A Data-Driven Dependency Parser for Romanian

Mihaela Călăcean and Joakim Nivre
Uppsala University
Department of Linguistics and Philology
E-mail: mcalacean@sdl.com
joakim.nivre@lingfil.uu.se

Abstract

We present the first data-driven dependency parser for Romanian, which has been developed using the MaltParser system and trained and evaluated on a dependency treebank for Romanian developed within the RORIC-LING project. The parser achieves a labeled attachment score of 88.6% (unlabeled 92.0%) when evaluated on held-out data from the treebank. We present a partial error analysis, focusing on accuracy for different parts of speech and dependencies of different length.

1 Introduction

Data-driven methods for syntactic parsing currently enjoy widespread popularity, mainly because of the relative ease and efficiency with which parsers can be developed, provided that appropriate data sets for training are available. In recent years, considerable attention has been given to data-driven parsers that use representations based on the notion of dependency, where syntactic structure is represented by a hierarchy of syntactic relations between words. Dependency representations have been claimed to be especially well suited for languages with free or flexible word order, and research has shown that data-driven dependency parsers that are both efficient and accurate can be developed with fairly modest amounts of data. The potential of data-driven dependency parsing has been demonstrated on a large scale in the CoNLL shared tasks in 2006 and 2007, where parsers have been trained and evaluated using data from some twenty languages [2, 12].

In this paper, we add to the increasing literature in this field by presenting what we believe to be the first data-driven dependency parser for Romanian. The parser has been trained and evaluated on a treebank developed within the RORIC-LING project, using the freely available MaltParser system, and achieves a labeled

attachment score of 88.6% on held-out test data. This seemingly impressive result, given a training set of less than 35,000 tokens, is partly explained by the data selection procedure for the treebank, where only short and simple sentences were included, and by the fact that gold standard part-of-speech tags were used in the input at testing time. Nevertheless, the results are promising enough to motivate further work to extend the capability of the parser.

The remainder of the paper is structured as follows. Section 2 introduces the treebank that has been used for training and evaluation, and section 3 describes the MaltParser system. Section 4 contains the experimental results, and section 5 gives conclusions and suggestions for future research.

2 A Dependency Treebank for Romanian

The treebank developed in the RORIC-LING[1] project consists of 36,150 tokens (punctuation excluded) and comprises newspapers articles, mostly on political and administrative subjects. It contains 4,042 sentences, having a mean sentence length of 8.94 tokens per sentence. The type/token ratio of 0.245 indicates a rather frequent repetition of certain types. The texts were chosen so as to offer a representative sample of modern written standard Romanian. However, texts including complex ambiguities were avoided as much as possible and removed from the treebank.[2]

The strong tradition of applying the Dependency Grammar (DG) formalism in linguistic research on Romanian and in teaching prescriptive grammar in Romanian schools justifies the choice of annotation style. A description of a DG approach to syntactic analysis with special reference to Romanian can be found in [5]. The texts were annotated with part-of-speech (POS) tags and information about the dependency relations (annotation of the head and the dependent) and dependency labels. All the dependency graphs for the sentences in the treebank are connected, projective, rooted, acyclic and any node has at most one head.

The annotation was performed completely manually by a Romanian linguist, using only the graphical interface tool Dependency Grammar Annotator (DGA).[3] Since there was only one annotator, the POS tags and dependency types were relatively coherently used throughout the whole material. The annotated texts are

[1] RORIC-LING is the Romanian part of BALRIC-LING, an Information Society Technologies project aimed at raising awareness on Human Language Technologies and possible scientific and industrial applications of linguistic resources in Bulgarian and Romanian. More at http://www.larflast.bas.bg/balric/index.html.

[2] The information regarding the choice of texts from the treebank was included in the Romanian version of the RORIC-LING Bulletin available at http://www.phobos.ro/roric/Ro/qa16.html.

[3] Downloadable at http://www.phobos.ro/roric/DGA/dga.html.

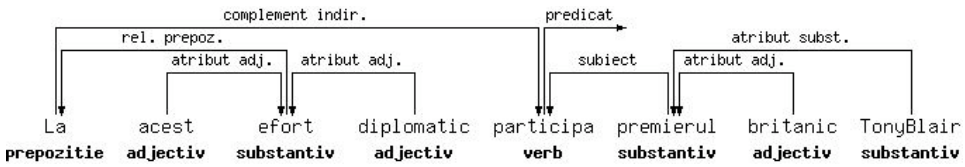

Figure 1: The sentence *La acest efort diplomatic participa premierul britanic Tony Blair* [Lit. 'The British prime-minister Tony Blair is part of this diplomatic effort'] annotated by DGA.

automatically saved by DGA in the XML format, using the ASCII character encoding, thus explaining the absence of the five Romanian diacritics (ă, â, î, ş and ţ) from the treebank.

The POS tagset used for annotating the treebank is relatively small and rather simple considering the morphosyntactic richness of Romanian. The dependency types set is exhaustive for the texts the treebank consists of. The problematic cases were most of the time avoided. All idiomatic and complex group structures were decomposed into simple elements, sometimes removing difficult structures. The representation of elliptical constructions in Romanian was considered unnecessary, mostly because the written language tends to eliminate them. Consequently, a shallow syntactic annotation was adopted. All complex-compound sentences (including coordinated sentences) were split into simple sentences, therefore there are no subordinate clauses in the treebank. Dates and measure phrases do not receive any kind of special annotation. There are no punctuation marks (except for a few hyphens, brackets and slashes in words like, e.g., *e-mail* or *NATO/Rusia [ro]*). Proper names consisting of two or more elements were collapsed into one lexical element (E.g., *Tony Blair* becomes a lexical unit *TonyBlair*). Figure 1 shows a sentence from the treebank developed in the RORIC-LING project.

In order to evaluate the accuracy of manual annotation, we randomly selected 3% of the total number of sentences (i.e., 122 sentences) and manually corrected all the errors, thus creating a gold-standard material. The original annotation and the gold standard were compared, and the agreement was found to be 98.9% for the dependency annotation, as measured by the labeled attachment score (LAS) of the original annotation with respect to the new gold standard. This result shows that the quality of the material is satisfactory. The errors occurring in the treebank include incorrect dependency labels and head identification problems.

Even if the annotation of the material was performed by a single annotator, inconsistencies still occur, especially within the annotation scheme. Four of the twenty POS tags and one dependency type appear only in the first 6% of the material, reducing significantly the POS tagset for the rest of the material. For instance,

verbs and adjectives in participle form are annotated as such only in the first part of the material. On the other hand, the definite article POS tag is present only in the last 90% of the material. In order to eliminate these inconsistencies, we have mapped the POS tagset used in the first part to the one used in the larger part of the treebank.

The treebank was intended first of all as a step towards linguistic resources in Romanian and, secondly, for training and evaluating statistical dependency parsers for Romanian. As far as we know, this is the first time the treebank has been used for the second purpose.

3 MaltParser

MaltParser [13] is a language-independent system for data-driven dependency parsing, based on a transition-based parsing model [10]. More precisely, the approach is based on four essential components:

- A transition-based deterministic algorithm for building labeled projective dependency graphs in linear time [11].
- History-based feature models for predicting the next parser action at nondeterministic choice points [1, 7, 15].
- Discriminative classifiers for mapping histories to parser actions [6, 19].
- Pseudo-projective parsing for recovering non-projective structures [14].

Given that all dependency structures in the Romanian treebank are strictly projective, the pseudo-projective parsing technique has not been used in the experiments and is not further described in this paper.

3.1 Parsing Algorithm

The parser uses the deterministic algorithm for labeled dependency parsing first proposed in [11]. The algorithm builds a labeled dependency graph in one left-to-right pass over the input, using a stack to store partially processed tokens and adding arcs using four elementary actions (where TOP is the token on top of the stack and NEXT is the next token):

- **Shift**: Push NEXT onto the stack.
- **Reduce**: Pop the stack.
- **Right-Arc(*r*)**: Add an arc labeled *r* from TOP to NEXT; push NEXT onto the stack.

	POS	LEX	DEP
TOP	*	*	*
TOP+1	*		
TOP−1	+		
TOP−2	+		
NEXT	*	*	
NEXT+1	*	*	
NEXT+2	*		
NEXT+3	*		
NEXT−1	+		
HEAD(TOP)	+	*	
LDEP(TOP)	+		*
RDEP(TOP)	+		*
LDEP(NEXT)	+		*
RSIB(LDEP(TOP))		+	

Table 1: History-based features used in the experiments, divided into default features (*) and features added specifically for Romanian (+). Symbols: TOP = token on top of stack; NEXT = next input token; HEAD(w) = head of w; LDEP(w) = leftmost dependent of w; RDEP(w) = leftmost dependent of w; RSIB(w) = next right sibling of w. Positive subscripts on TOP indicate relative position in the stack; other subscripts indicate relative position in the input string (negative = left, positive = right).

- **Left-Arc**(r): Add an arc labeled r from NEXT to TOP; pop the stack.

Parser actions are predicted using a history-based feature model (section 3.2) and SVM classifiers (section 3.3).

3.2 History-Based Feature Models

History-based parsing models rely on features of the derivation history to predict the next parser action [1]. The features used are all symbolic and defined in terms of three different token attributes:

- POS = part of speech
- LEX = word form
- DEP = dependency type

Features of the type DEP have a special status in that they are extracted during parsing from the partially built dependency graph and are updated dynamically during parsing. The other two feature types (LEX, POS) are given as part of the input to the parser and remain static during the processing of a sentence. The use of POS features presupposes that the input has been preprocessed by a part-of-speech tagger but for the experiments reported below we use the gold standard tags from the treebank.

Starting from the default feature model in MaltParser, we have performed forward feature selection to adapt the parser for Romanian. Table 1 shows the features used in the experiments, where rows identify tokens in a parser state, columns identify token attributes, and cells identify features defined by the row token and the column attribute. Cells marked * correspond to features in MaltParser's default model, while cells marked + indicate features added during feature selection for Romanian. Results for the different feature models are presented in section 4.

3.3 Discriminative Classifiers

We use support vector machines [18] to predict the next parser action from a feature vector representing the history. More specifically, we use LIBSVM [3] with a quadratic kernel $K(x_i, x_j) = (\gamma x_i^T x_j + r)^2$ and the built-in one-versus-all strategy for multi-class classification. Symbolic features are converted to numerical features using the standard technique of binarization, and we use MaltParser's default settings for all parameters associated with the learning algorithms.

4 Experiments

The total data set was split into a training set, a development set and an evaluation test set. The development set was used for preliminary testing and tuning the parsing models and the feature model specific for Romanian. The test set of 404 sentences, unseen during the development phase, was used for a final test run, training on 90% of the data (approximatixely 32,500 sentences). All results presented in this paper are on the final test set.

On the whole, there were several modifications brought to the original treebank data. First of all, the training and testing files used in the experiments were converted from XML into the CoNLL data format.[4] Secondly, some errors and inconsistencies detected in the material were fixed, as described in section 2. Thirdly, the presence of blanks and punctuation characters inside the strings of the original

[4]More information about the CoNLL format for dependency treebanks can be found at http://nextens.uvt.nl/depparse-wiki/DataFormat.

Feature Model	LAS	UAS
Default	87.9 (±2.1)	91.5 (±1.8)
Optimized	88.6 (±2.0)	92.0 (±1.7)

Table 2: Parsing accuracy measured in labeled and unlabeled attachment score (LAS and UAS).

POS tags and dependency labels led to the decision to map the original symbols to shorter, more compact but absolutely equivalent symbols. Finally, the special root node of the dependency graphs for sentences in the treebank was modified from being an extra dummy word at the end of each sentence to being an extra dummy word at the beginning of the sentence. Since the special root node does not correspond to any real token of the sentence, we have considered it a pure notational convention, having no theoretical or practical consequences.

As indicated in section 3, we have used two feature models in the experiments: the default model and a feature model optimized for Romanian. The results of the experiments are presented in Table 2, showing how the two feature models influence the parsing accuracy, evaluated with respect to labeled and unlabeled attachment score (LAS and UAS). The labeled attachment score represents the percentage of tokens that have been assigned both the correct head and the correct dependency label, while the unlabeled score considers only whether the head has been assigned correctly. Scores are reported with a 95% confidence interval, indicating that the differences between the default model and the optimized models are not statistically significant.

The scores for Romanian are very similar to those obtained using MaltParser 0.4 in the CoNLL 2007 Shared Task for configurational languages like English, Italian and Catalan [4]. Languages with rich morphology and flexible word order like, for instance, Czech have lower results. Given that Romanian can be considered a language characterized by flexible word order, plus a relatively rich morphology, these results are rather unexpected, especially given the limited amount of data available for training. However, as described in section 2, the texts were strictly selected, and complex syntactic structures were eliminated or simplified. This clearly facilitates the parsing task and explains the seemingly high accuracy, compared to results obtained for other languages and treebanks.

Table 3 presents the accuracy for different parts of speech, considering the percentage of tokens grouped under a certain part of speech for which both the head and the dependency relation to the head are predicted correctly. Predicative verbs have almost one hundred percent accuracy, most probably due to the fact that the treebank contains only simple sentences with only one verb acting as a predicate,

Part of Speech	LAS
Adjectives	95.1
Adverbs	90.8
Coordinating conjunctions	65.2
Nouns	89.1
Predicative verbs	99.5
Prepositions	69.2
Pronouns	89.1

Table 3: Parsing accuracy for different parts of speech

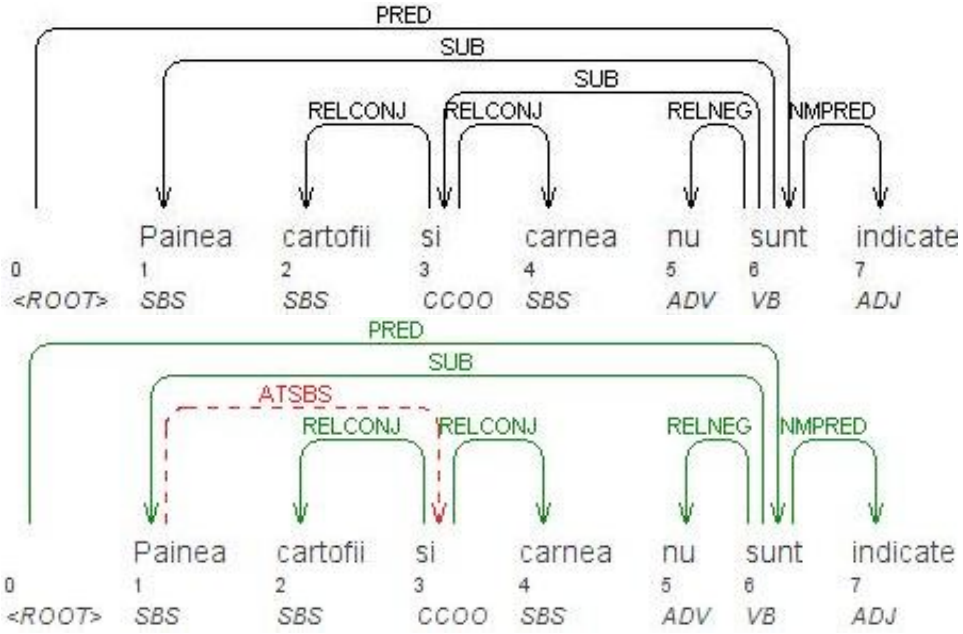

Figure 2: Gold standard (top) and predicted graph (bottom) for the sentence *Painea carnea si cartofii nu sunt indicate* [Lit. 'Bread, meat and potatoes are not advisable']. Correct dependencies are represented by solid arrows, incorrect dependencies by dashed arrows.

Length	P	R
ROOT	99.5	99.5
1	95.9	96.3
2	92.3	89.7
3–6	83.6	81.9
7–	60.8	71.3

Table 4: Labeled precision (P) and recall (R) for dependencies of different lengths; ROOT = dependents of the special root node.

also the root of the sentence. At the other end, prepositions and coordinating conjunctions have the lowest accuracy, being the hardest to attach correctly.

Figure 2 exemplifies the kind of errors the parser produces regarding coordinating conjunctions. In the gold standard annotation for the sentence *Painea carnea si cartofii nu sunt indicate*, the coordinating conjunction *si* ('and') is the head of two nouns: *cartofii* ('the potatoes') and *carnea* ('the meat'), while the conjunction itself is a dependent of the main verb, acting as a subject for the predicate *sunt* ('are') of the sentence. However, as the figure shows, the parser fails to predict this structure and instead analyzes the coordinate structure *carnea si cartofii* ('meat and potatoes') as a modifier of the noun *Painea* ('bread').

Table 4 shows the precision and recall for dependencies of different lengths (with dependents of the special root node in a separate category ROOT). The precision is the percentage of predicted dependencies of a certain length that are actually correct; the recall is the percentage of true dependencies of a certain length that are correctly predicted by the parser. As expected, both precision and recall decrease as dependencies get longer, a pattern that is well attested for other languages and treebanks [10]. However, the drop is less drastic than for many other data sets, with labeled precision and recall remaining above 80 for dependencies up to length 6, a result that can again probably be explained by the relative simplicity and homogeneity of the sentences in the treebank.

5 Conclusion

We have presented the first empirical results on parsing Romanian using the treebank developed in the RORIC-LING project. Using the freely available MaltParser system with a feature model adapted for Romanian, we achieve a labeled attachment score of 88.6% and an unlabeled attachment score of 92.0% on held-out test data from the treebank, using gold standard part-of-speech tags in the input to the parser.

The empirical results look very promising, but it must be remembered that the data in the treebank has been selected in such a way that only short and simple sentences have been included. This means that the parsing accuracy reported, while perfectly valid for the type of sentences included in the treebank, is not representative for the harder task of parsing unrestricted text in Romanian. The most important direction for future research is therefore to investigate different techniques for extending the capability of the parser to more complex sentences. Besides the obvious approach of simply annotating a larger training set, including sentences of arbitrary complexity, it is worth considering whether the existing parser can be used to speed up the process, using a weakly supervised approach involving active learning [17, 16] and/or self-training [8, 9]. In order to parse unrestricted text, a part-of-speech tagger for Romanian must also be developed.

References

[1] Ezra Black, Frederick Jelinek, John D. Lafferty, David M. Magerman, Robert L. Mercer, and Salim Roukos. Towards history-based grammars: Using richer models for probabilistic parsing. In *Proceedings of the 5th DARPA Speech and Natural Language Workshop*, pages 31–37, 1992.

[2] Sabine Buchholz and Erwin Marsi. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL)*, pages 149–164, 2006.

[3] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: A Library for Support Vector Machines*, 2001. Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

[4] Johan Hall, Jens Nilsson, Joakim Nivre, Gülsen Eryiğit, Beáta Megyesi, Mattias Nilsson, and Markus Saers. Single malt or blended? A study in multilingual parser optimization. In *Proceedings of the CoNLL Shared Task of EMNLP-CoNLL 2007*, 2007.

[5] Florentina Hristea and Marius Popescu. A dependency grammar approach to syntactic analysis with special reference to Romanian. In Florentina Hristea and Marius Popescu, editors, *Building Awareness in Language Technology*. University of Bucharest Publishing House, 2003.

[6] Taku Kudo and Yuji Matsumoto. Japanese dependency analysis using cascaded chunking. In *Proceedings of the Sixth Workshop on Computational Language Learning (CoNLL)*, pages 63–69, 2002.

[7] David M. Magerman. Statistical decision-tree models for parsing. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 276–283, 1995.

[8] David McClosky, Eugene Charniak, and Mark Johnson. Effective self-training for parsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 152–159, 2006.

[9] David McClosky, Eugene Charniak, and Mark Johnson. When is self-training effective for parsing? In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING)*, pages 561–568, 2008.

[10] Ryan McDonald and Joakim Nivre. Characterizing the errors of data-driven dependency parsing models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 122–131, 2007.

[11] Joakim Nivre. An efficient algorithm for projective dependency parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*, pages 149–160, 2003.

[12] Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task of EMNLP-CoNLL 2007*, pages 915–932, 2007.

[13] Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülşen Eryiğit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13:95–135, 2007.

[14] Joakim Nivre and Jens Nilsson. Pseudo-projective dependency parsing. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 99–106, 2005.

[15] Adwait Ratnaparkhi. A linear observed time statistical parser based on maximum entropy models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1–10, 1997.

[16] Min Tang, Xiaoqiang Luo, and Salim Roukos. Active learning for statistical natural language parsing. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 406–414, 2002.

[17] Cynthia A. Thompson, Mary Elaine Califf, and Raymond J. Mooney. Active learning for natural language parsing and information extraction. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML)*, pages 406–414, 1999.

[18] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

[19] Hiroyasu Yamada and Yuji Matsumoto. Statistical dependency analysis with support vector machines. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*, pages 195–206, 2003.