Lexicalization of Grammars with Parameterized Graphs

Olivier Blanc and **Matthieu Constant**
Université of Marne-la-Vallée
5, bd Descartes
Champs-sur-Marne
77 454 Marne-la-Vallée, France
{oblanc,mconstant}@univ-mlv.fr

Abstract

This paper is about the use of large coverage syntactic lexicon for text parsing. Our work focuses on the construction of a lexicalized unification grammar using the fine-grained syntactic information encoded in the lexicon-grammar tables built at LADL (France). We present a method to generate this grammar from a hand-built meta-grammar composed of parameterized graphs. For each lexical item of our lexicon, a specialized grammar is generated by resolving the parameters referring to syntactic properties encoded in the lexicon-grammar tables. We also show that our method can be adapted to a more complex lexicon in the form of relational tables.

1 Introduction

Over the past ten years, interest in the development and use of Language Resources (LR) have increased dramatically and become a global concern. This interest is not confined to corpora, but extends to lexicons and grammars. For instance, as interaction between descriptive linguistics and language engineering is growing, Natural Language formalisms are now being adapted to the interaction between lexicon and grammars such as LTAG (Schabes et al. 1988; Abeillé 2002; XTAG Group Research) and related frameworks (Carroll et al. 1998) or HPSG (Pollard et al. 1994).

Our goal is to develop a robust syntactic parser dealing with real texts. This involves the construction of a fine-grained lexicalized grammar. In this paper, we present a method inspired by Roche (1993) to build such a grammar semi-automatically by using large-coverage lexicon-grammar resources (Gross 1994) and a system of parameterized graphs.

This paper will be preliminary devoted to a brief description of the Language Resources used (section 2) and then a detailed introduction to our grammar formalism (section 3). The last sections (4 and 5) will focus on the lexicalization process and some extensions.

2 Language Resources

Over the last thirty years, the informal network RELEX of laboratories in the domain of Linguistics and Computational Linguistics (http://infolingu.univ-mlv.fr), has been constructing hand-built lexical resources in several languages (French, English, Portuguese, Spanish, German, Korean, Thai, ...). Especially, their effort focused on the construction of exhaustive syntactic dictionaries in the framework of the lexicon-grammar methodology initiated by Gross (1975). The lexical entries are predicative elements, either verbs, nouns or adjectives (simple words or multiword expressions). For each predicate, a set of syntactic properties is systematically examined such as:

- number and nature of the arguments (e.g. complemental clause, infinitive, human noun phrase, ...),
- appropriate prepositions,
- accepted transformations (e.g. passivation, argumental alternation, pronominalization, etc.),
- some co-reference resolutions.

All these properties are encoded into syntactic dictionaries in the form of tables called lexicon-grammar tables. Each row corresponds to a lexical value and each column corresponds to a syntactic property. A boolean value at the intersection of a row and a column indicates whether a given lexical entry verifies a syntactic property. Each table gathers predicative elements that have some syntactic similarities according to definitional criteria (Gross 1975). An example of a lexicon-grammar table is given in figure 1[1]; it represents a subset of French verbs with the definitional construction *N0 V que P* (N0 V that S)[2],

[1] A true value is represented by the symbol + (- for false)

[2] These verbs have a noun phrase as subject and are followed by a complemental clause

such as the verb *empêcher* (to prevent).

The French lexicon-grammar currently contains 15,000 simple verbs and 10,000 predicative nouns and adjectives. In addition, there is a dictionary of frozen sentences (composed of 30,000 entries). This linguistic work is still in progress.

N0=Nhum	N0=Nnr	N0=queP	N0=V1W	entry	N0 V	N1=quePind	quePind=deV0W	N1=quePsubj	quePsubj=deV0W	quePsubj=V0W	(N1)(deV-infW)	queP=Ppv	N1=Nhum	N1=N-hum	N1=le fait que P
+	-	-	-	détester	-	-	-	+	+	+	+	-	+	+	+
+	+	+	+	empêcher	-	-	-	+	-	-	+	+	-	+	-
+	+	+	+	encenser	+	-	-	+	-	-	+	-	+	+	+
+	-	-	-	envier	-	-	-	+	-	-	+	-	+	+	+
+	-	-	-	estimer	-	-	-	+	-	-	+	-	+	+	+
+	+	+	+	exalter	-	-	-	+	-	-	+	+	+	+	+
+	-	-	-	exécrer	-	-	-	+	+	+	+	-	+	+	+
+	-	-	-	fêter	-	-	-	+	-	-	+	-	+	+	+
+	-	-	-	flétrir	-	-	-	+	-	-	+	-	+	+	+
+	-	-	-	fustiger	-	-	-	+	-	-	+	-	+	+	+
+	-	-	-	haïr	-	-	-	+	+	+	+	-	+	+	+
+	-	-	-	honnir	-	-	-	+	+	-	+	-	+	+	+

Figure 1: sample of a lexicon-grammar table

3 Decorated RTN as a grammatical formalism

Our current research focuses on the exploitation of those accurate and systematic subcategorization descriptions and transformational properties encoded in the lexicon-grammar tables for large coverage text parsing. For this purpose, we are currently constructing a lexicalized unification grammar for French, which is generated semi-automatically from the syntactic tables using the methods described in the next section.

Our grammar is a syntagmatic grammar represented by a Recursive Transition Network (RTN) (Woods 1970) augmented with feature structure constraints. The different realizations of each syntactic constituent of the grammar are described in recursive finite state automata; those descriptions are decorated with functionnal equations that help formalize various linguistic phenomena such as the agreement between two constituents or the extraction of a grammatical item and long distance dependencies.

This formalism is actually very close to the Lexical Functional Grammar model (LFG) (Bresnan 1982), both models being equivalent from the point of view of their descriptive and computational capacity. The main difference is that, in our case, context-free rules are replaced by linguistic descriptions encoded in finite-state graphs.

Many phrases such as semi-frozen expressions (e.g. time adverbials, numerical determiners, ...) or named entities frequently occur in texts and exhibit lexical and syntactic local constraints that can be easily described in the form of finite state graphs (Silberztein 1994; Gross 1997). Such local grammars permit efficient recognition and can be integrated well as part of our whole grammar framework with RTN-parsing as a basis.

Moreover, the representation of syntactic constituents into recursive finite state automata allow a grammar writer to relate with ease syntactic constructions which are considered transformationally equivalent, like passivation, argument alternation, nominalization of a finite clause.

We believe such transformations cannot be considered as general syntactic rules but are strongly related with some specific lexical elements. Thus, in this context, each transformation must be described on a case by case basis for each predicative element, as described in the tables. The complexity of such systematic description can be greatly reduced by the use of parameterized graphs as will be shown in the next section.

For instance, the graph in figure 2 represents different realizations of French clauses having as main predicate, the verb *empêcher* (to prevent) as described in the lexicon-grammar table given in figure 1. On the left, we describe the possibility to have the subject as a NP or a sentential complement like a subjunctive clause introduced by the conjunction *que* (that) or an infinitive:

> *(Lea+Que Lea ait quitté Max+Boire du café) empêche Luc de dormir.*
> ((Lea+That Lea left Max+Drinking coffee) prevents Luc from sleeping)

The right part of the graph presents the possible realizations of the second argument which is a predicative NP (SN in French) or a subjunctive clause. The bottom path describes the possibility of raising the subject of the *que*-complement clause in position of direct object:

> (1) a. *Le soleil empêche que Luc travaille*
> = b. *Le soleil empêche Luc de travailler*
> (The sun prevents Luc from working)

In our formalism, labels prefixed with a colon (such as <:SN>, <:P> or <:V>) are non-

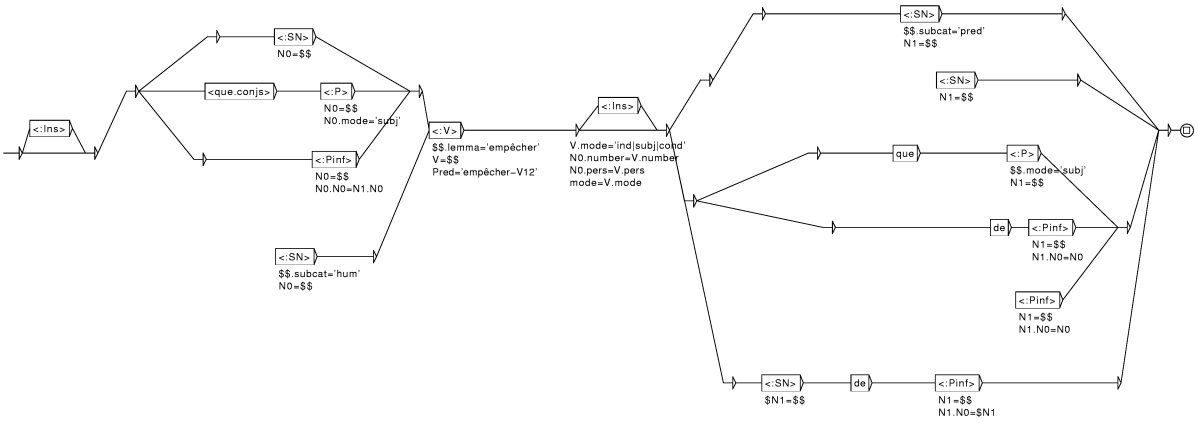

Figure 2: sentence constructions anchored by the verb *empêcher*

terminal symbols referring to syntactic constituents described in other graphs. For example, the label <:V> in the center of the figure refers to a graph describing the verbal complex of the sentence (which is the verb *empêcher* that might be modified by some adverbs, or modal and aspectual auxiliaries). Finally, the functional equations are given under the boxes and permit among others

- to verify the agreement in number and person between the verb and its subject (e.g. *N0.number=V.number*),
- to resolve some co-references, by identifying the subject of the infinitive (e.g. *N1.N0=N0*),
- and to identify the semantic predicate of the sentence with its arguments, while verifying that their natures are compatible with its subcategorization properties (e.g. *$$.subcat='hum'*[3]).

The result of the sentence analysis consists of a syntactic tree associated with a feature structure which contains all that information. Figure 3 is a simplified version of the feature structure resulting from the parsing of sentence 1.a and presents the semantic predicates with their essential arguments identified in the text.

$$\begin{bmatrix} CAT : P \\ Pred : empecher \\ N0 : \begin{bmatrix} CAT : SN \\ head : soleil \end{bmatrix} \\ N1 : \begin{bmatrix} CAT : Pinf \\ Pred : travailler \\ N0 : \begin{bmatrix} CAT : SN \\ head : Luc \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 3: simplified version of the feature structure obtained by parsing sentence 1.a with the automaton given in figure 2

4 Construction of a Lexicalized Grammar

We are currently building a lexicalized grammar for French using the formalism described above. This grammar is semi-automatically generated from lexicon-grammar tables. The construction of specialized grammars for each predicative element requires the construction of meta-grammars by hand. A meta-grammar is associated with a table and is composed of a set of parameterized graphs.

Each parameterized graph describes a syntactic consituent (finite or infinite clause, clause missing an extraposed element, etc.) whose predicate element is a variable which will be instantiated during the lexicalisation stage. Informally, a meta-grammar (i.e. the set of parameterized graphs associated with a table) can be seen as the specialized grammar for an abstract entry of the table, that would verify all the properties encoded. Each path is identified with a parameter referring to the property encoded in the corresponding table. A parameter has the following format *@X@*, where X is the name of the column referring to a syntactical property.

Once the meta-grammar of a table is con-

[3]Symbol $$ refers to the feature structure associated with the item in the box above

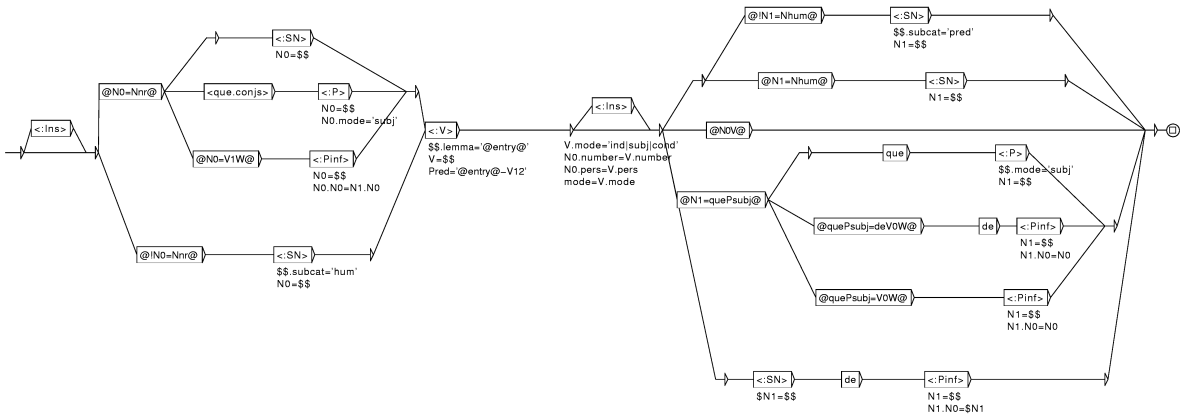

Figure 4: parameterized graph of declarative sentences for the table given in figure 1

structed, for each lexical entry, the generation process creates a specialized grammar where only the paths corresponding to the properties verified by the entry are kept. When the properties are not verified, the corresponding paths are removed. Columns can also contain textual value; in this case, the parameter is replaced by this value. It is also possible to negate a parameter: *@!X@* means that the paths corresponding to the property *X* are kept only if the value is false. For instance, figure 4 presents one of the parameterized graphs associated with the table given in figure 1. The lexicalized graph in figure 2 specialized for the verb *empêcher* has been generated from it. For instance, parameter *@N1=Nhum@* refers to the column indicating if the transitive complement can be a human Noun Phrase; parameter *@entry@* refers to the column providing the graphical form of the verb and parameter *@N0V@* refers to the column indicating whether this verb accepts the direct object ellipsis.

Note that it is theoretically possible to automatically produce the meta-grammars from the tables. However, this process is not straightforward because some syntactic properties encoded in the tables are specific to few tables only, and the meaning of a property can vary from a table to another. Moreover, some properties aren't explicitely encoded because they are accepted (or rejected) uniformally for all the verbs in a table. So we decided to write for each table, its associated meta-grammar manually.

Once the lexicalized graphs are automatically generated, we compute the union of the graphs for each syntactic constituent. Then, epsilon-transition removal, determinization and minimization are computed to obtain a grammar optimized for parsing. The construction of the whole lexicalized grammar for French is a long process. At this stage of our work, we only achieved the convertion of 17 tables (15 tables of verbs and 2 tables of nouns) which is about 15% of the whole set of tables and represent 2468 lexical entries. In its current state, the grammar, obtained from 137 parameterized graphs, contains 38,000 states and 70,000 transitions.

5 Extensions

5.1 Relational Tables

Standard syntactic dictionaries are in the form of simple tables. Nevertheless, it is sometimes more convenient to use *relational tables* to avoid duplication: for instance, this method has been used to represent time adverbials (Maurel 1990), geographical locative prepositional phrases (Constant 2003). A system of relational tables is composed of a set of tables (which includes a main table) and a set of relations between these tables. A relation is a special property that refers to a set of other properties in another table. This type of dictionaries, though similar to the standard ones, cannot be used straightforwardly in the lexicalization process described above and needs slightly different parameterized graphs. Actually, as information is split into multiple tables, a parameter should not only refer to a syntactic property (a column) but instead to the sequence of relations needed to reach the information pointed by the parameter. More detailed explanations can be found in (Constant 2003).

5.2 Meta-meta-grammars

The construction of the whole lexicalized grammar involves the construction of a parameterized graph for each type of constituents for each lexicon-grammar table. This process is costly because it requires many manual duplications. A more convenient way to deal with this would be to generate automatically every parameterized graphs related to a table from the same source. This source could be another kind of parameterized graph, let's call it meta-parameterized graph. The process of generation of the parameterized graphs from such a meta-meta-grammar requires a special table. Each row correspond to a type of constituent to be built, each column describes a property of those constituents such as the verbal tense, or the non-existence of a complement. Another approach using higher-level parameterized graphs has been studied in (Paumier 2003).

6 Conclusion

The need for fine-grained linguistic descriptions for parsing has become a reality with the development of more and more effective parsers. In this paper, we presented a method for interfacing a large-coverage syntactic dictionary with a grammar. We are currently using this method for the construction of a large-coverage unification grammar for French. It has been designed to deal with other languages studied within the lexicon-grammar framework. We think that it could be also adapted to other linguistic description frameworks such as the Proton (Eynde et al. 2001) or COMLEX Syntax (Grishman et al. 1994) projects.

References

(Abeillé 2002) Abeillé, Anne, 2002. *Une grammaire électronique du français*, CNRS Editions, Paris.

(Bresnan 1982) Bresnan, Joan, 1982, *The Mental Representation of grammatical relations*, MIT Press.

(Carroll et al. 1998) Carroll, John, Nicolas Nicolov, Olga Shaumyan, Martine Smets and David Weir, 1998, *LexSys Project*, Proceedings of the 4th International Workshop on Tree-adjoining Grammars and Related Frameworks (TAG+'98), Philadelphia, USA, pp.29-33

(Constant 2003) Constant, Matthieu, 2003, *Grammaires locales pour l'analyse automatique de textes : Méthodes de construction et outils de gestion*, Thèse de doctorat, Université de Marne la Vallée.

(Eynde et al. 2001) Eynde, Karel van den and Piet Mertens, 2001, *La syntaxe du verbe, l'approche pronominale et le lexique de valence PROTON* Preprint 174, Departement of Linguistics, K.U.Leuven

(Grishman et al. 1994) Grishman, Ralph, Catherine Macleod and Adam Meyers, 1994, *Comlex Syntax: building a computational lexicon*, Proceedings of the 15th conference on Computational linguistics, Kyoto, Japan, pp.268-272

(Gross 1975) Gross, Maurice, 1975, *Méthodes en syntaxe*, Hermann, Paris.

(Gross 1994) Gross Maurice, 1994, *Constructing Lexicon-grammars*, In Computational Approaches to the Lexicon, Atkins and Zampolli (eds.), Oxford Univ. Press, pp. 213-263

(Gross 1997) Gross Maurice, 1997, *The construction of Local Grammars*, in E. Roche and Y. Schabes Eds., *Finite State Language Processing*, Cambridge, Mass., MIT Press, pp. 329-352

(Maurel 1990) Maurel Denis, 1990, *Adverbes de date : étude préliminaire à leur traitement automatique*, Lingvisticae Investigationes, XIV:1, John Benjamins, pp 31-63

(Paumier 2003) Paumier Sébastien, 2003, *De la reconnaissance de formes linguistiques à l'analyse syntaxique*, Thèse de doctorat, Université de Marne-la-Vallée.

(Pollard et al. 1994) Pollard C. and I.A. Sag, 1994, *Head-Driven Phrase Structure Grammar*, University of Chicago Press and CSLI Publications.

(Roche 1993) Roche, Emmanuel, 1993, *Analyse syntaxique transformationnelle du français par transducteurs et lexique-grammaire*, Thèse de Doctorat, Paris, Université Paris 7.

(Schabes et al. 1988) Schabes, Yves, Anne Abeillé and Aravind K. Joshi, 1988, *Parsing strategies with 'lexicalized' grammars: Application to tree adjoining grammars*, In Proceedings of the 12 International Conference on Computational Linguistics (COLING'88), Budapest, Hungary, August 1988.

(Silberztein 1993) Silberztein, Max D., 1993, *Dictionnaires électroniques et analyse automatique de textes. Le système INTEX*, Paris, Masson, 234 p.

(Woods 1970) Woods, W.A., 1970, *Transition Network Grammars for Natural Language Analysis*, in Communications of the ACM, Vol 13:10