Adding a Syntactic Annotation Level to the Corpus of Contemporary Romanian Language

Andrei Scutelnicu[1,2], Cătălina Mărănduc[1,3], Dan Cristea[1,2]

[1] Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași
[2] Institute of Computer Science of the Romanian Academy, Iași branch
[3] "Iorgu Iordan – Al. Rosetti" Institute of Linguistics, Romanian Academy, Bucharest

{andreis, catalina.maranduc, dcristea}@info.uaic.ro

Abstract

In this paper we present an experiment of augmenting the Corpus of Contemporary Romanian Language (CoRoLa) with the syntactic level of annotations, which would allow users to address queries about the syntax of Romanian sentences, in the Universal Dependency model. After a short introduction of CoRoLa, we describe the treebanks used to train the dependency parser, we show the evaluation results and the process of upgrading CoRoLa with the new level of annotations. Out of three variants of parsers trained on manually built treebanks, the one displaying the best accuracy with respect to the recognition of heads and relations was chosen. A number of examples showing types of queries addressing the syntactic level are also presented.

Keywords: syntactic annotation, Romanian treebank, dependency corpus, Universal Dependency, MaltParser, CoRoLa, DeReKo, KoRAP, Poliqarp, DRuKoLa, EuReKo.

1. Introduction

Introducing syntactic annotation in existing corpora is useful for many endeavours: as training data for parsers, as a support layer for addressing syntactic queries to the corpus, as a source for extracting patterns of noun phrases, verb phrases and other types of sub-syntactic compounds, as a complement layer for extracting verb roles, semantic relations, etc. In this paper we describe the process of upgrading the Computational Corpus of Contemporary Romanian Language (CoRoLa[1]) with the syntax level.

At the end of the project, in November 2017, the CoRoLa Corpus had the following parameters:

- almost 400,000 files,
- around 1.26 billion tokens (including punctuation),
- approx. 900 million word occurrences,
- more than 3 million surface unique forms,
- 198,800 words with frequency higher than 50,
- 121,091 lemmas with frequency higher than 50,
- 2,346,546 unique lemma forms, out of which 2,136,391 were lowercase lemmas.

In CoRoLa each document is paired with a metadata file (marking title, authors, year of publication, publishing house, document style, domain, ISBN/ISSN, etc.). The annotations include segmentation to paragraphs, sentences and tokens, while lemmas, POS and morphological features are indicated for each token, and have been obtained with the NLPCube annotator, an end-to-end Natural Language Processing framework (Boroș et al., 2018). Based on recurrent neural networks, the framework[2] performs sentence splitting, tokenization, compound word expansion, lemmatization, tagging and parsing.

The main search frontend of CoRoLa is KorAP (Bański et al. 2012, 2013; Diewald and Margaretha, 2016). Designed and realised at the Leibniz Institute for the German Language[3] since 2011, KorAP, and its user interface Kalamar, were built with the intention to be used as the corpus analysis platform and query frontend for the Reference Corpus of the German Language, DeReKo[4], a corpus that in 2018 counted already 43 billion words (Kupietz et al. 2018). Kalamar's default query language is Poliqarp[5] (Przepiórkowski et al. 2004), which is both powerful for complex annotation queries and easy to use for non-specialists. Being based on regular expressions, Poliqarp allows the user to combine different features in the query, thus exploiting the internal structure of the tags that accompany the primary tokens. Examples are: queries addressing the lexical level, sensible to the orthographical form of (sequences of) words, including endings, prefixes and inner strings of characters, queries addressing the morphological level, regarding lemmas, part of speeches and any combination of features (in both morphosyntactic description tags and category tags), queries exploiting the metadata level, as well as any combinations of these levels. The infrastructure also allows the generation of sub-corpora that observe combinations of search constraints.

Our work is a follow up pursuit of a German-Romanian initiative, the DRuKoLa project[6] (Kupietz et al., 2019), which aimed to create the linguistic data[7] and the

[1] Priority project of the Romanian Academy (RA), realised in collaboration by two institutes of RA: the Research Institute in Artificial Intelligence, in Bucharest, and the Institute of Computer Science, in Iași. The query frontend, the project members and a comprehensive list of papers on CoRoLa can be found at http://corola.racai.ro.

[2] https://github.com/adobe/NLP-Cube

[3] *Leibniz-Institut für Deutsche Sprache* (IDS)

[4] *Deutsches Referenzkorpus*

[5] Created in the *Instytut Podstaw Informatyki Polskiej Akademii Nauk* (http://www.ipipan.waw.pl).

[6] A project funded by the Alexander von Humboldt Foundation, which run in the period 2016–2018. DRuKoLa is an acronym from *Deutsch-Rumänische korpuslinguistische Analyse*.

[7] Actually, the Romanian language data, since the German corpus, DeReKo, was running and in continuous development even before 2010, when it counted 4 billion words – according to Kupietz, and Lüngen (2014).

technological basis for performing German-Romanian contrastive linguistics analyses, itself part of a larger international undertaking, having as goal the development of a common platform of comparable corpora, EuReKo (Kupietz et al., 2017).

2. Training the Parsers

We use the MaltParser tool (Nivre et al., 2007; Gómez-Rodríguez and Nivre, 2010) to train the syntactic parser. Three gold treebanks were used during training. The first represents a Romanian translation of a part of George Orwell's novel "1984", with 900 sentences (referred to in the following as the ORWELL set). The annotations, done manually by one of the authors, were in line with the Universal Dependency (UD) conventions[8] (Nivre et al., 2016). The second is a treebank of 9,524 sentences developed at RACAI (Barbu Mititelu et al., 2016), also following the UD conventions (here called the RACAI[9] set). Finally, the third is a treebank developed at UAIC[10], following conventions specific to the Romanian language, which are richer in details than those from UD (Mărănduc and Perez, 2016), out of which we have extracted 8,444 sentences that do not include neither old documents nor chats (we will refer here to this corpus as the UAIC set). Apart from a different set of relations names, other major differences between the two sets of conventions are related to structure and granularity. For instance, relational words in UD are subordinated to the full-semantic word, while in UAIC they are placed above them (see Figure 1[11]). Also, UAIC has 14 types of circumstantial modifiers, while only one type is used in UD.

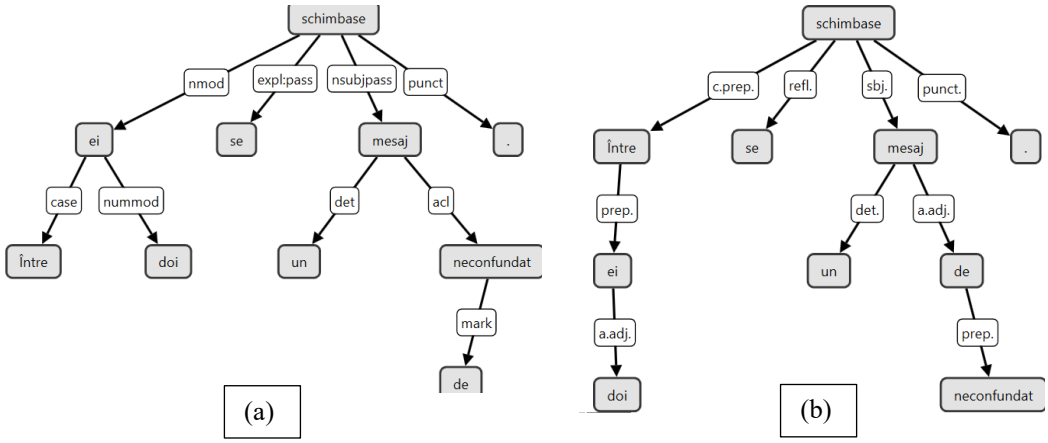

Figure 1: Analysis of the sentence *Între ei doi se schimbase un mesaj de necomfundat.* (*Between them two was exchanged an unmistakable message.* – topic kept). The UD (a) and the UAIC (b) notation of prepositions (*Între* and *de*): they are headed by the semantic word in UD (*ei*, respectively *neconfundat*) and they head the semantic word in UAIC.

Different front-ends have been used to develop these gold treebanks. Sometimes they have been expanded iteratively, using a bootstrapping approach: at each step, an initial corpus, manually annotated, was used to train the parser and then the errors were corrected, producing the next step of the corpus; see details in (Popa, 2010; Perez, 2014). The annotations created by the interfaces are represented as XML files (Moruz, 2008; Mărănduc et al., 2017) and a script was developed for transforming the gold files from the XML format into the CONLL-U format, used by MaltParser. Then, the MaltParser service was called and its output was converted back into XML, the format supported by both the front-ends and the CoRoLa corpus.

Moreover, a script named Treeops[12] (Mărănduc et al., 2018) was used for transforming the Romanian annotation format into the UD one (mainly by performing surgery operations on the structure, merging more relations into one and renaming relations). Treeops runs error free.

In order to derive training data for the classifier, an oracle is used to reconstruct a valid transition sequence for every dependency structure in the training set. The learning problem in transition-based parsing, as implemented in MaltParser, is to induce a classifier for predicting the next transition, given a feature representation of the current parser configuration. The training is optimised using the LIBLINEAR built-in machine learning package[13].

3. Evaluating the Parsers

Before actually upgrading the corpus on the public server with the new level of annotations, a number of tests were performed locally in order to select the most appropriate training data, to prove the accuracy of the new level of annotation and to test how it responds to queries.

All evaluations were done by using a 10-fold strategy for assessing the accuracy of the dependency parser, actually by comparing its output against parts of the gold corpora. As already shown, we used three different gold corpora, two respecting the UD annotation format (ORWELL and RACAI) and one adopting stipulations specific to the Romanian language (UAIC). We notice as well that, at the moment these experiments were performed, the three mentioned gold corpora we used to train the parser were not yet part of CoRoLa. In principle, at least, there are no major differences between the criteria used in gathering the

[8] https://universaldependencies.org/u/overview/syntax.html
[9] Romanian Academy Institute for Artificial Intelligence "Mihai Drăgănescu"
[10] University "Alexandru Ioan Cuza" of Iași
[11] All figures of dependency trees are generated with Treebank Annotator (Mărănduc et al., 2017)
[12] https://ufal.mff.cuni.cz/tlt16/
[13] http://www.maltparser.org/api/index.html

texts for inclusion in CoRoLa, as a corpus of contemporary Romanian, and the texts used for training the parsers, which would hamper to include in CoRoLa also these textual data. Indeed, literary styles, domains, years of writing and other criteria are rather similar, and IPR constraints are also observed, so this will be the next step to proceed.

The accuracy of the parser for different training data is presented in Table 1. The results reported for UAIC refer to the original tag set, not the version mapped to universal dependencies.

The treebank	Head	Relation	Average
ORWELL (UD)	0.896	0.866	0.881
RACAI (UD)	0.642	0.687	0.665
UAIC (non-UD)	0.881	0.910	0.896

Table 1: Accuracy (number of true positives out of the total number of heads or relations) of the dependency parser trained on different gold treebanks. The last column shows the average of the preceding two numbers

We did also a comparison with another dependency parser, which runs as a component of the NLP-Cube, evaluated in CoNLL's "Multilingual Parsing from Raw Text to Universal Dependencies 2018" Shared Task (Boroș et al., 2018). The accuracy of the parser, as reported in the competition, showed lower values than our UAIC-trained parser: for the head – an accuracy of 0.850 and for the syntactic relation – 0.701 (with an average of 0.775).

4. Upgrading the Syntactic Level in CoRoLa

Having these results, it became clear that the best choice for the syntactic annotation of CoRoLa is to use the solution given by the MaltParser tool trained on the non-UD conventions (the UAIC corpus). Let us also note that we can think of two different syntactic annotations of CoRoLa: one following the Romanian conventions and the second – the UD conventions.

To proceed with the addition of this new annotation level in the query platform of the CoRoLa corpus, the steps we must follow are: 1. transposing of already annotated text (token, POS, lemma) from the XML format into the CONLL-U format; 2. parsing the corpus with MaltParser trained on the UAIC treebank; 3. transforming the CONLL-U format that is returned in output by this process back into XML, and 4. converting this format to the one accepted by KoRAP[14], the query platform of the corpus. This pipeline will upgrade CoRoLa with the UAIC syntactic format. To take advantage of the better accuracy of the parser when trained with UAIC data, in order to build the variant following the UD format, a conversion from the UAIC format into the UD format (actually, a simplification) is preferred to the solution of directly adopting a UD-trained parser. The supplementary conversion should be introduced as a step 2', included between steps 2 and 3. Figure 2 shows an extract from the corpus including both UD and UAIC attributes.

```
<S id="1" offset="0">
 <W LEMMA="lui" MSD="Tf-so" POS="DET"
deprel-ud="det" head-ud="1.2" deprel-
uaic="det." head-uaic="1.2"
id="1.1">Lui</W>
 <W LEMMA="Winston" MSD="Np" POS="PROPN"
deprel-ud="iobj" head-ud="1.4" deprel-
uaic="c.i." head-uaic="1.4"
id="1.2">Winston</W>
 <W LEMMA="el" MSD="Pp3-sd--------w"
POS="PRON" deprel-ud="expl" head-ud="1.4"
deprel-uaic="c.i." head-uaic="1.4"
id="1.3">îi</W>
 <W LEMMA="displăcea" MSD="Vmil3s"
POS="VERB" deprel-ud="root" head-ud="1.0"
deprel-uaic="null" head-uaic="1.0"
id="1.4">displăcuse</W>
 <W LEMMA="fată" MSD="Ncfsry" POS="NOUN"
deprel-ud="nsubj" head-ud="1.4" deprel-
uaic="sbj." head-uaic="1.4"
id="1.5">fata</W>
 <W LEMMA="acesta" MSD="Dd3fsr---o"
POS="DET" deprel-ud="det" head-ud="1.5"
deprel-uaic="a.adj." head-uaic="1.5"
id="1.6">asta</W>
…
</S>
```

Figure 2: Concatenation of attributes for head and relation in UD and UAIC notation, for the segment *Lui Winston îi displăcuse fata asta…* (*Winston had disliked this girl…*)

We are currently working to annotate the whole CoRoLa corpus within the described technology. Thus, the syntax level of CoRoLa will become accessible through KoRAP, in the same way this GUI allows addressing queries referring the syntax level in DeReKo, the German reference corpus.

5. Querying the Syntactic Level of CoRoLa

When this endeavour will be finished, linguists will be able to search remotely for verbal, nominal or other kinds of dependencies, querying the corpus for evidences of language use that touch controversary syntactic issues. Some examples follow.

Studying linearization of attributive adjectives inside the nominal phrase, linguists try to answer the question: is it that sequences of attributive adjectives are strictly ordered, according to a functional projections rule that sees them as cognitive categories (Sproat and Shih, 1988; Cornilescu and Cosma, 2019)?

As shown in Cristea et al. (2019), inventorying types of configurations of syntactic subordinates that a particular word can have is important in the process of elaboration of dictionaries of verbal patterns. As such dictionaries put in evidence typical syntactic-semantic structures for verbs (Levin, 1993; Pană Dindelegan, 1974; Barbu Mititelu,

[14] https://korap.ids-mannheim.de/

2018), they are useful to both linguists and computational linguists. The patterns revealed by searching the corpus could, for instance, be incorporated into a parser as constraints for determining the dependencies associated to particular words.

Another interesting search could be the second-degree dependencies of a word, i.e. the sub-tree linked to a certain word. One example are nested noun dependencies, which are second-degree dependencies that redefine the head (Figure 3).

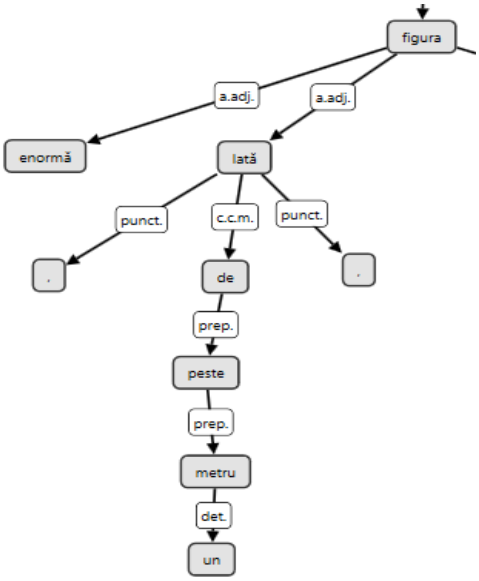

Figure 3: Analyses of the sentence segment: … *enormă, figura, lată de peste un metru,…* (… *enormous, the figure, flat for more than one meter,…*), in which *un metru* (*one meter*) is a sub-constituent of the adjective *lată* (*flat*), and this one – a sub-constituent of *figura* (*the figure*).

Other types of noun dependencies are the appositive dependencies, in which the apposition refers to the same entity as the nominal phrase preceding it. In many types of nominal dependencies, the dependent adds information, narrowing or widening the scope of the head.

A search can be restricted to put in evidence solely elements of the core of the clause (subject, direct or indirect object) or optional elements (adverbial modifiers, nominal modifiers, oblique dependencies). Both situations when the dependencies are expressed by a word or by a clause can be evidenced through a query. Searching the corpus for patterns that relate the complexity of construction of the dependent in correlation with its head can configure parser constraints for future enhancements.

In researches of pragmatic linguistics one can be interested to know the actors involved in a communication act, and the vocative clearly puts in evidence one such direct actor. Here, again, UD conventions differ from UAIC, in UD the words in the vocative case being clearly annotated as belonging to a different syntactic structure.

A whole class of queries and their relevance for linguistic research addressing Romanian syntax is described in (Cristea et al., 2019).

6. Conclusions

The paper presents an experiment of upgrading CoRoLa, the Corpus of Contemporary Romanian Language, with a new level of annotations. To the one already existent, which refers to morphology, the syntactic level will allow users to address queries in terms of heads and dependency relations. Syntax annotation in corpora, following the Universal Dependency model or other models, has been extensively described in the literature. This paper insists on the elaboration strategy of such a level of annotation by making heavy use of the NLP technology. To suggest the degree of applicability of the upgraded corpus, a number of possible queries addressing syntactic dependency structures of Romanian language are also sketched.

There remain many issues to be solved, on which we will concentrate in the near future. First, the technology that we describe here should be made functional on KoRAP, the query infrastructure that supports CoRoLa. Then, we ought to verify to what extend CoRoLa will be now even more useful as an empirical basis for syntactic studies, a question that has been uttered already when CoRoLa had no syntax inside (Cornilescu and Cosma, 2019) and which should be put again now when the corpus is enriched with the dependency syntax, while we are also aware that errors are inherently left behind by the annotation technology. Because of the extremely free word order in Romanian, it is possible for the syntactic head to be separated from some of its dependents by various other subordinates. However, being extremely difficult to parse, with an error rate still high for these long-distance dependencies, we might consider leaving them unparsed. Then the issue is to implement a filtering decision criterion. One possible criterion for this filter could be the computation of a confidence score for a parsed sentence, that would take into consideration individual accuracy scores for different types of relations and the relative word-head distance. The calibration of such a score can easily be done by comparing automatic parses with their equivalents in the gold files.

Finally, we want to keep our promise to augment the corpus itself with the texts used in training.

It is our sincere belief that the addition of this level of annotation will create new opportunities for Romanian language research. Moreover, we hope that the experience described here for the implementation of syntax in this large Romanian corpus could be inspiring for similar endeavours addressing other languages.

7. Acknowledgements

The work described in this paper was performed as part of the research plan of the NLP team in the Institute of Computer Science of the Romanian Academy, Iași branch, and was partially supported by a grant of the Ministry of Research and Innovation, Program PN-III-P1-1.2.-PCCDI, nr. 73/2018 - the ReTeRom project.

8. References

Bański, P., Fischer, P. M., Frick, E., Ketzan, E., Kupietz, M., Schnober, C., Schonefeld, O. and Witt, A. (2012). The New IDS Corpus Analysis Platform: Challenges and

Prospects. In N. Calzolari, K. Choukri, T. Declerck, M. Doğan, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis (Eds.), Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012), Istanbul, Turkey, May 2012. European Language Resources Association (ELRA), p. 2905–2911.

Bański, P., Bingel, J., Diewald, N., Frick, E., Hanl, M., Kupietz, M., Pęzik, P., Schnober, C. and Witt, A. (2013). KorAP: the new corpus analysis platform at IDS Mannheim. In Z. Vetulani, H. Uszkoreit (Eds.), Human Language Technologies as a Challenge for Computer Science and Linguistics, Proceedings of the 6th Language and Technology Conference, Poznań, Fundacja Uniwersytetuim. A., p. 586–587.

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

Barbu Mititelu, A.M. (2018). Valence Dictionary For Romanian Language In Printed Version And Xml Format. In V. Păiş, D. Gîfu, D. Trandabăţ, D. Cristea, D. Tufiş (eds), Proceedings of The 13th International Conference "Linguistic Resources And Tools For Processing The Romanian Language", Iaşi, November 22-23, p. 101–112.

Nivre, J. (2010). Statistical parsing. In Nitin Indurkhya and Fred J. Damerau (Eds.), Handbook of Natural Language Processing. Second Edition, CRC Press, Taylor and Francis Group, p. 237-266.

Boroș, T., Dumitrescu, Ș. and Burtică R. (2018). NLP-Cube: End-to-End Raw Text Processing With Neural Networks. 10.18653/v1/K18-2017.

Cristea, D., Diewald, N., Haja, G., Mărănduc, C., Barbu-Mititelu, V. and Onofrei, M. (2019). How to Find a Shining Needle in the Haystack. Querying CoRoLa: Solutions and Perspectives. In Revue Roumaine de Linguistique, București, vol. 64, no. 3, p. 279–292.

Cornilescu, A. and Cosma R. (2019). Linearization of attributive adjectives in Romanian. In Revue Roumaine de Linguistique, București, vol. 64, no. 3, p. 307–323.

Diewald, N. and Margaretha, E. (2016), Krill: KorAP search and analysis engine. In M. Kupietz, A. Geyken (Eds.), Corpus Linguistic Software Tools, Journal for language technology and computational linguistics (JLCL) 31 (1), Berlin, GSCL, p. 73–90.

Gómez-Rodríguez, C. and Nivre J. (2010). A transition-based parser for 2-planar dependency structures. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, p. 1492-1501.

Kupietz, M., Cosma, R. and Witt, A. (2019). The Drukola Project. In Revue Roumaine de Linguistique, Bucharest, vol. 64, no. 3., p. 255–263.

Kupietz, M., Witt, A., Bański, P., Tufiş, D., Cristea, D. and Váradi, T. (2017), EuReCo – Joining Forces for a European Reference Corpus as a sustainable base for cross-linguistic research. In P. Bański, M. Kupietz, H. Lüngen, P. Rayson, H. Biber, E. Breiteneder, S. Clematide, J. Mariani, M. Stevenson, T. Sick (Eds.), Proceedings of the Workshop on Challenges in the Management of Large Corpora and Big Data and Natural Language Processing, Mannheim, IDS, p. 15–19.

Kupietz, M. and Lüngen, H. (2014). Recent developments in DeReKo. In N. Calzolari, K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, S. Piperidis (Eds.), Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014), Reykjavik, p. 2378–2385.

Kupietz, M., Lüngen, H., Kamocki, P., and Witt, A. (2018). The German Reference Corpus DeReKo: New Developments – New Opportunities. In N. Calzolari, K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, T. Tokunaga (Eds.), Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, European Language Resources Association (ELRA), p. 4353–4360.

Levin, B. (1993). English Verb Class and Alternations: A Preliminary Investigation, University of Chicago Press.

Mărănduc, M. and Perez, C.-A. (2016). A Resource for the Written Romanian: the UAIC Dependency Treebank. In Proceedings of ConsILR, Mălini, 27-29 Oct., p. 79-90.

Mărănduc, C., Mititelu, C. and Bobicev, V. (2018). Syntactic Semantic Correspondence in Dependency Grammar. In Proceeding of 16th International Workshop on Treebanks and Linguistic Theories Prague, p. 167-180.

Mărănduc, C., Hociung, F., Bobicev, V. (2017). Treebank Annotator for multiple formats and conventions. In Proceedings of the 4th Conference of Mathematical and Computer Science Society of the Republic of Moldova, Chișinău, June 28 – July 2, p. 529-534.

Moruz, A. (2008). Developing a Functional Dependency Grammar (FDG) annotator for Romanian. Master thesis, A. I. Cuza University, Faculty of Computer Science, Iași.

Nivre, J., Hall, J., Nilsson, J. and Chanev, A. (2007). MaltParser: A language-independent system for data-driven dependency parsing. Natural Language Engineering, Cambridge University Press, volume 13, p. 95-135.

Nivre, J., de Marneffe, M.C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C.D., Mc Donald R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R. and Zeman, D. (2016). Universal Dependencies v1: A Multilingual Treebank Collection. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC-2016), European Language Resources Association (ELRA), Portoroz Slovenia, p. 1659–1666.

Pană Dindelegan, G. (1974). Sintaxa transformațională a grupului verbal în limba română, Bucureşti, Editura Academiei.

Perez, C.-A. (2014). Linguistic Resources for Natural Language Processing. Ph.D. dissertation, A. I. Cuza University, Faculty of Computer Science, Iași.

Popa, C. (2010). FDG Parser for Romanian language. Master thesis, A. I. Cuza University, Faculty of Computer Science.

Przepiórkowski, A., Krynicki, Z., Dębowski, L., Woliński, M., Janus, D. and Bański, P. (2004). A search tool for corpora with positional tagsets and ambiguities. In Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004, p. 1235–1238.

Sproat, R. and Shih, C. (1988). Prenominal Adjectival Ordering in English and Mandarin. In J. Blevins, J. Carter (Eds.), Proceedings of NELS 18, Amherst, MA: GLSA, vol. 2, p. 465–489.