Revista Romana de Interactiune Om-Calculator 9 (4) 2016, 285-304 © MatrixRom

Improving parsing using morpho-syntactic and semantic information

Verginica Barbu Mititelu[1], Radu Ion[1], Radu Simionescu[1], Andrei Scutelnicu[2,1], Elena Irimia[1]

[1] Research Institute for Artificial Intelligence "Mihai Drăgănescu" – Romanian Academy
13 Calea 13 Septembrie, 050711, Bucharest
E-mail: {vergi, radu}@racai.ro, radsimu@gmail.com, elena@racai.ro

[2] Faculty of Computer Science – "Alexandru Ioan Cuza" University
16 General Berthlot, 700483, Iasi
E-mail: andreiscutelnicu@gmai.com

Abstract. In this paper we present the efforts of creating a syntactic parser with a very good performance on Romanian sentences. Instead of creating a parser from scratch, we decided to test the freely available existing ones and to subject them to tuning, feeding them with linguistic information in the form of features (transitivity/intransitivity, semantic class, subcategorization frames, etc.). The parsers are trained and tested on the Romanian treebank in the Universal Dependencies format. We present here, as on-going work, some partial results of our endeavour: after including only several features, we got encouraging results. We also discuss some other features that can be added to the parser in order to further improve its performance, with the final aim of attaining a reliable tool for syntactic analysis of sentences, as a task per se, and also for their use in various applications involving natural language processing.

Keywords: Romanian, syntactic parsing, derivational relations, wordnet, subcategorization frames.

1. Introduction

The need for a syntactic analyser of Romanian, i.e. a syntactic parser, ideally freely available and reliable, was expressed, on the one side, by the linguists who are in search for a large corpus of Romanian annotated at various levels, and, on the other side, by the computer scientists who are interested in creating applications and tools involving natural language processing (e.g.: machine translation, summarization, speech synthesis, etc.). We are now trying to fill this gap: we have created several resources (a treebank, a valence dictionary, word clusters based on word vectors) and we are using some results from previous work (derivational relations between words) in order to improve the results of parsers on Romanian.

The analysis of errors made by parsers in general showed that both syntax- and semantics-aware analysers are necessary, as we will demonstrate below.

In what follows, we present the current state of developing parsers for the Romanian language (section 2), then we present the results a parser trained on our treebank has on Romanian without any tuning (section 3). We enumerate and briefly describe the linguistic resources we use to improve parsing (section 4) and the new results are then presented and discussed (in section 5), before outlining the future work (in section 6) and concluding the paper.

2. Related work

Several attempts at creating parsers for Romanian are known. Călăcean & Nivre (2009) reported the use of the treebank developed by Hristea & Popescu (2003) to train the statistical MaltParser. However, this small treebank (containing 4042 sentences and less than 35,000 tokens) contains short sentences (8.94 words/sentence on average) selected so as to contain only main clauses, lacks punctuation and diacritics. The reported labelled attachment score is 88.6% and the unlabelled attachment score is 92%. However, the authors admit that the impressive results are influenced by the characteristics of the sentences in the treebank: short and simple.

Seretan et al. (2010) announced some preliminary work of adapting the Fips constituency rule-based parser for Romanian and this is accessible online (http://www.latl.unige.ch/). The authors reported partial analyses of sentences and no evaluation of the parser.

Colhon & Cristea (2016) use patterns to detect relations within the Romanian noun phrases, with no interest for the verb phrase.

Another syntactic parser using dependency grammar (Perez et al, 2015) reports 78.04% labeled attachment score on a corpus of approximately 8000 sentences at that time. This parser uses a set of syntactic labels that are quite numerous, given the semantic richness of the inventory of syntactic relations (there are distinctions among various semantic adjuncts: manner, time, place, result, cause, etc.).

None of these attempts make use of semantic information for syntactic parsing. However, at the international level, it has been proven that semantic information in the form of subcategorization frames (Caroll et al, 1998; Zeman, 2002), semantic classes (Agirre et al, 2008, 2011) and features extracted by a named entity tagger (Ciaramita & Attardi, 2010) can all help syntactic parsing, even if to a moderate extent. In this paper we will support this line of research with the results we have obtained so far.

3. Results of parsing Romanian with standard features

There are many open-source dependency parsers available that were mainly developed for and tested on English. The top-performing ones are based on neural networks (Andor et al., 2016) or on graph algorithms like RBG (Lei et al, 2014) which uses sampling from a probability distribution over dependency trees to find the optimum dependency tree from the set of all possible projective or non-projective dependency trees of a sentence.

When developing the Romanian treebank (see below, subsection 4.1), we used the Malt Parser (Nivre et al., 2007) because of its flexibility (it allows both projective and non-projective parsing through the use of several algorithms) and its automatic optimization facility (Malt Optimizer (Ballesteros & Nivre, 2012) is a meta-algorithm which adapts the Malt Parser to a new language by running it with different parsing algorithms so as to maximize the parsing performance). The latter feature of Malt Parser proved quite important when attempting to parse a new language because a large portion of the parameter space is automatically searched for the optimal combination, while the search progress is presented in a human readable format to the engineer who can take the optimum model and further optimize it.

The English dependency parsing performance figures reported in the literature are not indicative of the parsing performance on a new language.

The performance of the dependency parsing depends, to a large extent, on the following factors:

- the correctness and consistency of the treebank annotation: the head-modifier relations appearing in similar contexts should have the same label;
- the size of the dependency label set: choosing from fewer labels is easier;
- the features used by the parser when making its parsing decisions.

Malt Parser is a transition-based dependency parser, meaning that the parser goes from one state to the next with the option of linking two words with a labelled dependency arc or of pushing words onto a stack, for later use (see (Nivre, 2007) for a detailed description of transition-based dependency parsing). The option of what dependency label to use or what words to link in a transition is based on a machine learned model that was trained on a portion of the human-annotated treebank. The learned model is based on features that are available to the parser beforehand (i.e. before the parsing task is executed). Malt Parser works with two machine learning classifiers for learning the arc-drawing and dependency-labelling models, namely LIBSVM (Chang & Lin, 2011) and LIBLINEAR (Fan et al, 2008).

3.1. Standard features

For any language, the "standard features" (i.e. the features that are readily available from the human-annotated treebank) on which to base linking and labeling decisions are:

- the part-of-speech (POS) tags of the words to be linked and/or of the words appearing before and after the words to be linked in a window of words of fixed size (usually 1, 2 or 3 words);
- the lemmas/word forms of the words to be linked and/or of the words appearing before and after the words to be linked in a window of words of fixed size;
- existing information about the existence of any types of dependency relations in the treebank between the two words (extracted from the "dependency grammar" learned from the treebank): this feature is called "a first order feature" in the literature;
- if any of the words to be linked is already linked in the partial tree analysis at a given moment in time, the partial analysis belonging to any of the candidate words can be used as a feature called "a second-order feature".

Beside the standard features, Malt Parser can be programmed to consider other features (e.g. semantic features that are generated using external programs) that can be useful when deciding if two words can be linked by a dependency relation:

- word clustering (see below, subsection 4.3);

- wordnet relations (see below, subsection 4.2).

We have tuned Malt Parser’s ‘arc eager’ linking algorithm features to better suit Romanian and, to this end, we have used lemmas and coarse-grained POS tags instead of the default word form and detailed POS tags when defining the standard features. The set of Romanian standard features[1] contains 35 atomic (e.g. the coarse-grained POS tag of the first word on the stack) and merged features (e.g. the feature obtained by concatenating the coarse-grained POS tag of the first word on the stack with the coarse-grained POS tag of the first word in the input). The linking algorithm uses LIBLINEAR as the learner of dependency linking decisions during parsing.

3.2. Romanian Malt Parser evaluation

The Romanian Malt Parser with standard features was evaluated using a 10-fold cross validation run on our treebank. That is, we randomly split our treebank into 90% train and 10% test sets, 10 times, and averaged results over the 10 test sets, making sure that there were no duplicate sentences in any of the 10 test sets. In order to measure performance, we have used the standard Labelled Attachment Score (LAS: the parser is penalized if the dependency relation is not correctly labelled - even if the head and the dependent are correct) and the Unlabelled Attachment Score (UAS: the parser is penalized only if the dependency relation, irrespective of its label, does not hold between the linked words). Both LAS and UAS report the correct percent of relations found by the parser (as compared to the ground-truth annotation) out of all relations found by the parser.

Table 1 reports the UAS and LAS scores and their mean over each of the 10 test sets, establishing the baseline performance: 84.22% the UAS average score and 77.59% the LAS average score. Section 5 below gives the UAS and LAS figures on the same 10 test sets, after the standard set of features were enriched with morpho-syntactic and semantic features.

[1] We will not give here the feature set for Romanian, e.g the contents of the ‘featuremodel’ XML element, because it is quite large. The model is available upon request.

Table 1: LAS and UAS for the Romanian Malt Parser with standard features.

Fold number	UAS	LAS		Fold number	UAS	LAS
1	84.4%	77.8%		Fold number	UAS	LAS
2	84.3%	77.4%		7	84.1%	77.6%
3	83.9%	77.4%		8	83.9%	77.1%
4	83.3%	77%		9	84.6%	77.8%
5	83.5%	76.9%		10	85.3%	78.7%
6	84.9%	78.2%		Mean	84.22%	77.59%

3.3. Error analysis

As already stated, the parsing errors are the cases when the relation is not established between the right nodes (irrespective of the correctness of its syntactic label) or when the relation is established between the right words but its label is incorrect.

Looking into the Romanian Malt Parser's output, we could identify several types of errors of the latter type, having as criteria the causes of errors and the possible solutions to prevent their occurrence. Among the causes of errors we mention:

- a too refined set of syntactic labels: for some syntactic relations there are subtypes in the treebank the parsers were trained on: they were introduced to cope with language-specific linguistic phenomena; however, the parser seems not to learn how to distinguish between a subtype and a type: the time nominal or time adverb modifiers (labeled with the relations `nmod:tmod` and, respectively, `advmod:tmod`[2] in the treebank) (see example (1)) cannot be distinguished from other nominal or, respectively, adverb modifiers (labeled with `nmod` and, respectively, `advmod`) (see example (2)), the prepositional phrases with the preposition selected

[2] The labels `nmod:tmod` and `advmod:tmod` are chosen to denote the fact that they are subtypes of the relations denoted by the labels `nmod` and `advmod`, respectively.

by a predicate (e.g., the preposition *de* "of" is selected by the verb *depinde* in example (3)) (semantically labeled with `nmod:pmod`[3]) cannot be distinguished from prepositional phrases in which the preposition is not a selectional restriction of a predicate (as is *de* in example (4)) (syntactically labeled with `nmod`), etc.;

(1) Vin dimineața.
Come-I morning-the
"I am coming in the morning."
(2) Vin încet.
Come-I slowly
"I'm coming slowly."
(3) Totul depinde de acest meci.
"All depends on this match."
(4) Nu știu nimic de tine.
Not know-I nothing of you.
"I do not know anything about you."

- errors in the previous annotation phases, especially in the POS tagging phase: the homonymy between different parts of speech makes it hard to distinguish between them: nouns and adjectives, adjectives and participles, etc.;
- annotation decisions generating confusing cases: the decision to analyse the animate direct object of ditransitive verbs (as is the noun *copii* in example (5)) as an indirect object is not beneficial for the parser, which learns that indirect objects can occur in the accusative case, which is rather strange for the Romanian grammar;

(5) Îi învăț pe copii un cântec.
Clitic3plural.Accusative teach on children a song
"I am teaching the children a song."

- a too general value for a morphologic attribute in the POS tag: for example, the same value (namely r) is used for both accusative and nominative cases, thus making it difficult to distinguish between direct objects and nominal subjects (see also the discussion about this confusion in subsection 6.2);
- language characteristics: for example, the relatively free word order characterising Romanian and the tendency to postpone the subject

[3] The relation labeled as `nmod:pmod` is a subtype of the relation labeled as `nmod`.

are further causes of confusing nominal subjects and direct objects.

Among the solutions we foresee we mention:

- lowering linguistic precision by using a less refined set of syntactic labels: as a subtype of a relation is frequently mistaken for the respective relation (e.g., `nmod:pmod`, a subtype of the relation `nmod`, is mistaken for the latter) and vice versa, we consider that reducing the ambiguity class will improve parsing accuracy;
- use of external linguistic resources in the form of lexicons (for example, to learn what the multiword expressions in the language are), valence dictionaries (which can help distinguish either between arguments and optional modifiers or between different types of arguments);
- adding rules to the parser, for instance to help distinguish between a doubling clitic and a non-doubling one (that is, an argument) or to enforce agreement between the subject and the predicate (thus helping the parser to distinguish between subjects and direct objects) or between the noun and its modifiers.

4. Linguistic resources for parsing

In this section we present the linguistic resources useful for training and testing the parsers presented in the previous chapters, as well as for improving their results. These resources are: the Romanian treebank (Mititelu et al, 2016) in Universal Dependency (universaldependency.org) format (see subsection 4.1), the inventory of derivationally related words in the Romanian wordnet (Mititelu, 2013) (subsection 4.2) and a list of automatically-derived word clusters (subsection 4.3).

4.1 The Romanian treebank in Universal Dependencies format

A treebank is a collection of sentences that are syntactically analysed, that is the relations between the components (both words and punctuation) are established and labelled with the appropriate syntactic relation from a set. Each sentence is thus represented as a syntactic tree: its root is the head of the sentences (usually the main verb in the main clause), while the other components attach as dependents to their syntactic head. In Figure 1 we show the tree-like representation of the sentence in example (6):

(6) Totul ținea de domeniul presupunerilor.
All held of domain-the presuppositions-the-Genit.
"It was all guesswork."

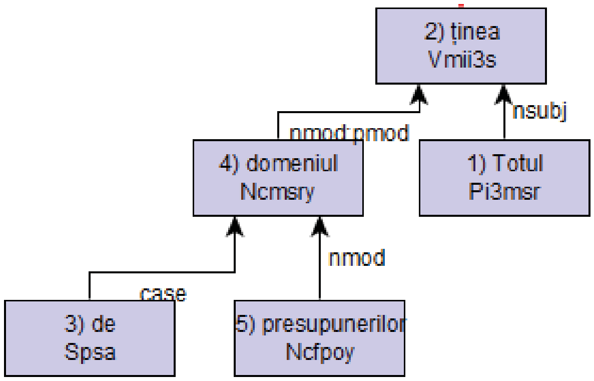


Figure 1. The tree representation of the sentence in example (6).

The treebank we worked with for training the parsers and for testing them is the Romanian treebank in Universal Dependencies format. Called RoRefTrees, it has been created in an effort to offer the community a resource with the following characteristics:

- large size (of 9523 sentences),
- containing longer sentences (23 tokens/sentence on average),
- correctly (i.e., with the right diacritics) written,
- larger coverage: it contains several text genres (literature - 1818 sentences, law - 1606 sentences, medical - 1210 sentences, FrameNet translations - 1092 sentences, academic writing - 950 sentences, news - 933 sentences, science - 362 sentences, wikipedia - 251 sentences, miscellanea - 1301 sentences),
- consistently annotated with an inventory of relations that is meant to be universal in the syntactic analysis of natural languages (however, allowing language-specific relations as subtypes of the universal ones) and obeying principles that have the open-class words at their centre, thus promoting nouns, verbs, adjectives and adverbs in head position and all function words being confined to the dependant position. The details about the principles of Universal Dependency

initiative and about their set of relations can be found on the project website (universaldependency.org),

- is free: the treebank is freely downloadable in CoNLL-U format, in a new release every six months, in May and November, at universaldependencies.org,
- can be queried online with user-friendly tools: http://clarino.uib.no/iness/page?page-id=iness-main-page, http://bionlp-www.utu.fi/dep_search, http://lindat.mff.cuni.cz/services/pmltq/#!/home

4.2 The Romanian wordnet

A wordnet (Fellbaum, 1998) is a lexical database in which words (nouns, verbs, adjectives and adverbs) are grouped, according to their senses, in synonymic series called synsets. The number of senses a word has equals the number of synsets it occurs in, provided that the wordnet offers a complete image of the language.

What makes the wordnet a greatly used resource in various applications processing texts is its organization and the set of relations interlinking the words. Nouns are organized[4] in hypo-/hyperonymic[5] hierarchies, with semantically more general words towards the top and the more specific ones to the bottom of the tree; the verbal hierarchies are based on the hyperonymy and troponymy[6] relations; descriptive adjectives are organized in clusters (containing adjectives with a similar meaning to the cluster head), which are further grouped in pairs based on the antonymy relation established between cluster heads; relational adjectives and all adverbs have no organization.

Besides these intra-part of speech relations, a wordnet has several inter-part of speech relation, some of them only of a semantic nature, others also morphologic. Among the latter ones are the derivational or morpho-

[4] We enumerate here only the main relations in the wordnet, also relevant for the work described here.

[5] The hyponymy relation is the relation from the more specific to the more general term: we say that *daffodil* is a hyponym of *flower*. It is also known as the IS-A relation. Its inverse relation is called hyper(o)nymy.

[6] The troponymy relation is a manner relation: a troponym of a verb elaborates on its meaning, as *toddle* is a troponym of *walk*, as it means “to walk unsteadily”.

semantic relations.

The Romanian wordnet (RoWN) (Tufis et al, 2013) (freely available on meta-share, at http://ws.racai.ro:9191) follows the structure presented above. Although containing only almost 60,000 synsets (thus being unable to mirror the semantic and lexical richness of Romanian), it can still be used in various applications.

In the work described here, we rely on two of its strong points: (i) the hyperonymic hierarchies among nouns and (ii) the derivational relations between nouns and verbs. We detail these here below.

(i) Such hierarchies of nouns can be used, together with valence frames for predicates, to correctly syntactically analyse arguments of these predicates. For more details on this, see subsection 6.1. below.

(ii) Derivational relations are established between two words: one of them is created from the other (called base) by means of adding or removing linguistic material (called affix) to or from its beginning or ending: the noun *vorbire* ("speaking") is derived from the verb *vorbi* ("speak") because the former is the result of adding the suffix *-re* to the latter. Besides this formal relation between the base and the derived word, there is also a semantic relation between them: in the example above the relation is called Event, as the noun designates the name of the action expressed by the verb.

Out of all types of derivational relations marked between words in RoWN (Barbu Mititelu, 2014) we were interested in those established between verb bases and derived nouns and marked with the semantic relation Event (see Section 5 below).

4.3. Word embeddings and clusters

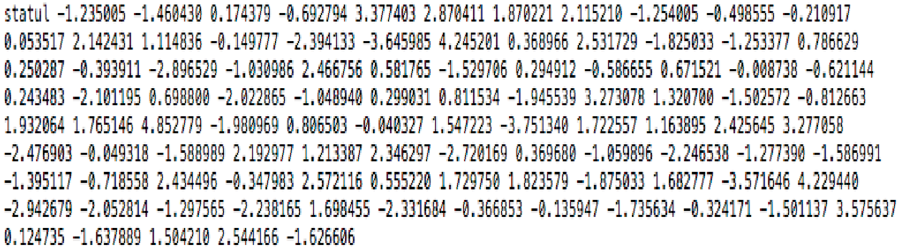
If two words are semantically related and uniquely identified as such (i.e. placed in the cluster with the same identification number (ID)), the generalization power of the parser increases as it can use the ID to learn certain attachments[7], e.g. [(NP) *cărțile* [(PP) *cu coperți roșii*]] ([(NP) books-the [(PP) with covers red-plural]] "the books with red covers") and [(NP) *cărțile* [(PP) *cu coperți albastre*]] (books-the with covers blue-plural "the books with blue covers") have the same attachment pattern if the colour denoting words *roșii* ("red") and *albastre* ("blue") are placed in the same cluster and the cluster ID is used to learn the attachment.

[7] The abbreviations stand for: NP = noun phrase, PP = prepositional phrase.

In order to use clustering to increase the generalization power of a dependency parser, we have to learn word clusters from a large corpus such that they cover as much of the language vocabulary as possible.

CoRoLa – the Computational Representative Corpus of Contemporary Romanian Language (Tufiș et al., 2016) is a resource developed in a priority project of the Romanian Academy. At the end of the project (2017), the corpus, built in collaboration by the Research "Mihai Drăgănescu" Research Institute for Artificial Intelligence in Bucharest and the Institute of Computer Science in Iași, will include texts totalizing 500 million Romanian words acquired from a broad range of domains (politics, humanities, theology, science, etc.) and covering all literary genres (prose, poetry, theatre, scientific texts, journalistic texts, etc.) (Tufiș et al, 2016). The texts, obtained on according to written protocols with our providers, are subject to a processing chain which, at this moment, consists of sentence segmentation, tokenization, lemmatization and POS tagging, but will also be syntactically analysed once our parser is ready. We have used about 155 million words (out of 280 million which make the corpus at the moment) to extract word embeddings and clusters for Romanian dependency parsing.

The word clusters are obtained by grouping words described as real-valued, N-dimensional vectors. One way of getting these vectors is to use the word embeddings generated by a program called `word2vec` (Mikolov et al., 2013). This algorithm takes as input the tokenized corpus and produces real-valued word vectors as output, with a fixed dimension given as a parameter. `word2vec` is basically a recursive neural network which can either be instructed to predict the next word in a sequence given the words in context (the bag-of-words context model) or the words in the context (in a fixed-length window) given the current word (the Skip-gram model). A by-product of training the neural network is *the matrix of weights* connecting the hidden layer with itself which, during training, averages the values of all contexts for all words in the vocabulary, thus *quantifying all contexts of a word* into a single real-valued vector of a fixed dimension. Figure 2 presents an example real-valued vector with 100 dimensions for the word *statul* (state-the, "the state") generated by `word2vec` with the Skip-gram model:



```
statul -1.235005 -1.460430 0.174379 -0.692794 3.377403 2.870411 1.870221 2.115210 -1.254005 -0.498555 -0.210917
0.053517 2.142431 1.114836 -0.149777 -2.394133 -3.645985 4.245201 0.368966 2.531729 -1.825033 -1.253377 0.786629
0.250287 -0.393911 -2.896529 -1.030986 2.466756 0.581765 -1.529706 0.294912 -0.586655 0.671521 -0.008738 -0.621144
0.243483 -2.101195 0.698800 -2.022865 -1.048940 0.299031 0.811534 -1.945539 3.273078 1.320700 -1.502572 -0.812663
1.932064 1.765146 4.852779 -1.980969 0.806503 -0.040327 1.547223 -3.751340 1.722557 1.163895 2.425645 3.277058
-2.476903 -0.049318 -1.588989 2.192977 1.213387 2.346297 -2.720169 0.369680 -1.059896 -2.246538 -1.277390 -1.586991
-1.395117 -0.718558 2.434496 -0.347983 2.572116 0.555220 1.729750 1.823579 -1.875033 1.682777 -3.571646 4.229440
-2.942679 -2.052814 -1.297565 -2.238165 1.698455 -2.331684 -0.366853 -0.135947 -1.735634 -0.324171 -1.501137 3.575637
0.124735 -1.637889 1.504210 2.544166 -1.626606
```

Figure 2: A 100-dimensional vector describing the word *statul* (the word followed by the space-separated values of the 100 dimensions)

`word2vec` can also use the generated word embeddings to do K-means clustering. In this mode, it outputs a vocabulary file with words and their corresponding cluster IDs, a word and cluster ID per line. The only variable here is the number of clusters to generate, which, according to the clustering literature, is unknown beforehand. But, as far as the dependency parsing is concerned, this is a parameter which can be tuned on a development set: choose the number of classes in increments of say 500, assign the cluster IDs to words in the treebank as additional features for the parser to learn from and measure UAS and LAS on the development set using these features, retain the number of classes for which you obtain the highest performance.

Returning to our colour example from the beginning of this section, running K-means on our CoRoLa subcorpus, we can see some very interesting (and not very frequent) colour names being detected as belonging to the same clusters: *galben-maroniu* ("brownish yellow"), *galben-pai* ("straw yellow"), *galben-untdelemn* ("oil yellow") and *galben-verzui* ("greenish yellow") or *roșu-cărămiziu* ("brick red"), *roșu-portocaliu* ("orange red"), roșu-violet ("red violet"), *roșu-zmeuriu* ("raspberry red") and *roșu-închis* ("dark red").

5. The results of parsing Romanian with additional morpho-syntactic and semantic information

The attachment of PPs is a pain in the neck for parsers. Some of these attachments can be coped with when parsers have access to the morphological (derivational) knowledge: nouns derived from verbs usually preserve from them the number of dependents and their grammatical realization. Among these, the nominalizations keep their verbal characteristics regarding the prepositional phrases that they can govern syntactically. Complex dependency structures can be formed around such nouns. In the following example (see Figure 3), *schimburi* ("exchanges") is derived (by the mechanism called back-formation[8]) from the verb *a schimba* ("to exchange"). The parsing difficulty can be observed: the parser must choose between linking *reprezentanții* ("the representatives") with *schimburi* ("exchanges") or with *procedează* ("proceed"), which is a highly ambiguous decision. When *schimburi* is exposed as a noun derived from the verb *a schimba*, the parser manages to make the correct decision.

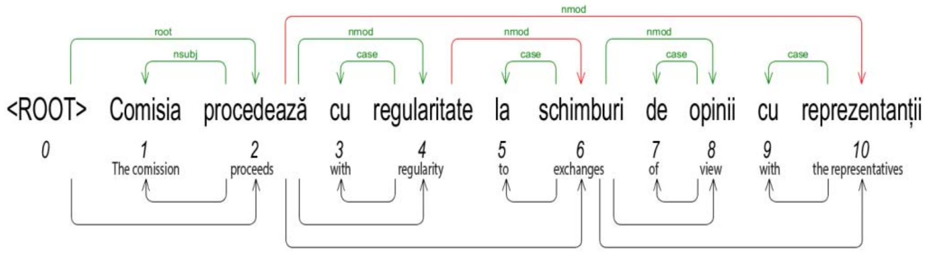


Figure 3: Example of a complex dependency structure.

Our experiments regarding the calibration of the parser have shown that the correlation between the lemmas of verbs and the prepositions that occur as their grandchildren, in UD representation[9], play a very important role in the overall precision of the system. Parsing nouns derived from verbs can also benefit from this fact, but for this, they must be identified and their root verb must be specified. In this manner, the preposition to verbal lemma attachment is observed and exploited by syntactic parsing not only in verbs but also in nouns derived from verbs. Moreover, this additional, semantic, information better separates the phenomena of prepositional attachment to

[8] A word is back-formed from a root when the affix is removed from the latter. In Romanian, the noun *schimb* is derived from the verb *schimba* by removing the suffix *-a*. This type of derivation, in which the length of the newly derived word is shorter than that of the root, is called back-formation.

[9] According to the UD annotation principles, prepositions are not heads of phrases. They depend on the words they precede: see in Figure 1 the attachment of the prepositions to their heads by means of the relation labeled as `case`.

verbs versus nouns.

MaltParser was configured to consider some additional features for the words:

a flag indicating if the word represents an action;
the lemma of the source verb, if applies;
a flag indicating that the source verb can be transitive, if applies.

The training and testing corpora have been preprocessed. The first two new features were populated using resources resulted from previous research (see subsection 4.2, under (ii)). The third feature was populated as indicated by the freely available Dexonline database (dexonline.ro). Even though the transitivity of a verb depends on its sense in every occurrence, merely indicating that some verbs may be transitive while others not has resulted in a slight precision increase in syntactic parsing. The following table shows the improvements rendered by these first semantic features, while parsing the same folds listed in Table 2.

Table 2: LAS and UAS for the Romanian Malt Parser using the described semantic features

Fold no.	**UAS**	**LAS**
1	85.4%	79.4%
2	85.1%	78.4%
3	84.7%	78.4%
4	84.5%	78.4%
5	84.7%	78.4%
6	85.7%	79.2%
Folder no.	**UAS**	**LAS**
7	85.1%	79%
8	84.4%	78.4%
9	85.7%	79.3%
10	86.4%	80.2%
Mean	**85.17% +0.95%)**	**78.91% (+1.32%)**

6. Further improvements

6.1. Expected improvements when using word clusters based on word vectors

As already mentioned above, one of the most frequent types of parsing errors in statistical systems is the wrong attachment of prepositional phrases. Let us consider the following examples (where V stands for verb):

(7) Maria (NP) mănâncă (V) ciorbă (NP) cu carne (PP).
Mary eats soup with meat.
"Mary eats meat soup."

(8) Maria (NP) mănâncă (V) ciorbă (NP) cu poftă (PP).
Mary eats soup with appetite.
"Mary eats soup with appetite."

Although the components of the two sentences and their order in the sentence are the same (NP+V+NP+PP), their structure is different, since the PP "cu carne" must be attached to the second NP in (7), while the PP "cu poftă" must be attached to the verb in (8). A statistical model, fundamentally based on previously seen component sequences but also on lexical features, can solve the PP-attachment problem in (8) if and only if the syntactical relation between the verb "mănâncă" and the prepositional phrase "cu poftă" already exists in the training data and this attachment probability is better than the probability of attaching it to "ciorbă". Similarly, if no occurrence of the verb "mănâncă" governing the PP "cu carne" exists in the training data, the parser attaches it to the nearer component, the NP "ciorbă".

Moreover, with alike but unseen examples like (9), the parser is again confused, since a lexico-syntactic model has no semantic knowledge, and therefore no means of identifying the semantic similarity between "cu dezgust" and "cu poftă". This limitation can be overcome by incorporating semantic class constraints in the parser: each content word in the training data is annotated with a corresponding semantic class in a predetermined set of semantic classes and this information is passed to the parser as semantic features of the syntactic model. Coming back to our example, if both "cu plăcere" and "cu dezgust" are associated (in the training data and at runtime, respectively, through an automatic mechanism) to the semantic class STATE, the parser will associate them the same attachment preference.

(9) Mănânc ciorbă cu dezgust.
Eat-I soup with disgust
"I eat soup with disgust."

We cannot report now the improvement of parser results when knowledge of the type described in subsection 4.3 is added as feature to it. However, we foresee that they will help the parser when confronted with sentences as those in the three examples above.

6.2. Expected results with subcategorization frames

Another frequent error, also mentioned above, comes from the syntactic ambiguity between the subject and the object of the verb, especially in Romanian, where the topical and morphological information are not sufficient for disambiguation: the language has a relatively free word order, while the distinction between accusative and nominative case is not marked morphologically. Subcategorization (or valency[10]) frames for verbs augmented with semantic class restrictions for their arguments and lexical constraints on the prepositions for prepositional arguments can solve this difficulty: provided as additional features to the training model, they serve as supplementary linguistic resource for the statistical parser.

Let us study another example (10) concerning the same verb, *a mânca* ("to eat"), for which a collection of subcategorization frames contains, among others, the following frame:

a mânca
f1. NP[nom, +ființă:1] NP[ac, +hrană:1]
(10) Mănâncă cu poftă morcovul iepurele.
Eats with appetite carrot-the rabbit-the .
"The rabbit eats the carrot with appetite."

The frame (f1) specifies that the verb combines with a nominative NP that is of the semantic type "*ființă*" ("being", i.e. a living thing) and with an accusative NP that is of the semantic type "*hrană*" ("food"). The power of such frames is given by their connection with the Romanian wordnet (see above, subsection 4.2.): any literal occurring in the wordnet in (direct or indirect) hyponymy relation to the synset containing the literal "*ființă*" with

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

sense number 1 can occupy the nominative NP position of the verb "*a mânca*". Similarly for nouns denoting food and occurring under the synsets containing "*hrană:1*" which can fill the accusative NP position of the same verb.

Because of the nominative-accusative homonymy, the statistical POS-tagger will provide the same morpho-syntactic description for both "iepurele" and "morcovul": *Ncms-y* (N=noun, c=common, m=masculine, s=singular, -=no case specific form, y=definite form). But the valency frame (f1) coupled with the mechanism mentioned before that is able to associate their respective semantic classes (*being* and *food*), can identify "iepurele" as the nominative NP, that is syntactically analysed as subject, and "morcovul" as the accusative NP, syntactically analysed as direct object.

We have created valence frames for verbs, but for the moment they are not introduced as features for the parser, thus no results can be reported here.

7. Conclusion

Syntactic parsing of sentences is a difficult task, given the general characteristics of natural languages. Moreover, language-specific characteristics add to this. Romanian, like any language, displays ambiguities at many linguistic levels. Its relatively free word order is a challenge for the parsing results. We need also consider its status as an under-resourced language in order to understand the efforts implied by our work, which presupposes, besides parser tuning, the development of resources that can help it.

The conclusion of our presentation is in line with the work done in the international context, especially on English: the addition of semantic information in parsing cannot but improve syntactic parsing results. We have quantitatively shown this for one type of semantic information (involved by derivation). We will continue to experiment in this direction, by adding information from other semantic resources, such as the wordnet and the verbal frames in order to further increase the quality of our parsing.

Acknowledgements

This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS–UEFISCDI, project number

PN-II-RU-TE-2014-4-1362.

References

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

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

Andor, D., Alberti, C., Weiss, D., Severyn, A., Presta, A., Ganchev, K., Petrov, S., Collins, M. (2016) Globally normalized transition-based neural networks. arXiv, https://arxiv.org/abs/1603.06042.

Ballesteros, M., Nivre, J. (2012) MaltOptimizer: An Optimization Tool for MaltParser. *Proceedings of the System Demonstration Session of the Thirteenth Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. Avignon, France, 23-27 April 2012.

Barbu Mititelu, V., Ion, R., Simionescu, R., Irimia, E., Perez, C.-A. (2016) The Romanian Treebank Annotated According to Universal Dependencies. *Proceedings of The Tenth International Conference on Natural Language Processing (HrTAL2016)* (in press).

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

Călăcean, M., Nivre, J. (2009) A Data-Driven Dependency Parser for Romanian, *Proceedings the Seventh International Workshop on Treebanks and Linguistic Theories*, 65-76.

Chang, C.-C., Lin, C.-J. (2011) LIBSVM : a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*.

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

Colhon, M., Cristea, D. (2016) Dependency parsing within noun phrases with pattern-based approaches. *Proceedings of the 12th International Conference "Linguistic Resources and Tools for Processing the Romanian Language"*, Mălini, 27-29 October, 51-60.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., Lin, C.-J. (2008) LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research* 9, 1871-1874.

Fellbaum, Ch. (ed.) (1998) *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.

Hristea, F., Popescu, M. (2003) A Dependency Grammar Approach to Syntactic Analysis with Special Reference to Romanian. In F. Hristea and M. Popescu (eds.), *Building Awareness in Language Technology*, Bucharest, University of Bucharest Publishing House, 9-16.

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

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J. (2013) Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems* 26, 3111–3119.

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

Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S., Marsi, E. (2007) MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering* 13(2), 95-135.

Perez, A.C., Mărănduc, C., Simionescu, R. (2016). Including Social Media – a Very Dynamic Style, in the Corpora for Processing Romanian Language. *Linguistic Linked Open Data: 12th EUROLAN 2015 Summer School and RUMOUR 2015 Workshop, Sibiu, Romania, July 13-25, 2015, Revised Selected Papers*, 139-153.

Seretan, V., Wehrli, E., Nerima, L., Soare, G. (2010) FipsRomanian: Towards a Romanian Version of the Fips Syntactic Parser, *Proceedings of LREC 2010*, 1972-1977.

Tufiș, D., Barbu Mititelu, V., Ștefãnescu, D., Ion, R. (2013) The Romanian Wordnet in a Nutshell. *Language Resources and Evaluation* 47, 1305-1314.

Tufiș, D., Barbu Mititelu, V., Irimia, E., Dumitrescu, S.D., Boroș, T. (2016) The IPR-cleared Corpus of Contemporary Written and Spoken Romanian Language. *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Portorož, Slovenia, 23-28 May 2016, 2516-2521.

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