third column)
    d. The phonetic and interpretative transcript of Manuscript 4389 (fourth column)
    e. The modern translation of *Septuagint*, used as an auxiliary tool for the understanding of the old translation variants (fifth column)
2. Philological notes
3. Biblical-philological and linguistic commentaries
4. An exhaustive index of words and variants which contains all of the lexical occurrences in the 1688 Bible
5. Facsimiles of the manuscripts.

Until the time of writing, the "Al. I. Cuza" University Publishing House has published 9 of the projected 25 volumes: I. Genesis (1988), II. Exodus (1991), III. Leviticus (1993), IV. Numeri (1994), V. Deuteronomium (1997), XI. Liber Psalmorum (2003), VI. Iosue. Iudicum. Ruth (2005), VII. Regum I, Regum II (2008), IX. Paralipomenon I, Paralipomenon II (2011).

Starting with *Regum I, Regum II* (Andriescu et al., 2008), the volumes are available in electronic format (Haja et al., 2008), (Patras et al., 2008), together with an exhaustive index for the printed text (the 1688 Bible), ordered by lemma; attached to each lexical

occurrence are the morphologic analysis, the translations into German and French, and the first attested use of the term.

3. Aligning old Romanian words

The alignment of the parallel translations of the Bible was partly inspired by the notion of alignment between parallel texts in different languages (Moore, 2002). This idea was successfully used in the creation of a lexical similarity equivalence database consisting of French and Romanian multi-word expressions aligned according to semantic similarity (Husarciuc, 2008). In order to create a similar taxonomy for old Romanian texts, we first need the world level alignment from which to extract the expression alignment.

In the case of parallel translations in the same language, the problem of aligning parallel texts is simpler, as it is to be expected that at least some of the words are common, even accounting for differences induced by differences in time and region. Because of this we have adopted a bootstrapping based approach that makes use of Levenshtein distances between words and a set of heuristics and resources to improve the alignment result.

3.1. Alignment algorithm

The input for the alignment system consists of pairs of pre-aligned verses extracted from the sources described in section 2. A general outline of the alignment process is given below:

1. Starting from the pre-aligned verse pair, we attempt to match words on the basis of lexical similarity, extracted by means of exact match and Levenshtein distance. Since there is the possibility that a verse contains the same word more than once, the alignment score is weighed by the distance between the positions of the two candidates in the verses. Usually, the words extracted at this stage are proper names and words that have little temporal or regional variance (e.g. “bătrînețe”, “Solomón” etc.), and can be counted upon to be semantically equivalent in a large majority of cases;
2. Given the previously extracted alignments, we consider the aligned words pivots and attempt to align further words on this basis. To this extent we consider that unaligned words that are found next to already aligned words and are lexically similar have a high probability of being aligned, and so their matching score is boosted;
3. In order to account for the semantic similarity of words that are not lexically similar, we have used an automatically generated lexical similarity equivalence database which has been validated by hand. It contains the word pairs that have been aligned by means of heuristics (such as those in step 2), and is manually validated in order to avoid incorrect matches. The score assigned to the alignment obtained by using the taxonomy is computed on the basis of match frequency in the aligned corpus, as a specific word can have multiple semantic equivalents (e.g. “domn - boiarin” and “căpetenie - boiarin”). The taxonomy is described in greater detail in subsection 3.2;

4. The process is repeated from step 2 until no new alignments can be carried out.

3.2. A lexical database for semantic similarity

Given the fact that no suitable lexicon for the old Romanian language exists, the most accessible manner for solving the issue of semantic similarity is the creation of a taxonomy that describes this relation. While solving this issue, such a database is an accomplishment in itself, as this is a valuable resource for the study of the Romanian language of the 17th century.

The database is populated by adding those words that have lexical similarity scores below an empirically determined threshold, but have been aligned at step 2 in the algorithm given above. These alignments are then manually validated by human annotators, and then a bootstrapping approach is applied, in order to add further pairs to the database. Since the texts on which alignment was carried out are not available in lemmatized form for the manuscripts and the modern translation, the words in the database are given in inflected forms, which greatly reduce the number of cases where a given relation can be inferred. A relation consists of the semantically similar words and an attached confidence score, which is assigned on the basis of frequency in the corpus (we only take into account the corpus represented by the validated alignments). Examples of such relations are given in Fig. 1 below:

fiu [is] fecior [score] 1
jîrtăvnicul [is] altariul [score] 1
ţiitoarea [is] posadnica [score] 1
astruca [is] îngropa [score] 1
boiêri [is] domni [score] 0.6
boiêri [is] căpetenii [score] 0.4
den sîmbătă în sîmbătă [is] în toate sîmbetele [score] 0.7
împărăţi [is] stătu împărat [score] 0.9

Figure 1: Entries in the Semantic Similarity Taxonomy

As can be seen in the examples above, a similarity relation usually holds between two words (one-to-one relations), but we have also allowed for the possibility of one-to-many and many-to-many relations, in order to model similarity for multi-word expressions.

4. Results and discussions

The algorithm described in section 3 was tested on the texts available in (Andriescu et al., 2008), which contains the books “Regum I” and “Regum II”, and the obtained results are described in this section. The reason for our using this particular volume was twofold: it is the first volume in the series to have an electronic index of semantically disambiguated words (Haja et al., 2008), (Patras et al., 2008) and it was available in a structured electronic format that allowed quick access to the aligned verses.

Also, these particular books are less difficult to align, since verses usually contain large numbers of proper nouns, which are easily matched; large numbers of high confidence

matches give high confidence in heuristic matches, and thus the lexical similarity equivalence database is more quickly populated. Once the database is already established and contains large numbers of relations, the alignment of verses that are not lexically similar becomes easier.

Table 1 below gives the results of the application of our algorithm on these books. The alignment was carried out on pairs of translation versions: the 1688 Bible represents version 1, manuscript 45 is version 2, manuscript 4389 is version 3 and version 4 is the modern translation (in order to determine the similarity of the modern translation to a 17^{th} century one, we have decided to also align manuscript 4389 to the modern translation). The numbers in the table represent the number of word alignments that have been determined.

Table 1: Alignment results for "Regum I" and "Regum II"

Match type	**1 – 2**	**2 - 3**	**3 – 4**
Lexical identity	40191	25389	14798
Levenshtein distance	5946	6913	4973
Lexical database	569	443	453
Unmatched words	6600	19055	28866

As can be seen in Table 1, the 1688 Bible and manuscript 45 are very similar, which is to be expected given the fact that one is based on the other. Manuscript 45 and manuscript 4389 are significantly different at the lexical level, as shown by the lower number of lexical matches, while manuscript 4389 and the modern translation have very few lexical similarities, which is to be expected due to the evolution of language. The low number of semantic taxonomy matches is due to the fact that, at the time of testing, the taxonomy contained approximately 100 pairs; given the fact that these pairs contain inflected words, their scope is limited, thus resulting in a low number of matches.

Examples of alignment cases which support the steps of the algorithm in section 3 are given below.

Regum I, 1, 10

B1688*: Și ea, amărîtă la suflet, și s-au rugat către Domnul și plîngînd au plîns*

Ms. 45*: Și ea, amărîtă la suflet, și s-au rugat către Domnul și plîngîndu au plînsu.*

Example 1: Lexical similarity based alignment

Example 1, given above, is a case of near perfect lexical match. This similarity is mainly due to the fact that B1688 is largely based on Ms. 45. For this particular case, step 1 of the algorithm in section 3 solves all of the alignments.

Regum I, 15, 5

B1688 *: Și veni Saul pînă la cetățile lui Amalic și* ***strejui*** *la pîrîu.*

Ms. 45*: Și veni Saul pănă la cetățile lui Amalic și* ***să aleșuiră*** *în părău.*

Example 2: Enriching the semantic similarity taxonomy using heuristics

In the case of example 2, all of the words in the verses are aligned at step 1, with the exception of those words which are highlighted. According to step 2 of the algorithm, since the highlighted words are surrounded by already aligned words, and since there

are no other words that are not aligned, the similarity score of the word pair is boosted, and alignment is found. The pair is inserted in the database candidate pool, awaiting manual validation.

Regum II, 24, 18

Ms. 45*: Și veni Gad către David întru dzua acêea și-i dzise lui: "Suie-te și pune Domnului jirtăvnic întru ariia lui Orna, ievuseului!".*

Ms. 4389*: Și veni Gad într-acea zi la David și-i zise: "Suie-te și pune altar lui Dumnezeu în arătura lui Iornei ievuseul".*

Example 3: Semantic similarity based alignment

In those cases where the semantic similarity taxonomy contains a word pair, alignment is carried out directly on the basis of that relation. Such is the case in example 3, where two alignments are carried out by this method. It is worth noting that without the taxonomy, the alignment would be very difficult to predict, given the low lexical similarity of the words in the vicinity. Table 2 shows the improvement brought by the lexical similarity equivalence database to the alignment process (between 1 and 2):

Table 2: Alignment results with and without the lexical similarity equivalence database

Run type	Exact	Ontology	Levenshtein	Not aligned
With database	40191	569	5946	6600
Without database	40191	0	6042	7073

Alignment is made more difficult by a series of translation inconsistencies within the variants. Such an inconsistency is given by the fact that parts of the original text in some verses are missing and are given in endnotes or not given at all. Such is the case in example 4, where part of the text in manuscript 45 is missing and is given in an endnote.

Regum I 20, 30

B1688*: Și să mînie cu urgie Saul pre Ionathan foarte și zise lui: "Fecior de fêtele cêle ce mergu de bunăvoie, au nu știu că părtaș ești tu cu* ***fiiul lui Iesei*** *întru rușinea ta și întru rușinea descoperirii maicii tale?*

Ms 45 *: Și să mînie cu urgie Saul preste Ionathan foarte și-i dzise lui: "Fiu a fêtelor ce mărgu de bunăvoie, au nu știu că părtaș ești tu[10] întru rușinea ta și întru rușinea dăscoperirei maicii tale?*

+note *: Marginal note in another hand: "****fiiului lui Iese****".*

Example 4: Missing text in verses

Another type of inconsistency which greatly hinders automatic word alignment is the use of proper and common nouns for denoting the same concepts (this occurs most commonly in the case of names of peoples). In extreme cases, the word forms in the translation variants are very different, as is the case in example 5 below. The verse from manuscript 45 is also missing some text, which does not exist even in the endnotes.

Regum I 15, 6*:*

B 1688*: Și zise Saul către Chineu: "ferêște-te și te abate den mijlocul* ***Amalichitului****, ca să nu te adaogă împreună cu el; și tu ai făcut milă cu toți fiii lui Israil cînd să suia ei den Eghipet". Și să abătu Chineul den mijlocul lui Amalic.*

***Ms. 45**: Și dzise Saul cătră Chineu: "ferêşte-te // și te abate den mijlocul **amalichitului**, ca să nu te adaog împreună cu el. Și tu ai făcut milă cu toți fiii lui Israil cînd să suia ei den Eghiptu".*

***Ms. 4389**: Și zise Saul lui Chinei: "Pasă și te dă în laturi den mijlocul **ammalitênilor** și nu te apropiia de dînșii să te concenesc depreună cu dînșii, că tu ai făcut milă cu feciorii lui Israil cînd ieșiia den Eghipet". Și se dêde Chinei într-o lature den mijlocul ammalitênilor.*

Example 5: Proper and common nouns for the same concept

5. Conclusions and future work

This paper proposes an algorithm for aligning translation variants of old Romanian texts by means of lexical and semantic similarity. It also proposes a method for extending existing semantic similarity taxonomies by using a set of heuristics. The results obtained prove the potential of our proposed method, as increases of the lexical similarity equivalence database are directly correlated to increases in the number of alignments.

The proposed alignment can be used for the extraction of inflection variants for the old Romanian language, which is useful for the creation of an old Romanian grammar; also, the alignment to the modern version of the translation allows for the observation of the evolution of words and expressions.

As future work we intend to apply our algorithm to a new volume in the MLD series, *Paralipomenon I, II*, in order to further test and enhance our algorithm. Also, for future alignments, we will use the B1688 version as a pivot, aligning the other variants only to it, mainly because most of the lexical semantic disambiguation has been carried out on this version. Also, the generic Levenshtein string matching algorithm should be modified in order to accommodate a series of linguistic phenomena, such as ignoring the final "u" in some words (e.g. "plînsu" vs. "plîns") or aligning "dz" to "z" ("zise" vs. "dzise").

Acknowledgements. This work was partly funded by the "Al. I. Cuza" University of Iasi and the Sector Operational Program for Human Resources Development through the project ―Development of the innovation capacity and increasing of the research impact through post-doctoral programs POSDRU/89/1.5/S/49944 and the METANET4U – Enhancing the European Linguistic Infrastructure project.

References

Andriescu, A., Miron, P., Haja, G. (coordinators) (2008). Monumenta linguae Dacoromanorum. Biblia 1688. Pars VI. Regum I, Regum II. *Iași, "Alexandru Ioan Cuza" University Publising House*, 560 p. + DVD. Authors: Tamara Adoamnei, Mădălina Andronic, Mioara Dragomir, Gabriela Haja, Elsa Lüder, Paul Miron, Alexandra Moraru, Mihai Moraru, Adrian Muraru, Veronica Olariu, Elena Tamba Dănilă. Scientific consultant: Eugen Munteanu. Electronic format on DVD by Vlad-Sebastian Patraș.

Cristea, D., Răschip, M., Moruz, A. (2009). Steps in Building the Electronic Version of a Thesaurus Dictionary of the Romanian Language. *Buletinul Institutului Politehnic din Iasi*. Sectia: Matematica. Mecanica Teoretica. Fizica, 1244-7863.

Haja, G., Munteanu, E. (2010). Monumenta linguae Dacoromanorum. 1688 Bible Project. *In Clarin, Newsletter of Clarin Project*, 8, 4-5 http://www.clarin.eu/files/cnl08_web.pdf.

Haja, G., Dănilă, E., Clim, M. R., Patraș, V. (2008). Contribuții la informatizarea cercetării filologice românești: Biblia 1688 și eDTLR. *Simpozionul internațional Distorsionări în comunicarea lingvistică, literară și etnofolclorică românească și contextul european*, Iași, 25-28 septembrie.

Husarciuc, M. (2009). Echivalarea în limba română a unităților frazeologice infinitivale din limba franceză. *În Lucrările Atelierului Resurse lingvistice și instrumente pentru prelucrarea limbii române* (Iași, 19-21 noiembrie 2008), Editura Universității "Alexandru Ioan Cuza" Iași, 115-124.

Moore, R. C. (2002). Fast and Accurate Sentence Alignment of Bilingual Corpora. *In "Lecture Notes In Computer Science"*, 2499 - Proceedings of the 5th Conference of the Association for Machine Translation in the Americas, on Machine Translation: From Research to Real Users, Springer-Verlag, London, ISBN: 3-540-44282-0, 135-144.

Patraș, V. S., Pavel, G., Haja, G. (2008). Resurse lingvistice în format electronic. Biblia 1688. Regi I, Regi II – probleme, soluții. *În volumul Resurse lingvistice și instrumente pentru prelucrarea limbii române*, editori Ionuț Cristian Pistol, Dan Cristea, Dan Tufiș, Iași, Editura Universității "Alexandru Ioan Cuza", 51-60.

Roselli del Turco, R. (2010). Filologia digitale: ragioni, problemi, prospettive di una disciplina. *III Incontro di Filologia Digitale*, Verona, 3-5 marzo.

Simionescu, R., Cristea, D., Haja, G., Minuț, A. M. (2012). Inferarea unei morfologii diacronice folosind eDTLR, to appear.

GRAPHIC COMPARABILITY LEVELS FOR COMPARABLE CORPORA

RADU ION

Research Institute for Artificial Intelligence, Romanian Academy
Calea 13 Septembrie nr. 13, Bucharest 050711, Romania

radu@racai.ro

Abstract

Comparable corpora are a notable solution to the parallel data acquisition bottleneck for under-resourced languages for statistical machine translation. A comparable corpus is inherently different from a parallel one in that the translations, should they exist, are scattered throughout the corpus. Given that such a corpus is usually very large, it is important to be able to estimate the level of parallelism of the comparable corpus before embarking into the task of parallel data mining which is a computationally intensive task.

1. Introduction

Parallel corpus collection from the Web or otherwise is a laborious task which, more often than not, is difficult to complete for under-resourced languages. By "under-resourced languages" we understand languages for which linguistic computational resources (such as parallel corpora of a satisfactory size) and tools are not readily available. Furthermore, the size of the document collection available on the Web is significantly lower than for "highly-resourced" (and computationally studied) languages such as English, Spanish or French.

Usually, parallel corpora are collected from either specialized institutions that employ translation on a regular basis (publishing houses, law offices, etc.) or, more conveniently, from the Web. Collecting parallel data from the Web requires that:

1. Websites containing parallel information in more than one language are identified;
2. The website structure is analyzed and a web crawler is programmed in such a way that it is able to extract the pages that are mutual translations. For instance, from the website http://ec.europa.eu/, the English side begins at http://ec.europa.eu/index_en.htm which has the Romanian counterpart at http://ec.europa.eu/index_ro.htm.

The direct consequence of this approach is that the web crawler that is developed for site A cannot be used without fundamental modifications for another site B. Furthermore, one has to pre-analyze the HTML page structure in order to find the relevant text parts that are parallel.

In contrast, comparable corpora are mainly collected monolingually. There are websites such as http://www.wikipedia.org/ which provide linked strongly comparable texts (Ion,

2011b) but the biggest advantage of collecting comparable corpora is that the two requirements listed above may be skipped. This effectively means that:

- Web crawlers developed to collect comparable corpora may be reused to collect any type of comparable corpora;
- No website identification is necessary nor complicated HTML page structure analysis is required;
- The comparable corpus can become very large rapidly, much larger than a parallel corpus collected with our sketched algorithm.

In the context of Statistical Machine Translation (SMT), it is important to estimate the amount of parallel data that exists in a comparable corpus collected from the Web. In order to train translation models, an SMT engine such as Moses (http://www.statmt.org/moses/) needs lots of pairs of parallel sentences/phrases and, if we are to extract these pairs from our collected comparable corpora, we need to know if the corpus actually contains parallel pieces of text. In other words, we need to estimate the "comparability level" of the corpus which, in our view, continuously ranges from "parallel" through "strongly comparable" to "weakly comparable" and "unrelated".

In what follows, we will review some of the work done in the area of estimating how comparable is a comparable corpus and we will propose an experimental methodology to identify the comparability level of a comparable corpus in a document-paired comparable corpus.

2. Considerations on the comparability of texts

In this section we will refer to bilingual comparable corpora as opposed to monolingual comparable corpora. Thus, we are aiming at defining comparability of two texts or collection of texts, the first one written in a source language and the second one in a target language.

Comparability of a corpus is rather difficult to define and Maia (2003) goes even further and states that "*to a certain degree, comparability is in the eye of the beholder*". This statement is made in the context of collecting and using comparable corpora for translation studies and reflects the opinion that the comparability of a corpus actually depends on the particular usage of that corpus. The EAGLES definition[1] of a comparable corpus is that "*A comparable corpus is one which selects similar texts in more than one language or variety.*" And "*The possibilities of a comparable corpus are to compare different languages or varieties in similar circumstances of communication, but* ***avoiding the inevitable distortion introduced by the translations.***" Finally, Munteanu and Marcu (2005) define a (bilingual) comparable corpus to be "*a set of paired documents that, while not parallel in the strict sense, are related and* ***convey overlapping information***".

It seems that the keywords in the proposed definitions of a comparable corpus are "similar" and "related". This is where the concept of a comparable corpus becomes fuzzy: how can one define the "similarity" or "relatedness" (which are not the same

[1] http://www.ilc.cnr.it/EAGLES96/corpustyp/node21.html

thing) of two texts or of two collections of texts, one in the source language and the other in the target language? To answer this question, one must realize that the similarity or relatedness of the texts of a comparable corpus is the foundation of the comparable corpus construction. Thus, the way the author of the comparable corpus has defined the "similarity" and/or "relatedness" of a pair of texts becomes the very nature of the comparable corpus. For instance, collecting sports news texts from the Web in a certain day and maybe when a major sporting event is unfolding (like the Olympic Games) has great chances of producing a comparable corpus in which the "similarity" of the documents is the membership to the news texts/sports genre and the "relatedness" is presumed to be the daily covering (independently and in multiple languages: who won, what discipline, etc.) of the competitions held that day. Thus chances are that:

- The athletes' opinions are quoted and translated in different languages;
- Specific terminology (names of the games themselves, sport equipment, places, etc.) are also translated in different languages.

At this point we would like to stress out that the EAGLES characterization of a comparable corpus to contrast (texts in different) languages without the distortion introduced by translation (see the bolded part from the definition) is unattainable when "similarity" and "relatedness" come into play. Of course that, between documents collected independently in two different languages (but which are bounded by specific similarity/relatedness requirements) translated spans of text (sentences/phrases) may be missing entirely but, at least, single word and multi-word terms have to be translated because of the fact that selected documents are similar/related in a specific way.

One formal approach at evaluating the comparability level of a corpus is due to Kilgarriff (2010). He uses content word frequency lists when he compares two collections of texts and, by a selection of top N ranked words for each language along with a translation dictionary, one can decide how much the two corpora are alike (by computing a correlation coefficient for instance).

The ACCURAT project (http://www.accurat-project.eu/) produced several comparability metrics for comparable corpora that correlate very well with the amount of parallel data that can be extracted from the corpus. The metric works on a pair of source and target documents and assigns a score between 0 (unrelated) and 1 (parallel). It has been shown (Su et al., 2011) that the amount of parallel data extracted from document pairs with high comparability scores is significantly larger than parallel data extracted from document pairs with a lower comparability score. The approach uses stemmed keyword vectors that are computed from the source and the target document and the metric is in fact, a cosine similarity between the vectors.

3. Graphic comparability levels

In the context of SMT, it makes sense to be able to learn beforehand if the bilingual comparable corpus is close to one of the 3 types of comparable corpora that we know of:

1. Parallel corpora: the corpus is a collection of document pairs, each source document is completely translated (paragraph by paragraph, sentence by

sentence) by the corresponding target document; an example of this type of corpus is the JRC Acquis corpus (http://langtech.jrc.it/JRC-Acquis.html);

2. Strongly comparable corpora: the corpus is a collection of document pairs, each source document has visible and easily detectable (at a manual inspection) sentences or paragraphs that are translated as such in some (unpredictable) part of the corresponding target document; an example of this type of corpus is the document collection from Wikipedia (http://www.wikipedia.org/);
3. Weakly comparable corpora: the corpus is a collection of document pairs, each source document is similar and related to the corresponding target document but the actual translations occur only at phrase/word level and are difficult to spot.

One reason for which we would like to know in the comparability level of the given document pair/corpus in advance is that the parallel data mining algorithms are CPU intensive. The typical parallel data mining algorithm performs the following steps:

1. Segments the source and target documents at the required granularity: paragraphs, sentences of phrases (we call these "textual units");
2. For each pair of textual units from the source and the target documents, it computes a translation similarity measure (e.g. score between 0 and 1) or a classification of the pair (e.g. parallel/non parallel).

The second step can be computationally expensive in terms of computation time and as such, waiting to process a not so strongly comparable corpus/weakly comparable corpus to obtain only a few parallel pairs is not acceptable.

Our methodology of detecting the comparability level of a comparable corpus relies on a method of pairing (aligning) the documents in the corpus such that the probability of encountering translations in each pair is maximized. For this purpose we implemented a tool called EMACC which uses an Expectation-Maximization algorithm to detect these pairs (Ion, 2011a). EMACC will assign a "translation strength" score to each document pair it finds and from here on, we assume that the studied comparable corpus is document aligned.

In connection with the document alignment in a comparable corpus, we also define the "document alignment productivity": if for a given document pair *x*, there are *m* different target documents that align to the source document and *n* different source documents that align to the target document, the document alignment productivity is

$$P(x) = 1 - \frac{1}{\sqrt{\max(m,n)}}$$

The "source average document alignment productivity" (denoted by "Avg. 1: n") is the average number of target documents that align to a single source document and the "target average document alignment productivity" (denoted by "Avg. n: 1") is the average number of source documents that align to a single target document.

With these notions at hand, the intuition behind detecting the comparability level of a corpus containing M documents in the source language and N documents in the target language is that:

- If the corpus is parallel (and complete), then M = N and an accurate document alignment methodology would have to provide only 1:1 document alignments with high alignment probabilities. It also means that the document alignment productivity would also have to be close to 1 for every document pair in the corpus;
- If the corpus is strongly comparable, M and N are not necessarily closer to one another, but the document alignment probabilities would still have to be high. The document alignment productivity is now greater than 1, but not much greater;
- Finally, in the case of weakly comparable corpus, M and N are very different, the document alignment productivity would be large for each document pair and relatively few pairs of documents will have their alignment probabilities greater than the average on the whole alignment set.

It is clear that the judgments above are heavily dependent on the ability of the document alignment technique (EMACC in our case) to actually detect and assign high alignment probabilities to parallel or strongly comparable document pairs and to assign low alignment probabilities to unrelated document pairs.

It turns out that if plotting the document alignment productivity (from the lowest to the highest) and the document alignment probabilities (from the highest to the lowest) for all pairs of documents in a comparable corpus, some interesting patterns occur with respect to our assumptions. The next figures depict these plots for a parallel corpus (an English-Romanian parallel corpus of news collected from http://ec.europe.eu/ website), a strongly comparable corpus (a subset of the English-Romanian Wikipedia corpus where the Romanian documents have been collected following the inter-lingual links) and a weakly comparable corpus (an English-Romanian corpus collected in the ACCURAT project).

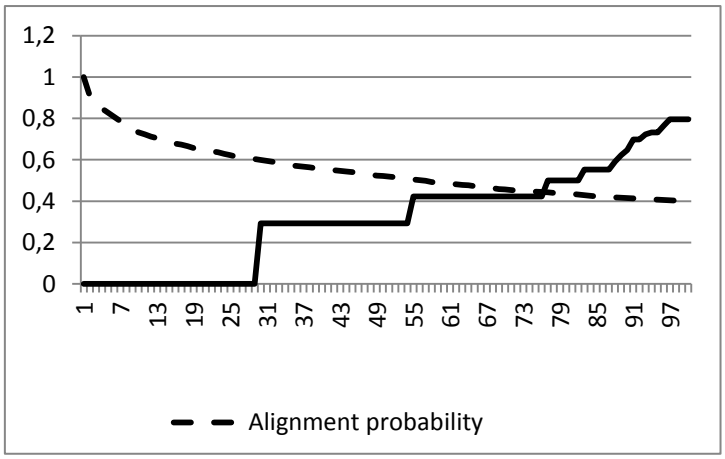

Figure 1: Comparability levels for a parallel corpus. M = N = 949, Avg. 1:n = 1.83, Avg. n:1 = 1.82

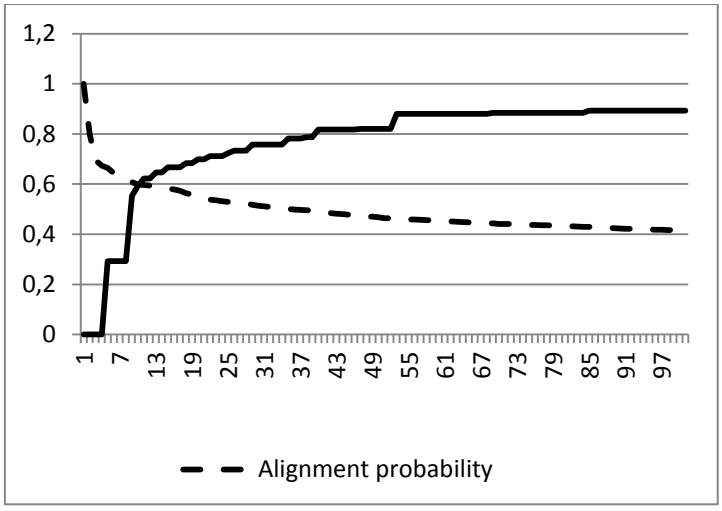

Figure 2: Comparability levels for a strongly comparable corpus. M = N = 1000, Avg. 1:n = 4.98, Avg. n:1 = 8.78

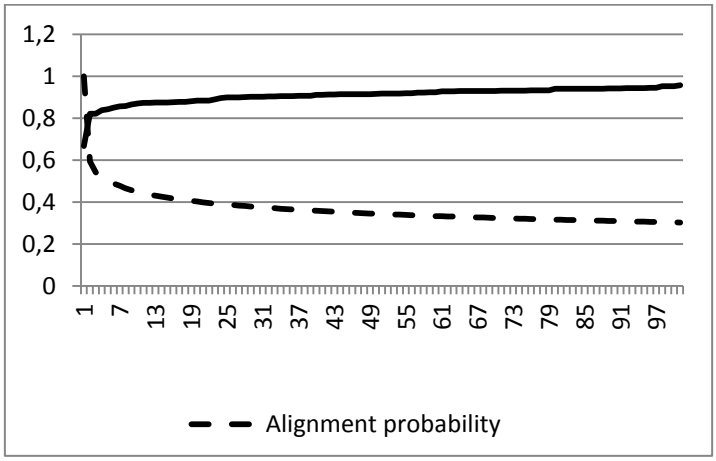

Figure 3: Comparability levels for a weakly comparable corpus. M = 5629, N = 11651, Avg. 1:n = 16.26, Avg. n:1 = 11.24

All the graphs in Figures 1-3 have been plotted by randomly selecting around 100 document pairs from the all the document pairs in the corpus. If we compare the graphs we find that:

- The alignment probability curve tends to skew when going from the parallel corpus to weakly comparable corpus: that is, fewer document pairs have high alignment probabilities. In the ideal case, when the document alignment probabilities are equal to 1 in for the parallel document pairs, we would have a straight line at y = 1;
- The document alignment productivity tends to become flat at y = 1 when going from the parallel corpus to the weakly comparable one. A value of the alignment productivity close to 1 indicates that the source document tends to align to a lot of target documents and/or the vice versa. For parallel corpora, the document alignment productivity is close to a straight line at y = 0.

Thus, plotting the document alignment probabilities and the document alignment productivity and interpreting the results along the lines presented in this section, one can rapidly get an idea of the comparability level of the given corpus.

4. Conclusions

We have presented a methodology of graphically identifying the comparability level of a given corpus. In order to be applied, this methodology requires that the studied comparable corpus is document aligned and that each document pair receives a "translation strength" probability that is high if the documents contain translated pieces of text and low in the opposite case. In this context, we have successfully validated our document alignment algorithm called EMACC in that its alignment probabilities and resulting document alignment productivities correlate well with our expectancies of their variations on different types of comparable corpora.

Acknowledgements. This work has been supported by the ACCURAT project (http://www.accurat-project.eu/) funded by the European Community's Seventh Framework Program (FP7/2007-2013) under the Grant Agreement n° 248347.

References

Ion, R., Ceaușu, A., Irimia, E. (2011a). An Expectation Maximization Algorithm for Textual Unit Alignment. *Proceedings of the 4th Workshop on Building and Using Comparable Corpora (BUCC 2011)*, Portland, USA, 24 June 2011, 128-135.

Ion, R., Tufiș, D., Boroș, T., Ceaușu, A., Ștefănescu, D. (2011b). On-Line Compilation of Comparable Corpora and their Evaluation. *Proceedings of the 7th International Conference Formal Approaches to South Slavic and Balkan Languages (FASSBL7)*, Croatian Language Technologies Society – Faculty of Humanities and Social Sciences, University of Zagreb, Dubrovnik, Croatia, 29-34, October 2010.

Kilgarriff, A. (2010). Comparable Corpora Within and Across Languages, Word Frequency Lists and the KELLY Project. *Proceedings of the 3rd Workshop on Building and Using Comparable Corpora (BUCC 2010)*, Valletta, Malta, 1-6.

Maia, B. (2003). What Are Comparable Corpora? In *Multilingual Corpora: Linguistic Requirements and Technical Perspectives*. *A Workshop on the Corpus Linguistics Conference*, Lancaster, UK.

Munteanu, D. S., and Marcu, D. (2005). Improving Machine Translation Performance by Exploiting Non-Parallel Corpora. *Computational Linguistics*, 31(4): 477–504.

Su, F., Babych, B., Paramita, M. and Gaizauskas, R. (2011). Evaluation and Elaboration of Metrics. ACCURAT Deliverable D1.3, version 1.0, December 31, 2011.

ROMANIAN DEEP NOUN PHRASE CHUNKING USING GRAPHICAL GRAMMAR STUDIO

RADU SIMIONESCU

University Al. I. Cuza Iaşi, Faculty of Computer Science– Romania

radu.simionescu@info.uaic.ro

Abstract

This paper introduces the Graphical Grammar Studio (GGS) software and describes a grammar that was built with it for deep Romanian noun phrase chunking. GGS is an open source java tool for NLP tasks which, unlike most grammar languages and matching tools, allows the graphical design of grammars, somehow resembling the old recursive transition networks. Such a grammar is usually used for finding sequences of words which respect certain conditions. GGS grammars can also be used to annotate the matched sequences. With GGS one can create complex grammars which can even work as standalone NLP tools. The second part of this paper presents a complex grammar which recursively detects and annotates noun phrase chunks for Romanian text.

1. Introduction

Graphical Grammar Studio (GGS) is a tool developed by the author and published as an open source project on SourceForge[1]. GGS offers the tools for creating and applying Recursive Transitional Networks (RTN). But unlike classic RTNs which consume one character at a time, GGS consumes one token at a time. For this reason GGS is a tool for finding and annotating sequences of tokens and it is somewhat oriented towards syntactic layer of NLP.

Like Finite State Automata (FSA), RTNs are recognizers (acceptors) of sentences generated by grammars. But, additionally, their states can be structured into subnetworks (or graphs of nodes), with the possibility to create jumps from one subnetwork to another. Recursive jumps can be created. Each graph / subnetwork has an initial and a final state and each RTN has a main graph. W. A. Woods claims that by adding a recursive mechanism to a finite state model, parsing can be achieved much more efficiently (Woods 1970).

A GGS grammar represents a RTN in a fashion which is more convenient for NLP tasks. Unlike classic RTNs whose arcs are labeled with terminal symbols (acceptance conditions) and their nodes with arbitrary symbols, GGS grammars only have their nodes labeled with acceptance conditions. The arcs have no labels in GGS representation. But one can easily consider the condition of a GGS arc to be the condition of the state that it points to.

[1] http://sourceforge.net/projects/ggs/

GGS grammars can generate output while matching an input text, thus creating annotations. A similar behavior in literature can be found in models called Finite State Transducers, which are classical Finite State Automata but which also generate an output while consuming an input. A GGS grammar could be thus considered a Recursive Transition Transducer. In one of his papers devoted to state machine techniques, Emmanuel Roche says: ”Finite-state transducers should appeal to the linguist looking for precise and natural description of complex syntactic structures[...] The parsing programs derived from this approach are both simple, precise linguistically and very efficient” (Roche, Parsing with finite state transducers 1997). In another paper, published in the same book, he describes an efficient method for deterministic part of speech tagging using transducers (Roche, Deterministic Part-of-Speech Tagging with Finite-State Transducers 1997).

2. Graphical Grammar Studio

GGS is a tool oriented towards syntactical analysis. GGS grammars are meant to consume sequences of tokens; a state can consume one token at a time. Each token can have an unlimited number of attributes, and a GGS grammar can relate to these to specify acceptance conditions for its nodes.

At its core, GGS is a tool very similar with Nooj (Silberztein 2004), but the latter is meant to become a comprehensive development environment for NLP tasks. Nooj is an improved version of the INTEX system (Silberztein, 1994). It has support for dictionaries, morphological grammars, paradigmatic representations etc. and can read an impressive number of input formats. It also provides means to create chains of grammars.

GGS on the other hand is best suited for matching and annotating sequences. It is a tool that can be easily integrated in various processing chains. It doesn’t have support for dictionaries, like Nooj, and it doesn’t require the presence of certain attributes for its input tokens. All token attributes are treated as key-value pairs which a GGS grammar can refer to, using regular expressions for both the keys and the values.

Being a tool specialized on matching and annotating sequences of tokens, GGS contains several features which Nooj lacks, like recursive depth and loop limits, or look ahead and look behind assertions. There is, though, a feature which GGS lacks in the present version, but which will be introduced with a future version: the possibility to define variables.

But before anything else, GGS is meant to be an open source project which can be used by anyone. Also, the Java platform might offer some technical advantages for some users, over the .NET platform.

GGS’s main component is the GGS Editor. A secondary component is a java library used for integrating the GGS engine in java code.

When a grammar is applied, the GGS engine compiles it first. This process transforms the RTN in a FST (Finite State Transducer) which consumes one token at a time, and which maintains a call stack of jumps between graphs. This conversion creates the possibility to efficiently check for inconsistencies in the grammar before actually running the state machine (e.g. infinite loops, left recursions).

At the moment, GGS accepts only xml input. It requires the name of the tags which represent text units (usually sentences), and expects to find sequences of token tags as the children of these xml nodes. A token tag can have an unlimited number of XML attributes.

Below is a sample of GGS input text.

```
<S>
<W LEMMA="hol" POS="NOUN" Type="common" Gender="masculine"
Number="singular" Definiteness="yes">Holul</W>
<W LEMMA="bloc" POS="NOUN" Type="common" Gender="masculine"
Number="singular" Definiteness="yes">blocului</W>
<W LEMMA="mirosi" POS="VERB" Type="main" Mood="indic."
Tense="imperfect" Person="third" Number="singular">mirosea</W>
<W LEMMA="a" POS="ADPOSITION" Type="preposition"
Formation="simple">a</W>
<W LEMMA="varză" POS="NOUN" Type="common" Gender="feminine"
Number="singular" Definiteness="no">varză</W>
<W LEMMA="călit" POS="ADJECTIVE" Type="qualificative"
Degree="positive" Gender="feminine" Number="singular"
Definiteness="no">călită</W>
<W Type="PERIOD" POS="PERIOD" LEMMA=".">.</W>
</S>
```

A node of a GGS graph can be one of three types:

- Token Matching Node
- Empty Node
- Jump Node

A **Token Matching Node** consumes the next input token only if the conditions that it imposes are met. **Empty Nodes** do not consume input tokens. They always match in the matching process and are used for organizing grammars and for other features of GGS. A **Jump Node** represents a transition to another graph.

Till the matching process reaches the final node of a grammar, a variable number of input tokens are consumed, representing the matched sequence. This sequence starts with the first token from the given input stream. When searching for multiple matches, some of which are starting in the middle of a sequence of tokens (sentence), GGS applies the matching process described repeatedly, each time offsetting the starting token by one index to the right.

In GGS, a matching / acceptance condition of a node is described as a sequence of key-value pairs which must be present or not in the attributes map of the next token from the input stream. Moreover, both the keys and the values can be specified using regular expressions, providing a great amount of flexibility. GGS comes with an user guide which contains many practical examples.

2.1. GGS features

To best describe the features of GGS, a few examples are provided. The main graph in Figure 1 matches all pair of words in which the first one is "the". The "<>" node matches any token because it doesn't impose any conditions.

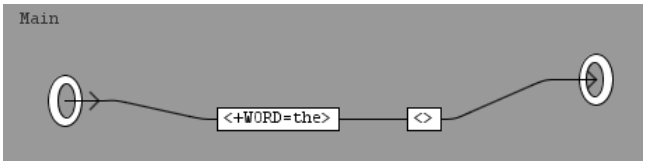

Figure 1: The main graph of a simple grammar

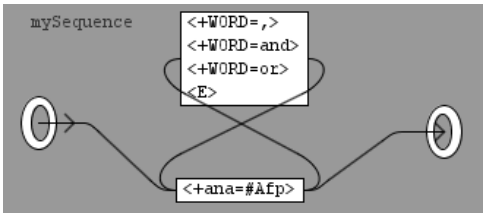

Figure 2: A secondary graph named "mySequence"

The graph in Figure 2 is named "mySequence" and matches sequences of adjectives (provided that ana=#Afp is the annotation for adjectives) which can be optionally separated by "or", "and" and comma. <E> stands for the empty node.

The main graph showed in Figure has two jumps to the "mySequence" graph, and matches adjectives sequences followed by nouns or nouns followed by the copulative verb "be" and adjectives.

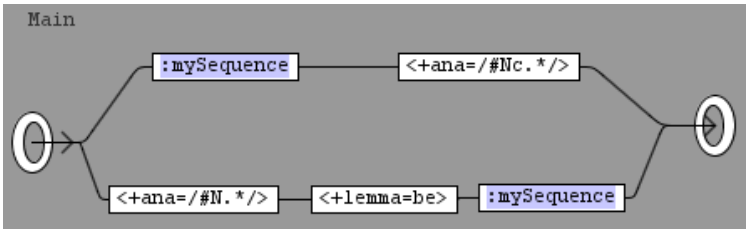

Figure 3: A main graph which contains jump nodes to mySequence

In the previous examples, the regex /#Nc.*/ is intended to match part of speech tags which stand for common noun. #Afp is intended for adjectives. GGS does not impose any particular tagsets for the annotations of the input tokens. The designer of the grammar is the one which establishes what attribute names and what tagsets is the grammar expecting to receive as input. If a grammar is applied on a text which is not annotated accordingly, then it will not behave as expected.

GGS grammars can also annotate matches. A GGS annotation marks a continuous sequence of tokens. It has a name, and it can contain an unlimited number of attributes. GGS nodes can be used to specify the start and end of annotations. When a path is matched, i.e. when the matching process reaches the finial node of the grammar, if the nodes of the parsed path contain annotation specifications, then these are written to the output. This output can then be serialized. The grammar in **Figure** annotates the sequences it matches.

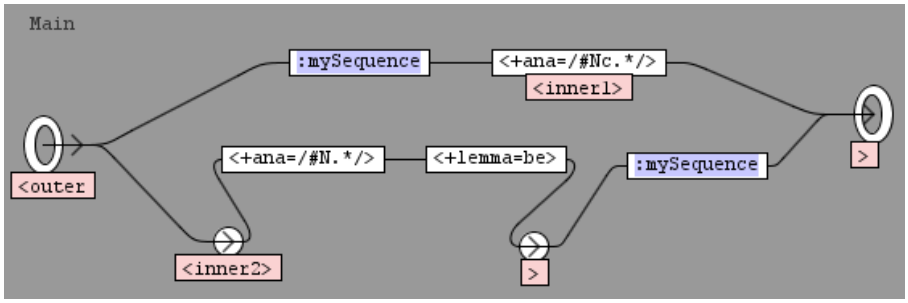

Figure 4: An example main graph containing annotation instructions

When the matching process reaches a node which leads to two or more possible paths, by default, the matching process will try to match the input by going on all possible paths, searching for the longest match. The user can set a priority on arcs emerging from the same node. If by going on a priority arc, the matching process manages to find a match (reaches the final node of the grammar), then the rest of the alternative arcs are ignored. This is very useful for creating efficient and precise recursive tools.

With GGS the user can also set recursive limits on certain nodes which play an important role in some recursive mechanism.

Inspired by the look behind and look ahead assertions from regular expressions, GGS is the first to offer an implementation of such a mechanism at the token level. An assertion acts as a node condition. The user can restrict the matching process from continuing after a certain node if a secondary grammar (assertion grammar) doesn't (or does) match the input text. Unlike classic look behind assertions from regular expressions, GGS look behind assertions are not restricted to only fix sized grammars (grammars which consume a constant number of tokens).

Obtaining certain behaviors like determining if a word is the first or the last one in a sentence can be achieved only by using assertions. For example, one can restrict a node from matching if there is a certain token present at a maximum token distance *d* to it's left, by using a negative look behind assertion.

3. Romanian prepositional and noun phrases

Noun phrase chunking (NP chunking) is a partial parsing which outputs the phrase structure of only the noun phrases, including usually some other internal phrasal constituent types, like prepositional phrases or verbal phrases (VP) (which both can contain other NPs and so on). Therefore a NP is recursive in structure. More rigorous formalizations support that the NP is even more complex, i.e. there are intermediate phrases such as Quantifier Phrase (QP) and Adjective Phrase (AP) (Abney 1987) which contribute to the structure of a NP.

In the context of this paper NPs can only contain PPs and other NPs. And PPs must contain at least a NP. Adjectives, articles, determiners etc are present in NPs, but not as separate constituents. For visual simplicity, the PPs will not be represented in the annotations, but only their inner NP will be visible as a child of another NP.

The general structure and chunking rules of NP is language dependent. Like in most highly flexional languages and rich in agreements, the order of the words is not very

strict in Romanian. The presence of the agreements makes it possible to have very complex NP structures.

A grammar which parses Romanian text in such a manner has been created successfully with GGS. This grammar requires part of speech tags and lemmas for the tokens given as input.

The grammar contains a structure of graphs which match different types of NPs. First, the NPs are classified into direct and oblique (Figure) cases based on their center noun. NP_direct and NP_oblic are each jumping to NPs centered around common nouns, proper nouns and pronouns (Figure). This manner of structuring the grammar allows to easily defining a different behavior for each case. For the cases of NPs centered around a direct noun, the problem is further fine grained into 4 different cases (all combinations of feminine/masculine and singular/plural) (Figure).

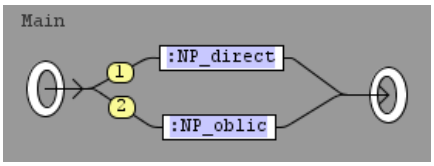

Figure 5: The main graph of the NP chunking grammar

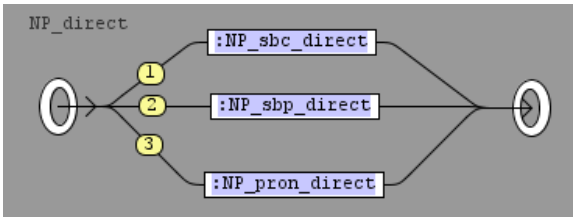

Figure 6: NPs are further categorized so they can be solved by different graphs

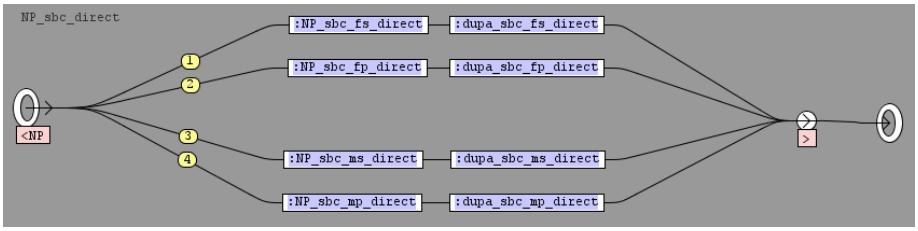

Figure 7: Different paths for solving with various post modifiers differently

There are thus 16 cases of noun based NPs for which the grammar is behaving differently. The number of cases handled differently for pronoun centered NPs is 28. This is due to various types of pronoun that can constitute NPs (in this grammar) each having its own particular behavior in Romanian: the personal, demonstrative, possessive, indefinite and negative pronoun.

For most of the parallel paths in the grammar there are priorities set up. This is for efficiency. Once the GGS matching process matches a type of NP it is useless to have it search for some other longer matches (the default priority policy finds the longest match which implies testing all the possible matching paths, in a depth first manner). Even though the grammar handles both pronoun and noun centered NPs, this paper focuses on the latter.

In Figure each type of NP is handled by two iterative nodes. The first column is composed of jumps to graphs which match the noun of the NP and eventual pre modifiers (Figure). The nodes from the second column (Figure) jump to graphs which match post modifiers. These can contain other NPs; recursive jumps are thereby present. These graphs are quite complex. They were designed to successfully annotate test cases which cover various linguistic phenomena of interest for the problem of NP chunking. Figure shows the graph which matches post modifiers for NPs centered on a feminine, singular, direct case, common nouns. To reduce the number of visual elements, the path priorities are not visible in this picture.

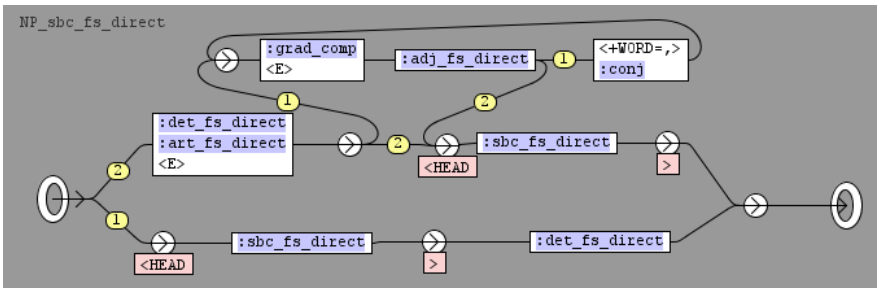

Figure 8: The graph which matches nouns with eventual premodifiers

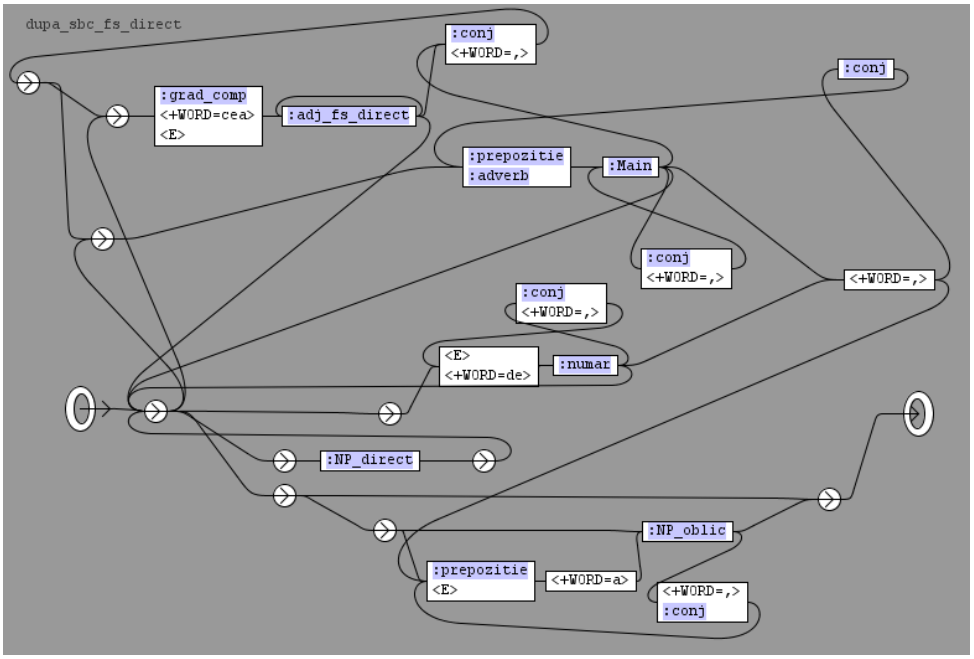

Figure 9: The graph which matches the post modifying sequence for feminine, singular, direct case, common nouns

Examples of linguistic phenomena covered by the post modifiers matching graph:

- A post modifying structure of a noun can be composed of a sequence of adjectival and prepositional phrase post modifiers, in any order.
- The possessive post modifier (matched by the bottom most branches) can be only the last in a sequence of post modifiers.

The grammar contains in total 122 graphs and 1,288 nodes from which 212 are token matching nodes and 466 are jump nodes. It contains in total 1,621 arcs. The grammar

has been applied on the 1984 corpus (6,726 sentences, 118,334 tokens) created in the Multext East Project[2] (Erjavec, 2004) and matched 28,308 NPs. Unfortunately, due to the lack of a rigorous testing corpus, an evaluation score can't be presented for now.

4. Implication of semantic information in NP chunking

There is an empirical rule which is obeyed by the grammar; i.e., in the case of a recursive NP sequence which ends with a post modifier, it is considered that this modifies the rightmost NP possible (in case of adjectival post modifiers, grammatical agreement must be satisfied). In the case of a prepositional phrase modifier it will be considered to modify the rightmost NP, because no grammatical agreement must be satisfied. This rule usually works well and seems to be the default manner in which the Romanian speaker understands NP post modifiers. Yet there are exceptions which suggest that human generation and parsing of NP structure also involves a semantic understanding.

The constructed grammar considers only morphological information but this is not sufficient for solving cases where semantic information is involved. The following example illustrates this in English:

- [Beds for [children with [iron legs]]] – incorrect (the output of the NP chunker)
- [Beds for [children] with [iron legs]] - correct

An idea is suggested towards making the grammar correctly solve this type of confusion: just like the adjectival post modifier is considered to modify the rightmost NP which satisfies *grammatical agreement*, in the same manner all post modifiers should be considered to modify the rightmost NPs which also satisfy a *semantic agreement*. The manner in which such a *semantic agreement* would be formalized or the detail of an actual implementation, remain the subject of future research.

GGS would still be a great solution for implementing such rules which also consider semantic agreement. A preprocessing module would be required to annotate the input with semantic annotation which would serve this purpose.

Another semantic implication was identified by observing failed cases of the NP chunker in which a prepositional phrase is actually modifying a verb, but because it is positioned immediately to the right of another NP (which usually has the role of object or complement) the NP chunker considers it a post modifier of this. Below is an example of such confusion illustrated in English.

- He greeted [the man with [the hammer]] – correct
- He broke [the stone with [the hammer]] – incorrect

The semantic implication is obvious and is deeply infiltrated in the logic of correctly parsing the second sentence. This simple example suggests that semantic information about the verbs, the prepositions of their objects and even the syntactic category of these might be required for correctly parsing the example.

[2] http://nl.ijs.si/ME

5. Conclusions

The first part of this paper presented recursive transitional networks to introduce Graphical Grammar Studio as a NLP tool for finding and annotating sequences of tokens which can easily be integrated in processing chains.The second part presents the details behind writing a complex grammar in general and more specifically creating rules for a Romanian NP chunker.

The NP chunking grammar created is quite impressive. All the 16 parallel noun centered NP matching graphs are very similar in structure, with only a few differences meant to check for agreement with various post and pre modifiers. The question would naturally rise: "wouldn't it be easier to have less such graphs and use some sort of parameters to check for agreement between different tokens attributes (gender, number etc.)?". The answer is yes; this will be possible using variables once they will be implemented in GGS. Another obvious question might be: "Nooj has support for variables, why not use it instead?" Unfortunately, Nooj doesn't provide the possibility to define the visibility of variables. All variables are global. Any value modification of any variable from a recursive level is visible from all its superior recursion levels.

Finally, a conclusion on the involvement of semantic understanding of language when it comes to NP chunking is underlined: by creating a morphologic based NP chunker and by observing the failed cases, situations of semantic information implication can be extracted.

References

Abney, S. (1987). The English Noun Phrase in its Sentential Aspect. *PhD Thesis*, MIT.

Erjavec, T. (2004). MULTEXT-East Version 3: Multilingual Morphosyntactic Specifications, Lexicons and Corpora. In Proc. of the Fourth Intl. Conf. on Language Resources and Evaluation, LREC'2004, ELRA.

Roche, E. (1997). Deterministic Part-of-Speech Tagging with Finite-State Transducers. E. Roche and Y. Schabes (eds.), *Finite-State Language Processing*, Cambridge, Mass./London, The MIT Press.

Roche, E. (1997). Parsing with finite state transducers. *E. Roche and Y. Schabes (eds.), Finite-State Language Processing*, Cambridge, Mass./London, The MIT Press, 241-281.

Silberztein, M. (1994). INTEX: a corpus processing system. *Kyoto, Japan: COLING 94 Proceedings*.

Silberztein, M. (2004). NooJ: an Object-Oriented Approach. INTEX pour la Linguistique et le Traitement Automatique des Langues. *C. Muller, J. Royauté M. Silberztein Eds, Cahiers de la MSH Ledoux*, Presses Universitaires de Franche-Comté, 359-369.

Woods, W. A. (1970). Transition network grammars for natural language analysis. *Communications of the ACM* 13 (10), 591-606.

CHAPTER 4
RESOUCES AND APPLICATIONS IN LEXICOGRAPHY

DERIVATIONAL-SEMANTIC NETWORK FOR ROMANIAN

VERGINICA BARBU MITITELU

Romanian Academy Research Institute for Artificial Intelligence

vergi@racai.ro

Abstract

The Romanian wordnet is quite a rich resource from more perspectives: the number of synsets, the information contained (not only glosses, relations between synsets, but also domain information and subjectivity annotation). However, it can still be enriched and improved in various ways. One of them, the adding of derivational relations, which is the focus of our presentation and was also considered by other teams developing wordnets for their languages as a means of quantitative or/and quantitative enrichment of their resources.

1. Introduction

The Romanian WordNet (RoWN), a resource under continuous development for more than 10 years already, is also the most valuable one at the Research Institute for Artificial Intelligence, thus used in many applications developed here. As a consequence, our concern for its enrichment with information that would make it more valuable for use in various tasks is understandable.

We begin this paper with a presentation of some quantitative data about the current status of our wordnet (section 2), then we describe our objectives (section 3), the derivational relations (section 4), the related work (section 5), the resources we use (section 6.1), the methodology we intend to follow (section 6.2), further work (6.3) and we reach conclusions afterwards.

2. The current status of the Romanian WordNet

Started in the BalkaNet project (2001-2004) and developed by two teams of specialists from Iasi (Faculty of Computer Science) and Bucharest (Research Institute for Artificial Intelligence), the RoWN has been under ceaseless development at the latter Institute. The merge approach adopted from the beginning is still our way of creating this resource and we still observe the two principles established for choosing the concepts to be implemented: the hierarchy preservation principle and the conceptual density principle. Details about the work strategy adopted can be found in Tufiș (2004).

At present, the RoWN has the following quantitative characteristics:

Table 1: Quantitative data about RoWN

No. of synsets	57895
No. of literals	51986
No. of senses	83860
No. of relations	120198

The approach adopted implied the transfer of relations from Princeton WordNet (PWN) (Fellbaum, 1998) into RoWN, according to the hierarchy preservation principle. We mention that PWN also contains derivational relations. However, due to their characteristics, which will be discussed below, we chose not to transfer them, but to add them internally.

3. Objectives

The main objective of the project presented here is to add derivational relations into the RoWN, alongside with semantic information in the form of semantic labels. Relying on these, linguistic studies on affixes, on lexical families, on derivation are possible, with plenty of examples.

4. Characteristics of derivational relations

At this point it is necessary to clarify what derivational relations are. They establish between two words, provided that one is obtainable from the other by adding a suffix or/and prefix to it. We say "obtainable", not "obtained" because we will treat similarly words that are analyzable borrowings and words created within Romanian. We should keep in mind that an affix entered our language by means of a set of borrowings containing it. After some time, the speakers became able to analyze the borrowings (i.e. detach the components: stem and one or more affixes) by linking them semantically to other words existing in use and, furthermore, they started to attach the respective affix to other words, thus being creative and enriching their language.

Examples of derivational relations include:

- *dormitor* "bedroom" – *dormi* "sleep": the former is derived from the latter by means of the suffix *-tor*;
- *extraurban* "outside the town" – *urban* "urban": the former is derived from the latter by means of the prefix *extra-*;
- *îmbuna* "soften up" – *bun* "good": the former is derived from the latter by means of the prefix *în-* and of the suffix *-a*.

So, derivational relations are relations between an affixed word and its stem, where by stem we understand a root or a root to which (an)other derivational affix(es) has/have been added previously. All the three examples above contain a stem that is also a root. However, a derived word like *împădurire* is obtained from the root *împăduri* by adding to it the suffix *-re*. But the root *împăduri* is not a stem; it is also a derived word from the stem *pădur* by means of the prefix *îm-* (phonetic version of *în-*) and the suffix *-i*.

Such relations are established between words, so they are valid within a language. Moreover, they are established at the word sense level, not at the level of words as clusters of meanings. Let us consider the following pair: the verb *visa* and the adjective *visător*. The former has three meanings: 1. to have a dream while sleeping; 2. to daydream; 3. to long for something. The latter has one proper meaning: who tends to daydream. (Another meaning, derived from this, is used for people lacking common sense.) Due to their semantics, it is reasonable to analyze the adjective derived from the verb considered in its second meaning. This could be represented as below:

1. *"to have a dream while sleeping"*

visa 2. *"to daydream"* <----> *"who tends to daydream" visător*

3. *"to long for sth."*

At a monolingual level, an affix can be polysemous. Attached at different stems, it can help create words with different semantics: for example, the suffix *-tor* behaves as follows:

dormi + -tor > dormitor (place)

vopsi + -tor > vopsitor (job)

transporta + -tor > transportator (company)

suci + -tor > sucitor (instrument)

At the same time, affixes can be synonymous: they help create words belonging to the same semantic class (co-hyponyms): for example, in order to create words designating jobs in Romanian, we use various suffixes: *-tor*, *-ăreț*, *-uș*, *-ant*, etc.

vopsi + -tor > vopsitor (job)

cânta + -ăreț > cântăreț (job)

căra + -uș > cărăuș (job)

manevra + -ant > manevrant (job)

At a multilingual level, it is interesting to notice that various languages appeal to an equivalent affix to render a certain semantic relation. If we consider the relation between the verb *dream* and the noun *dreamer*, we notice similar pairs in many other languages:

EN dream-dreamer

RO visa-visător

FR rêver-rêveur

BG mechta-mechtatel

DE träumen-Träumer

IT sognare-sognatore

However, if we deal with the pairs below:

EN cook-kitchen

RO bucătar-bucătărie

FR cuisinier-cuisine

BG gotvach-kuhnya

DE Koch-Küche

IT cuòco-cucina

we notice that: in some languages the relation between the words in focus is marked morphologically by a suffix (in Romanian and French), in others words belong to the same word family, although they were not derived, but inherited from older stages or languages (German and Italian), while in others (English and Bulgarian), there is no morphological relation between the words. Moreover, the situation is not similar in Romanian and French, as in the former the name of the place is derived from the name

of the person, while in French the name of the person is derived via suffixation from the place where this person works.

5. Related projects

The richer the information contained in a wordnet, the more valuable the linguistic resource. Thus, many teams developing wordnets undertook the task of adding derivational information to their semantic network.

The Princeton team (Fellbaum et al., 2007) adopted a semi-automatic procedure for identifying, marking and labeling derivational relations. That is, for a certain affix, all pairs of the type stem-derived were automatically extracted from the wordnet. Afterwards, they were subject to a manual validation and grouping of the valid pairs according to the semantics of the affix. In this step, the rich polysemy and synonymy of affixes were noticed. The morpho-semantic links were marked at synset level (so between word senses) and got a semantic label, as well, which is also syntactically motivated, whenever possible. The morpho-semantic relations are marked for pairs of synsets, out of which one is a verb, and the other one a noun. These are: *agent*, *material*, *instrument*, *location*, *by-means-of*, *undergoer*, *property*, *result*, *state*, *uses*, *destination*, *event*, *body-part*, *vehicle*. They are available in a standoff file at the address http://wordnet.princeton.edu/wordnet/download/standoff/.

A remark is worth being made at this point. PWN contains other derivational relations, too. These are available in the downloadable file. But, unlike the one mentioned above, they lack a semantic label. Thus, all wordnets developed by transferring the PWN relations contain such derivational relations, although they may not be valid within the respective languages, because of the language specificity of these relations.

The Czech team (Pala and Hlavackova, 2007) automatically generated derived forms from known stems by means of certain suffixes. (They did not deal with prefixes.) Such a process creates both "possible and existent" forms and "possible but inexistent" ones. That is why, manual validation follows as a necessary step. Using ten derivational relations (deriv-na, deriv-ger, deriv-dvrb, deriv-pos, deriv-pas, deriv-aad, deriv-an, deriv-g, deriv-ag, deriv-dem), they created "sub-nets" or "derivational nests" containing, in fact, words from the same lexical family. An important issue brought forth in the presentation of their work is the non-directionality of these links, due to the fact that the direction of the derivation is not always clear.

For Bulgarian (Koeva, 2008) and Serbian (Koeva et al., 2008) the developing teams reported the transfer of the derivational relations available in the PWN at the level of the synsets in their own resources. However, applying manual validation proved that they are not always valid in the target languages, thus they added a note containing this information at the synset level.

For Turkish (Bilgin et al., 2004), in which derivation is extremely productive and predictable, an automatic process in which, by means of regular expressions and relying on a list of productive affixes, pairs stem-derived are found and then validated. It is the lexicographer's task to decide among which synsets containing those literals a derivational relation exists and what semantic label it has. The possible semantic labels used are: *become*, *acquire*, *be-in-state*, *someone-with*, *something-with*, *someone-from*,

someone-without, something-without, pertains-to, with, reciprocal, causes, is-caused-by, cat-of, manner.

For the Estonian wordnet (Kahusk et al., 2010) a small size automatic experiment is presented, using two very productive suffixes that are added to stems for creating derived words. However, they need manual validation to decide about possible but inexistent derived words, abstract senses and metaphors.

The Polish wordnet is reported (Piasecki et al., 2009) to contain two types of derivational relations (*pertainimy* and *related-to*), both transferred from PWN.

The approaches presented here are of two types: one is a way of enriching a mature wordnet qualitatively, with derivational and semantic information (see the PWN experiment), and the other one is a way of enriching wordnets under development both quantitatively and qualitatively, with synsets that are primarily derivationally motivated (see the Czech, the Estonian, etc. experiments).

6. The Romanian project

6.1. Resources

For marking derivational relations in the RoWN, we intend to make use of a list of Romanian affixes that can help us to automatically find pairs stem-derived word in the RoWN. The list of affixes was created on the basis of the rich bibliography dedicated to derivation in Romanian linguistics. The main papers from which we extracted the affixes are: *Formarea cuvintelor în limba română* (1970, 1978, 1989), Pascu (1916), Coteanu (2007), Philippide (2011), Tudose (1978), Iordan (1939), *Studii și materiale despre formarea cuvintelor în limba română* (1959, 1967, 1969). As in wordnet there are only words belonging to open parts of speech, when we created our lists we focused only on affixes specific to nouns, verbs, adjectives and adverbs. These are, in fact, most of the affixes irrespective of the language. For Romanian we found 492 affixes: 83 prefixes and 409 suffixes.

A part of the found stem-derived word pairs can be validated automatically by means of the etymological tags in our electronic version of the explanatory dictionary of Romanian. However, we will go beyond this, because we want to link words that are borrowed but are analyzable in Romanian (that is, there are a stem and an affix into which we can say that the borrowed word can be split) to the stem they contain. For example, *veselie* (En. cheerfulness) is a Slavic borrowing, just like *vesel* (En. cheerful). But we will link these two words, because the former can be analyzed as made up of the adjective *vesel* and the suffix *-ie*, as this pattern of derivation is common in Romanian (an example would be the noun *hărnicie* (En. diligence) derived from the adjective *harnic* (En. diligent) by means of the suffix *-ie*; a common vowel mutation also occurs: *a:ă*).

6.2. *Methodology*

In order to accomplish our aims we start by automatically identifying in the RoWN pairs of the type stem-derived word. We make use of the list of Romanian affixes mentioned above. We are not interested in roots (the minimal analysable and

meaningful unit of a word), but in stems (root plus any derivational affix attached to it), because we want to identify the whole derivational process. For instance, we want to mark *reîmprospătare* (En. refreshment) as derived from *reîmprospăta* (En. refreshen) (by means of suffixation), derived in its turn from *împrospăta* (En. freshen) (by means of prefixation), which, in its turn, is derived from *proaspăt* (En. fresh) by prefixation. So, there are direct derivational links between stems and the words derived from them, but there are also indirect derivational links, like the one between *reîmprospătare* and *proaspăt*, which is constructed from the direct derivational links. Choosing this work method is the most appropriate for the derivational process that takes place in steps: affixes are usually attached one after the other.

We choose not to use directed links, because we aim at a similar treatment of both proper derivation and back formation (i.e. creation of a word by deleting a string of letters that can be analyzed as an affix of another word; e.g. *picta* (En. paint) is back formation from *pictor* (En. painter) and *pictură* (En. painting)).

Ignoring proper nouns and all compound literals in RoWN, we identified 2862 words derived by prefixation and 13556 by suffixation. We did not deal with words derived by means of a prefix and a suffix at the same time.

The next step is to semi-automatically validate the pairs extracted before, relying on subgroups of affixes created according to the part of speech of the stem and that of the derived word. Thus, the list of 13556 suffixed words was reduced to 9123. Usually, prefixes alone do not change the part of speech of the word they attach to. Out of the 2862 pairs of base-derived words only 2621 obeyed this constraint. However, we found a prefixed word that has a different part of speech than its root: *anticancer* (adjective) – *cancer* (noun). Other cases have also been reported in the literature (see Petic, 2011).

One more method for automatic validation is using the lemmatized glosses, relying on the assumption that words from the same lexical field can be defined with the help of derivationally related words. Manual validation must follow, anyway. We have not applied this validation method yet.

We calculated the precision and recall of the methods used for finding prefixed and suffixed words. Precision is the fraction of retrieved pairs that are relevant (i.e., are pairs of base-derived words). Recall is the fraction of relevant pairs that are retrieved. The data are included in Table 2 below.

Table 2: Precision and recall of finding derived words.

Affix	Precision	Recall
Prefix	70%	96%
Suffix	71%	89%

This means that in the case of prefixed words we are able to find almost all pairs of base-derived words (we miss only 4% of them) and we are less accurate in the case of suffixed words.

Precision is very difficult to improve: many false suffixes and prefixes cannot be spotted unless the semantics of the words is considered. Many short words (two or three

letters) can be recognized within plenty other longer words, either at their beginning or at their ending, without being their roots.

In fact, we are more interested in increasing recall, that is in automatically finding as many pairs base-derived words as possible. Searching through wordnet for such pairs is unconceivable.

6.3. Further work

All synsets containing the correct pairs are automatically extracted from the RoWN. As we want to mark the derivational relations at the word sense level, we intend to use other assumptions further on: derived word senses are used in the same domain, have a common hypernym and others. A manual validation will follow every time.

Whenever possible, the semantic label will be attached automatically, otherwise we will add it manually. One of the heuristics that will be used is that nouns derived with the suffix *-tor* and having person as (direct or indirect) hypernym are semantically labelled as Agents. At the moment we do not have a complete list of the semantic labels we intend to use. In fact, we consider that it is adjustable during the annotation and only at the end of it we can say it is fixed. However, further quantitative enrichment of RoWN may require adjusting this list.

We choose not to transfer the derivational relations from another wordnet for two important reasons. On the one hand, there is no complete coverage of such relations in any wordnet. On the other hand, by transferring them we might miss some relations, because a certain meaning expressed by means of derivation in one language can be expressed by morphologically unrelated words in another language (see above the example with *bucătar-bucătărie* and *cook-kitchen*).

7. Conclusions

There are three levels at which the importance of marking morpho-semantic relations is evident. First, at the monolingual level, the density of relations in a wordnet increases, between words with the same part of speech, but especially between words of different parts of speech. For example, the lexical family made up of *pădure* (forest), *pădurar* (forester), *pădurice* (grove), *păduros* (wooded), *împăduri* (afforest), *împădurire* (afforestation), *reîmpăduri* (reafforest), *reîmpădurire* (reafforestation), there are only two derivational links between words of the same POS (i.e. *pădure-pădurar* and *pădure-pădurice*), but there are four derivational links between words of different POSes (noun-verb: *pădure-împăduri*, *împădurire-împăduri*, *reîmpădurire-reîmpăduri*; noun-adjective: *pădure-păduros*).

Second, at the multilingual level, the semantic labels associated with the derivational relations are established at the synset level, so they hold among concepts and could be transferred from one wordnet into another, provided that they are aligned with each other. The more wordnets with such relations, the more numerous and interesting comparative studies can be done: one can analyse how a certain semantic relation is morphologically realized in various languages: if it has a morphologic counterpart or not, what affixes express it, etc.

Third, at the applications level, a wordnet enriched with morpho-semantic relations turns into a knowledge base useful for various tasks such as question answering, information retrieval and others.

The method described here will only deal with morpho-semantic relations between literals already in the RoWN. Our future work could be concerned with adapting our tools for marking these relations at the moment when new synsets are implemented in Romanian.

Acknowledgements. The work reported here is supported by the Sectorial Operational Programme Human Resources Development (SOP HRD), financed from the European Social Fund and by the Romanian Government under the contract number SOP HRD/89/1.5/S/59758.

References

Bilgin, O., Cetinoglu, O., Oflazer, K. (2004). Morphosemantic relations in and across wordnets: A study based on Turkish. In P. Sojka, K. Pala, P. Smrz, C. Fellbaum, P. Vossen (eds.), *Proceedings of GWC*.

Coteanu, I. (2007). Formarea cuvintelor în limba română: derivarea, compunerea, conversiunea. Narcisa Forăscu, Angela Bidu-Vrănceanu (eds.), *București: Editura Universității din București*.

Fellbaum, C. (ed.) (1998). WordNet. An Electronic Lexical Database. *Princeton Mass: MIT Press*.

Fellbaum, C., Osherson, A., Clark, P. E. (2007). Putting Semantics into WordNet's "Morphosemantic" Links. *Proceedings of the 3rd Language and Technology Conference*, Poznan.

Graur, Al., Avram, M. (eds.). (1970, 1978, 1989). Formarea cuvintelor în limba română. *București: Editura Academiei*, 3 vols.

Iordan, I. (1939). Sufixe românești de origine recentă (neologisme). *Buletinul Institutului de filologie română "Alexandru Philippide"*, VI, Iași, 1-59.

Kahusk, N., Kerner, K., Vider, K. (2010). Enriching Estonian WordNet with Derivations and Semantic Relations. *Proceeding of the 2010 conference on Human Language Technologies – The Baltic Perspective*.

Koeva, S. (2008). Derivational and Morphosemantic Relations in Bulgarian Wordnet. *Intelligent Information Systems*.

Koeva, S., Krstev, C., Vitas, D. (2008). Morpho-semantic Relations in Wordnet – a Case Study for two Slavic Languages. *Proceedings of the Fourth Global WordNet Conference*.

Pala, K., Hlavackova, D. (2007). Derivational relations in Czech Wordnet. *Proceedings of the Workshop on Balto-Slavonic*.

Pascu, G. (1916). Sufixele românești. *București: Librăriile Socec&Co*, C. Sfetea, Pavel Suru.

Petic, M. (2011). Automatizarea procesului de creare a resurselor lingvistice computaționale. *PhD Thesis*. Institutul de Matematică și Informatică al AȘM.

Philippide, Al. (2011). Istoria limbii române. *Iași: Editura Polirom*.

Piasecki, M., Szpakowicz, S., Broda, B. (2009). A Wordnet from the Ground up. *Wroclaw: Oficyna Wydawnicza Politechniki Wroclawskiej*.

(1959, 1967, 1969). Studii şi materiale privitoare la formarea cuvintelor în limba română. *Bucureşti: Editura Academiei*.

Tudose, C. (1978). Derivarea cu sufixe în româna populară. *Bucureşti: Editura Universităţii din Bucureşti*.

Tufiș, D. (ed.) (2004). Romanian Journal on Information Science and Technology. *Special Issue on BalkaNet*, 7, Romanian Academy.

CLRE. THE ESSENTIAL ROMANIAN LEXICOGRAPHIC CORPUS

ELENA TAMBA[1], MARIUS-RADU CLIM[1],
MĂDĂLIN PĂTRAȘCU[2], ANA CATANĂ-SPENCHIU[1], MARIUS RĂSCHIP[2]

[1] *"A. Philippide" Institute of Romanian Philology,*
The Romanian Academy, Iași Branch
[2] *Faculty of Computer Science, "Alexandru Ioan Cuza" University, Iași;*

isabelle.tamba@gmail.com; marius.clim@gmail.com;
madalin.patrascu@gmail.com; anaspenchiu@gmail.com

Abstract

This paper aims at highlighting the importance of creating a Romanian Essential Lexicography Corpus. The purpose of the project presented in this paper is the exploitation of certain results from the complex project which is eDTLR by using, as reference text for the alignment, the Thesaurus Dictionary in electronic format and by creating a Romanian lexicographic corpus, which will contain 100 dictionaries (since the 16th century until now) at the entry level.

1. Introduction

One of the specific elements of the globalization process which is currently fund in the society is represented by the capacity of sending and receiving information, without spatial and temporal barriers, the only limitation, only partially overcome, being the linguistic one. One of the objectives of the European policies is the preservation and the exploitation of the national linguistic identity, as long as there is a general tendency to use languages which are privileged by the existence of (electronic) promotion means. In Romania, the natural response was represented by measures that were taken have for creating electronic tools and resources that are necessary for supporting the Romanian language and culture on a cross border level, in the general context of computerization of the fundamental academic research.

Electronic dictionaries and texts corpora, structured as databases, are important for multiple reasons. On one hand, they facilitate the possibility of knowing, preserving and maintaining the cultural identity on a linguistic level and, on the other hand, they allow the inclusion of a national language in the field of digitalized research of natural languages, at a global level.

The Romanian academic specialists in linguistics and informatics, as well as in computational linguistics, have initiated research projects through which they want to exploit the non-digitized resources by transposing them in electronic format and to create new resources and tools for the automatic language processing. Thus, within a series of previous projects, the objective was to acquire, in electronic format, the *Thesaurus Dictionary of the Romanian Language* (14 tomes, 36 volumes, over 17.000 lexicon printed pages, having between 7000 and 11000 characters / page), which constitutes, in a synthetic form, the Romanian spirituality manifested in language, under all its aspects, from the first writings until the present time. That is why the creation of

an electronic type, which would be accessible to scientists and to all those interested in learning or studying the Romanian language, in our country or abroad, became, in the computerized multicultural society, an absolutely necessary step.

The main objectives of the complex project - *eDTLR Dicționarul tezaur al limbii române în format electronic* (2007-2010) – are the following: to achieve the complete form of the Romanian Language Dictionary - *Dicționarul limbii române* - in electronic format and to obtain a corpus which integrates all the texts part of the *Dictionary*'s Bibliography (scanned and text); to parse the DTLR and to obtain the semantic structure of each entry, results that will bring the possibility of complex consultations of the *Dictionary,* as well as editing and updating activities.

In this context, our project - CLRE. "Corpus lexicografic românesc esențial. 100 de dicționare din bibliografia DLR aliniate la nivel de intrare" (ERLC. Essential Romanian Lexicographic Corpus. 100 dictionaries from DLR Bibliography aligned by entries) - is a natural continuation of the projects related to the digitizing of the Romanian Language Dictionary and this also proves the capacity of exploitation of some results in the complex project eDTLR, project that has initiated a series of techniques and methodologies for the achievement and use of electronic data for the great Dictionary.

The research team is represented by three lexicographers (Elena Tamba, Marius-Radu Clim, Ana-Veronica Catană-Spenchiu) and two IT specialists (Marius Iulian Răschip, Mădălin Pătrașcu).

2. CLRE's objectives

The objectives of CLRE:

1. Obtaining a database which contains the essential dictionaries from the DLR Bibliography, at entry level.

2. Creating software which would allow the interactive consulting of this corpus that represents a modern work frame for the lexicographic research, easily adaptable for various objectives.

3. Achieving a quasi-exhaustive list of words, for the Romanian language, starting from the aligned corpus.

4. Promoting the project results and increasing the visibility of the results for the Romanian language, by the IT and linguistics specialists.

This project has the following purposes: achieving a scanned corpus, with the reference dictionaries of DLR (taking into account the current copyright legislation); scanning and processing these dictionaries (by OCR – optical character recognition – the conversion from image to text; parsing the text at entry); achieving an on-line interface for validating/correcting the text after parsing it (= automatic identification of the entries from previously scanned and converted dictionaries), as well as validating the alignment between the text of the *Romanian Language Thesaurus Dictionary* (in electronic format, from the eDTLR project) and the reference dictionaries from DLR Bibliography.

The CLRE project will include three types of specific activities: 1. elaborating techniques for digitizing the dictionaries of the DLR Bibliography, a software for identifying the fields of a dictionary entry, aligning and organizing them into a database, an interface which would allow the correction and searching through this aligned corpus – activities which would be carried out by the IT specialist; 2. lexicographic activities (transliteration of the title-words from the dictionaries written in Cyrillic and transition alphabets, validation for the final alignment); 3. disseminating the final product – activity carried out by all the researchers in the project.

In order to exemplify the types of dictionaries taken into account on this project, we selectively present some titles:

1. General dictionaries:

DA = *Dicționarul limbii române*, tom I-II, Tipografia ziarului "Universul", București, Imprimeria Națională, 1907-1944;

DLR = *Dicționarul limbii române*, Serie nouă, tom VI-XIV, București, Editura Academiei, 1965-2010;

DEX = *Dicționarul explicativ al limbii romane*. București, Editura Academiei, 1975;

DEXI = *Dicționarul explicativ ilustrat al limbii române*, Autori: Eugenia Dima, Doina Cobeț, Laura Manea, Elena Dănilă, Gabriela E. Dima, Andrei Dănilă, Luminița Botoșineanu, Chișinău, Editurile Arc și Gunivas, 2007;

MDA = *Micul dicționar academic*. Vol. I–IV. București, Editura Univers Enciclopedic. Volumul I: A–C (2001); volumul al II-lea: D–H (2002); volumul al III-lea: I–Pr (2003); volumul al IV-lea: Pr–Z (2003);

NDU = Ioan Oprea, Carmen-Gabriela Pamfil, Rodica Radu, Victoria Zăstroiu, *Noul dicționar universal al limbii române*. București – Chișinău, Litera Internațional, 2006.

2. Auxiliary dictionaries (which are strictly required for editing the Thesaurus Dictionary)

A. de Cihac, *Dictionnaire d'etymologie daco-romane*. Vol. I. *Elements latins, comparés avec les autres langues romanes*, Francfort A.-M., Ludolphe St. Goar; Berlin, A. Asher; Bucarest, Socec, 1870. Vol. II. *Elements slaves, magyars, turcs, grecs-moderne et albanais*, Francfort, Ludolphe St. Goar; Berlin, S. Calvary; București, Sotschek, 1879

Alexandru Ciorănescu, *Dicționarul etimologic al limbii române*. Ediție îngrijită și traducere din limba spaniolă de Tudora Sandru-Mehedinți și Magdalena Popescu Marin. București, Editura Saeculum I. O., 2002.

*** *Dicționarul ortografic, ortoepic și morfologic al limbii române*. Ediția a II-a revăzută și adăugită, București, Univers Enciclopedic, 2005.

Florin Marcu, *Noul dicționar de neologisme*. București, Editura Academiei Române, 1997.

3. Special dictionaries (encyclopaedic or special dictionaries chosen according to the importance for the diachronic perspective of the language):

Dicționar enciclopedic. [Vol.] I: A–C (1993), [vol.] II: D–G (1996), [vol.] III: H–K (2000), [vol.] IV: L–N (2001), [vol. V]: O–Q (2004). [vol.] VI: R–S (2006). [vol.] VII: T–Z (2009). București, Editura Enciclopedică;

I.-Aurel Candrea – Gh. Adamescu, *Dicționarul enciclopedic ilustrat. Partea I: Dicționarul limbii române din trecut și de astăzi* de I.-Aurel Candrea. *Partea II: Dicționarul istoric și geografic universal* de Gh. Adamescu. București, Editura Cartea Românească, [1926–1931];

Lexiconul tehnic român. I ș. u. Elaborare nouă. București, Editura Tehnică, 1957 ș. u.

3. Technical resources

In order to achieve the project objectives, it was necessary to use advanced equipment in order to facilitate the achievement of electronic dictionaries and also the software used in processing scans, character recognition which should allow the smooth implementation of the database. Further on, we shall detail some of the equipment and software used on the project.

At the beginning, a special scanner for books was purchased – Atiz Book DIY[1]. This proved to be the best solution for digitizing books, in terms of costs and efficiency.

The Atiz scanner has two Canon EOS 450D with 35 mm lens cameras. The EF 35mm lens allows a better focus and they are specially used for A3 or A2 book format.

BookDrive DIY has a great advantage to produce accurate images with no page curvature and the books are not damaged. The book is placed face up at 120°on the v-shaped cradle. Atiz scanner is available with two programs used for capturing and processing images. Thus, BookDrive Capture is the application that controls the cameras. It supports a wide range of Canon EOS SLR cameras and allows you to change camera settings directly from the software (e.g. shutter speed, aperture and ISO values). After scanning a book, the scanned images are converted by using another program, BookDrive Editor Pro. With this program, the scanned pages are processed and transformed in PDF (single-page or multi-page files), TIFF (LZW and CCITT Group 4, single-page or multi-page file) and JPEG formats fit for distribution or archiving and ideal for OCR text conversion. This program replaces an unwanted tinted page background common in old books with a bright and clear background free from speckles and ink stains. Other features include rotation, de-skew, crop, auto level, brightness and contrast adjustment, sharpen, black border removal, image resize and DPI adjustment and also the saving is made in different formats for each page or in folders for more pages.

For recognition accuracy and text conversion capabilities, an Abby Fine Reader Engine was purchased, which includes an optical character recognition program (OCR), an intelligent character recognition, an optical mark recognition (OMR), a barcode recognition (OBR), a document imaging and PDF conversion. This program turns scans, PDF documents and digital photographs into various searchable and editable

[1] Futher information about this product is available on the website http://diy.atiz.com/.

documents. We can add that this program has a conversion utility that instantly turns the different elements of formatting such as content, titles, footnotes, page number, headings into various electronic formats, including Microsoft Word, Excel and PDF, because of the ADRT (Adaptive Document Recognition Technology) for intelligent reconstruction of the logical structure and format documents. All this equipment and the computer software facilitate ensure the accurate processing of the lexicographical material in question.

3.1. CLRE – Information Index

In the first stage of the project, which involves aligning entry-level dictionaries, it is necessary to store the lexicographic resources in electronic format in order to establish connections. This operation was performed in three steps:

- Dictionary scanning with a vertical edge scanner, which gives extra quality results;
- Applying OCR (Optical Character Recognition) program on the obtained images. The process involves recognition of graphic signs and their electronic storing. For a more efficient version, the ENGINE Abby Fine Reader program has been used;
- Storing all this information in a database, both in image format and as alphanumeric structure.

3.2. CLRE – Entry identification

Due to the diversity of graphical formats of the dictionaries, a less traditional idea was engaged, which involved giving up writing the parsed versions. To solve this problem, we used machine learning concepts to extract definitions. This was possible by using clustering algorithms, applied according to a specified criteria for groups of dictionaries with similar graphic format. This way, geometrical shapes are built, which include the words that constitute a definition.

4. Conclusions

The present project continues and refines the new research methods in Romanian lexicography by offering, in addition of the results of eDTLR, a modern way for editing and updating of the great Romanian dictionary (DLR), the possibility of interactively consulting the dictionaries from DLR Bibliography by any Romanian or foreign philologist/linguist/lexicographer and, why not, by any Romanian language user.

Thus, this project proposes both classic / traditional linguistic methods (for example, transliterating the entries into Cyrillic or into the alphabet of transition or the comparative study of the dictionaries on the semantic level), as well as new, lexicographic-computational methods.

The project results and especially the elaboration of a corpus in which the alignment is done at an entry level will allow the development of vast applications regarding the semantic of words, entries selections with the purpose of elaborating new specialized dictionaries (etymologic, semantic etc.), the correlation with other linguistic or media

resources, fact that should take the Romanian lexicography at a level close to the European lexicography (see *Le rayon des dictionnaires*, http://www.atilf.fr/ – a collection of digitized French dictionaries, from the 16th to the 20th century or *Nuevo tesoro lexicográfico de la lengua española*, http://buscon.rae.es/ntlle/SrvltGUILoginNtlle – the database containing the facsimiled versions of all dictionaries edited and published by Real Academia Espagnola).

The result of the project will be available on-line.

The final result of this project is an essential Romanian Lexicographic Corpus, which will include an important number of essential Romanian language dictionaries, aligned according to their form and semantic, fact that will offer the Romanian specialists an excellent working tool and will set the basis for future research.

References

DA = *Dicționarul limbii române*, tom I-II, Tipografia ziarului "Universul", București, Imprimeria Națională, 1907-1944.

DLR = *Dicționarul limbii române*, Serie nouă, tom VI-XIV, București, Editura Academiei, 1965-2010.

DRAE = *Diccionario de la lengua espanola de la Real Academia Española* – http://buscon.rae.es/draeI/

TLFI = *Le Trésor de la Langue Française Informatisé* – http://atilf.atilf.fr/

TLIO = *Tesoro della lingua italiana delle origini* – http://tlio.ovi.cnr.it/TLIO/index2.html

OED = *Oxford English Dictionary* – http://www.oed.com/

DWB = *Deutsches Wörterbuch "der Grimm"* – http://germazope.uni-trier.de/Projects/DWB

ROMANIAN ASSOCIATIVE DICTIONARY

VICTORIA BOBICEV, VICTORIA MAXIM

Technical University of Moldova, Computers, Informatics and Microelectronics Faculty

victoria_bobicev@rol.md *maxivica@yahoo.com*

Abstract

The paper reports about an experiment of creation and development of an associative dictionary for the Romanian language. It outlines the first phase of the experiment when word associations were collected using questionnaire surveys. The second phase includes online interface creation and expanding the dictionary via internet. Several technical issues of the second phase are discussed. Some preliminary statistics are presented and a brief analysis of the obtained database is made. The created dictionary can be used in lexicography and studying Romanian language. At this stage of work we however are more interested in the richest and the most representative database of word association; the detailed analysis is postponed to forthcoming research.

1. Introduction

Semantics is proven to be the nuclear element of language. Lexical semantic networks are of great importance in Computational Linguistics of our days. The WordNet (Miller, 1995) wide popularity is the argument which proves the utility of semantic lexicons. One of the WordNet shortcomings is a small number of semantic relations. Other semantic lexicons like EuroWordNet[1] and Simple[2] were created to solve this problem. Semantic relations in these lexicons are well considered by the competent linguists and based on various lexical theories.

Our lexicon is created relying on different principles. The source of relations is the first main difference. Relations between words are obtained directly from the native speakers of the language as their free associations. The second main difference is the type of the involved relations. We do not name these relations or classify them; these are just relations of free associations in the human's mind. In psychology, free associations are the first words that come to the mind of a native speaker when he or she is presented with a stimulus word, presumably retrieved from associative memory (Nelson et al., 1998). The word presented to respondent is called "stimulus" and the word that comes to the mind is called "response". This type of relations is being studied in various domains of research, such as psychology, artificial intelligence, computational linguistics and natural language processing.

Associative dictionary is a collection of the word pairs "stimulus - response" and represents the language in a somehow unusual form - not in the form of continuous text, as in a novel or newspaper article, not in the form of a systematic description, as in

[1] http://www.illc.uva.nl/EuroWordNet/
[2] http://www.ub.edu/gilcub/SIMPLE/reports/simple/Site_simple.htm

grammar or dictionary, but as pair (combination) of words or word groups that serve as building material for the detailed phrases for constructing sentences.

Thus, any word in our minds, in memory, just like in the speech chain, does not exist in isolation. Any word requires a kind of "extension", looking for its pair, wants to become "a model of two words." And such possible "extensions", such models of two words - typical, easily reproducible, believable and understandable to native speaker - are recorded in the associative dictionary. In addition, each pair of stimulus-response is not a complete sentence, but a necessary component of it – it is either a grammatically formalized part, or only the core of a future statement which will be given the completed form (Уфимцева 2004).

The paper reports on an experiment of the word's associations for Romanian database creation. It outlines the first phase of the experiment while word associations were collected using questionnaire surveys. Next, the second phase is described which include online interface creation and augmentation of the dictionary via internet. Several technical issues of the second phase are discussed. Some preliminary statistics is presented and a brief analysis of the obtained database is made. At this stage of the work, we are interested in the richest and the most representative database of word association; the detailed analysis is postponed for the upcoming research.

2. Related work

There are a number of semantic lexicons with a various relations between words. The most popular is WordNet which contain a relatively small number of relations; it is considered one of its disadvantages. EuroWordNet authors revised and enlarged this set of relations. Simple uses Qualia structure theory as a source of semantic relations in lexicon (Pustejovsky, 2010). The attempt to code as much relations as possible has its negative effect; these lexicons are difficult to process. Fairly sophisticated algorithms are required to obtain the necessary information in a plausible time.

Knowledge bases are the other types of semantic information sources. Well-known CYC[3] include the lexicon as part of the knowledge base. Words in the lexicon are connected with knowledge base concepts thus obtaining semantic capacity. The number of concepts and relations is one of the largest among various resources of this kind. On the contrary, ConceptNet[4] describes only 20 types of relations; some of them are similar with other resources. It is the only resource which is created not by skilled linguists but by volunteers via online interface. This method of knowledge acquisition has several advantages: no need of professional linguists with special training, which leads to less cost and higher growing rate.

Associations between words are obtained also from people without any Special knowledge of linguistics; the only demand is that they should be native speakers of language. Though word association experiments are a usual psychological practice, the obtained results are of great interest in various domains of research, as for example in cognitive science. The most important among these is the understanding that the association is a fundamental mechanism underlying human knowledge (Cramer, 1968,

[3] http://www.cyc.com/
[4] http://conceptnet5.media.mit.edu/

Dees, 1965). This notion is compatible with a number of statements in the field of natural language processing research such as the notion of mutual information (Church and Hank, 1990) as a measure of the importance of an association between two words (Hirst, 2004) and confirmation of the fact that a lexicon can often be a useful basis for the creation of practical ontologies. Lexical networks, represented by lexical nodes (Collins, Loftus, 1975) are the basic points of many connection patterns of human thoughts.

Recently, word associations have being studied by a number of researchers in the domain of cognitive science (Nelson et al., 2005; Steyvers et al., 2004). All these studies use The University of South Florida *Word Association, Rhyme, and word fragment Norms* (Nelson et al., 1998), which is the largest database of American English words associations, comprising nearly 5,000 words and a response average of 149 for each word collected from more than 6000 participants during the years 1975-2000.

There are various sources of word associations for different languages. The already mentioned largest database of word association for English[5]. We should also mention Edinburgh Associative Thesaurus (Kiss et al, 1973) freely available database for English[6]. Among the resources for other languages the Russian associative dictionary (Караулов, 2003), Bulgarian associative dictionary (Балтова 2003), the integrated Slavic dictionary (Уфимцева, 2004) are to be referred. All these resource were collected manually using questionnaire surveys. The more recent resources have been creating using online interface are the Large-Scale Database of Japanese Word Associations (Joyce, 2005), French associative dictionary[7], word association game for English[8], online interface for Russian associative dictionary[9].

3. The first step of Romanian word associations database creation

The first collection of Romanian word associations was created by the direct interrogation. 150 stimulating words were selected from the list of the most frequent Romanian words. The frequency list was created for the corpus described in (Vlad, 2005). The corpus was created on the base of 93 books of various genres: Romanian and foreign fiction, religious literature, philosophy, medical texts, history, law, and others. The authors' aim was to include in the corpus as much types of literature as it was possible. The corpus overall volume is 8.8 million words; the corpus frequency dictionary consists of more than 200 000 words. It is well known that the most frequent words in text are so called "stop-words": articles, prepositions, conjunctions, pronouns and some others which do not carry much semantic information and are used for syntactically correct sentences formation. We obviously were not interested in these words; we selected the most frequent 50 nouns, 50 adjectives and 50 verbs. This list of 150 words arranged in the first column of a table was presented to respondents. They

[5] freely available at http://w3.usf.edu/FreeAssociation
[6] http://www.eat.rl.ac.uk/
[7] http://dictaverf.nsu.ru/fr
[8] http://wordassociation.org
[9] http://tesaurus.ru/dict/dict.php

had to write in the second column of the table the word they were associating in mind while reading the word from the first column of the table.

The respondents were 50 students aged between 19-21 years. Each of them was given a MSWord document with the described above table and they completed the second column of the table. We were interested in the statistical results and the inquiries were anonymous.

Table 1: The strongest associations from the Romanian word association database.

Stimulating word	Association	Number of respondents	Number of respondents providing this association
forța	putere	50	29
ciudat	straniu	50	22
ceas	timp	50	21
noaptea	întuneric	50	21
bucurie	fericire	50	18
istoria	trecut	50	18
târziu	noapte	50	18
moment	clipa	50	17
nevoie	necesitate	50	17
bucătărie	mîncare	50	15
frig	iarna	50	15
piatra	tare	50	15

The obtained data was analyzed using a Perl script. Our main goal was to find the most frequent associations for each word so we calculated the number of times the same association was written for the word. For example, for the word “bucurie” (joy) 18 of 50 respondents indicated “fericire” (happiness), 7 respondents indicated “zâmbet” (smile), 6 respondents indicated “veselie” (fun), other associations were different and had frequency less than 3. Thus the strongest associations for the word “bucurie” (joy) were “fericire” (happiness), “zâmbet” (smile) and “veselie” (fun). We preserved all the associations provided even those with the frequency equal to one keeping in mind the aim to enlarge our associative dictionary.

The overall results are presented in the table 1 and figures 1, 2. Table 1 contains 12 most frequent pairs of stimulating word and associated word. For example, the pair “forța - putere” ("force-power") has the highest frequency: 29 respondents provided this association. In general a great number of associations were synonyms or near-synonyms (9). Even if the association was not synonym as in example “bucătărie - mîncare” (kitchen - food) the association in most cases the same part of speech as the stimulating word. There is a small number of exceptions as, for example, “piatra - tare” (stone - hard).

4. The second step of creating the Romanian word association database

After the first phase of the dictionary creation, we had 150 words-stimulus and 50 responses for each of these words. This information was organised in MySQL database

which we intended to enlarge. In order to obtain more word-associations we created an online interface for our dictionary using PHP[10]. The interface is presented in the figure 3. It can be accessed on http://lilu.fcim.utm.md/asociere/.

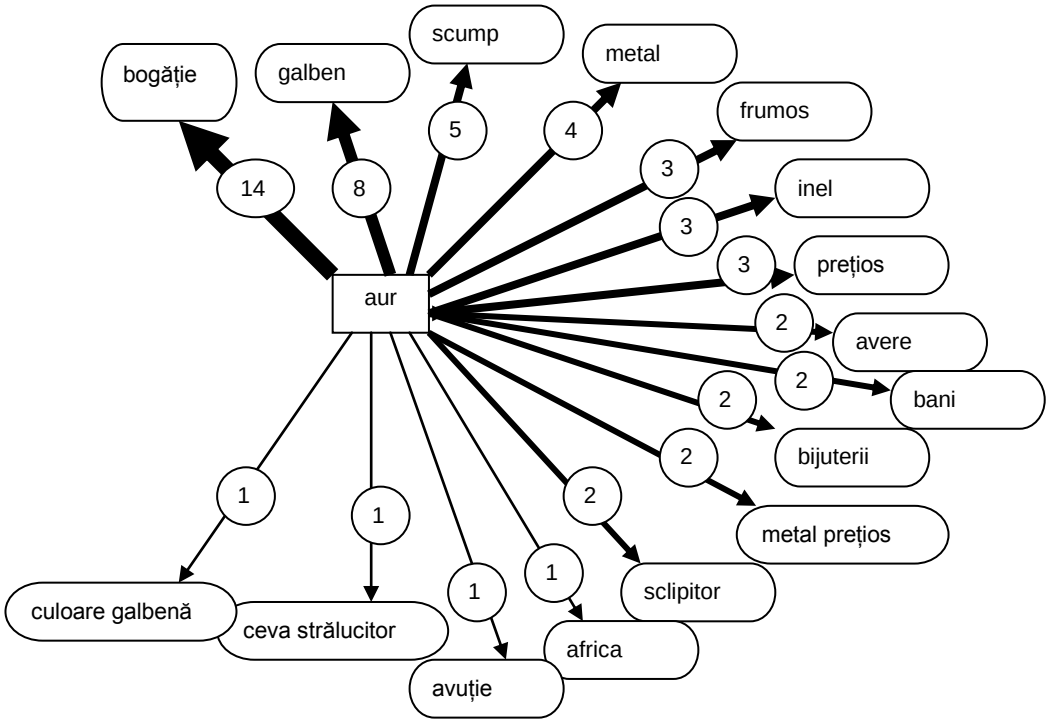

Figure 1: The set associated to the word "gold" that contains 16 associations. The figures on the arrows show the number of respondents who gave this answer.

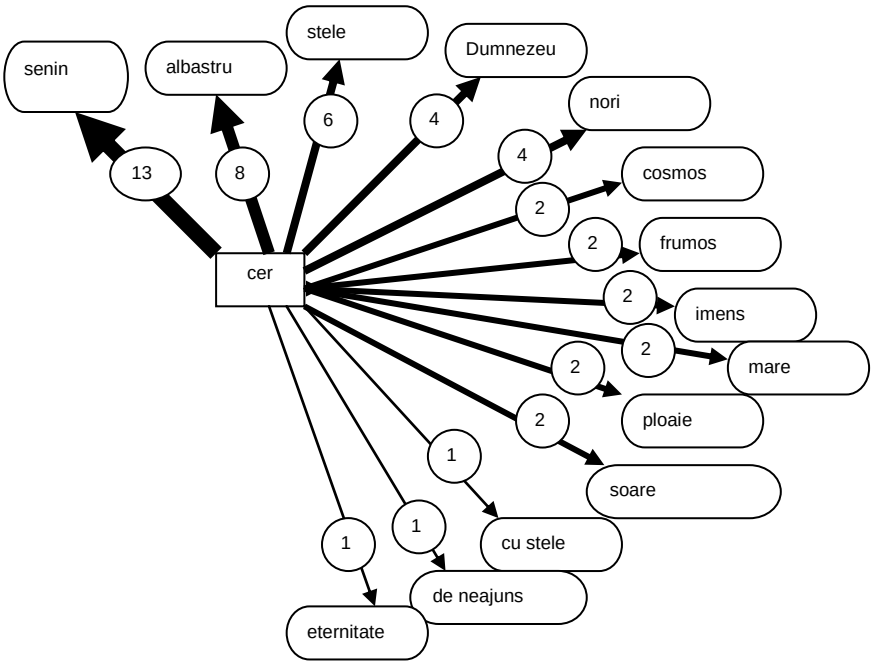

Figure 2: The set associated with the word "sky" which contains 14 associations.

[10] Data base and interface are impemented by our former student Ion Badan.

In order to obtain associations, a user has to type the word in the input field and press the button. An example of result for the word “frumos” (beautiful) is presented in figure 4. The list of all associations is sorted initially by the frequency in descending order; it can also be sorted by any other column of the table, in descending or ascending order.

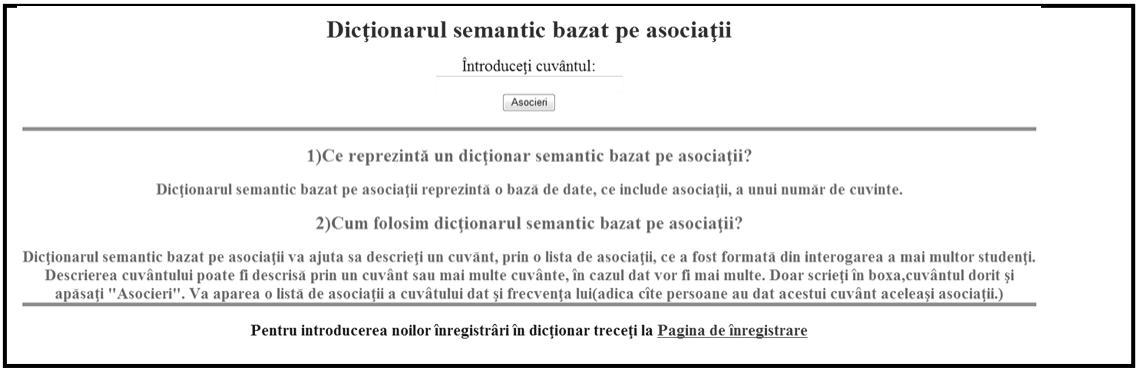

Figure 3: The interface for the Romanian associative dictionary interrogation.

Asocierile pentru cuvântul: “frumos”

#	Asociere ↑/ Asociere ↓	Asociere ↑/ Asociere ↓	Frecvenţa ↑ / Frecvenţa ↓
1	femeie	**frumos**	4
2	**frumos**	placut	4
3	zâmbet	**frumos**	4
4	aur	**frumos**	3
5	corpul	**frumos**	3
6	**frumos**	atragator	3
7	**frumos**	copil	3
8	viitor	**frumos**	3
9	ceas	**frumos**	2
10	cer	**frumos**	2

Rezultate de la 1 la 10 din 66.

1 | 2 | 3 | 4 | 5 | 6 | 7 | Următorul »

Figure 4: The associations for the word “frumos” (beautiful) extracted from Romanian associative dictionary.

There are two types of relations between words in the associative dictionary: *direct relation* from stimulus toward response, and the *inverse relation* from response to stimulus; these relations are not symmetrical. Thus, for the stimulus “aur” (gold) three responses were “frumos” (beautiful), but if “frumos” (beautiful) was stimulus no one response was “aur” (gold).

The resulting table for the interrogation contains both types of relations for the introduced word; it can be seen in figure 4. The first column contains words-stimuli; the second one contains words-responses. The word “frumos” (beautiful) appears in both columns; in the first column as the stimulus and in second as response.

The last line of text in the interface presented in figure 3 contains the link to the page created for introduction of the new records in the associative dictionary. This page is

presented in figure 5. A random word is presented to the user, and the user has to introduce the associated word in the input box. After clicking the button "Asociază", the user is informed that the introduced association was added in the data base. For example:

"Baza de date a fost actualizată cu success pentru înregistrarea lemn <-> foc"

(The database was successfully updated for the record wood<->fire)

Word – stimulus is selected randomly from the list of al words in the database both stimuli and responses. Thus the number of stimuli is also growing more than these 150 words selected initially.

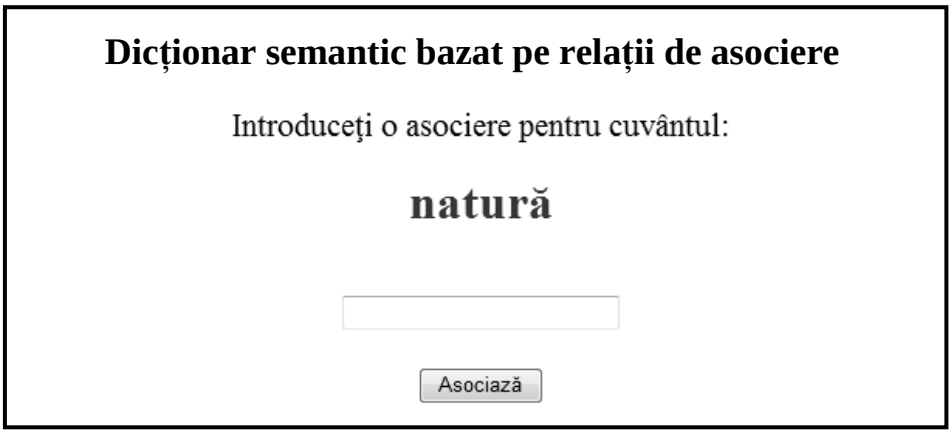

Figure 5: The interface for the Romanian associative dictionary augmentation.

The first version of the association database obtained after processing the questionnaires contained almost 7 500 stimulus-response pairs. We had to remove some of responses for different reasons. Some respondents were not accurate and missed some words, some wrote long phrases instead of words as responses, which we had to remove. After preprocessing we obtained 5821 different pairs; 4152 pairs had frequency equal to 1. Since the database was installed online, it has been augmenting. Statistics until 11 of November 2011 is the following: 10092 pairs total, 6163 different pairs, 4464 pairs had frequency equal to 1.

There are several problems which still remain to be solved. First, the words added online should be verified. A user can add wrong information, a word with grammatical errors or even a combination of letters without any sense. Automatic verification against a dictionary can discard words which are not in our dictionary and if the word is written with a grammatical error, it is extremely difficult to correct it automatically. Diacritic signs represent another problem. Some users introduce words with these signs; some ignore them as it is a usual practice while writing online. The same word typed in two forms, with diacritic signs and without them, is considered as two different words in the database. For example, the stimulus word "zice" (say, speak) has three variants of word "vorbește" (talk) as a response: "vobeste", "vorbeste" and "vorbește". The first one has one letter missed and no diacritics, the second one is correct but without diacritics and the third one is absolutely correct. All of them are stored as three different responses in the current version of the association database.

5. Conclusion

The paper reports about the experiment of an associative dictionary for Romanian language creation. It outlines the first phase of the experiment when *word associations* were collected using questionnaire surveys. The second phase includes online interface creation and expanding of the dictionary via internet. Several technical issues of the second phase are discussed. Some preliminary statistics is presented and a brief analysis of the obtained database is made. The created dictionary can be used in lexicography and Romanian language studying. At this stage of work, we however are more interested in the richest and the most representative database of word association; the detailed analysis is postponed to forthcoming research.

Acknowledgements. The authors are grateful to anonymous reviewers for the profound analysis of our paper and helpful comments.

References

Church, K. W., Hanks, P. (1990). Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16, 22-29.

Collins, A. M., Loftus, E. F. (1975). A spreading-activation theory of semantic processing. *Psychological Review*, 82, 407-428.

Cramer, P. (1968). Word association. *New York & London: Academic Press*.

Deese, J. (1965). The structure of associations in language and thought. *Baltimore: The John Hopkins Press*.

Edmonds, P., Hirst, G. (2002). Near-synonymy and lexical choice. *Computational Linguistics* 28:2,105-144.

Joyce, T. (2005). Constructing a large-scale database of Japanese word associations. *Special issue edited by Katsuo Tamaoka: Corpus Studies on Japanese Kanji*. Glottometrics, 10, 82-98.

Hirst, G. (2004). Ontology and the lexicon. *In Steffen Staab, & Rudi Studer, (Eds.), Handbook of ontologies*. Berlin, Heidelberg, & New York: Springer-Verlag, 209-229.

Kiss, G. R., Armstrong, C., Milroy, R., Piper, J. (1973). An associative thesaurus of English and its computer analysis. *In Aitken, A.J., Bailey, R.W. and Hamilton-Smith, N. (Eds.), The Computer and Literary Studies. Edinburgh: Edinburgh University Press*.

Miller, G. A. (1995). WordNet: A Lexical Database for English. *Communications of the ACM*, 38: 11, 39-41.

Nelson, D. L., McEvoy, C. L., Schreiber, T. A. (1998). The University of South Florida word association, rhyme, and word fragment norms. http://www.usf.edu/ FreeAssociation/.

Pustejovsky, J. (2010). Qualia Roles. The Cambridge Encyclopedia of the Language Sciences. *Ed. Patrick Hogan*, Cambridge, UK: Cambrige University Press.

Steyvers, M., Shiffrin, R. M., Nelson, D. L. (2004). Word association spaces for predicting semantic similarity effects in episodic memory. *In A. F. Healy, (Ed.),*

Experimental cognitive psychology and its applications. (Decade of behavior). Washington, D.C.: American Psychological Association, 237-249.

Vlad, A., Mitrea, A., Mitrea, M. (2005). Limba română scrisă ca sursă de informație. *Paideia, România*.

Балтова, П., Ефтимова, А., Липовска, А., Петрова, К. (2003). БАС 2003: Български асоциативен речник. *София: Изд. СУ* "Св. Кл. Охридски".

Караулов, Ю. Н., Черкасова, Г. А., Уфимцева, Н. В., Сорокин, Ю. А., Ярошинская, В. Н. (2002, 2003). РУССКИЙ АССОЦИАТИВНЫЙ СЛОВАРЬ. Том I. От стимула к реакции. Том II. От реакции к стимулу. Астрель, АСТ, 784 (992) стр.

Уфимцева, Н. В. (2004). Славянский ассоциативный словарь: русский, белорусский, болгарский, украинский. *Институт языкознания РАН*, 790 стр.

DEACC- LEXICAL DICTIONARY EXTRACTOR FROM COMPARABLE CORPORA

ELENA IRIMIA

Institutul de Cercetări pentru Inteligență Artificială, Academia Română, București, România

elena@racai.ro

Abstract

The paper describes a tool developed in the context of the ACCURAT project (Analysis and evaluation of Comparable Corpora for Under Resourced Areas of machine Translation). The purpose of the tool is to extract bilingual lexical dictionaries (word-to-word) from comparable corpora which do not have to be aligned at any level (document, paragraph, etc.) The method implemented in this tool is introduced by Rapp (1999). The application basically counts word co-occurrences between unknown words in the comparable corpora and known words from a Moses extracted general domain translation table (the base lexicon). We adapted the algorithm to work with polysemous entries in the translation table, a very frequent situation which is not treated in the standard approach.

1. Introduction

The task of extracting translation equivalents from bilingual corpora has been approached in different manners, according to the degree of parallelism between the source and target parts of the corpora involved. For a well sentence aligned parallel corpora one can benefit from reducing the search space for a candidate translation to the sentence dimension and external dictionaries are not required. In the case of comparable corpora, the lack of aligned segments can be compensated by external dictionaries (Rapp, 1999) or by finding meaningful bilingual anchors within the corpus based on lexico-syntactic information previously extracted from small parallel texts (Gamallo, 2007).

The word alignment of parallel corpora has been received significant scientific interest and effort starting with the notorious paper of Brown et al. (1990) and continuing with important contributions like (Gale & Church, 1993), (Kay & Roscheisen, 1993), (Och, et al., 1999), etc. and many more recent approaches. They are already various free software aligners used in the industry and research, from which we mention only the famous GIZA++ (Och & Ney, 2003). Moreover, the error rate goes down to 9% in experiments made with some of these approaches (Och & Ney, 2003). By comparison, the efforts and results in extracting bilingual dictionaries from comparable corpora are much poorer. Most of the experiments are usually done on small test sets, containing words with high frequency in the corpora (>99) and the accuracy percentages are not rising above 65%.

The most popular method to extract word translations from comparable corpora, on which we based the construction of our tool, is described and used in (Fung &

McKeown, 1997), (Rapp, 1999), (Chiao & Zweigenbaum, 2002). It relies on external dictionaries and is based on the following hypothesis: *word **target1** is a candidate translation of **source1** if the words with which **target1** co-occur within a particular window in the target corpus are translations of the words with which **source1** co-occurs with in the same window in the source corpus.* The translation correspondences between the words in the window are extracted from the external dictionaries, being seen as seed word pairs.

Gamallo & Pichel (2005) used as seed expressions pairs of bilingual lexico-syntactic templates previously extracted from small samples of parallel corpus. This strategy leaded to a context-based approach, reducing the searching space from all the target lemmas in the corpus to all the target lemmas that appear in the same seed templates. In the improved version of the approach (Gamallo, 2007), the precision-1 (the number of times a correct translation candidate of the test word is ranked first, divided by the number of test words) and precision-10 (the number of correct candidates appearing in the top 10, divided by the number of test words) scores go up to 0.73 and 0.87 respectively.

In the following we will describe the algorithm implemented by our tool as introduced by Rapp (1999) and we will highlight the modifications and the adaptations we made, based on the experimental work we conducted.

2. Short presentation of the original approach

In a previous study, Rapp (1995) had already proposed a new criterion (the co-occurrence clue) for word alignment appropriate for non-parallel corpora. The assumption was that "there is a correlation between co-occurrence patterns in different languages" and he demonstrated by a study that this assumption is valid even for unrelated texts in the case of English-German language pair.

Starting from a more or less small seed dictionary and with the purpose of extending it based on a comparable corpus, a co-occurrence matrix is computed both for the source corpus and for the target corpus. Every row in the matrix corresponds to a type word in the corpus and every column corresponds to a type word in the base lexicon. For example, the intersection of a row *c* (associated to a word in the corpus) and a column d (associated to a word in the dictionary) in the co-occurrence matrix of the source corpus contains a value *sourcecooc(c,d) = frequency of common occurrence of word c and word d in a window of pre-defined size* (see Figure 1 for a graphic of a generic co-occurence matrix). The target and source corpus are lemmatized and POS-tagged and function words are not taken in consideration for translation (they are identified by their POS closed class tags: pronouns, prepositions, conjunctions, auxiliary verbs, etc.).

For any row in the source matrix, all the words with which the co-occurence frequency is bigger than 0 are sent for translation to the seed lexicon. An entry in the seed lexicon is identified by a unique identifier id. It is not clear what was the solution of Rapp (1999) for the polysemous words in the seed dictionary. The unknown words (absent in the lexicon) are discarded and a vector of co-occurrence for the word correspondent to the row is computed versus the list of ids resulted after translation. The same procedure is applied to all the rows in the target matrix.

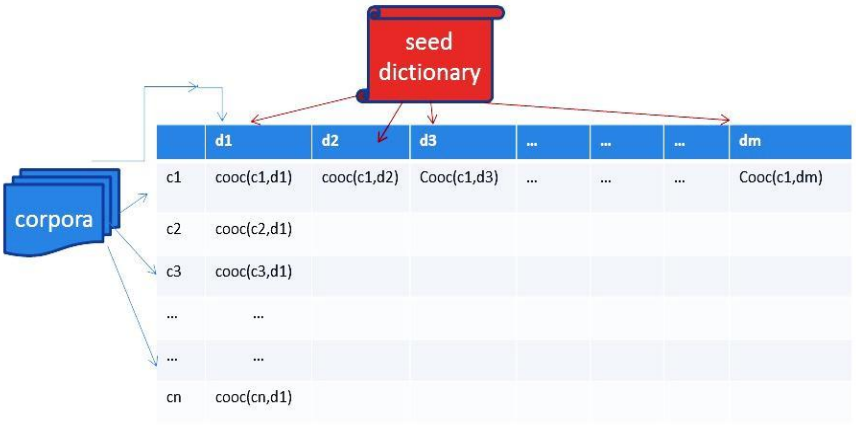

Figure 1: A generic co-occurrence matrix

Experiments conducted to the need of replacing the co-occurence frequency in the co-occurence vectors by measures able to eliminate word-frequency effects and favour significant word pairs. Measures with this purpose were previously based on mutual information (Church & Hanks, 1989), conditional probabilities (Rapp, 1996), or on some standard statistical tests, such as the chi-square test or the log-likelihood ratio (Dunning, 1993). In the approach we based our tool on, the measure chosen was the log-likelihood ratio computed as below:

$$LL(w_1, w_2) = \sum_{i,j \in \{1,2\}} k_{ij} \log \frac{k_{ij} N}{C_i R_j} = k_{11} \log \frac{k_{11} N}{C_1 R_1} + k_{12} \log \frac{k_{12} N}{C_1 R_2} + k_{21} \log \frac{k_{21} N}{C_2 R_1} + k_{22} \log \frac{k_{22} N}{C_2 R_2}$$

where

$$C_1 = k_{11} + k_{12}$$
$$R_1 = k_{11} + k_{21}$$
$$C_2 = k_{21} + k_{22}$$
$$R_2 = k_{12} + k_{22}$$
$$N = k_{11} + k_{12} + k_{21} + k_{22}$$

and

k_{11} = frequency of common occurence of w_1 and w_2 in a specific window in the corpus

k_{12} = corpus frequency of word w_1 - k_{11}

k_{21} = corpus frequency of word w_2 - k_{11}

k_{22} = size of corpus - corpus frequency of word w_1 - corpus frequency of word w_2

Finally, similarity scores are computed between all the source vectors and all the target vectors computed in the previous step, thus setting translation corrpespondencies between the most similar source and target vectors. Different similiarity scores were used in the variants of this approach; see (Gamallo, 2008) for a discussion about the efficiency of several similarity metrics combined with two weighting schemes: simple occurrences and log likelihood.

3. Our approach

3.1. Adaptations of the Rapp's algorithm

With the aim of obtaining a dictionary similar to a translation table of the type a decoder like Moses would need to produce its translation, we decided that the lines and columns of the matrixes will be populated in our approach by word forms and not by lemmas, as in the standard approach. The option for lemma entries in the matrix was assumed also by works like (Gamallo & Pichel, 2005) and (Gamallo, 2008).

As the purpose of this tool (and of all the other tools in the ACCURAT project) was to extract from comparable corpora data that would enrich the information already available from parallel corpora, it seemed reasonable to focus (just like Rapp (1999) did) on the open class (versus closed class) words. Because in many languages, the auxiliary verbs can also be main verbs, frequently basic concepts in the language (see "be" or "have" in English), and most often the POS-taggers don't discriminate correctly between the two roles, we decided to eliminate their main verb occurrences as well. For this purpose, the user is asked to provide a list of all these types with all their forms in the languages of interest (parameters in the configuration file: `sourceamverblist`, `targetamverblist`).

We gave the user the possibility to specify the length of the text window in which co-occurrences are counted by modifying a parameter in the configuration file (parameter in the configuration file: `window`). As our experiment conducted to good results for a text window of length five, this is the default value of the parameter.

Being based on word counting, the method is sensitive to the frequency of the words: the bigger the frequency, the better the performance. In previous works, the evaluation protocol was conducted on frequent words, usually on those with the frequency bigger than 100. Even in works like (Gamallo, 2008), where the evaluation was made on a list of nouns whose recall was 90% (those nouns that together come to the 90% of noun tokens in the training corpus), this corresponded to a bilingual lexicon constituted by 1,641 noun lemmas, each lemma having a token frequency>=103, for a bilingual comparable corpus of around 15 million tokens for each part. It doesn't seem too efficient to extract only a small amount of tokens from a big size corpus. Therefore, even if it brings loss of precision, the frequency threshold must be lowered when we are interested in extracting more data. In our tool, this parameter can be set by the user, according to his/her needs, but it should be bigger than 3 (our minimal threshold) and it should take into account the corpus dimension.

As we mentioned in the previous section, the polysemy in the seed lexicon is not discussed in the standard approach. Our seed lexicon is based on a general domain translation table automatically extracted (with GIZA++) and this is consistent with the idea that we want to improve translation data obtained from parallel corpora. But as a consequence, we deal with high ambiguity and erroneous data in the seed lexicon. In the following table (Table 1) you can see an excerpt from the base lexicon displaying all the possible translation for the word form "creates" with their translation probabilities. Only the first three entries are exact translations of the word form "creates" while 3 of them ("instituie", "stabilește" and, in a lesser extent, "ridică" are acceptable translations in

certain contexts). The two bold entries, "naștere" (birth) and "duce" (carries), may seem wrong translations learned from the training data, having a translation probability score similar to some correct translation (like "creând" or "crea"), but they also can be acceptable translations in certain contexts. We think we need to have acces to all these possible translations as the semantic content of a linguistic construction is rarely expressed in another language through an identical syntantic or lexical structure. This is true especially in the case of a comparable corpus.

Our solution was to distribute the log-likelihood of a wordpair (w1, w2) in the source language to all the possible translations of w2 in the target language as follows:

$$LL(w_1, w_2) = \sum_i LL(w_1, w_2) * p(w_2, t_i),$$

where $p(w_2, t_i)$ is the probability of a word w_2 to be translated by t_i and

$$\sum_i p(w_2, t_i) = 1.$$

Every translation pair (w_2, t_i) is identified in the base lexicon by an unique *id*, making it possible to compute a similarity score across the languages.

Table 1: An excerpt from the base lexicon with the possible candidate translations of the word "creates" and the distribution of this word LL according to the translation probabilities of the candidates

id	***word***	***translation***	***translation score***	***LL distribution: LL (man, creates)=12 becomes***
72083	creates	creea	0.0196078	LL(man, 72083) = 12*0.0196078 = 0.2352936
72084	creates	creează	0.686275	LL(man, 72084) = 12*0.686275 = 8.2353
72085	creates	creând	0.0196078	LL(man, 72085) = 12*0.0196078 = 0.2352936
72086	creates	**duce**	0.0196078	LL(man, 72086) = 12*0.0196078 = 0.2352936
72087	creates	instituie	0.117647	LL(man, 72086) = 12*0.117647 = 1.411764
72088	creates	**naștere**	0.0196078	LL(man, 72086) = 12*0.0196078 = 0.2352936
72089	creates	ridică	0.0392157	LL(man, 72089) = 12*0.0392157 = 0.4705884
72090	creates	stabilește	0.0196078	LL(man, 72086) = 12*0.0196078 = 0.2352936

Previous to the LLs distribution, there is a step of LL filtering, in which all the words that occur with an LL smaller than a threshold are eliminated (the threshold is set by the `ll` parameter in the configuration file). This was motivated by the need to reduce the space and time computational costs and is also justified by the intuition that not all the words that occur at a specific moment together with another word are significant in the general context of our approach and the LL score is a good measure of this significance.

Following the conclusions of Gamallo's (2008) experiments, we used as a vector similarity measure the DiceMin function, where *Sids* and *Tids* are the sets of dictionary entries identifiers with which w_1 and w_2 co-occur:

$$Dicemin(w_1, w_2) = \frac{2\sum_{k \in Sids \cap Tids} min\ (LL(w_1, k), LL(w_2, k))}{\sum_{i \in Sids} LL(w_1, i) + \sum_{j \in Tids} LL(w_2, j)}$$

In computing the similarity scores, we did not allowed the cross-POS translation (a noun can be translated only by a noun, etc.); the user can decide if he/she allows the application to cross the boundaries between the parts of speech, through a parameter modifiable in the configuration file: `crossPOS`. Each choice has its rationales, as we know that a word is not always expressed through the same part of speech when translated in another language. On the other hand, putting all the words in the same bag increases the number of computations and the risk of error.If the user's machine has multiple processors, the application can call a function that splits the time consuming problem of computing the vector similarities and runs it in parallel. This function is activated by the user through a "multithreading" parameter in the configuration file. To avoid overloading the memory, the application gives the user the opportunity to decide how many of the source/target vectors are loaded in the memory at a specific moment, through the `loading` parameter, activated only for `multithreading:yes`; take into account that setting this parameter to a value smaller than the matrix size can bring an important time delay, so it's in user's hands to set properly the parameters and balance advances and disadvantages according to the time constraints and according to the available memory resources.

3.2. Experiments and results

Tests have been conducted on different sizes and different types/registers of comparable corpora:

I. A comparable corpora of small size representing the civil code of Romania (184,081 words) vs. the civil code of Canada (199,401 words).

II. A corpus of articles extracted from Wikipedia: 743,194 words for Romanian, 809,137 words for English.

III. The corpus compiled by Sheffield in this project (1,396,009,747 words for English, 3,764,654,484 words for Romanian).

The evaluations are in progress, therefore only a small part will be presented here. We manually compiled a gold standard lexicon of around 1,500 words (common nouns, proper nouns, verbs and adjectives) from the Wikipedia corpus. In the conditions described by the default parameters in the configuration file, the precision-1 and precision-10 scores introduced earlier were computed and presented in Table 2:

Table 2: P-1 and P-10 for the 1,500 test words from Wikipedia corpus

POS	**Precision-1**	**Precision-2**
commnon nouns	0.5739	0.7381
proper nouns	0.6957	0.7338
adjectives	0.4944	0.6292
verbs	0.6621	0.8276

In Table 3 we present translation results obtained for 4 adjectives, marking the correct translations in bold characters. In 3 of the examples, possible correct translations are ranked by the tool on the first position. In the case of *significant*, multiple solutions are offered: synonims (like "important", "principal", "semnificativ") in different morphological forms (feminine, masculin, singural, plural); this approach is in accordance with the basics of statistical machine translation, where a language model is used to select the appropriate form from a pool of possible translations. In the last example, modern has as translation in Romanian a perfect cognat, ranked in an inferiour position in the list of possible translations (8). This can be remedied by a cognate score which will boost the term in the first position.

Table 3: Sample of the result file for the adjective translations; the correct translations are bolded.

additional^af	**significant^af**
suplimentari^af 0.1268#	**importante^af** 0.0468#
general^af 0.0014#	**semnificativă^af** 0.0427#
financiare^af 0.0011#	mari^af 0.0418#
referitor^af 0.0010#	**principalele^af** 0.03902#
nouă^af 0.0008#	prezente^af 0.0367#
mari^af 0.0008#	**importantă^af** 0.0367#
indian^af 0.0007#	economice^af 0.0346#
comună^af 0.0007#	culturale^af 0.03423#
medie^af 0.0006#	**semnificative^af** 0.0339
nordică^af 0.0006#	**semnificativ^af** 0.0309#
religious^af	**modern^af**
religioase^af 0.06583#	considerată^af 0.0457#
culturale^af 0.0448#	veche^af 0.0423#
politice^af 0.0412#	cunoscut^af 0.0403#
religioasă^af 0.0400#	antică^af 0.0390#
umane^af 0.0370#	roman^af 0.03790#
economice^af 0.0369#	engleză^af 0.0377#
diferite^af 0.0369#	vechi^af 0.0372#
administrativ^af 0.03474#	**modern^af** 0.0319#
sociale^af 0.0335#	latină^af 0.0314#
economic^af 0.0330#	importante^af 0.0310#

4. Conclusions

For strongly comparable corpora like I. and II. in the precedent section, the results are fairly good and the dictionaries look usable in augmenting a more general dictionary like the one we used as a seed dictionary.

For the third corpus, the outcome is not very encouraging. We intent to experiment with different values for the parameters, especially increasing the frequency threshold, but our guess is that we will not attain performances similar with those in the previous section because of the weaker comparability of the corpus.

Acknowledgements. The work reported here is funded by the ACCURAT project funded from the European Union Seventh Framework Programme (FP7/2007-2013), grant agreement no 248347.

References

Brown, P., Cocke, J., Della Pietra, S. A., Della Pietra, V. J., Jelinek, F., Lafferty, J. D., Mercer, R. L., Rossin, P. S. (1990). A statistical approach to machine translation. *Computational Linguistics,* 16:2, 79-85.

Chiao, Y.-C., Zweigenbaum, P. (2003). Looking for candidate translational equivalents in specialized, comparable corpora. *Coling 2002*, Taipei, Taiwan.

Church, K. W., Hanks, P. (1989). Word association norms, mutual information, and lexicography. *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics.* Vancouver, British Columbia, 76-83.

Dunning, T. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19:1, 61-74.

Fung, P., McKeown, K. (1997). Finding terminology translations from non-parallel corpora. Proceedings of the Fifth workshop on Very Large Corpora. ed. Joe Zhou and Kenneth Church, 18 August 1997, Tsinghua University, Beijing, China, 20 August, Hong Kong University of Science and Technology, Hong Kong,192-202.

Gamallo, P., Pichel, J. R. (2005). An Approach to Acquire Word Translations from NonParallel Texts. *Lecture Notes in Computer Science,* 3808. SpringerVerlag

Gamallo, P. (2007). Learning bilingual lexicons from comparable English and Spanish corpora. *In Machine Translation SUMMIT XI,* Copenhagen, Denmark.

Gamallo, P. (2008). Evaluating two different methods for the task of extracting bilingual lexicons from comparable corpora. *Proceedings of LREC 2008 Workshop on Comparable Corpora*, Marrakech, Marroco, ISBN: 2-9517408-4-0, 19-26.

Gale, W. A., Church, K. W. (1993). A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19:3, 75-102.

Kay, M., Roscheisen, M. (1993). Text-Translation Alignment. *Computational Linguistics*, 19:1, 121-142.

Och, F. J., Ney, H. (2003). A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29:1, March, 19-51.

Och, F. J., Tillmann, C., Ney, H. (1999). Improved alignment models for statistical machine translation. *Joint SIGDAT conference on Empirical Methods in Natural Language Processing and Very Large Corpora*. Proceedings ed. Pascale Fung and Joe Zhou, 21-22 June, University of Maryland, College Park, MD, USA, 20-28.

Rapp, R. (1995). Identifying word translations in nonparallel texts. In: *Proceedings of the 33rd Meeting of the Association for Computational Linguistics.* Cambridge, Massachusetts, 320-322.

Rapp, R. (1996). Die Berechnung von Assoziationen. *Hildesheim: Olms*.

Rapp, R. (1999). Automatic identification of word translations from unrelated English and German corpora. *ACL-1999*: 37th Annual Meeting of the Association for Computational Linguistics. *Proceedings of the conference, 20-26 June 1999*, University of Maryland, College Park, Maryland, USA, 519-526.

EXTRACTING PARALLEL TERMINOLOGY FROM COMPARABLE CORPORA

DAN ȘTEFĂNESCU

Research Institute for Artificial Intelligence (*RACAI*), *Romanian Academy*

danstef@racai.ro

Abstract

This paper describes RACAI's current methodology for extracting parallel terminology from comparable corpora. The techniques involved are language independent and are currently employed within the ACCURAT project with the purpose of improving the translation models used by statistical machine translation systems. In addition, parallel terminology can be used for constructing dictionaries or indexing technical or domain-restricted documents.

1. Introduction

Statistical Machine Translation Systems require two types of statistical models: language models and translation models, whose parameters are, more often than not, derived from the analysis of parallel corpora. Since large parallel corpora are only available for several languages with rich resources (English, French, German, Spanish, etc.), there is an increasing necessity in gathering parallel data for under resourced languages. Taking into account the huge amount of data existing over the internet, one of the recent approaches for acquiring parallel data is to extract it from comparable corpora. Such corpora consist in documents covering the same topic or subject and using more or less similar expressions, named entities or terminology. For example, one can easily find *Wikipedia*[1] or *news* articles, which are examples of strongly and, respectively, weakly comparable documents. The objective is to extract the existing parallel data and use it to enrich poor translation models.

This paper describes RACAI's current methodology for extracting parallel terminology from comparable corpora. The techniques involved are language independent and are currently employed within the ACCURAT[2] project with the purpose of improving the translation models used by statistical machine translation systems. Outside this scope, parallel terminology can be used for constructing dictionaries or indexing technical or domain-restricted documents.

Our approach works in two steps. First, the terminology is monolingually extracted, taking into consideration both single and multi-word terms, while in the second step, the extracted terms are mapped based on lexical similarity and the existing dictionaries. The methods described are language independent as long as language specific parameter data is provided.

[1] http://en.wikipedia.org/wiki/Romania vs. http://ro.wikipedia.org/wiki/Rom%C3%A2nia
[2] http://www.accurat-project.eu/

The paper is structured as follows: the next section presents the monolingually terminology extraction, while section 3 describes the terminology mapping. Experiments and results are presented in section 4. The paper ends with conclusions, acknowledgements and references sections.

2. Terminology extraction

Terminology extraction is the subtask of Information Extraction which refers to extracting terms from a given corpus, relevant to the genre / domain of the corpus. This task dates back to the 70s and it was most studied in the 90s. This latter period saw an explosion of various approaches (Schütze, 1998) based on raw frequency and part of speech filters (Dolby et al., 1973; Justeson & Kats, 1995), low variance in relative position for multi-word terms (Smadja, 1993), hypothesis testing and mutual information (Church & Hanks, 1989), likelihood ratios on assumed distributions (Dunning, 1993), inverse document frequency on assumed distributions (Church, 1995), finite-state automaton parsing (Grefenstette, 1994), full parsing (Bourigault, 1993; Strzalkowski, 1995), semantic analysis (Pustejovsky et al., 1993), etc.

We make a clear distinction between single-word and multi-word terms, since their identification and extraction is usually performed by using different approaches.

2.1. Single-word terminology extraction

We approached the task of single-word terminology extraction by improving Damerau's method (Damerau, 1993) as it has been reported to yield very good results (Schütze, 1998; Paukkeri et al., 2008). Damerau's approach compares the relative frequency in the documents of interest (user corpus – C_U) to the relative frequency in a reference collection (reference corpus – C_R). The original formula for computing the score of a word is:

$$score(word) = \frac{f(word, C_U)}{|C_U|} \div \frac{f(word, C_R)}{|C_R|},$$

where $f(word_i, C_j)$ is the frequency of the word i in corpus j, and $|C_j|$ is the total number of words in C_j. One can immediately notice that the score for a word is calculated according to the likelihood ratios of occurring in both corpora (that of the user and the reference). The main idea is to compare the maximum likelihood estimates (MLE) computed on the user corpus to the ones on the reference corpus. Consequently, the reference corpus should be a large, balanced and representative corpus for the language of interest. Essentially, the MLE on such a corpus is equivalent with an unigram language model:

$$P_{MLE}(word) = \frac{f(word, C_R)}{|C_R|}.$$

In practice, such models are usually used in information retrieval to determine the topic of documents. Thus, Damerau's formula works by comparing two unigram language models.

It has been proven however, that due to data sparseness, the unigrams language models constructed only by the means of MLE behave poorly and that a proper smoothing

should be performed (Chen & Goodman, 1998). To do this, we employ a variant of Good-Turing estimator smoothing (Kochanski, 2006):

$$P_{GT}(word) = \frac{f(word, C_R) + 1}{|C_R| + |V_R|} \cdot \frac{E(f(word, C_R) + 1)}{E(f(word, C_R))}$$

where V_R is the vocabulary (the unique words in C_R) and $E(n)$ is the probability estimate of the word to occur exactly *n* times.

Let us consider a slightly modified example from (Kochanski, 2006): let us say we have a (reference) corpus with 40,000 English words which contains only one instance of the word "*unusualness*": $f(word, C_R) = 1$. Let us also say that the corpus contains 10,000 different words that appear once and so, $E(1)$ = 10,000 / 40,000, and that we have 5,500 words that appear twice, giving $E(2)$ = 5,500 / 40,000. Again, let us consider that the total number of the unique words in the corpus is 15,000 ($|V_R|$ = 15,000). The Good-Turing estimate of the probability of "*unusualness*" is:

$$P_{GT}(unusualness) = \frac{1+1}{40{,}000 + 15{,}000} \cdot \frac{5{,}500/40{,}000}{10{,}000/40{,}000} = \frac{2}{55{,}000} \cdot \frac{5{,}500}{10{,}000} = \frac{1}{50{,}000}.$$

But using MLE, we would have had a larger value:

$$P_{MLE}(unusualness) = \frac{1}{40{,}000}.$$

Because the sum of the probabilities must be 1, we have a remaining probability mass to be assigned to the unseen words (U). Consequently, the probability of an unseen word depends on the estimated number of unseen words:

$$P_{GT}(unseen) = \frac{E(1)}{(|C_R| + |V_R|) \cdot |U|} = \frac{10{,}000/40{,}000}{55{,}000 * |U|}.$$

Going back to Damerau's formula, we have now that:

$$score(word) = \frac{f(word, C_U)}{|C_U|} \div P_{GT}(word\ in\ C_R).$$

The words having the highest scores are terminological terms.

In case C_U is a large corpus, we can also compute Good Turing estimators for the numerator. For small corpora, this is however impractical since one cannot compute the estimates E(n) with high enough confidence.

This approach can be improved by additional preprocessing of the corpora involved. First, for better capturing the real word distribution, it is better to use word lemmas (or stems) instead the occurrence forms. Second, the vast majority of the single terminological terms are nouns and so, one can apply a POS filtering in order to disregard the other grammatical categories. Both can be resolved by employing stand-alone applications that can POS-tag and lemmatize the considered texts. As our research is mainly focused on English and Romanian, we usually make use of the TTL preprocessing Web Service (Justeson & Katz, 1995; Tufiș et al., 2008) when dealing with these languages.As reference corpora we used the *Agenda* corpus (Tufiș & Irimia, 2006) and *Wikipedia* for Romanian and *Wikipedia* for English.

The method presented above can be reinforced with the well-known *TF-IDF* (*term frequency – inverse document frequency*) approach (Spärck Jones, 1972), provided that

the corpus of interest is partitioned into many documents or that this partitioning can be automatically performed.

2.2. Multiple-word terminology extraction

Terminology extraction does not limit to the single-word terms and so, one must be able to extract multi-word terminology, too. Smadja was among the first to advocate that low variance in relative position is a strong indicator for multi-word terminological expressions (Smadja, 1993), which can be found among the collocations of a corpus. These are expressions which cannot be translated word-by-word using only a simple dictionary and a language model, because they are characterized by limited compositionality – the meaning of the expression is more than the sum of the meaning of the words composing the collocation.

Different methods have been proposed for finding collocations. Some counted the occurrences of bigrams and then used a part-of-speech filter in order to rule out those bigrams which cannot be phrases (Justeson & Krats, 1995). Smadja employed a method based on the mean and the variance of the distances between pairs of words (Smadja, 1993), while others (Church et al., 1991) used *t Test*, *chi square Test*, *Log-Likelihood* or *Mutual Information* for finding pairs of words which appear together in the text more often than expected by chance.

Our approach for the identification and extraction of collocations has been described in several papers (Ștefănescu et al., 2006; Todirașcu et al., 2009; Ștefănescu, 2010). Basically, a collocation is a pair of words for which:

– the distance between them is relatively constant;

– they appear together more often than expected by chance: *Log-Likelihood*.

One component of our solution for collocation identification is based on Smadja's method that uses the average and the standard deviation computed for the distances between pairs of words in the corpus, in order to identify those words which appear together in a fixed relation. Collocations can be found by looking for pairs of words for which the standard deviations of distances are small.

In order to find terminological expressions, we employ a POS-filtering, computing the standard deviation for **only** the *noun-noun* and *noun-adjective* pairs within a window of 11 non-functional words length, and we keep all the pairs for which standard deviation is smaller than 1.5 – a reasonable value according to (Manning & Schütze, 1999). This method allows us to find good candidates for terminology. However, we still want to further filter out some of the pairs so that we keep only those composed by words which appear together more often than expected by chance. We do this by computing Log-Likelihood (LL) score for all the above obtained pairs. We compute the LL scores by taking into account only the occurrences of the words having the selected POS-es. We take into consideration the pairs for which the LL values are higher than 9, as for this threshold the probability of error is less than 0.004 according to the *chi square* tables.

We further keep as terminological expressions only those for which at least one of the terms forming them can be found between the *single-word* terminological terms, disregarding their context.

3. Terminology mapping

Lately, automatic terminology mapping has been well-studied using methods like compositional analysis (Grefenstette, 1999; Daille & Morin, 2008) or contextual analysis (Fung & McKeown, 1997). Still, terminology mapping for languages with scarce resources is less researched (Weller et al., 2011).

Our terminology mapping tool was developed under the name TEA (TErminology Aligner). Given two lists containing monolingually extracted terminology, it is designed to find (in those lists) pairs of expressions which are reciprocal translations. In order to do this, TEA analyzes candidate pairs, assigning them translation scores based on (i) the translation equivalents and (ii) the cognates that can be found in those pairs:

$$translationScore(pair) = max\big(ete(pair), ecg(pair)\big),$$

where *ete*(*pair*) is the translation equivalence score and *ecg*(*pair*) is the cognate score for the expressions forming the candidate pair.

The translation equivalence score for two expressions is computed based on the word-level translation equivalents existing in the expressions. Each word w_s in the source terminological expression e_s is paired with its corresponding word w_t in e_t such that the translation probability is maximal, according to a Giza++ like translation dictionary. The score should be normalized with the length of expression e_s. Still, we modify the denominator in order to penalize the candidate pairs according to the length difference between source and target expressions:

$$ete(e_s, e_t) = \frac{\sum_{w_s \in e_s} \max_{w_t \in e_t} wte(w_s, w_t)}{length(e_s) + \frac{|length(e_s) - length(e_t)|}{2}}.$$

The cognate score for two expressions is computed as a modified *Levenshtein* distance (LD) between them. The expressions are normalized by removing double letters and replacing some character sequences: "*ph*" by "*f*", "*y*" by "*i*", "*hn*" by "*n*" and "*ha*" by "*a*". This type of normalization is often employed by spelling and alteration systems (Ștefănescu et al., 2011). Moreover, the score takes into account the length of the longest common substring of the two expressions, normalized by the maximum value of their lengths:

$$ecg(e_s, e_t) = \frac{1 - \frac{LD\big(normalize(e_s), normalize(e_t)\big) + 1}{\min(length(e_s) + 1, length(e_t) + 1)} + \frac{length(LCS(e_s, e_t))}{\max\,(length(e_s), length(e_t))}}{2}.$$

The values of *ete*(*pair*) and *ecg*(*pair*) are taken into account only if they are higher than a threshold, the value of which regulates the tradeoff between precision and recall.

4. Experiments and results

Experiments on extracting parallel terminology require the existence of a *Gold Standard* (GS) containing bilingual mapped terminology relevant to a collection of bilingual comparable texts. The only freely available such GS we know of is Eurovoc. This is "*the thesaurus covering the activities of the EU and the European Parliament in particular*" and it has been described in (Steinberger et al., 2002). We performed two

experiments: the first one was designed to assess the performance of the monolingually terminology extraction, while the second one, the performance of the mapping.

In the first experiment we considered the newest 500 documents from 2006 for both English and Romanian. In order to accurately compute the accuracy figures, we generated lists containing only those Eurovoc terms that appeared in these 500 documents for each language and counted how many of the recognized terms were found in these corresponding restricted lists. The results are summarized in Table 1.

Table 1: Eurovoc terms identified as terminological

	English	**Romanian**
Number of documents	500 / 7972 (06.27%)	500 / 5792 (08.63%)
Size of preprocessed collection	33.8 MB	4.0 MB
Eurovoc terms identified out of those found in the collection having at least 1 occurence	816 / 2140 (38.13%)	117 / 483 (24.22%)
… 10 occurences	490 / 1523 (32.17%)	75 / 324 (23.15)
… 100 occurences	269 / 1039 (25.90%)	44 / 185 (23.79%)
… 500 occurences	79 / 525 (15.04%)	8 / 51 (15.68%)

Regarding this evaluation methodology, we must observe that the list of Eurovoc terms is neither exhaustive nor definitive and as such, there may be terms that the application discovers that are not in Eurovoc. Examples for English include "*Basel convention*", "*standards on aviation*", "*Strasbourg*", "*national safety standards*", etc.

For the second experiment, we considered the ideal case in which the monolingual terminology, for every language involved, contains only the Eurovoc terms. The results are summarized by Table 2.

Table 2: Terminology Mapping Performance for English-Romanian

Threshold	**Precision**	**Recall**	**F1**
0.1	56.31	06.84	12.20
0.2	42.61	10.11	16.34
0.3	56.18	19.35	28.78
0.4	75.89	29.51	42.49
0.5	90.41	35.65	51.13
0.6	96.44	29.82	45.56
0.7	98.64	21.56	35.91
0.8	99.60	15.12	26.26
0.9	99.46	08.36	15.43

5. Conclusions

This paper describes RACAI's current methodology for extracting parallel terminology from comparable corpora. Our objective is to provide a domain specific resource that can be used to improve the automatic alignment process of comparable corpora, which finally aims at developing better translation models for statistical machine translation systems. We have to underline the fact that the mapping module is the basic mapping tool in the ACCURAT project and it is currently used to map terminology extracted for all the languages involved: English, Estonian, German, Greek, Croatian, Latvian, Lithuanian, Romanian and Slovenian. Future work will be focused on improving this

approach, as well as on assessing its performance on every possible pair that can be formed with the above mentioned languages.

Acknowledgements. This work has been supported by the ACCURAT project (http://www.accurat-project.eu/) funded by the European Community's Seventh Framework Program (FP7/2007-2013) under the Grant Agreement no. 248347.

References

Bourigault, D. (1993). An endogenous corpus-based method for structural noun-phrase disambiguation. *In Proceedings of EACL-93,* 81–86.

Chen, S.F., Goodman, J. (1998). An empirical study of smoothing techniques for language modeling. *Technical Report TR-10-98,* Harvard University.

Church, K. (1995). One term or two?. *In Proceedings of SIGIR-95*, 310–318.

Church, K., Hanks, P. (1989). Word Association Norms, Mutual Information, and Lexicography. *In Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*.

Church, K., Gale, W., Hanks, P., Hindle, D. (1991). Parsing, word associations and typical predicate-argument relations. *Current Issues in Parsing Technology*. Kluwer Academic, Dordrecht.

Daille, B., Morin, E. (2008). Effective Compositional Model for Lexical Alignment. *In Proceedings of the 3rd International Joint Conference on Natural Language Processing*, Hyderabad, India.

Damerau, F. (1993). Generating and evaluating domain-oriented multi-word terms from text. *Information Processing and Management,* 29:4, 433–447.

Dolby, J. L., Ross, I. C., Tukey, J. W. (1973, 1973, 1975, 1973). Index to Statistics and Probability. 1:1, The Statistics Cumindex, 2 Citation Index, 3-4 Permuted Title, 5 Locations and Authors. R and D Press, Los Altos.

Dunning, T. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics* 19:1, 61–74.

Fung, P., McKeown, K. (1997). Finding terminology translations from non-parallel corpora. *In Proceedings of the 5th Annual Workshop on Very Large Corpora*, 192–202.

Grefenstette, G. (1994). Explorations in Automatic Thesaurus Discovery. *Kluwer Academic Press*, Boston.

Grefenstette, G. (1999). The World Wide Web as a resource for example-based machine translation tasks. *Translating and the Computer* 21, London, UK.

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

Justeson, J.S., Katz, S.M. (1995). Technical Terminology: some linguistic properties and an algorithm for identification in text. In Natural Language Engineering (1), Cambridge University Press, 9-27.

Kochanski, G. (2006). Lecture 4 - Good-Turing probability estimation. Oxford: http://kochanski.org/gpk/teaching/0401Oxford/GoodTuring.pdf.

Manning, C., Schütze, H. (1999). Foundations of Statistical Natural Language Processing. *MIT Press*, Cambridge.

Morin, E., Prochasson, E. (2011). Bilingual Lexicon Extraction from Comparable Corpora Enhanced with Parallel Corpora. *ACL HLT 2011*, 27.

Paukkeri, M., Nieminen, I. T., Pöllä, M., Honkela, T. (2008). A Language-Independent Approach to Keyphrase Extraction and Evaluation. *In Proceedings of COLING-08*.

Pustejovsky, J., Bergler, S., Anick, P. (1993). Lexical semantic techniques for corpus analysis. *Computational Linguistics*, 19:2, 331–358.

Schütze, H. (1998). The Hypertext Concordance: A Better Back-of-the-Book Index. *In Proceedings of Computerm '98* (Montreal, Canada, 1998), D. Bourigault, C. Jacquemin, and M.-C. L'Homme, Eds., 101–104.

Smadja, F. (1993). Retrieving Collocaionts from Text: Xtract. *In Computational Linguistics 19*, 143–175.

Spärck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation* 28:1, 11–21.

Steinberger, R., Pouliquen, B., Hagman, J. (2002). Cross-lingual Document Similarity Calculation Using the Multilingual Thesaurus Eurovoc. *Springer-Verlag*.

Strzalkowski, T. (1995). Natural Language information retrieval. *IP&M*, 31 :3, 397–417.

Ștefănescu, D. (2010). Intelligent Information Mining from Multilingual Corpora. *PhD thesis* (Romanian), Romanian Academy, Bucharest.

Ștefănescu, D., Ion, R., Boroș, T. (2011). TiradeAI: An Ensemble of Spellcheckers. *In Proceedings of the Spelling Alteration for Web Search Workshop*, Bellevue, USA, 20–23.

Ștefănescu, D., Tufiș, D., Irimia, E. (2006). Automatic Identification and Extraction of Collocations from Texts. *In the 2nd Romanian Workshop for Linguistic Tools and Resources Volume*, 3 Nov. 2006, Bucharest, Romania.

Todirașcu, A., Gledhill, C., Ștefănescu, D. (2009). Extracting Collocations in Contexts. *In Human Language Technology. Challenges of the Information Society*, Lecture Notes in Computer Science Series, Springer Berlin/Heidelberg. ISSN: 0302-9743 (Print) 1611-3349 (Online), Volume 5603/2009, ISBN 978-3-642-04234-8, 336–349.

Tufiș, D., Irimia, E. (2006). RoCo_News - A Hand Validated Journalistic Corpus of Romanian. *In Proceedings of the 5th LREC Conference*, Genoa, Italy, 869–872.

Tufiș, D., Ion, R., Ceaușu, A., Ștefănescu, D. (2008). RACAI's Linguistic Web Services. *In Proceedings of the 6th Language Resources and Evaluation Conference* – LREC-08, Marrakech, Morocco, 28–30.

Weller, M., Gojun, A., Heid, U., Daille, B., Harastaniv, R. (2011). Simple methods for dealing with term variation and term alignment. *In Proceedings of TIA 2011: the 9th International Conference on Terminology and Artificial Intelligence*, November 8-10, 2011, Paris, France.

LEXICOGRAPHIC MODELING AND PARSING EXPERIMENTS FOR THE DICTIONARY OF MODERN LITERARY RUSSIAN LANGUAGE

NECULAI CURTEANU[1], SVETLANA COJOCARU[2], ALEX MORUZ[1, 3]

[1]*Institute of Computer Science, Romanian Academy, Iași Branch*
[2]*Institute of Mathematics and Computer Science, Chișinău, Republic of Moldova*
[3]*Faculty of Computer Science, "Al.I. Cuza" University, Iași, Romania*

ncurteanu@yahoo.com, svetlana.cojocaru@math.md, alex.moruz@gmail.com

Abstract

The purpose of this present paper is twofold: (a) to discuss problems and innovative solutions for parsing the **DMLRL** (Dictionary of Modern Literary Russian Language) (Dictionary of Modern Literary Russian Language, 1994), relying on the already achieved lexicographic modeling (Curteanu et al., 2012) of this very large thesaurus-dictionary, and (b) to contribute with this further successful experiment to the development of the general and efficient parsing technology of SCD (Segmentation-Cohesion-Dependency) *configuration* (Curteanu et al., 2008), (Curteanu et al., 2010b), which can provide the computational background for designing a new, general and comprehensible DTD (Document Type Description) within the TEI standard for dictionaries (XCES TEI, 2007).

1. Introduction

This paper extends the experience of parsing other *five*, sensibly different, Romanian, French, and German largest dictionaries, to **DMLRL** (Dictionary of Modern Literary Russian Language) (Dictionary of Modern Literary Russian Language, 1994), using the optimal and portable *parsing method* of SCD (Segmentation-Cohesion-Dependency) *configurations* (Curteanu et al., 2010b), (Curteanu et al., 2010a), (Curteanu et al., 2008). In the papers (Curteanu et al., 2008), (Curteanu et al., 2010a), (Curteanu et al., 2010b) have been analyzed mainly the first and second SCD configurations (see next sections) of the following five thesaurus-dictionaries: **DLR** (The Romanian Thesaurus – new format) (Cristea et al., 2007), (Curteanu et al., 2008), **DAR** (The Romanian Thesaurus – old format) (Pușcariu, 1906), **TLF** (Le Trésor de la Langue Française) (Le Trésor de la Langue Française informatisé, 2010), **DWB** (Deutsches Wörterbuch – GRIMM) (Das Woerterbuch-Netz, 2010), and **GWB** (Göthe-Wörterbuch) (Das Woerterbuch-Netz, 2010). The paper (Curteanu et al., 2012) contains an extended analysis of complete lexicographic modeling for **DMLRL**, with sense marker classes, dependency hypergraphs for the three considered SCD configurations, illustrative examples etc. The next three sections summarize the **DMLRL** lexicographic modeling, necessary for the parsing method of SCD configurations, described at large in (Curteanu et al., 2012). The fifth section deals with **DMLRL** parsing experiments with this original and efficient method.

2. Lexicographic modeling of DMLRL

The homonymic entries in **DMLRL** (**D**ictionary of **M**odern **L**iterary **R**ussian **L**anguage) are discriminated by indexing each of the homonyms with Arabic numerals followed by dot, all in *Arial font*, *Regular* and *Bold* format. These indexes are positioned in front of each homonym-word lemma, enumerating increasingly all the homonyms of the same word-lemma in the dictionary. An example of *four* homonymic entries of the word "**БЫЧОК**" is present in **DMLRL** [7 :860-861].

The first SCD configuration has to recognize the *lexicographic segments* of a DMLRL entry. **DMLRL** comprises (at least) five types of lexicographic segments / packages (Curteanu et al., 2012): (1) a *morpho-lexical* package / segment, (2) the *sense description* segment, (3) a *TildaDef* package or segment of definitions, (4) the *morpho-syntactic variant* segment, and (5) the *etymology* segment of the word-lemma. Examples of **DMLRL** homonymic entries and lexicographic segments within an entry are provided in (Curteanu et al., 2012). We notice that the structure of lexicographic segments for large thesaurus-dictionaries, recognized within the first SCD configuration is, in general, linear and simple (Curteanu et al., 2010b), (Curteanu et al., 2010a). However, remarkable exceptions are the oldest dictionaries studied (Curteanu et al., 2010b), namely the German **DWB** (Das Woerterbuch-Netz, 2010) and the Romanian **DAR** (Pușcariu, 1906).

The following three SCD configurations are outlined (Curteanu et al., 2012): the first one has to separate the *lexicographic segments* of **DMLRL** dictionary, the second SCD configuration concentrates on the SCD *marker classes* and their *hypergraph hierarchy* for DMLRL *primary* and *secondary senses*, while the third SCD configuration is descending and refining the same modeling process for the *atomic sense definitions* and their *examples-to-definitions*. The dependency hypergraph of the *third* SCD *configuration*, interconnected to the one of the second SCD configuration, is specified completely at this atomic sense level ***for the first time***, in (Curteanu et al., 2012) and Fig. 2 below, exceeding the SCD modeling experiences for the other *five* thesaurus-dictionaries, *i.e.* **DLR**, **DAR**, **TLF**, **DWB**, **GWB**, described in (Curteanu et al., 2010b), (Curteanu et al., 2010a).

3. DMLRL dependency hypergraph of the second SCD configuration

The *primary sense* markers in **DMLRL** pointed out so far by the lexicographic analysis in (Curteanu et al., 2012) are: **(1)** capital Roman numerals followed by a dot (**I.**, **II., III.,...**etc.), in bold (*LatCapNumb_Mark*), and **(2)** Arabic numerals followed by a dot (**1., 2., 3.,...** etc.), in bold (*ArabNumb_Mark*). The markers of these classes are positioned at new paragraph (*NewPrg* marker), except for the *first-sense* (**I.**,… **1.**, …) or *root-sense markers*, which usually does not occur at new paragraph (*NewPrg*).

For **DMLRL**, the secondary senses are introduced by two sense markers, *viz.* the *two-oblique-bars* "//" and the *empty-diamond* "◊". They are similar and correspond to the **DLR** secondary sense markers *filled-diamond* "♦" and, respectively, *empty-diamond* "◊" (Curteanu et al., 2008), (Curteanu et al., 2010b). In the entry **ВЕДУЩИЙ** of **DMLRL** (Dictionary of Modern Literary Russian Language, 1994), which follows, the secondary sense "//" is refined through literal enumeration. In analogy with the **DLR**

hypergraph of sense dependencies, we associate the **DMLRL** "//" marker with the **DLR** "♦" sense marker: they are both secondary sense markers and subsume the similar secondary sense marker denoted in both dictionaries by the *empty-diamond* "◊" (see below) (Curteanu et al., 2008), (Curteanu et al., 2010b), (Dictionary of Modern Literary Russian Language, 1994).

ВЕДУЩИЙ, а я, е е. **1.** Идущий впереди; головной. *Ведущий самолет*. □ *Каждый из ведущих броненосцев больше всего осыпался неприятельскими снарядами.* Нов.-Прибой, Цусима. // *В знач. сущ.* В е д у щ и й, е г о, *м,* В е д у щ а я, е й, *ж.* **а)** Тот, кто ведет, возглавляя какую-л. группу. *В тайге заблудиться легко, если к тому же окажется самонадеянным и не очень опытным ведущим.* Ворон. Волев. прием. *Последнее время Драченко ходит у нас в качестве разведчика. А теперь думаем посылать его ведущим.* Кудреватых, Стр. нашей жизни. **б)** Летчик, летящий на головном самолете, направляющий действия своего ведомого. *За те немногие минуты, что они провели в воздухе, Петров сумел оценить уверенную и поистине мастерскую манеру полета своего ведущего.* Б. Полев. Пов. о наст. чел. ...

We associate the secondary sense maker set {♦, ◊} in **DLR-DAR** with the corresponding set of markers {//, ◊} in **DMLRL**, relying mainly on the facts supported by the current stage of our **DMLRL** investigation: lexical-semantics *subsumption* properties of the senses involved by the corresponding sense markers, and the *semantic granularity* of the senses within the dictionary entry text. The following dependency hypergraph of the *primary*, *secondary*, and literal enumeration for sense marker classes in **DMLRL** is proposed in Figure 1 (Curteanu et al., 2012).

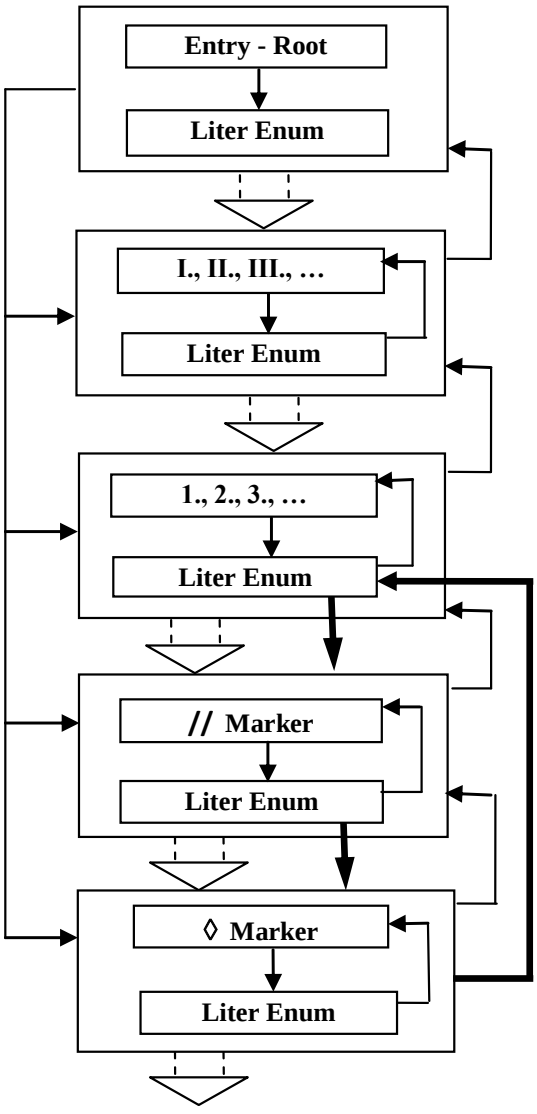

Figure 1: The dependency hypergraph at primary and secondary sense marker classes in **DMLRL**

The problem of *literal enumeration* in **DMLRL** is, for the moment, the most challenging one concerning the sense dependencies introduced by **DMLRL** marker classes of the *second* SCD *parsing configuration*. This is because one may find entry samples that display a recursion between the *literal enumeration* and the *secondary senses* "//" and "◊" (at least these markers). This level of recursion can be raised towards the higher (primary) senses, or may step down to the atomic senses / definitions. The solution of reducing these recursions to a finite number of cycles, and disambiguation of the cyclic application of secondary sense markers and of the literal enumeration should be consistent with the

possible extension of the literal enumeration recursion to the higher or lower levels on the **DMLRL** hypergraph of marker class dependencies, pre-established for **DMLRL** (Fig. 1). The entry excerpts of "**БЫ**" [7: 844] illustrate this special situation, displayed in **Example 5.3** (Section 5) with spans of both the entry text and its parsed sense tree.

4. DMLRL Dependency Hypergraph of the Third SCD Configuration

Trying to keep as close and unitary as possible to the already existing lexicographic SCD modeling of the atomic definitions and examples for **DLR-DAR**, **TLF**, **DWB-GWB**, we outline the following **DMLRL** atomic senses definitions, examples-to-definitions, their sense markers and dependencies (Curteanu et al., 2012), (Curteanu et al., 2008), (Curteanu et al., 2010b), (Curteanu et al., 2010a). Each atomic sense definition is classified accordingly to the taxonomies proposed in previous papers and in (Curteanu et al., 2012), (Curteanu et al., 2010a), (Curteanu et al., 2010b), outlined here as *obligatory* and *optional* definitions (the first taxonomy), completed with *autonomous* and *contingent* definitions (the second taxonomy). We found (until the current stage of **DMLRL** lexicographic investigation) that it is necessary to operate with the following **DMLRL** atomic sense definitions and examples-to-definitions:

(D1) *MorfDef* (*Morphological*-derivation *Definition*); *Obligatory* and *Contingent* definition. When non-present, it should to be inherited from a regent or a higher-level sense. It is written with Times New Roman, Italics font. **(D2)** *SpecDef* (*Specification Definition*); *Contingent* and *Optional* definition. This is a *modifying* type definition applied in a cyclic or recursive manner to an autonomous definition. Written with Times New Roman, Italic font, the *SpecDef* expressions are usually abbreviated, reserved words. **(D3)** *SpSpecDef* (*Spaced Specification Definition*); Similar to *SpecDef* but written with spaced-characters. *Internal reference* (inside the same entry), *external reference* (to another **DMLRL** entry), *morphological suffixes* or *lexical variants* are written, in certain contexts, with spaced-characters. **(D4)** *RegDef* (*Regular Definition*); *Autonomous* and *Obligatory* definition. It is written with Times New Roman, Regular font. This is the basic scheme to describe the lexicographic meaning of an entry sense / subsense in **DMLRL** (and in the largest majority of other dictionaries). **(D5)** *TildaDef* (*Tilda-marker Definition*); *Autonomous* and *Optional* definition. The *TildaDef* definition is introduced by the **DMLRL**-specific marker *tilda* "**~**", being written in *bold italics* Times New Roman font, at the end of a sense / subsense description. The *TildaDef* package is not *NewPrg*-marked when attached to an entry having just the sense-root or to the proper subsenses of the word-lemma, but *NewPrg*-marked when assigned to the root-sense of an entry with proper subsenses. **(D6)** *RefDef* (*Reference Definition*); *Autonomous* and *Optional* definition. *RefDefs* are *external references*, frequently met as constitutive part of the *TildaDef* definition package, or *internal references* to an entry sense (including its root-sense) inside which such a reference is used. We notice that all *RefDefs* are *SpSpecDefs* but the reverse is not true. **(D7)** *LexVarDef* (*Lexical-Variant Definition*); *Contingent* and *Optional* definition, used to provide lexical variation(s) to the entry-word. It is written with *bolded font*, and met within a *MorfDef*, when the meaning of the lexical variant is the same as that of the word-lemma. **(E8)** *DictExem* (*Dictionary* authors' *Example*); *Contingent* and *Optional* example. This type of examples is given by the **DMLRL** dictionary authors to support

the refinement of semantic explanations to autonomous definitions assigned to entry senses. *DictExems* are separated by the *traverse* "□" (empty square) **DMLRL** specific marker from *DefExems* that follow. **(E9)** *DefExem* (*Definition Example*); *Contingent* and *Optional* example. It is very similar to *DefExem* in **DLR-DAR** dictionaries (Curteanu et al., 2008), (Curteanu et al., 2010a), (Curteanu et al., 2010b). This type of examples represents quotations, text excerpts from bibliographic sources, with the role of refining and completing the meanings of autonomous definition(s) assigned to a (sub)sense of an entry. To each *DefExem* is associated a *sigle*, *i.e.* the *reference* of *DefExem* citation excerpt to its *bibliographic source*(*s*) or authorship.

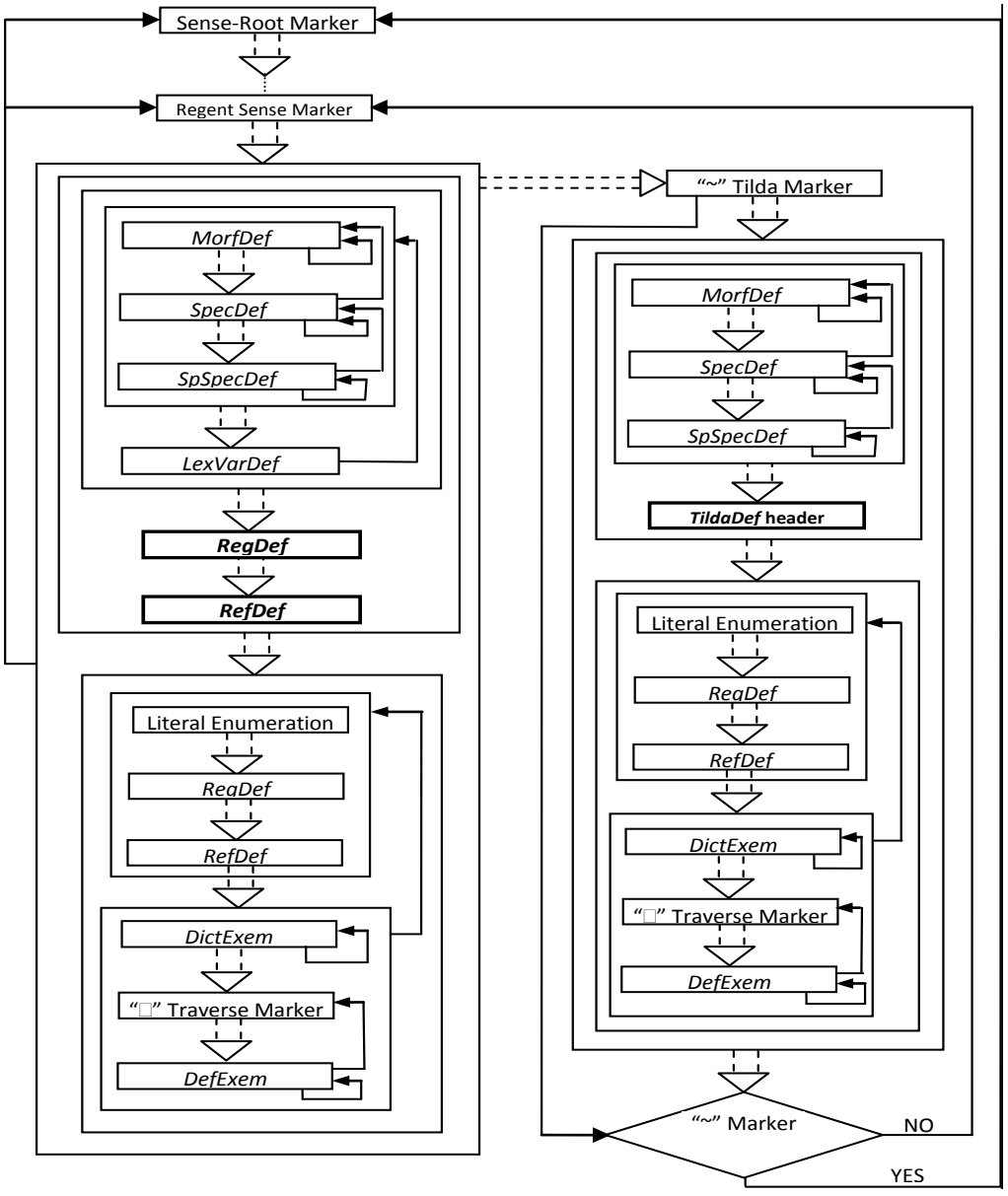

Figure 2. *RegDef* block and *TildaDef* block sequences and dependencies for **DMLRL** atomic sense definitions and examples-to-definitions

The dependency hypergraph containing the sequences and dependencies of the main *RegDef* and *TildaDef* autonomous definition blocks for **DMLRL** atomic sense definitions and examples-to-definitions, corresponding to the *third* SCD *parsing configuration* (*for the first time* at this modeling level), is presented in Figure 2.

5. DMLRL Parsing Examples and Comments

Each example gives the **DMLRL** entry, its sense marker sequence, and some excerpts of the parsed sense tree (XML format), with significant sense dependencies involved.

Example 5.1. The **DMLRL** entry **БЫТЬ** shows the sense marker subsequence "II. 1. ◊… 2. а) б) в) 3." involving primary, secondary, and literal enumeration dependencies:

<entry>
<list>БЫТЬ, I. 1. ◊ ◊ // ◊ 2. ◊ // // 3. ◊ ◊ // 4. ◊ ◊ ◊ 5. 6. ◊ ◊ II. 1. ◊ ◊ ◊ ◊ ◊ ◊ 2. а) б) в) 3. 4. 5. ◊ n-39</list>
<sense value="БЫТЬ, " class="0">
<definition>наст. не употр. кроме 3 л. ед. есть и (устар.) 3 л. мн. суть, буд. буду, будешь, прош. был, была, было (с отрицанием: не был, не была, не было), повел. будь (те), прич. действ. прош. бывший, деепр. будучи, несов., неперех. </definition>
<sense value="I." class="2">
<definition> Как самостоятельный глагол означает: </definition>
<sense value="1." class="4">
… … … … … … … … … .. … … … … … … … … … … … .. … … … … … … … … …
<sense value="II." class="2">
<definition> В знач. вспомогательного глагола или связки. </definition>
<sense value="1." class="4">
<definition> Употр. в составном именном сказуемом (в настоящем времени обычно опускается). В то время был еще жених Ее супруг. Пушк. Е. О. … … … </definition>
<sense value="◊" class="8">
<definition><spaced> Б у д ь!</spaced>Прост. Употр. при прощании как усеченная форма от будь здоров! - Проводишь? - Вопрос! - воскликнул Евгений. - Конечно, с удовольствием...
… … … … … … … … … .. … … … … … … … … … … … .. … … … … … … … … …
</sense>
</sense>
<sense value="2." class="4">
<definition> Употр. в сложном сказуемом. </definition>
<sense value="а)" class="5">
<definition> С кратким прилагательным должен.- Наши отношения должны быть такие, какие они всегда были. Л. Толст. Анна Карен. … … Я должен был сказать жестко и непреклонно: - Нет, я не согласен. Кузьмин, Круг царя Соломона. </definition>
</sense>
<sense value="б)" class="5">
<definition> С предикативами (можно, надо, нужно и т. п.). Быть можно дельным человеком И думать о красе ногтей. Пушк. Е. О. Надо было видеть, с какой радостью наш кинооператор выскочил из дома, едва услышав гул машины. Ефрем. Дорога ветров, </definition>
</sense>
<sense value="в) " class="5">
<definition> С личными формами глаголов мочь, хотеть.- Он хочет быть как мы цыганом; Его преследует закон. Пушк. Цыганы. [Гражданин:] Поэтом можешь ты не быть, Но гражданином быть обязан. Некр. Поэт и Гражданин.</definition>
</sense>
</sense>
<sense value="3." class="4">
<definition> Употр. в формах будущего времени с неопределенной формой глаголов несовершенного вида для образования будущего времени изъявительного наклонения.- … … …
… … … … … … … … … .. … … … … … … … … … … … .. … … … … … … … … …

Example 5.2. The entry **БРАТЬ** contains the sense marker subsequence "7. ◊ ◊ // ◊ // ◊ ◊ 8." with its dependency interplay between primary and secondary **DMLRL** senses.

<entry>
<list>БРАТЬ, 1. ◊ ◊ ◊ ◊ ◊ ◊ ◊ ◊ ◊ ◊ // ◊ ◊ // ◊ // ◊ // // // 2. ◊ ◊ // // 3. ◊ // 4. ◊ ◊ ◊ ◊ ◊ ◊ // ◊ // // // // // ◊ // // 5. // 6. // ◊ // // // 7. ◊ ◊ // ◊ // ◊ ◊ 8. ◊ ◊ // // 9. ◊ ◊ ◊ 10. ◊ ◊ ◊ ◊ ◊ ◊ ◊ ◊ ◊ 11. ◊ ◊ ◊ ◊ ◊ // ◊ ◊ // ◊ // ◊ // ◊ 12. ◊ 13. ◊ // // 14. ◊ ◊ ◊ ◊ ◊ ◊ ◊ ◊ ◊ ◊ ◊ ◊ а) б) в) ◊ n-120</list>
<sense value="БРАТЬ," class="0">
<definition><spaced> б е р у,</spaced><spaced> б е р ё ш ь,</spaced> прош. <spaced> б р а л, л а</spaced>, <spaced> л о</spaced>, несов., перех. и неперех., (сов. <spaced> в з я т ь).</spaced></definition>
<sense value="1." class="4">
… … … … … … … … … .. … … … … … … … … … … … .. … … … … … … … … …
<sense value="7." class="4">
<definition> Перех. Овладевать чём-л., захватывать что-л. Брать крепость, город. □ Шесть раз мы уступали поле Врагу и бреши у него. Лерм. Поле Бородина. На всех каменных зданиях по углам были поставлены пулеметы Нужно было брать дом за домом. Вершит. Люди с чист, совестью. </definition>
<sense value="◊" class="8">
<definition><spaced> Б р а т ь</spaced> штурмом, приступом; <spaced> б р а т ь</spaced> с боем. Корнилов настоял: продолжать поход на Екатеринодар, чтобы брать город штурмом. А. Н. Толст. 18-й год. </definition>
</sense>
<sense value="◊" class="8">
<definition><spaced> Б р а т ь</spaced> власть, управление <spaced> и т.</spaced> д. в свои руки. … … … </definition>
</sense>
<sense value="//" class="6">
<definition>Разг. Выигрывать (ход, взятку в карточной игре, фигуру в шахматном, шашечном состязании).- Мне ходить..- Нельзя ли погодить? - Беру.- Кругом нас обыграла. Пушк. Наброски к замыслу о Фаусте. </definition>
<sense value="◊" class="8">
<definition><spaced> Б р а т ь </spaced>приз. Отец любил вспоминать, какие призы брал на скачках, пока не сломал руку. Атаров, А я люблю лошадь. </definition>
</sense>
</sense>
<sense value="//" class="6">
<definition>Окружать чем-л., заключать во что-л. </definition>
<sense value="◊" class="8">
<definition><spaced> Б р а т ь</spaced> во что-л. Брать слово в скобки. Брать цитату в кавычки. </definition>
</sense>
<sense value="◊" class="8">
<definition><spaced> Б р а т ь</spaced> в окружение, в кольцо. Наш I-й Украинский фронт своим правым флангом обходил Берлин, беря его в кольцо. Б. Полев. Самые памятные. ~ Брать кого-л. в клещи. См. <spaced> К л е щ и.</spaced>
… … … … … … … … … .. … … … … … … … … … … … .. … … … … … … … … …

Example 5.3. The entry **БЫ** [7: 844] contains the marker subsequence "3. а) ◊ ◊ // ◊ ◊ б) ◊ ◊ ◊ в) г) ◊ ◊", which shows (in the entry text below) the occurrence of **DMLRL** secondary senses under the literal enumeration, whose parsing solution is based on the *closing enumeration condition* (Curteanu et al., 2010b), (Curteanu et al., 2012), used also to parse the entry **CAL** of **DAR** (Curteanu et al., 2010b).

БЫ (сокращенно **Б**), *частица.* В сочетании с глаголами в форме прошедшего времени образует сослагательное наклонение. **1.** Употр. для обозначения предположительной (или условно-предположительной) возможности действия.

… … … … … … … … … .. … … … … … … … … … … … .. … … … … … … … … …

3. Обозначает различные оттенки желаемости действия; **а)** Собственно желаемость. *Учился бы сын. Были бы дети здоровы.* ◊ Если б ы , когда б ы , хоть б ы и т. п. О, *если бы когда-нибудь*

Сбылась поэта сновиденья! Пушк. Посл. к Юдину. [Николка:] *Хоть бы дивизион наш был скорее готов.* Булгаков, Дни Турб. ◊ С неопр. ф. глаг. *Полететь бы пташечке К синю морю; Убежать бы молодцу в лес дремучий.* Дельв. Пела, пела пташечка.. [Настя:] *Ах, тетенька, голубок! Вот бы поймать!* А. Остр. Не было ни гроша…**—** *Жара, дедушка Лодыжкин .. Нет никакого терпения! Искупаться бы!* Купр. Бел. пудель. // Употр. для выражения опасения по поводу какого-л. нежелательного действия (с отрицанием). *Не заболел бы он.* ◊ С неопр. ф. глаг., имеющей перед собой отрицание. **—** *Гляди, — говорю, — бабочка, не кусать бы тебе локтя! Так-таки оно все на мое вышло.* Леск. Воительница. ◊ Только б ы (б) не. — *По мне жена как хочешь одевайся, .. только б не каждый месяц заказывала себе новые платья, а прежние бросала новешенькие.* Пушк. Арап Петра Вел. [Варя:] *Не опоздать бы только к поезду.* Чех. Вишн. сад. **б)** Пожелание. *Условие я бы предпочел не подписывать.* Л. Толст. Письмо А. Ф. Марксу, 27 марта 1899. ◊ С неопр. ф. глаг. *Поохотиться бы по-настоящему, на коня бы денег добыть, — мечтал старик.* Г. Марков, Строговы. ◊ В сочетании с предикативными наречиями со знач. долженствования, необходимости, возможности. … … ◊ Только б ы (б), лишь б ы , Употр. со знач. желательности действия. [Скалозуб:] *Мне только бы досталось в генералы.* Гриб. Горе от ума. **в)** Желание-просьба, совет или предложение (обычно при мест. 2л.). [Марина:] *И чего засуетился? Сидел бы:* Чех. Дядя Ваня. — *Пошел бы ты к ним счетоводом, полковник.* Павлен. Счастье. — *Ты бы, Сережа, все-таки поговорил с Лидией:* Пришв. Кащ. цепь. **г)** Желаемость целесообразного и полезного действия. ◊ С неопр. ф, глаг. *Вам б ы вступиться за Павла-то! — воскликнула мать, вставая. — Ведь он ради всех пошел.* М. Горький, Мать. ◊ С неопр. ф. глаг., имеющей перед собой отрицание. [Лиза:] *А вам, искателям невест, Не нежиться и не зевать б ы*. Гриб, Горе от ума.

~ Во что б ы то ни стало. См. С т а т ь . ***Как б ы не так.*** См. К а к . ***Кто б ы ни б ы л , что б ы ни было, как б ы то ни было.*** *См.* Б ы т ь … … …

— Срезневский: б ы ; Лекс. 1762: б ы .

```
<entry>
 <list>БЫ 1. ◊ ◊ ◊ ◊ ◊ 2. 3. а) ◊ ◊ // ◊ ◊ б) ◊ ◊ ◊ в) г) ◊ ◊ n-23</list>
 <sense value="БЫ" class="0">
  <definition> (сокращенно <b>Б</b>), частица. В сочетании с глаголами в форме прошедшего времени образует сослагательное наклонение. </definition>
  <sense value="1." class="4">
… … … … … … … … … … .. … … … … … … … … … … … .. … … … … … … … … …
  <sense value="3." class="4">
   <definition> Обозначает различные оттенки желаемости действия; </definition>
   <sense value="а)" class="5">
    <definition> Собственно желаемость. Учился бы сын. Были бы дети здоровы. </definition>
    <sense value="◊" class="8">
     <definition> Если <spaced> б ы</spaced>, когда <spaced> б ы</spaced>, хоть <spaced> б ы</spaced><spaced> и</spaced> т. п. О, если бы когда-нибудь Сбылась поэта сновиденья! Пушк. Посл. к Юдину. [Николка:] Хоть бы дивизион наш был скорее готов. Булгаков, Дни Турб. </definition>
    </sense>
    <sense value="◊" class="8">
     <definition> С неопр. ф. глаг. Полететь бы пташечке К синю морю; Убежать бы молодцу в лес дремучий. Дельв. Пела, пела пташечка.. … … </definition>
    </sense>
    <sense value="//" class="6">
     <definition> Употр. для выражения опасения по поводу … … … </definition>
     <sense value="◊" class="8">
      <definition> С неопр. ф. глаг., имеющей перед собой отрицание. <b>- </b>Гляди, - говорю, - бабочка, не кусать бы тебе локтя! Так-таки оно все на мое вышло. Леск. Воительница. </definition>
     </sense>
     <sense value="◊" class="8">
      <definition> Только <spaced> б ы</spaced> (б) не. - По мне жена как хочешь одевайся, .. только б не каждый месяц … … … </definition>
     </sense>
    </sense>
   </sense>
   <sense value="б)" class="5">
```

<definition> Пожелание. Условие я бы предпочел не подписывать. Л. Толст. Письмо А. Ф. Марксу, 27 марта 1899. </definition>
<sense value="◊" class="8">
<definition> С неопр. ф. глаг. Поохотиться бы по-настоящему, на коня бы денег добыть, - мечтал старик. Г. Марков, Строговы. </definition>
</sense>
<sense value="◊" class="8">
<definition> В сочетании с предикативными наречиями со знач. долженствования, необходимости, возможности. … … … </definition>
</sense>
<sense value="◊" class="8">
<definition> Только <spaced> б ы</spaced> (б), лишь бы, Употр. со знач. желательности действия. [Скалозуб:] Мне только бы досталось в генералы. Гриб. Горе от ума. </definition>
</sense>
</sense>
<sense value="в)" class="5">
<definition> Желание-просьба, совет или предложение…. … … … </definition>
</sense>
<sense value="г)" class="5">
<definition> Желаемость целесообразного и полезного действия. </definition>
<sense value="◊" class="8">
<definition> С неопр. ф, глаг. Вам бы вступиться за Павла-то! - воскликнула мать, вставая. - Ведь он ради всех пошел. М. Горький, Мать. </definition>
</sense>
<sense value="◊" class="8">
<definition> С неопр. ф. глаг., имеющей перед собой отрицание. [Лиза:] А вам, искателям невест, Не нежиться и не зевать бы. Гриб, Горе от ума. ~ Во что бы то ни стало. См. <spaced> С т а т ь.</spaced> Как бы не так. См. <spaced> К а к.</spaced> … … … Хоть бы хны. См. <spaced> Х о т ь.</spaced> Хоть бы что. См. <spaced> Х о т ь.</spaced></definition>
</sense>
</sense>
</sense>
</sense>
<EtymologicalPart>
- Срезневский: <spaced> б ы</spaced>; Лекс. 1762: <spaced> б ы</spaced>.
</EtymologicalPart>
</entry>

6. Conclusions

(a) DMLRL parsing problems solved: sense dependencies of the *second* SCD *configuration* (hypergraph in Fig. 1), *i.e.* primary, secondary, literal enumeration, and mutual calling between literal enumeration and secondary senses (**Ex. 5.3**). For the 35 **DMLRL** entries (of all sizes), the parser provided a very sound parsing percentage, at this level. **(b) DMLRL** still unsolved, but tractably solvable: homonymic entries (with prefix indexes, *e.g.* **БЫЧОК** [7: 860-861]), recognition of *TildaDef* definitions. **(c)** A more complex problem to be incorporated into the next version of **DMLRL** parser is to solve the dependencies of the *third* SCD *configuration* (hypergraph in Fig. 2), interconnected with the second one (hypergraph in Fig. 1). **(d)** The present parsing experiment and **DMLRL** modeling (Curteanu et al., 2012), integrated with those for the other *five* largest thesaurus-dictionaries exposed in (Curteanu et al., 2010b), represent the computational basis to design a new, optimal and comprehensible DTD for dictionaries within the TEI standard (XCES TEI, 2007), based on the lexicographic modeling and parsing technology of SCD configurations.

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.

Curteanu, N., Moruz, A., Trandabăț, D. (2008). Extracting Sense Trees from the Romanian Thesaurus by Sense Segmentation & Dependency Parsing. *Proceedings of CogAlex-I Workshop, COLING 2008*, Manchester, United Kingdom, ISBN 978-1-905593-56-9, 55-63.

Curteanu, N., Moruz, A., Trandabăț, D. (2010a). Comparative Parsing of the Romanian, French, and German Thesaurus-Dictionaries. *In Proceedings of the Workshop on Linguistic Resources and Instrument for Romanian Language Processing*, (Ed. A. Iftene, H.N. Teodorescu, D. Cristea, D. Tufiș), Editura Univ. "Al.I. Cuza" Iași, ISSN: 1843-911X, (in Romanian), 113-122.

Curteanu, N., Trandabăț, D., Moruz, A. (2010b): An Optimal and Portable Parsing Method for Romanian, French, and German Large Dictionaries. *Proceedings of COGALEX-II Workshop*, COLING-2010, Beijing, China, 38-47.

Curteanu, N., Cojocaru, S., Burcă, E. (2012). Parsing the Dictionary of Modern Literary Russian Language with the Method of SCD Configurations. The Lexicographic Modeling. *Computer Science Journal of Moldova*, Academy of Sciences of Moldova, 20: 1-2 (part I. + II.).

Das Woerterbuch-Netz (2010). http://germazope.uni-trier.de/Projects/WBB/woerterbuecher/

Dictionary of Modern Literary Russian Language (20 volumes - 1994). Словарь современного русского литературного языка. В 20 томах. Издательство: М.: Русский язык; Издание 2-е, перераб. и доп. 864 страниц; 1991 - 1994 г. ISBN: 5-200-01068-3 (in Rusian).

Le Trésor de la Langue Française informatisé (2010). http://atilf.atilf.fr/tlf.htm

Pușcariu, S. *et al.* (1906). Dictionary of the Romanian Language (Dictionary of the Romanian Academy – **DAR**), Bucharest, Edition 1940 (old format).

XCES TEI Standard, Variant P5 (2007): http://www.tei-c.org/Guidelines/P5/

INDEX OF AUTHORS