# A strategy for the syntactic parsing of corpora: from Constraint Grammar output to unification-based processing

# Toni Badia, Àngels Egea

Institut Universitari de Lingüística Aplicada Universitat Pompeu Fabra La Rambla, 30-32, 08002 Barcelona, Spain badia\_toni@trad.upf.es angels@slc.ub.es

#### Abstract

This paper presents a strategy for syntactic analysis based on the combination of two different parsing techniques: lexical syntactic tagging and phrase structure syntactic parsing. The basic proposal is to take advantage of the good results on lexical syntactic tagging to improve the whole performance of unification-based parsing. The syntactic functions attached to every word by the lexical syntactic tagging are used as head features in the unification-based grammar, and are the base for grammar rules.

# 1. Introduction

There currently are two main practices in the syntactic parsing techniques, that lead to two different results: lexical syntactic tagging (whose result is a flat analysis where syntactic functions and dependencies are indicated with lexically attached tags) and (partial) syntactic parsing (whose result is a tree, and where dependencies are built among constituents). So far, syntactic tagging and partial syntactic parsing (which builds a partial tree) have given some remarcable results when confronted with unrestricted text, in contrast to full syntactic parsing, whose results are not satisfatory enough because of two principal weaknesses to deal with unrestricted text: lack of robustness and lack of efficiency (the process is very time-consuming).

Nevertheless, we think that the performance of a full syntactic parsing system can be notably improved if part of the syntactic work has previously been done, thus following a strategy that is in accordance with recent parsing proposals, which prefer to split the syntactic analysis in several levels.

Our proposal, which is being implemented for Catalan, is based on the idea that every stage of analysis must be treated with the most robust and efficient strategy, and that the results obtained at each stage must be the input of the next one. In our environment, part-of-speech tagging is done with a two-level morphological analyser (Koskeniemi, 1984) written for Catalan (Badia & Tuells, 1997), which provides the possible lemmas and tags for every word. Partof-speech disambiguation, syntactic mapping (which provides every tagged word with all potential syntactic tags) and syntactic disambiguation (which eliminates wrong syntactic tags according to the context) are steps done with a Constraint Grammar (Karlsson et al., 1995) specially written for Catalan.

The last and more complex syntactic step is build into a unification-based grammar and implemented in ALEP (Advanced Language Engineering Platform) (Groenendijk, 1994; Simpkins, 1995)<sup>1</sup>. Our unification-based grammar preserves all the information that comes from the Constraint Grammar as values of head attributes and distributes it in the resulting feature structures. The amount of information present in the part-of-speech tags makes it possible to reduce to the minimum the lexicon (exactly to a single entry per tag) and this consequently eases the lexical lookup, which increases the overall efficiency of the system. On the other hand, the syntactic tags simplify the writting of phrase-structure rules as they make it possible to use the same rule: a) to build structures for different syntactic relations (verb-direct object, verb-indirect object, etc.), while keeping the syntactic function represented in the analysis tree, b) to avoide much of the lexical look-up, c) to simplify the identification of non-local dependencies, d) to leave some structural ambiguities unresolved with a portmanteau tag to be solved in later stages (such as semantic refinement), etc.

Let us finally mention that the particular architecture of the ALEP platform allows to foresee further parsing steps (so-called "refinement" steps) in which a real lexicon lookup is produced so that deep syntactic and lexical semantic information is available in the construction of the semantic representation.

### 2. Description of the implementation

This section focuses on the strategy that we followed to use the results of the Constraint Grammar as input of the unification based grammar implemented in ALEP. We will explain in some detail the process step by step, and some of the results we achieved. In section 2.1 we describe the information obtained from the previous processing (morphological analysis and syntactic tagging) and the format the text must take to be usable into ALEP. In section 2.2 we describe the typed feature structures the unification-based grammar is build on, which are the base for grammar development.

<sup>&</sup>lt;sup>1</sup>We have chosen the ALEP environment for grammar devel-

opment because it is specially designed to facilitate the linguist's job. But we believe that our proposal could be implemented in any system that makes it possible to develop head-driven phrase structure grammars.

In section 2.3 we explain the way the information on the input file is projected into a minimal feature structure according to the previous grammar description. In section 2.5 a description of the lexicon is done, where every lexical entries correspond to morphological tags, and we can see how the information contained into every tag is expanded into the feature structures.

Finally, in section 2.5 we can see the way this information is used for grammar developing and some of the results obtained.

#### 2.1. The input file of our unification parser

The file used as input for the ALEP grammar has undergone a previous linguistic processing which consists on these basic steps:

- morphological analysis, under a two-level morphological analyser for Catalan
- morphological disambiguation, under a Constraint Grammar (which is still under development)
- syntactic tagging, under a Constraint Grammar (still under development)

The resulting text file (the output of Constraint Grammar) must be automatically edited to provide a more suitable format to be used as input file for ALEP, where every type of information is to be included in SGML tags. The formal characteristics of the input file are illustrated in the following example, corresponding to the sentence *Uns gats corren sobre els terrats* ("Some cats are running over the roof"), where an SGML tag points to the beginning (<S>) and ending () of every sentence, and another SGML tag points to the beginning (</m>

<S> <w pos="amp", lemma="un", dn="yes">Uns</w> <w pos="n5mp", lemma="gat", subj="yes">gats</w> <w pos="vdr3p", lemma="córrer", fmv="yes"> corren</w> <w pos="p", lemma="sobre", advl="yes">sobre</w> <w pos="amp", lemma="el", dn="yes">els</w> <w pos="n5mp", lemma="terrat", cp="yes">terrats</w> .

As can be observed, the tag for word beginning has an internal structure consisting on three features: *pos* (whose value is a part of speech tag obtained from the morphological analysis), *lemma* (whose value is the canonical form of the token, obtained from the morphological analysis), and a third argument corresponding to the name of the syntactic tag (obtained from the Constraint Grammar syntactic tagging). The word token is placed between the beginning and ending word tags.

Part of speech tags, represented as values of feature *pos*, are a condensed way to express several morphosyntactic information. For example, *amp* stands for 'adjective masculine plural'; *n5mp* stands for 'common noun masculine singular'; *vdr3p* stands for 'finite verb indicative mood third person singular'; *p* stands for 'preposition'.

Syntactic tags are represented in different features, instead of using a single feature with different values, as we have done with the rest of the elements. This representation makes it possible to use a single phrase structure rule for several syntactic functions, as we will show later in section 2.5. The syntactic tags used in this explanation are: advl (adverbial complement), an (adjective modifying a noun to the right), atr (atribut), cd (direct object), ci (indirect object), cp (complement to a preposition), dn (determiner), ep (preposition introducing a nominal phrase), na (adjective modifying a noun to the left), subj (subject), fmv (finite main verb).<sup>2</sup>

#### 2.2. Grammar design: Type declarations

A unification-based grammar is a phrase structure grammar augmented with feature structures in the nodes. Feature structures are complex representations of linguistic signs that make possible to pass the information content from daughter nodes to mother nodes. The information content of every feature structure must be able to restrict the application of phrase structure rules to those cases where they can be applied felicitously. The basic operation on feature structures is unification. So feature structures must include as many information as is needed for a correct application of grammar rules.

The linguistic formalism supplied with ALEP (Alshawi, 1991) enables to design feature structures as type declarations expressed in the form of appropriateness conditions. Type declarations provide a mean to describe and restrict the content of valid feature structures. All possible features for every type must be declared, as well as all possible values for every feature. Before proceeding, we must say that our grammar design is roughly based on the HPSG proposal (Pollard and Sag, 1994).

For ease of explanation, we present a simplified version of our feature structures, where some features have been removed (like specifier features for grammar partition -see Simpkins, 1995- nonlocal and semantic features -see Pollard and Sag, 1994-, and some syntactic functions for relative pronouns and conjunctions introducing subordinate phrases).<sup>3</sup> Every sign (lexical or phrasal) is described by the features and values shown in the feature structure below, where the nature of the value is indicated inside the brackets.

The feature SYNSEM represents all relevant syntactic and semantic information. So far, we have only taken into account syntactic features, as semantics is left for future grammar development. Syntactic features are divided into sc head features (which contains morphosyntactic properties that a syntactic head daughter shares with its mother node), COMPL (which describes the syntactic properties of signs that can be complements of the head), MODIFIES (which describes the syntactic properties of a sign selected

<sup>&</sup>lt;sup>2</sup>The syntactic tags used in the Catalan Constraint Grammar are similar to the ones used in the English Constraint Grammar (Karlsson et al., 1995).

<sup>&</sup>lt;sup>3</sup>The notation for type declarations and rules used in this paper does not correspond to the ALEP notation, which is more cumbersome.

by an adjunct), SPEC (which describes the syntactic properties of a sign specified by a specifier, that is to say, by a minor category).<sup>4</sup>

The value of STRING is intended to take the orthographic form of the sign (lexical o phrasal). In the case of lexical (terminal) signs, it will take the content of the word tag (<w pos="n5pm" lemma="gat" subj="yes"> gats</w>). The value of REST is intended to take the rest of lexical elements of the sentence no yet analized.

The value of POS (part of speech) is intended to take the morphological tag provided for feature *pos* of the complex word tag (<w pos="**n5pm**" lemma="gat" subj="yes"> gats</w>). The value of LEMMA is intended to take the content of the feature lemma of the complex word tag (<w pos="n5mp" lemma="gat" subj="yes"> gats</w>).

ORTHO	STRING	<u>`</u>			
$\begin{bmatrix} REST & (list of atoms) \end{bmatrix}$					
MORPHO	POS (atom)				
	LEMMA (atom)				
	Γ	Γ		(yes,no)]]]	
SYNSEM	HEAD	SF		(yes,no)	
				(yes,no)	
			CD	(yes,no)	
			CI	(yes,no)	
			СР	(yes,no)	
			DN	(yes,no)	
			EP	(yes,no)	
			NA	(yes,no)	
			SUBJ	(yes,no)	
			FMV	(yes,no)	
		CAT (boolean)			
		TYPE (atom)			
	COMPL (list of synsem)				
	MODIFIES (synsem) SPECIFIES (synsem)				
L					

The feature HEAD is subtyped into different kinds of heads, as HEAD\_V (verbal head), HEAD\_ADJ (adjective head), HEAD\_NOUN (nominal head), HEAD\_PADV (prepositional and adverbial heads). A verbal head has the following

features (apart from the ones mentioned, which are common to every sign):

Γ	VTYPE
	VFORM
HEAD_V	VOICE
	TNS_MOOD
	AGREE

The feature VTYPE is intended to represent the verbal type (which can be pronominal or non pronominal). The feature VFORM is intended to represent the verbal form (which can have one of these values: finite, infinitive, gerund and participle). The feature VOICE can take one of these values: active or passive. The feature TNS\_MOOD is intended to represent the values for tense (present, past, future, etc.) and mood (indicative, subjunctive, imperative). The feature AGREE shows the agreement properties that a verb shares with its subject: person (p1, p2, p3), gender for past participle (f, m) and number (singular, plural).

A noun or adjective head has the feature INFL (inflection), which is equivalent to the verbal feature AGREE. All this information must be placed in the feature structure corresponding to every word if phrase structure rules are to be applied on them in order to combine terminal nodes and build a tree for every sentence. The specification of this information for every word constitutes the lexicon of our grammar, which contains all lexical entries needed to analyze a text.

#### 2.3. Lifting rules

Every sentence must undergo a lifting operation before applying phrase structure rules. Lifting is an ALEP action that takes a text chunk (corresponding to a sentence) in the appropriate format and converts it in a partial linguistic structure. *Partial* means that non immediate dominance between constituents is allowed. In other words, lifting provides minimal feature structures for every word in the sentence and a mother feature structure which dominates every word.

Lifting applies lift rules, which specify the way the elements of a text structure (as the one showed in section 2.1.) must be distributed into a partial linguistic structure, according to the type declarations. Lifting rules are called ts\_ls\_rules in the ALEP formalism:

ts\_ls\_rule=> 'w',[pos=>POS, lemma=>LEMMA, subj=>SUBJ],A).

Capital letters are variables, and they are used to place every piece of information as a value of the corresponding feature in the lift structure. Remind that we use a different feature for every syntactic tag. So we need as many lift rules as syntactic tags. The lift rule above builds the feature structure for every word in the text input that corresponds to the structure showed in the line below the structure. In that case, the rule lifts only words with a SUBJ feature in the word tag, as in the following example:

<sup>&</sup>lt;sup>4</sup>For a better understanding of these features, see Pollard & Sag, 1994.

### <w pos="n5pm" lemma="gat" subj="yes">gats</w>

In this way the value of the feature subj in the text word structure is placed as a value of the feature SUBJ in the lifted structure, as the identification of variables shows. The rest of syntactic features in the lifted structure are assigned a value *no*.

There is the possibility that Constraint Grammar output be not completely disambiguated for every word. In these cases a word can have two different syntactic tags attached. If the syntactic ambiguity have no structural consequences for tree-building (like, for exemple, adverbial complement and indirect complement, which both are verbal complements), we replace the two tags for a port-manteau tag and we introduce a new lifting rule where the values for both syntactic tags are set to *yes*. With this strategy we can proceed with the full syntactic analysis while preserving the ambiguity information into the analysis tree.

If the syntactic ambiguity have structural consequences (like adverbial complement and noun complement; that is the well known problem of PP-attachment), we do not create any port-manteau tag, so two different lifted structures are automatically generated that will give rise to two different trees.

Once the whole sentence is lifted, a partial linguistic structure is obtained, and lexical rules and phrase structure rules can be applied to it. The application of lexical rules (lexical entries) will complete some other values of these feature structures according to their morphosyntactic tag. The conversion of a partial linguistic structure into a linguistic structure, where immediate dominance is showed, is a matter of phrase structure rules.

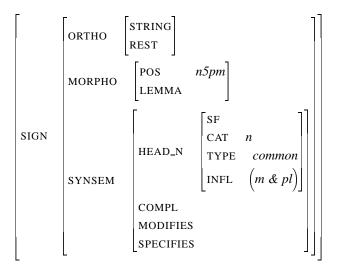
#### 2.4. Lexical entries

As we do not want to restrict the application of this grammar to any kind of text, the lexicon must contain all words needed to analyze unrestricted text. In most of the language applications of large coverage, lexicons must contain several thousands of lexical entries -possibly, a lot of thousands of entries- in order to be able to give account of the biggest part of the words that appear in real texts.

In our unification-based grammar the same effect can be achieved with about a hundred entries. This dramatic reduction in the lexicon size is possible if we use morphosyntactic tags as lexical entries instead of using word tokens or word types (we must remind that input text has already undergone morphological analysis and morphological disambiguation, so that a morphosyntactic tag and lemma is provided for every text word).

The side effect of the lexicon size reduction is a more efficient look-up during processing, and a more easy maintenance if entries have to be modified in some way.

The information contained in a morphosyntactic tag can be expanded in the following way, where every piece of information is placed as a value of the corresponding feature:



In order to express the information for every entry in a more compact and readable format, we use macros, one of the facilities of the ALEP formalism (Simpkins, 1995):

n5mp m\_sign[m\_noun[ n, com, (m&pl)], 'n5mp'].

where *n5mp* is the entry and the macro m\_sign stands as an abbreviation for the large feature structure above. The macro m\_sign has two arguments, that must be placed as values of the following features:

> macro:m\_sign[A,B], SIGN [MORPHO [POS B]] SYNSEM A ]]

The first argument is another macro, m\_noun, which stands for:

macro:m\_noun[CAT,NOUNTYPE,INFL]  

$$\begin{bmatrix} CAT & CAT \\ HEAD_N & TYPE & NOUNTYPE \\ INFL & INFL \end{bmatrix}$$

Macros are placed in a macro file and are expanded (they generate the structures they represent) at compile time. We need as many macros as lexical entries, that is to say, as many macros as morphosyntactic tags.

#### 2.5. Phrase structure rules

According to the ALEP formalism (Alshawi, 1991), phrase structure rules are expressed as a mother linguistic description followed by a list of daughter linguistic descriptions, that is: a feature structure describing the mother node and a list of feature structures describing the daughter nodes.

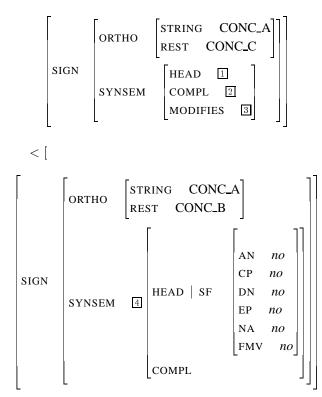
The aim of phrase structure rules is to show the hierarchical structure between constituents, and the flow of information between nodes, expressed in feature structures. Before proceeding, some remarks must be done in order to understand the phrase structure rules we present: a) Catalan is a pro-drop language, which means that subject can be omitted and only verbal inflection shows some of its morphosyntactic properties (person and number); b) subject and verbal complements show a quite free order in Catalan; both can be placed before or after the verb.

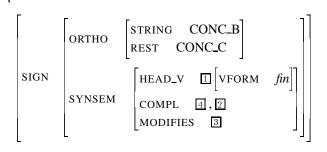
In our grammar, subject and verbal complements are treated as the same thing, with the only difference of the SF feature content. Thus we avoid having to cope with empty subjects.

One only rule is needed to combine a verbal head daughter with its subject or its complements (direct complement, indirect complement, adverbial complement). Syntactic features are used to restrict the application of this rule to the correct cases.

In the following example, every feature structure contains one piece of phrase structure rule: the first structure contains the mother node (the node that is created as a result of the combination of the daughters, which happens if constraints on the daughters are fullfiled). Capital letters and indexes show structure sharings.

The second and third structures correspond to the ordered daughters. The whole rule combines a verbal head daughter with a nonhead daughter whose SUBJ value must be subject (subj) or verbal complement (direct complement, indirect complement, adverbial complement), if the subject or verbal complement are placed before the verb (as <[] indicates). Note that capital letters and indexes express structure sharings, so that the values with the same variable must unify for rule application be successful. Note that the index 1 of the head mother is structure-shared with the head value of the head daughter (the second daughter in the rule), which has to be a verbal head with a value *fin* (finite verb) for VFORM feature. All these properties are passed to the head mother. This ensures the application of the Head Feature Principle (Pollard and Sag, 1994).





As there could be more than one complement, the index 2 of the head mother, which is the value of the COMPL feature, is structure-shared with the value of the head daughter, which is the tail value of the list in the COMPL feature. This makes it possible for a head daugther to be combined with as many complements as needed. Notice that, as we have no valence information in the lexicon, we cannot know in advance the complements a verb can combine with. But instead we know the complements a verb have in a particular sentence, and this information is used to build the analysis tree.

The index 4 of the head daughter, which corresponds to the complement that is to be combined with the head daughter, is structure-shared with the SYNSEM value of the nonhead daughter. The structure sharing happens if the nonhead daughter has value *no* in every of the syntactic features shown, while the value can be yes for the ones that are not mentioned: SUBJ, CD, CI, ADVL.

We need another rule with the same structure except that the head daughter has to be placed before the nonhead daughter. With both rules we can combine any main verb with its complements and subject no matter if they appear before or after the verb.

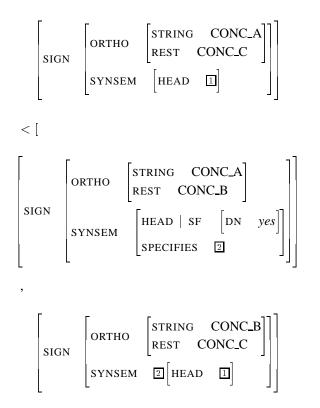
In order to avoid the generation of more than one structure when combining complements, we introduced one more feature to force a right to left application. We called this feature BAR and it can have three possible values:

- 0 if the verbal head has not been combined with any complement
- 1 if the verbal head has been combined with a complement (or subject) to the right
- 2 if verbal head has been combined with a complement to the left.

Complements on the right can only combine with verbal heads with BAR value 0 or 1, and the resulting verbal head takes a BAR value 1. Complements on the left can only combine with verbal heads with BAR value 0 or 1, and the resulting verbal head takes a BAR value 2. In this way, the feature BAR prevents the cancellation of left complements before the cancellation of right complements.

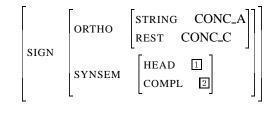
In order to analize our sentence example, we need some more rules: a rule that enables to combine a determinant with the noun it introduces (*uns gats*), and another rule that enables to combine a preposition with its nominal phrase (*sobre els terrats*). Otherwise, the whole analysis fails.

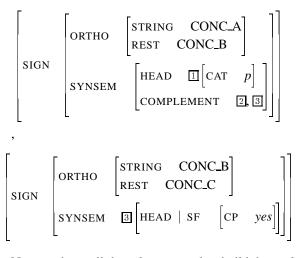
A determinant introducing a noun has always the syntactic tag DN. Determinants are treated as specifiers of their heads, according to HPSG. Specifiers are minor categories that specifies their head through SPECIFIES feature; that is to say: they select their heads. This is expressed in the rule below.



The first structure corresponds to the mother node, whose head is structure shared with the third structure, corresponding to the head daughter. The SYNSEM value of the head daughter is structure-shared with the nonhead daughter SPECIFIES value, which is a special feature for minor categories to select their heads. The nonhead daughter must have the DN value (syntactic tag for determinat) set to *yes*. This rule enables to combine the nominal phrases *uns gats* (some cats) and *els terrats* (the roofs) of our sentence example.

Now we only need another rule to combine the adverbial preposition sobre (over) with the nominal phrase it introduces. In our syntactic tagging, prepositions introducing an adverbial phrase are treated as heads, so they receive the tag corresponding to the syntactic function of the whole prepositional phrase (ADVL in our example). Nouns into a prepositional phrase are considered complements of the preposition (CP). This view is in accordance with the HPSG treatment of prepositions, where nominal phrases are complements of prepositions.





Now we have all the rules we need to build the analysis tree for the sentence *Uns gats corren sobre els terrats*. We can see the whole representation in the tree below. It must be noticed the importance of syntactic tags as head features: this ensures that the syntactic tag of the head daughter will allways be passed to the head mother, so that head feature principle holds and application of further rules on a constituent is possible because the syntactic tag of the head is preserved. Thus, the syntactic tag of the head mother of the whole sentence is FMV.

As a result of the unification operation, the terminal nodes of the tree contain the same information as their corresponding projections. This is specially important for verbs valence, which can be known from the analysis of a significant group of sentences. In our example, the verb *córrer* (run) in the bottom of the tree is structure shared with its projection of the top. This is a very important effect of unification-based grammars. Other lexical nodes, as nouns or prepositions, can be worthwhile to study from this point of view.

## 3. Preliminary results

While it is still soon for a global valoration of the results that can be obtained with the combination of syntactic tagging and unification-based processing, as both modules are still under development, we want to point out some aspects that make this approach attractive.

First of all, we have taken into account that the Constraint Grammar output cannot be completely disambiguated, either because of the limitations of the formalism or because some of the information that is needed appears outside the scope of the sentence being analysed. In some cases, these ambiguities have structural consequences, but in some other ones a single structure corresponds to a set of syntactic tags. The following sentences show the two possibilities:

- (1) a. Han enviat una carta a la direcció have sent a letter to the management
  - b. Han arribat alguns turistes del Japó have arrived some tourists from the Japan

In the first example the ambiguity is placed in the verbal complement *a la direcció*, which keeps two syntactic tags

< [

correspondig to adverbial complement and to indirect object, as the preposition *a* can introduce any of these complements. If semantic information is added to the lexical elements then it might be possible to do a safe disambiguation and chose the adverbial complement as the good reading. But at this stage semantic information is not yet available, so we have to carry on with this kind of ambiguities until later stages. This ambiguity is not structural, as in both cases the constituent depends on the verb.

In the second example however the ambiguity is placed in *del Japó*, which can be either a verbal or a nominal complement. Here the preposition *de* can introduce either nominal phrases that are complements of a noun or nominal phrases that are verbal complements. As Catalan shows a quite free order of its constituents, we cannot know by sure if the tourists of the example arrive from Japan (but they are not probably Japanese) or are Japanese (even though they arrive from any other place). In this example, the ambiguity is structural, as the constituent can depend on the verb *arrive* or on the noun *tourists*. Some of these ambiguities might be resolved with more lexical information; other (like this particular one) are true ambiguities that can only be solved if contextual information is taken into account.

However the existence of these ambiguities has no fatal effects on the performance of the unification-based parser. As we have noted earlier, if the ambiguity is not structural, a port-manteau tag is created and analysis can proceed normaly while keeping the ambiguity to be resolved if it is necessary in further steps of language processing. In the first example above, we create the port-manteau tag ADVL\_CI instead of the two different tags ADVL and CI.

On the other hand, if the ambiguity is structural, the parser builds two different trees corresponding to the different syntactic readings. In the second example above, we cannot merge the two tags into a single one, as analysis would fail because of conflicting information. In these cases both tags are left and two tree structures are built by the parser.

Thus, we can say that the unification-based parser performs as a robust system when attached to a syntactic tagger. Note that it will even be able to reduce some of the ambiguities resulting of the syntactic tagging when rich syntactic or semantic information is supplied.

The simplicity on the grammar writing is another aspect that must be mentioned as a consequence of the strategy presented in this paper. As mentioned above, we do not need much of the lexical information required in the theoretical proposals (like HPSG). Canonical form, morphological tag and syntagtic tag is all what we need to build an analysis with such an unification grammar. Recall that we only need a lexical entry for every morphological tag in order to disseminate the information throughout the feature structure.

It is worth noting that, as a result of a basic processing (the basic syntactic analysis we have explained), we can obtain lexical information on subcategorization that can be used for further grammar development. This information could be used to disambiguate unresolved ambiguities.

Similarly, long distance dependencies can be treated with our approach. Of course, no verbal complement to be cancelled is threaded through the structure as a slash feature (Pollard & Sag, 1994) until an element that feeds the slash is found. We do not know the verbal valence in advance. But we have the relative pronouns marked with a syntactic and morphological tags that indicate their relative nature and their syntactic function(s). Consequently, we have written rules that combine these elements in a similar way to other grammar rules: they take care of the order differences and they fill the value of the nonlocal attribute.

There still possibly are more advantages in this approach that we have not yet discovered. And probably also some problems will appear in the future development of our grammar. But we believe that the start is promising and worthwhile exploring.

### 4. Conclusions

In this paper we have shown some possibilities that the combination of different strategies in syntactic analysis offers. From our point of view, syntactic tagging and unification-based syntactic parsing can be seen as two different steps, where the robustness of the former remedies some of the deficiencies of the second. On the other hand, the unification-based approach is richer in the linguistic description, it builds the dependencies between constituents, and provides a way to treat other levels of linguistic analysis that we have not yet developed, as semantics or even pragmatics.

### 5. References

Alshawi et al., 1991. Eurotra ET6/1: Rule Formalism and Virtual Machine Design Study. Final Report. SRI International.

Badia, T. & Tuells, T., 1997. CATMORF: Multi-two level steps for Catalan morphology. In *Proceedings of the Conference on Applied Natural Language Processing*. Washington, 1997.

Groenendijk, 1994. *Environment Tools Guide. ALEP-2.* European Commission.

Karlsson et al., 1995. (eds.) *Constraint Grammar, a language-independent system for parsing unrestricted text.* Berlin and New York: Mouton de Gruyter.

Koskenniemi, 1984. A General Computational Model for Word-form Recognition and Production. In *Proceedings of the 10th International Conference on Computational Linguistics (COLING-84)*, Stanford.

Pollard, Carl; Sag, Ivan A. 1994. *Head-Driven Phrase Structure Grammar*. Chicago & London: University of Chicago Press. (Studies in Contemporary Linguistics).

Simpkins, 1995. *Linguistic Development and Processing. ALEP-2*. European Commission.