Extending a Multilingual Symbolic Parser to Romanian

Seretan, Violeta; Wehrli, Eric

How to cite

SERETAN, Violeta, WEHRLI, Eric. Extending a Multilingual Symbolic Parser to Romanian. In: Multilinguality and Interoperability in Language Processing with Emphasis on Romanian. Tufiş, D. & Forăscu, C. (Ed.). Bucharest : Romanian Academy Publishing House, 2010. p. 112–131.

This publication URL: https://archive-ouverte.unige.ch/unige:109245

Violeta Seretan and Eric Wehrli. Extending a multilingual symbolic parser to Romanian. In Dan Tufiş and Corina Forăscu, editors, Multilinguality and Interoperability in Language Processing with Emphasis on Romanian. Romanian Academy Publishing House, Bucharest, Romania, 2010.

EXTENDING A MULTILINGUAL SYMBOLIC PARSER TO ROMANIAN

Violeta SERETAN, Eric WEHRLI

LATL – Language Technology Laboratory
University of Geneva, Switzerland
E-mail: {violeta.seretan, eric.wehrli}@unige.ch

Abstract: A syntactic parser (a system that analyses the structure of natural language sentences) is a fundamental tool for any language, providing information that is essential in virtually any other language application. With a single exception (Călăcean & Nivre 2009), such a tool was missing from the otherwise vast repertory of language tools available for Romanian. In this paper, we report on ongoing work aimed to develop a symbolic syntactic parser able to fully analyse unrestricted Romanian text—in contrast, the existing parser provides an analysis in terms of dependency relations, is data-driven, and was only trained on simple sentences. Our parser is based on the Fips multilingual parsing architecture (Wehrli, 2007). We present the preliminary tasks that enabled the implementation of the Romanian version, i.e., lexicon compilation and grammar specification. We describe the current status of the parser and present experimental results, both on parsing a collection of journalistic text, and on using the parsed data in a collocation extraction application.

Key words: Syntactic parsing, symbolic methods, grammar, lexicon, multilingualism.

1. INTRODUCTION

Syntactic parsing is arguably one of the most important natural language processing applications, as its role is to provide the essential structural information that is required by virtually all other language applications in order to produce reliable results. Syntactic parsing, either symbolic (rule-based) or stochastic (statistical), has been shown to considerably improve the results of many NLP applications, e.g., term extraction (Maynard & Ananiadou, 1999), semantic role labelling (Gildea & Palmer, 2002), semantic similarity computation (Padó & Lapata, 2007).

A syntactic parser is central tool for any language. However, developing such a tool is an extremely complex task, as natural languages, as opposed to formal languages, pose notoriously difficult challenges. Besides, this task demands, depending on the approach, a substantial amount of various language resources, which are more or less difficult to obtain. For example, lexicons containing detailed morphosyntactic information associated to the items of a language are the typical resource used by natural language parsers, but their construction may be prohibitively time-consuming.

In this paper, we report on the efforts undertaken at the Department of Linguistics of the University of Geneva over the past years – since 2004 – to build a lexicon and a syntactic parser for Romanian. This work is part of a long-term parsing project that led, since its inception in the 1990s,

to the development of a large-scale, robust syntactic parser, called *Fips*, which is now available in six major indo-European languages: French, English, German, Spanish, Italian, and Greek (Wehrli, 1997; Wehrli, 2007; Wehrli & Nerima, 2009). A number of other languages have also been considered and the corresponding parsers reached different stages of development.

The work on Romanian has mainly been done in the framework of a recent project on multilingual extension of the Fips parser to four new languages, including another Romance language (Romansh), another major indo-European language (Russian), and a language much more distant than all the other languages currently supported by Fips (namely, Japanese). The aim of the project was also to challenge the extent to which Fips succeeded to evolve, over the years, from a parser supporting a single language, French (*Fips* stands for *French Interactive Parsing System*), to a multilingual parsing architecture separating the core language-independent implementation from the language-specific extensions.

The extension to Romanian was the one that advanced the most rapidly, thanks to the sustained efforts of lexicon compilation and grammar description made previously by different collaborators. An important role was also played by the availability of suitable lexical resources (such as comprehensive electronic dictionaries). The two main preliminary tasks required for building the Romanian version, i.e., lexicon compilation and grammar specification, recently approached completion. Therefore, the implementation of the grammar itself could be started, and the Romanian parser took shape. Although its development is far from completed, the parser is operational and could already be used in a specific application.

In the remaining sections, we describe the current state of FipsRomanian and present its first results. We first review, in Section 2, the previous work aimed at developing syntactic parsers for Romanian. In Section 3, we introduce the Fips multilingual parser, specify the kind of information it provides and briefly present its parsing algorithm. In Section 4, we focus on the development of the Romanian version and discuss the preparation of the resources needed, as well as the language-specific issues encountered. In Section 5, we present some preliminary results obtained by parsing a one-million word collection of newspaper articles. We also show how we used these results in an application concerned with the identification of a particular subtype of multi-word expressions. The last section provides concluding remarks and points out directions for future work.

2. RELATED WORK

Romanian can be considered as a rather privileged language, due the high number and variety of lexical resources and morphological tools that are available. Dictionaries, thesauri (e.g., WordNet, FrameNet), word aligners, annotated (parallel) corpora, POS taggers, and automated processing architectures are some examples of resources that have been produced over the past years and that are being successfully used for different purposes: computational lexicography, question answering, word sense disambiguation, anaphora resolution, and textual entailment, among many others. A brief review of the resources and tools that exist for Romanian can be found in Cristea (2009). A more detailed report is given in Cristea & Forăscu (2006). In addition, the proceedings the workshop series *ConsILR - Consortium for the Romanian Language: Resources & Tools*[1] provides a complete picture on the advances in Romanian language processing.

In contrast with this situation, little or no resources exist for Romanian insofar as the syntactic level is concerned. Despite the recent efforts made by Călăcean & Nivre (2009) to adapt a stochastic dependency parser to Romanian, there is currently no large-scale syntactic parser for Romanian, able to parse unrestricted text. To a certain extent, shallow parsing is already available:

[1] For example, (Tradabăţ et all., 2008).

not only NP and VP chunking, i.e., identification of word sequences constituting noun phrases and verb phrases (cf. Cristea & Forăscu, 2006), but also identification of syntactic relations by using regular expressions applied to POS-tagged text. The latter[2] is work in progress aimed at extending a lexicographic corpus tool, the Sketch Engine (Kilgarriff et al., 2004), to Romanian.

The work of Călăcean & Nivre (2009) can be considered as the most significant, to date, in the area of syntactic parsing for Romanian. It concerns the application to Romanian of a language-independent dependency parser, the MaltParser[3] (Nivre et al., 2007). MaltParser is data-driven, i.e., it uses syntactically-annotated corpora (treebanks) to induce a parsing model. Călăcean & Nivre (2009) made use of a small Romanian dependency treebank developed in the framework of the BALRIC-LING project, the *Balkan Regional Information Centers for HLT*.[4] This treebank consist of about 36000 tokens, corresponding to slightly more than 4000 sentences with an average length of 8.94 tokens. Only simple sentences were included, and the treebank is reported as rather homogenous.

The authors reported a high level of precision for the Romanian parser, similar to that achieved by MaltParser for English, Italian, and Catalan (88.6% for labelled attachments and 92.0% for unlabelled attachments). The evaluation was performed on a held-out portion (of 10%) of the same treebank. The parser has not been tested on more complex sentences, thus, in spite of the promising results obtained, it is still unclear to what extent the parser can be applied on unrestricted text. In addition, the reported accuracy holds for perfect POS-tagged data, as found in the treebank, whereas, in a more realistic evaluation scenario, one should consider data with automatically assigned POS categories. Figure 1 shows a sample dependency structure produced by MaltParser for the Romanian sentence *La acest efort diplomatic participa premierul britanic Tony Blair*[5] ("The British Prime Minister Tony Blair is part of this diplomatic effort").

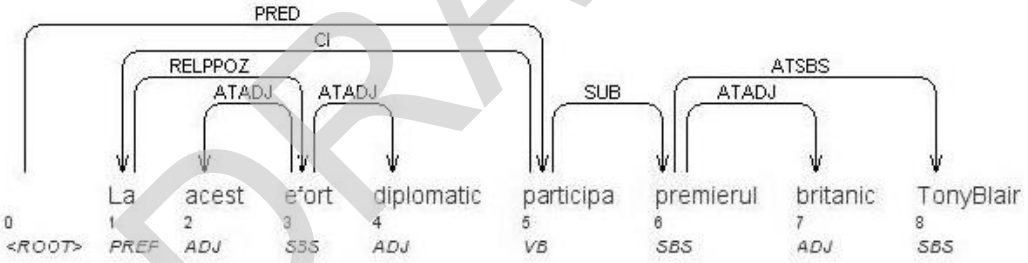

Fig. 1 Sample output of the Romanian dependency parser of Călăcean & Nivre (2009).

In a project with similar goals, a new dependency treebank is under construction at the University of Iaşi[6], which is expected to overcome the severe limitations imposed by the above-mentioned treebank.

Finally, another related report is given in Şaupe et all. (2009) on work done in the framework of a project on sentence analysis for Romanian. Despite the title, this work is not on syntactic parsing proper, since it is not concerned with sentence structure. It is limited to the lexical level, and provides a preliminary shallow lexical analysis aimed at identifying paragraph, sentence and word boundaries.

[2] Adam Kilgarriff, personal communication. April, 2010.
[3] MaltParser is freely available at http://maltparser.org/. Accessed May, 2010.
[4] http://www.larflast.bas.bg/balric/. Accessed March, 2010.
[5] Diacritics are unfortunately absent from the treebank used by Călăcean & Nivre (2009).
[6] Dan Cristea and Augusto Perez, personal communication, April 2010.

To summarize, there is currently no large scale syntactic parser available for Romanian, but work is in progress for developing both syntactically annotated resources and (shallow) analysis tools. In this paper, we present the first steps towards building a symbolic parser for Romanian, able to fully analyse unrestricted text. Unlike previous work concerned with dependency relations, phrase chunks, or specific syntactic relations in a sentence (as in the Romanian Sketch Engine), our parser aims to create a complete syntactic structure for the input sentence. In the cases when this is not possible, the parser returns disconnected parse trees for the parts of the input sentence it succeeded to analyse.

3. THE *FIPS* MULTILINGUAL PARSER

Fips (Wehrli, 1997; Wehrli, 2007; Wehrli & Nerima, 2009) is a deep symbolic parser developed at the Language Technology Laboratory of the University of Geneva. It is based on an adaptation of generative grammar concepts, as inspired by the Minimalist Program (Chomsky, 1995), the Simpler Syntax model (Culicover & Jackendoff, 2005), and the Lexical Functional Grammar (Bresnan, 2001). Each syntactic constituent is represented as a simplified X-bar structure of the form [$_{XP}$ L X R] with no intermediate levels. X stands for one of the following lexical or functional categories: N – noun, A – adjective, D – determiner, V – verb, Adv – adverb, P – preposition, Conj – conjunction, Interj – interjection, C – complementizer, T – tensed VP (head of a sentence), and F – functional phrase (representing predicative objects). L and R stand for (possibly empty) lists of left and right sub-constituents, respectively.

The lexical level contains detailed morphosyntactic and semantic information available from the manually-built lexicons, namely, selectional properties, subcategorization information, and syntactico-semantic features likely to influence the syntactic analysis. Thus, the parser relies on a strong lexicalist grammar framework.

Written in Component Pascal, Fips adopts an object-oriented implementation design that enables the coupling of language-specific processing modules to a generic module. The generic module defines the basic data types and is responsible of the parser's main operations, *Project* (assignment of constituent structures to lexical entries), *Merge* (combination of adjacent constituents into larger structures), and *Move* (creation of chains by linking surface positions of "moved" constituents to their corresponding canonical positions).

The parsing algorithm proceeds in a left-to-right, bottom-up fashion, by applying at each step one of the main operations enumerated above. The application of these operations is constrained by both language independent grammar rules (implemented in the generic module) and language-specific rules (defined for each language supported by the parser). Thus, the application of the *Merge* operation, in which a left or right sub-constituent is attached to the current structure, is constrained by language-specific licensing rules, like the agreement rules. Moreover, the attachments can only be made to a node that is active, i.e., a node that accepts sub-constituents. The alternatives are pursued in parallel, and several pruning heuristics are employed for limiting the search space.

Given a sentence, the parser provides both the phrase structure representation (same as the constituent structure, *c*-structure, in LFG) and the representation of grammatical functions for constituents, in the form of a predicate-argument table (similar to the *f*-structure in LFG). The parser also provides an interpretation for clitics, *wh*-elements, and relative pronouns, a process which can be assimilated to intra-sentential pronoun resolution. Since it adopts the theory of movement, the parser considers that words may leave their original "deep" (or canonical) position, and move to their final surface position, due to grammatical transformations (for instance, interrogation and relativization). The parser therefore builds co-indexation chains which link

extraposed (moved) elements to their empty original position, where empty constituents are created to mark the "trace" left by the movement.

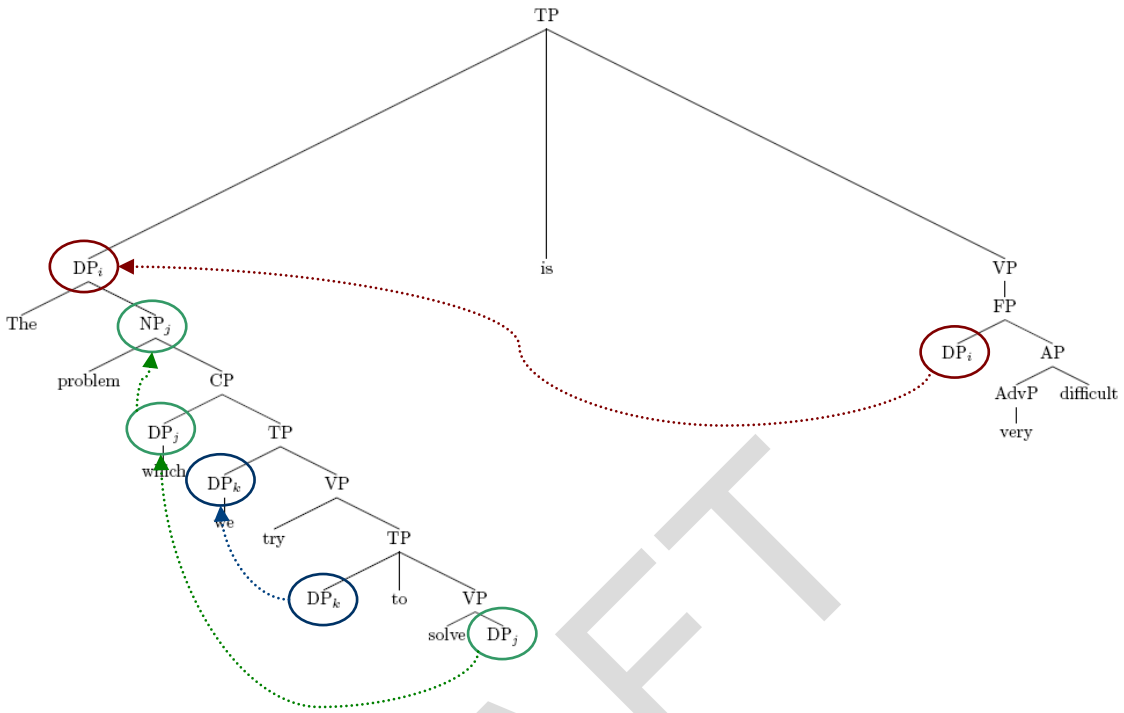

Fig. 2 Sample output of FipsEnglish showing co-indexation chains.

Figure 2 above displays graphically the constituent structure returned by Fips for the English sentence *The problem which we try to solve is very difficult.* The extraposed noun *problem* leaved its "deep" canonical position of direct object for the predicate *solve* and took the surface position of subject, due to a relativization transformation. The parser links the trace of the noun (DP_j) to the relative pronoun *which* and further to the noun in the subject position (NP_j). Therefore, in the predicate-argument table associated with *solve*, *problem* will be found on the direct object position.

Similarly, the parser builds a co-indexation chains between the empty subject of *solve* (the lower DP_k) and the overt subject of *try* (the higher DP_k). The pronoun *we* will thus be found on the subject position in the predicate-argument table of *solve*. The co-indexation denoted by *k* therefore allows us to infer that the subject of the verb *solve* is in fact the subject of the c-commanding (higher) verb *try*.

Fips is currently available for English, French, Spanish, Italian, German, and Greek, and has a high grammatical coverage for these languages (other languages are also supported to a certain extent). Fips is able to handle a wide range of constructions in these languages, such as the ones illustrated below for English:

- relativization: *the record which he broke …*
- passivization: *the record was broken …*
- interrogation: *which record did he break ?*
- cleft constructions: *it is the world record that was broken …*
- coordination: *the record set by X and later broken by Y…*
- apposition: the *record, previously held by X, was broken …*

One of the key features of Fips is its robustness. It can process large text collections at a reasonable speed, of approximately 150 tokens/s. Its precision is currently being measured in the framework of two parsing evaluation campaigns for French, namely, *EASy – Evaluation des Analyseurs*

SYntaxiques[7], and *PASSAGE – Produire des Annotations Syntaxiques à Grande Échelle*[8]. For the time being, no definitive evaluation results have been made available.

4. EXTENDING *FIPS* TO ROMANIAN: TWO MAIN TASKS

The language-specific part of the Fips parser consists, on the one hand, of the grammar rules of a given language, and, on the other hand, of the lexicon of that language. The grammar rules specify under which condition the parser's main operations, Project, Merge and Move (cf. Section 3), may apply in order to enable the creation of a parse tree, given an input sentence. The lexicon of the language contains entries for simple lexemes along with complex lexemes (i.e., compound words, collocations and idioms), which are enriched with morphosyntactic and semantic information whose role is to guide the parser. Extending Fips to a new language therefore means performing two main preliminary tasks: grammar specification, and lexicon compilation. In this section, we detail this process for Romanian.

4.1. Grammar specification

A description of the Romanian syntax for the purposes of Fips, consistent with a traditional grammar (Popescu, 2004) and a generative grammar (Dobrovie-Sorin, 1994) for Romanian and adapted to the theoretical model used by Fips, has been made available by Soare (2005). Based on this description, grammar rules for Romanian have been specified in a pseudo-formalism specific to Fips. This formalism has the advantage that it is easy to adopt by linguists and, at the same time, it is close to the program code of the parser. Most rules in the grammar specification refer to the conditions under which two adjacent constituents can be attached by the Merge operation in order to yield a larger constituent. An example of such a rule is provided in Figure 3.

```
D+T                                subjectAttachment
    a.HasCase(nominative)
    a.AgreeWith(b, {number, person})
    b.IsTensed
```

studentul scrie
student.the writes
the student writes

Fig. 3 Sample left attachment rule.

This rule describes the subject attachment, in which a DP, denoted by *a*, is attached as a left sub-constituent of a tensed VP, denoted by *b*, if the case of *a* is nominative and the conditions of agreement in number and person between *a* and *b* are satisfied.

```
D+N
    a.IsType(indefinite)
    b.IsType(commonNoun)
    a.AgreeWith(b, {gender, number})
```

un ***student***
a student

Fig. 4 Sample right attachment rule.

Figure 4 shows the example of a right attachment, a type of attachment which is relatively more frequent in Romanian. This rule enables the attachment of a NP as a right sub-constituent of a DP, when the determiner is indefinite, the noun is common, and there is agreement in gender and number. In the current Romanian specifications, about 100 grammar rules are described, a quarter of which concern left attachments and the others right attachments. This number is comparable to that for other languages supported by the parser, and the coverage of these rules is judged, at least in

[7] http://www.technolangue.net/article198.html. Accessed May, 2010.
[8] http://atoll.inria.fr/passage/. Accessed May, 2010.

principle, thorough. However, the implementation of these rules is far from complete. Currently, about half of the attachment rules are fully implemented and tested, and the status of the Romanian parser is that of work in progress.

Apart from the attachment rules, the implementation of the grammar also requires accounting for the syntactic transformations possible in Romanian, and, correspondingly, for the creation of co-indexation chains (cf. Section 3). These processes are already dealt with in FipsRomance, the grammatical component of Fips that models the Romance family of languages, to which Romanian belongs. Some refinements are however needed, because in certain respects the Romanian language exhibit distinctive properties.

A case in point is the clitic system, which is richer in Romanian than in other Romance languages, such as French or Italian (Monachesi, 2000). The Romanian clitic complex involves pronouns, negation, auxiliaries as well as a restricted subclass of monosyllabic adverbs (*mai* "again", *cam* "little", *prea* "too", *şi* "also", *tot* "still"). The order of these elements is rigid: negation is the leftmost element, preceding the clitics and the auxiliary; dative clitics precede accusative clitics, and monosyllabic adverbs fill a position between the auxiliary and the participial verb (cf. Example 1a).

(1) a. *nu ţi l-am mai prezentat pe Dan*
not you.cl.Dat he.cl.Acc-haveagain introduce PE Dan
"I didn't introduce Dan to you anymore"
b. ***nu ţi-am mai prezentat pe Dan***

Romanian is similar to Spanish but differs from French and Italian in that the Accusative DP, marked by the preposition *pe*, must be clitic doubled (cf. Example 1b and 2).

(2) a. *L-am văzut pe Ion.*
He.cl.Acc-have seen PE Ion.
"I have seen Ion."
b. ***Am văzut pe Ion.***

Clitic doubling is, however, optional for full Dative DPs, as shown in Example 3.

(3) (*Le)-am dat fetelor flori.*
They.cl.Dat-have given girls.Dat flowers.
"I gave the girls flowers."

In Romanian, unlike in French and Italian, no material can intervene between the auxiliary and the participial verb, except for the monosyllabic adverbs mentioned above. Our implementation is consistent with Monachesi's (2000) theoretical account, which postulates a compound structure for the Romanian auxiliary verb system, rather than a flat structure which is suitable for other Romance languages.

Another difference between Romanian and other Romance languages pertains to the *wh*-elements (Soare, 2009). Similarly to Slavic languages, the Romanian language permits multiple *wh*-fronting, which is not possible, for instance, in French or Italian (cf. Example 4a-c). The *wh*-phrases are rigidly ordered, obey the Superiority Condition[9], and no material can intervene between them (cf. Example 4d).

[9] Broadly speaking, the Nominative precedes the Dative, and the Dative precedes the Accusative.

(4) a. Cine cui ce a spus?
Who.who.Dat what has said?
"Who said what to whom?"
b. *Qui à qui quoi a dit ?
c. *Chi a chi cosa ha detto?
d. *Ce cine cui a spus?

To sum up, the Romanian grammar is fully described in our project in a formalism adapted to Fips; its implementation, currently in progress, accounts for whole gamut of possible constituent attachments, and, while building upon the existing Romance parsing module for dealing with grammatical transformations, it pays particular attention to the proper treatment of language-specific phenomena.

4.2. Lexicon compilation

The Romanian language has a rich morphology, since it inherited, in part, the Latin declension system. There nominal case system comprises the nominative-accusative and the dative-genitive syncretic cases, as well as the vocative. Inflected forms are obtained by suffixation. A particularity of Romanian, among all other languages of the same family, is that the definite declension is also obtained through suffixation. There are three genders, masculine, feminine, and neuter; the neuter gender behaves like masculine for singular and feminine for plural. Adjectives also exhibit declension in case, number, gender and definiteness. Verbs are highly inflected for person, number, tense, and mood.

The Fips parser conveniently includes a morphological generation module, which produces all the inflected forms of a lexeme, according to the appropriate declension paradigm. Generation rules are specified in a Fips-specific format, automatically processed by the morphological generator. A rule accounts for the production of a specific form, which will be inserted in the lexicon together with the applicable morphological features. Thus, given a base word form, its lexical category and a numerical code representing the inflection class, the morphological generator reads all the rules applying to that class and prepares the appropriate lexical entries. This procedure greatly simplifies the construction of the lexicon.


```
INFL
- "it" = (cat:V, inflClass:18, base:1, tense:pastPart,
   gender:{masc, neut}, pers:{1, 2, 3}, num:sing).
```

Fig. 5 Sample inflection rule used for morphological generation.

Figure 5 above gives the example of a simple inflection rule, used for generating the past participle form for verbs of a given inflection class—in this case, 18—by appending the suffix *-it* to the root (the root is obtained from the present infinitive). This rule specifies the morphological features for this form, e.g., masculine or neuter genders and singular number. Other rules will be used for other gender-number combinations, leading to the production of different forms. The present rule will generate, for instance, the past participle *venit* of the verb *a veni* "to come", once it is stated that *a veni* is a verb belonging to the inflection class 18.

The process of compilation of the FipsRomanian lexicon went through several stages. A list of base word forms was first obtained from the DEX dictionary (DEX, 1998). An inflection class number was automatically assigned to most nouns and adjectives, based on their suffix. This, in conjunction with the inflection rules defined for each class, allowed for the automatic generation of inflected forms for these lexemes. A part of the remaining nouns and adjectives were manually entered into the lexicon, as were pronouns, determiners, and the most common verbs.

Verbs, in particular, require detailed specific information about subcategorization, selectional features, grammatical function for arguments, thematic functions, and other information (for instance, on aspect), which can only be specified manually. Nonetheless, the morphological generation process is still very useful, since verb paradigms contain very numerous forms, and it would be too time-consuming to add them manually.[10] For the time being, the FipsRomanian lexicon contains slightly more than 3600 verbal lexemes, a low coverage relative to that of nouns (above 38000 lexemes) and adjectives (above 14000 lexemes).

As far as non-content words are concerned, part of them have been added automatically (e.g., interjections), the others semi-automatically or manually. The human intervention was necessary in order to specify, for each entry, detailed information (like, for instance, subtype for adverbs: negation, location, time, etc.) which is not available in DEX.

In addition to the entries compiled from DEX, the FipsRomanian lexicon also contains proper nouns, mostly for places (cities, countries, rivers, mountains) and persons (the most usual first names and surnames, separately). The majority of them has been collected from different repositories available on the Internet[11], and has been automatically entered in the lexicon. Table 1 displays the current composition of the FipsRomanian lexicon, by lexical category.

Table 1: Distribution of lexical categories in the FipsRomanian lexicon (May, 2010).

Lexical category	**Inflected forms**	**Lexemes**
Noun (common)	254410	38635
Noun (proper)	9269	9241
Pronoun	155	74
Clitic	22	5
Adjective	76203	14280
Verb	49340	3611
Adverb	879	882
Interjection	413	413
Preposition	114	116
Conjunction	72	76
Determiner	161	57
Total	391038	67390

In addition to single-word entries, the lexical database of the Fips parser also contains multi-word entries. A first category of such entries is represented by compound words (for instance, complex prepositions, conjunctions and adverbs: *de jur împrejurul* "around", *dat fiind că* "given that", *până când* "until", as well as a few complex proper nouns: *Câmpulung Moldovenesc*). These have, however, the status of single lexemes, since they behave like single words.

A second category is represented by collocations, which cover, as a special case, the idioms. Collocations are (semi-)compositional, language-specific restricted combinations of words, like *a atrage atenţia*, "to draw the attention" (lit., *to attract attention*). Like idioms, collocations pose production problems to non-native language speakers, but unlike idioms, they do not really pose comprehension problems. Idioms (e.g., *a pune la punct* "to fix', lit., *to put to point*) are the semantically non-compositional extreme of the collocations continuum (McKeown & Radev, 2000). On the other extreme, one finds collocations that are more similar to free combinations, like *mare importanţă* "high importance" (lit., *big importance*).

Since collocations allow the insertion of additional material between the component items, they cannot be stored in the lexicon in the same way as compounds. They are stored as binary associations of lexemes, where each item can be either a single word, or an existing multi-word

[10] There are about 35 inflected forms for a Romanian verb.

[11] For instance, http://ro.wikipedia.org/ and http://www.archeus.ro/. Accessed October, 2009.

entry (in particular, a compound or another collocation). Thanks to the availability of FipsRomanian, we recently could run our first experiments of collocation extraction from corpora, based on syntactically informed methods (Seretan, 2008). The results obtained allowed us to the put the basis of the Romanian collocation lexicon, which currently contains a few hundred collocations. This work will be detailed in a later section.

Figure 6 displays the interface to the lexical database of the parser. It shows the two entries that exist in the Romanian lexicon for the word *venit*, corresponding to the two possible readings: as past participle of the verb *a veni* "to come", and as a common noun, *venit* "income". The left-hand side present word-related information (applying to the inflected form), and the right-hand side lexeme-related information (pertaining to the corresponding lexeme). There are multiple lexemes for the verb *a veni*, with different subcategorization, as shown in the upper list. The features displayed below apply to the currently selected lexeme. Among other things, the interface displays the collocations in which a lexeme participates.

Information about the coverage of the Romanian lexicon, when tested on a collection of texts, will be provided in the Section 5 that is dedicated to the experimental results.

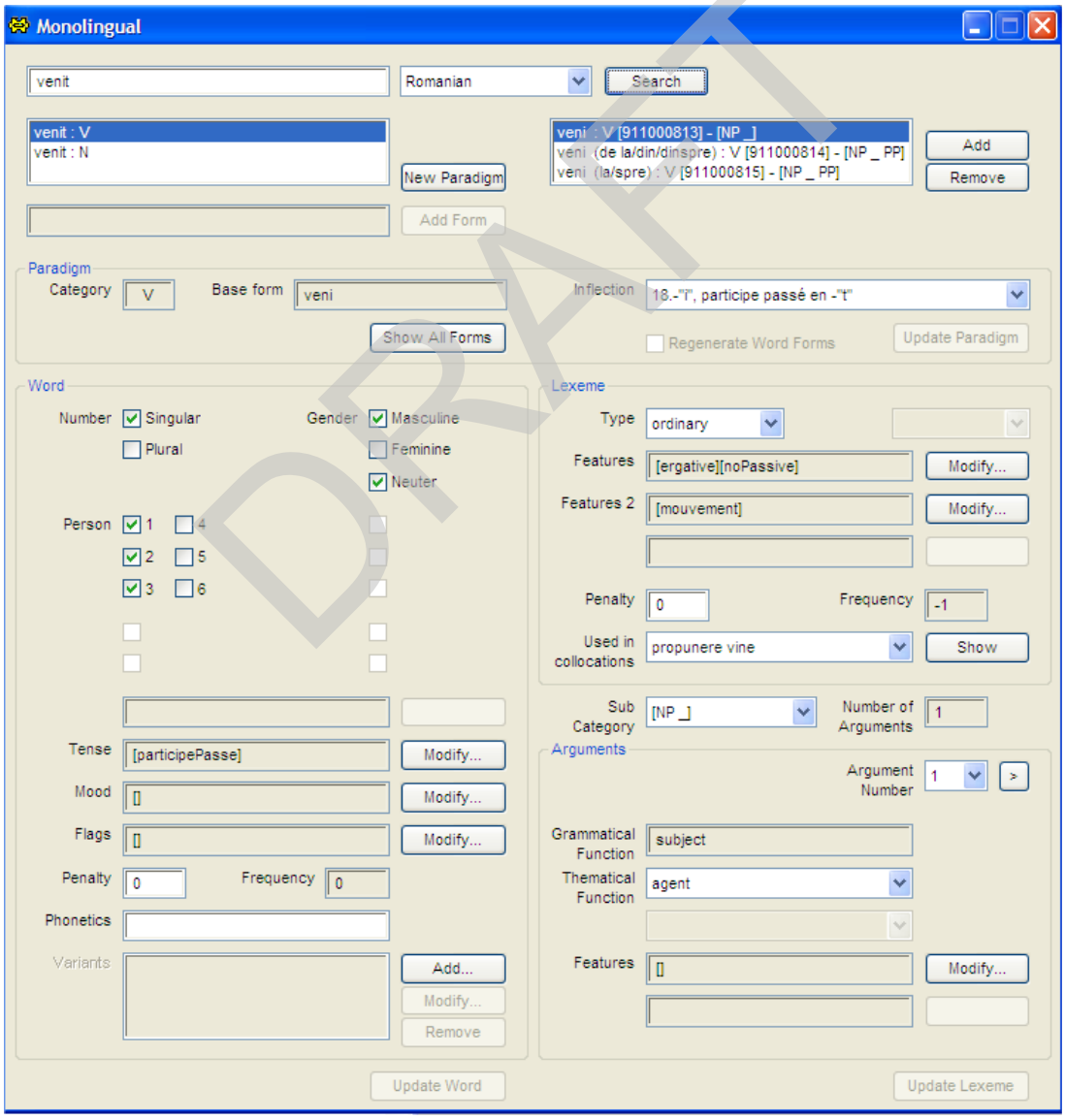

Fig. 6 Lexicon interface (screen capture).

5. EXPERIMENTAL RESULTS

As mentioned in the previous section, the preparation of the resources needed for extending Fips to Romanian (namely, the lexicon and the grammar) has already reached completion; however, the implementation of the grammar is still at an incipient stage. We started with simple structures for phrases, short declarative sentences, and then moved to more complex structures (subordination and coordination of clauses, interrogation, etc). Example 5 illustrates the kind of structures that FipsRomanian is able to analyze correctly. The parse trees are shown in the typical parenthetical notation.

(5) a. *un copac cu flori*
[DP un [NP copac [PP cu [NP flori]]]]
b. *Copiii scriu scrisori părinţilor.*
[TP [DP Copiii] scriu [VP [NP scrisori][DP părinţilor]]]
c. *Ghici cine vine mâine la cină.*
[TP [DP] Ghici [VP [TP[NP cine] vine [VP [AdvP mâine][PP la [NP cină]]]]]]
d. *Toţi angajaţii societăţii participă la această sedinţă de informare şi socializare.*
[TP[DP Toţi [DP angajaţii [DP societăţii]]] participă [VP [PP la [DP această [NP sedinţă
[PP de [NP [ConjP[NP informare] şi [NP socializare]]]]]]]]]

Figure 7 shows, in graphical form, the output obtained for a sentence involving a relative construction. As can be seen, a co-indexation chain is created that links the canonical position of subject in the subordinated clause to the relative *care* and then to the surface position in the matrix clause (see also the discussion in Section 3).

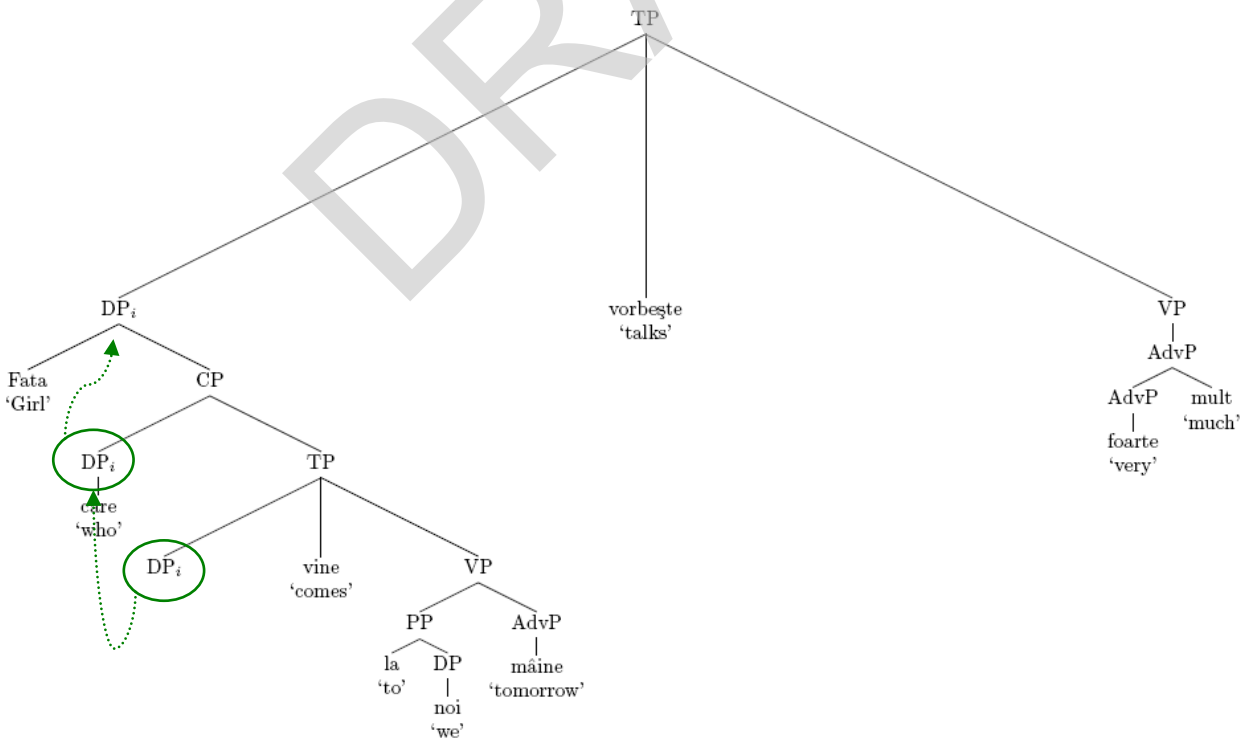

Fig. 7 Output of FipsRomanian for a relative construction, showing co-indexation.

As a by-product of parsing, a POS-tagger version of the output is available, which lists detailed information for each input token. A sample POS output is provided in Example 6 below.[12] As can be seen, the POS-tagging output also contains information on the predicate-argument structure.

(6) Mama NOM-COM-SIN-FEM-NOM-ACC 911000300 mamă SUBJ
spune VER-INF-PRE-3-SIN 911000443 spune SUB:mama DO:poveste IO:copilului
o DET-IND-SIN-FEM-NOM-ACC 911067636 o OBJ
poveste NOM-COM-SIN-FEM-NOM-ACC 911054772 poveste
copilului NOM-COM-SIN-MAS-DAT-GEN 911023143 copil IND-OBJ
. PONC-point 0

Even though FipsRomanian did not yet reached maturity, we were interested in evaluating to which extent the current version can be applied on real data, if it is robust enough, and if the results obtained would prove already useful. The remaining of this section presents the experiment performed to this end.

5.1. Data

The FipsRomanian parser was applied on collection of newspaper articles, totalling 1.2 million tokens (corresponding to 1.05 million words), which have been gathered from various sources on the Internet. These included the *BBC Romanian website*, *Gândul*, *Adevărul*[13] (articles from 2006), and, as the main source, the newspaper *Mesagerul*[14] (articles from the period 2006–2008). The collection of articles has been done manually. The main criterion for choosing the source sites was the proper encoding of diacritics.[15]

A few more statistics on these texts could be derived after parsing was performed: the average sentence length is 26.9 tokens, including punctuation marks, and there are in average 113.2 sentences per file (the whole collection contains 393 files). The corpus was not pre-processed in any way, therefore there are spans of texts corresponding to the article header (e.g., title, date), which might affect the parser since they do not constitute full sentences.

5.2. Results

FipsRomanian succeeded in processing the whole text collection, at a speed of 429 tokens/s. The processing took slightly less than one hour and was done on a standard PC configuration. A total number of 44483 sentences have been indentified and analysed. The parser could build a complete parse tree in 16.2% of the cases. For the others, it returned multiple disconnected parse trees. The average length of the partial parses is 5.3 tokens. The percentage of fully parsed sentences is much lower that that obtained by the parser Fips for other languages; for comparison, the English and French versions, the most developed, achieve around 80% complete analyses on journalistic data.

[12] Note that, contrary to many shallow parsing systems, the POS tags is a product, not a prerequisite of parsing.

[13] http://www.bbc.co.uk/romanian/, http://www.gandul.info/, http://www.adevarulonline.ro/. Accessed 2006.

[14] http://www.mesagerul.ro/. Accessed 2006–2008.

[15] In the Romanian alphabet, there are a few symbols that use diacritics (ă, â, î, ş, ţ). The online Romanian mass-media is not consistent in using these diacritics. Only a minority of online newspapers currently uses the correct alphabet.

As for the lexical coverage, this can be judged as rather satisfactory, as only 6.5% of the input tokens were not found in the Romanian lexicon[16]. The token-type ratio is 4.02. Moreover, about half of the unknown words are a priori proper nouns, as they begin with a capital letter (39.2% of them). Fips adopts a guessing strategy for the unknown words, trying to assign them a category compatible with the analysis pursued, and to attach them in the output tree. Thus, the impact of the unknown words on the success rate of the parser is rather reduced. The key factor for improving FipsRomanian is continuing the implementation of the grammatical component.

For the time being, given the incipient stage of implementation, we cannot provide more meaningful evaluation results for the parser. This will be a concern of our future work, in which we plan to test the precision and recall of FipsRomanian on the dependency treebanks mentioned in Section 2.

5.3. Application

In order to test to what extent the (often fragmented) parsing analyses produced by FipsRomanian are useful from a practical point of view, we attempted to use them in specific application: as suggested in Section 4, the application considered was a lexical acquisition application, whose goal is, more specifically, to detect collocations from parsed corpora. Until now, the collocation extraction application (Seretan, 2008) has systematically been run on corpora in the other languages supported by Fips, and the results have been used as raw material for manual inclusion in the lexical database. Furthermore, bilingual correspondences for the collocations indentified are being added into bilingual lexicons and used in an in-house rule-based machine translation system. Collocations are also being used, in turn, in further parsing processes, in order to inform attachment disambiguation decisions. But the most directly perceived application of collocation is the language generation, since collocations constitute a major means to ensure language fluency. Collocations not always translate literally: compare *faire attention* (in French; lit., "make attention") to the English equivalent *pay attention*, or *Entscheidung treffen* (in German; lit., "encounter a decision") to the English counterpart *make a decision*.

The extraction application applies a hybrid procedure, combining syntactical information and statistical methods. Broadly speaking, it first builds a list of syntactically-valid candidates, and then applies association measures to retain the candidates likely to constitute typical expression of a language. Applied to Romanian, this procedure collected pairs of words in predefined syntactical relations (like adjective-noun, verb-object etc.) from both complete and incomplete parse trees. These pairs have been scored using log-likelihood ratio, a measure typically used for collocation extraction.

We manually investigated the top 2000 extraction results with the help of our concordance tool (shown in Figure 8). Among the top 2000 pair types, a number of 606 have been retained as lexicographically interesting and have been entered in our collocation database. Thus, the extraction precision[17] is 30.3%. This is a large difference from the precision of 65.9% obtained for the top 500 pair types in an experiment for French. However, despite the fact that Romanian version of the parser is comparatively much less developed, our finding suggests that the results are already useful for practical applications.

[16] The lexical coverage of Fips for French data is about 98%. Only 2% of the tokens in a corpus are unknown.

[17] The extraction precision is the percentage of collocations in the results investigated. In contrast, the grammatical precision, i.e., the percentage of grammatical pairs in the results investigated, is much higher. We did not quantify it, as the focus of our work was to collect collocations.

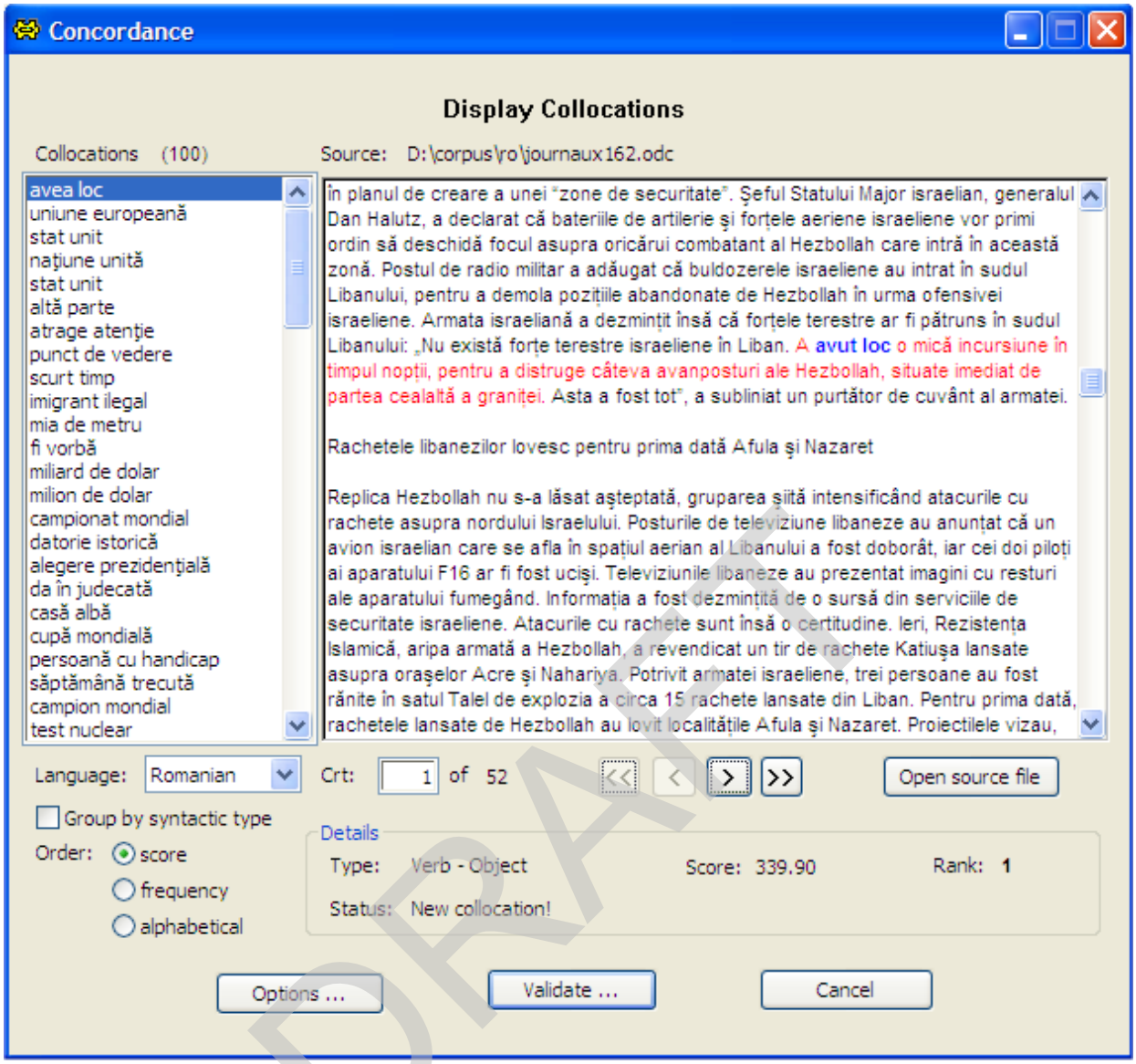

Fig. 8 Interface of the collocation concordancer, showing the top extraction results for Romanian.

6. CONCLUSION

A syntactic parser is a central tool for any language. Largely absent from the otherwise vast repertory of Romanian language resources, such a tool is under development at the University of Geneva, as part of a project of multilingual extension of a symbolic parser. We reported on the preliminary efforts made to build the necessary resources for the Romanian version, namely, the compilation of the lexicon and the specification of the grammar. We also described the current state of the grammar implementation. Although this is still in an incipient stage, we could report on the first results obtained by parsing a large amount of unrestricted text, and showed that these are useful for a practical application concerned with lexical acquisition. In future work, we will continue consolidating the lexicon; we will concentrate on the implementation of the grammar, and will perform appropriate evaluation experiments, in order to provide more insightful information on the performance of the parser.

ACKNOWLEDGEMENTS

This work has been supported by the Swiss National Science Foundation (grant no. 100012-117944). The authors acknowledge the contribution of Gianina Aonofriesei, Maria Husarciuc, Luka Nerima, Gabriela Soare and Diana Tradabăţ to this undertaking. Particular thanks are due to the two anonymous reviewers and the audience of the PROMISE 2010 workshop in Iaşi, for useful suggestions that helped enrich the content of this paper.

REFERENCES

1. BRESNAN, J. *Lexical Functional Syntax*. Blackwell, 2001.
2. CHOMSKY, N. *The Minimalist Program*. MIT Press, 1995.
3. CRISTEA, D. Romanian language resources and tools, http://www.clarin.eu/files/cnl04_web.pdf, 2009, accessed October, 2009.
4. CRISTEA, D. and FORĂSCU, C. Linguistic Resources and Technologies for Romanian Language, *Computer Science Journal of Moldova*, **14**, 1(40), 2006.
5. CĂLĂCEAN, M. and NIVRE J. A data-driven dependency parser for Romanian, In *Proceedings of the 7th International Workshop on Treebanks and Linguistic Theories (TLT 7)*, Groningen, Holland, pp. 65–76, 2009.
6. CULICOVER, P. and JACKENDOFF, R. *Simpler Syntax*, Oxford University Press, 2005.
7. DEX. *Dicţionarul explicativ al limbii române*, Academia Română, 1998.
8. DOBROVIE-SORIN, C. *The Syntax of Romanian: Comparative Studies in Romance*, Mouton de Gruyter, 1994.
9. GILDEA, D. and PALMER, M. The necessity of parsing for predicate argument recognition, In *Proceedings of ACL 2002*, Philadelphia, 2002.
10. KILGARRIFF, A., RYCHLY, P., SMRZ, P., and TUGWELL, D. The Sketch Engine, In *Proceedings of the Eleventh EURALEX International Congress*, Lorient, France, pp. 105–116, 2004.
11. MAYNARD, D. and ANANIADOU, S. A linguistic approach to terminological context clustering, In *Proceedings of the Natural Language Pacific Rim Symposium (NLPRS '99)*, Beijing, China, pp. 346–351, 1999.
12. MCKEOWN, K. R. and RADEV, D. R. Collocations, In Robert Dale, Hermann Moisl, and Harold Somers, editors, *A Handbook of Natural Language Processing*, pp. 507–523, Marcel Dekker, 2000.
13. MONACHESI, P. Clitic placement in the Romanian verbal complex, In B. Gerlach and J. Grijzenhout (eds.), *Clitics in Phonology, Morphology and Syntax*, John Benjamins, 2000.
14. NIVRE, J., HALL, J., NILSSON, J., CHANEV, A., ERYIĞIT, G., KÜBLER, S., MARINOV, S. and MARSI, E. Maltparser: A language-independent system for data-driven dependency parsing, *Natural Language Engineering*, **13**, pp. 95–135, 2007.
15. PADÓ, S. and LAPATA, M. Dependency-based Construction of Semantic Space Models, *Computational Linguistics*, **33**(2), pp. 161–199, 2007.
16. POPESCU, Ş. *Gramatica practică a limbii române*, Editura TEDIT FZH, 2004.
17. SERETAN, V. *Collocation Extraction Based on Syntactic Parsing*, Ph.D. thesis, University of Geneva, 2008.
18. SOARE, G. *Romanian syntax*, Technical report, Language Technology Laboratory, University of Geneva, 2005.
19. SOARE, G. *The Syntax-Information Structure Interface: A Comparative View from Romanian*. Ph.D. thesis, University of Geneva, 2009.
20. ŞAUPE, A., TEODORESCU, L. R., ORDEAN, M. A., BOLDIZSAR, R., ORDEAN, M. and SILAGHI, GH. C. Efficient parsing of Romanian language for text-to-speech purposes, In V. Matouček and P. Mautner (eds.), *Text, Speech and Dialogue 2009*, pp. 323-330, Springer-Verlag, 2009.
21. TRANDABĂŢ, D. M., CRISTEA, D. and TUFIŞ, D. (eds.) *Proceedings of the Workshop Linguistic Resources and Tools for Processing Romanian Language* (in Romanian: *Lucrările atelierului*

"Resurse lingvistice şi instrumente pentru prelucrarea limbii române"), University "Alexandru Ioan Cuza" Publishing House, 2008.
22. WEHRLI, E. *L'analyse syntaxique des langues naturelles: Problèmes et méthodes*. Masson, 1997.
23. WEHRLI, E. Fips, a "deep" linguistic multilingual parser, In *ACL 2007 Workshop on Deep Linguistic Processing*, Prague, pp. 120–127, 2007.
24. WEHRLI, E. and NERIMA, L. L'analyseur syntaxique Fips, In *Proceedings of the IWPT 2009 ATALA Workshop: What French parsing systems?*, Paris, France, 2009.