Integration of syntactic and lexical information in a hierarchical dependency grammar

Cristina Barbero and Leonardo Lesmo and Vincenzo Lombardo
Dipartimento di Informatica
Università di Torino - Italy

Paola Merlo
Université de Genève - Switzerland
IRCS - University of Pennsylvania

Abstract

In this paper, we propose to introduce syntactic classes in a lexicalized dependency formalism. Subcategories of words are organized hierarchically from a general, abstract level (syntactic categories) to a word-specific level (single lexical items). The formalism is parsimonious, and useful for processing. We also sketch a parsing model that uses the hierarchical mixed-grain representation to make predictions on the structure of the input.

1 Introduction

Much recent work in linguistics and computational linguistics emphasizes the role of lexical information in syntactic representation and processing.

This emphasis given to the lexicon is the result of a gradual process. The original trend in linguistics has been to individuate categories of words having related characteristics – the traditional syntactic categories like verb, noun, adjective, etc. – and to express the structure of a sentence in terms of constituents, or phrases, built around these categories. Subsequent considerations lead to a lexicalization of grammar. Linguistically, the constraints expressed on syntactic categories are too general to explain facts about words – e.g. the relation between a verb and its nominalization, "*destroy the city*" and "*destruction of the city*" – or to account uniformly for a number of phenomena across languages – e.g. passivization. In parsing, the use of individual item information reduces the search space of the possible structures of a sentence. From a mathematical point of view, lexicalized grammars exhibit properties – like finite ambiguity (Schabes, 1990) – that are of a practical interest (especially in writing realistic grammars). Dependency grammar is naturally suitable for a lexicalization, as the binary relations representing the structure of a sentence are defined with respect to the head (that is a word).

Pure lexicalized formalisms, however, have also several disadvantages. Linguistically, the abstract level provided by syntactic rules is necessary to avoid the loss of generalization which would arise if class-level information were repeated in all lexical items. In parsing, a predictive component is required to guarantee the *valid prefix property*, namely the capability of detecting as soon as possible whether a substring is a valid prefix for the language defined by the grammar. Knowledge of syntactic categories, which does not depend on the input, is needed for a parser to be predictive.

In this paper we address the problem of the interaction between syntactic and lexical information in dependency grammar. We introduce many intermediate levels between lexical items and syntactic categories, by organizing the grammar around the notion of *subcategorization*. Intuitively, a subcategorization frame for a lexical item L is a specification of the number and type of elements that L requires in order, for an utterance that contains L, to be well-formed. For example, within the syntactic category VERB, different verbs require different numbers of nominal dependents for a well-formed sentence. In Italian (our case study), an intransitive verb such as *dormire*, "sleep", subcategorizes for only one nominal element (the subject), while a transitive verb such as *baciare*, "kiss", subcategorizes for two nominal elements (the subject and the object) [1]. Grammatical relations such as subject and object are primitive concepts in a dependency paradigm, i.e. they directly define the structure of the sentence. Consequently, the dependency paradigm is particularly suitable to define the grammar in terms of constraints on subcategorization frames.

Our proposal is to use subcategories organized in a hierarchy: the upper level of the hierarchy corresponds to the syntactic categories, the other levels correspond to subcategories that are more and more

[1] We include the subject relation in the subcategorization, or valency, of a verb – cf. (Hudson, 1990) (Mel'cuk, 1988). In most constituency theories, on the contrary, the subject is not part of the valency of a verb.

specific as one descends the hierarchy. This representation is advantageous because of its compactness, and because the hierarchical mixed-grained organization of the information is useful in processing. In fact, using the general knowledge at the upper level of the hierarchy, we can make predictions on the structure of the sentence before encountering the lexical head.

Hierarchical formalisms have been proposed in some theories. Pollard and Sag (1987) suggested a hierarchical organization of lexical information: as far as subcategorization is concerned, they introduced a "hierarchy of lexical types". A specific formalisation of this hierarchy has never reached a wide consensus in the HPSG community, but several proposals have been developed – see for example (Meurers, 1997), that uses head subtypes and lexical principles to express generalizations on the valency properties of words.

Hudson (1990) adopts a dependency approach and uses hierarchies to organize different kinds of linguistic information, for instance a hierarchy including word classes and lexical items. The subcategorization constraints, however, are specified for each lexical item (for instance STAND → STAND-intrans, STAND-trans): this is highly redundant and misses important generalizations.

In LTAG (Joshi and Schabes, 1996), pure syntactic information is grouped around shared subcategorization constraints (tree families). Hierarchical representations of LTAG have been proposed: (Vijay-Shanker and Schabes, 1992), (Becker, 1993), (Evans et al., 1995), (Candito, 1996), (Doran et al., 1997). However, none of these works proposes to use the hierarchical representation in processing – just Vijay-Shanker and Schabes (1992) mention, as a possible future investigation, the definition of parsing strategies that take advantage of the hierarchical representation.

The goal of our hierarchical formalism is twofold. On one side, we want to provide a hierarchical organization to a lexicalized dependency formalism: similarly to the hierarchical representations of LTAG, the aim is to solve the problems of redundancy and lexicon maintenance of pure lexicalized approaches. On the other side, we want to explore how a hierarchical formalism can be used in processing in order to get the maximum benefit from it.

The paper is organized as follows: in section 2 we describe a lexicalized dependency formalism that is a simplified version of (Lombardo and Lesmo, 1998). Starting from this formalism, we define in section 3 the hierarchy of subcategories. In section 4, we sketch a parsing model that uses the hierarchical grammar. In section 5, we describe an application of the formalism to the classification of 101 Italian verbs. Section 6 concludes the paper.

2 A dependency formalism

The basic idea of dependency is that the syntactic structure of a sentence is described in terms of binary relations (*dependency relations*) on pairs of words, a *head* (or parent), and a *dependent* (daughter), respectively; these relations form a tree, the *dependency tree*. In this section we introduce a formal dependency system, which expresses the syntactic knowledge through dependency rules. The grammar and the lexicon coincide, since the rules are lexicalized: the head of the rule is a word of a certain category, namely the lexical anchor. The formalism is a simplified version of (Lombardo and Lesmo, 1998); we have left out the treatment of long-distance dependencies to focus on the subcategorization knowledge, which is to be represented in a hierarchy.

A *dependency grammar* is a five-tuple $<W, C, S, D, H>$, where
W is a finite set of words of a natural language;
C is a finite set of syntactic categories;
S is a non-empty set of categories ($S \subseteq C$) that can act as head of a sentence;
D is the set of *dependency relations*, for instance SUBJ, OBJ, XCOMP, P-OBJ, PRED;
H is a set of dependency rules of the form
$x{:}X\ (<r_1Y_1> \ldots <r_{i-1}Y_{i-1}>\ \#\ <r_{i+1}Y_{i+1}> \ldots <r_mY_m>)$
1) $x \in W$, is the *head* of the rule;
2) $X \in C$, is its syntactic category;
3) an element $<r_jY_j>$ is a *d-pair* (which describes a dependent); the sequence of d-pairs, including the special symbol # (representing the linear position of the head), is called the *d-pair sequence*. We have that
3a) $r_j \in D$, $j \in \{1, \ldots, i-1, i+1, \ldots, m\}$;
3b) $Y_j \in C$, $j \in \{1, \ldots, i-1, i+1, \ldots, m\}$;

Intuitively, a dependency rule constrains one node (head) and its dependents in a dependency tree: the d-pair sequence states the order of elements, both the head (# position) and the dependents (d-pairs). The grammar is lexicalized, because each dependency rule has a lexical anchor in its head ($x{:}X$). A d-pair $<r_iY_i>$ identifies a dependent of category Y_i, connected with the head via a dependency relation r_i.
As an example, consider the grammar [2]:

$G = <$
W : {gli, un, amici, eroe, lo, credevano}

[2] We use Italian terms to label grammatical relations – see table 1. Since subcategorization frames are language-dependent, we prefer to avoid confusions due to different terminology across languages. For example, the relation *Termine* – see the caption of figure 4 – actually corresponds to the indirect object in English. However *I-Obj* undergoes the double accusative transformation into *Obj*, while *Termine* does not.

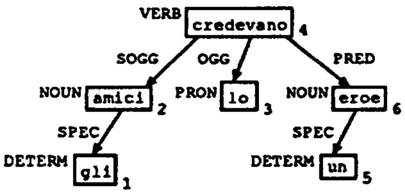

Figure 1: Dependency tree of the sentence *Gli amici lo credevano un eroe*, "The friends considered him a hero", given the grammar G. The word order is indicated by the numbers $1, 2, \ldots$ associated with the nodes – *amici*, "friend", is a left dependent of the head, as it precedes the head in the linear order of the input string, *eroe*, "hero", is a right dependent.

C : {Verb, Noun, Determ}
S : {Verb}
D : {Sogg, Ogg, Pred, Spec}
H >,

where H includes the following dependency rules:

1. gli: Determ (#);
2. un: Determ (#);
3. amici: Noun (<Spec Determ> #);
4. eroe: Noun (<Spec Determ> #);
5. lo: Pron (#);
6. credevano: Verb (<Sogg Noun> <Ogg Pron> # <Pred Noun>);

By applying the rules of the grammar, we obtain the dependency tree in figure 1 for the sentence *Gli amici lo credevano un eroe*, "The friends considered him a hero".

3 A hierarchy of subcategories

The formalization of dependency grammar illustrated above, like all lexicalizations, suffers from the problem of redundancy of the syntactic knowledge. In fact, for each $w \in W$, a different rule for each configuration of the dependents for which w can act as a head must be included in the lexicon. Some tool is required to represent lexical information in a compact and perspicuous way. We propose to remedy the problem of redundancy by using a hierarchy of subcategorization frames.

3.1 A basic hierarchy

The description of the dependency rules is given on the basis of a hierarchy of subcategories, each of which has a subcategorization frame associated [3]. Each subcategorization frame is, in turn, a compact representation of a set of dependency rules. The formal definition of the hierarchy is the following.

A *subcategorization hierarchy* is a 6-tuple $<T, L, D, Q, F, \leq_T>$, where:

T is a finite set of *subcategories*;

L is a mapping between W (the words, defined in the grammar) and sets of subcategories, $L : W \rightarrow 2^T - \{\}$. That is, each word can "belong" to one or more subcategories;

D is a set of *dependency relations* (as in section 2);

Q is a set of *subcategorization frames*. Each subcategorization frame is a total mapping $q : D \rightarrow R \times 2^T$, where R is the set of pairs of natural numbers $<n_1, n_2>$ such that $n_1 \geq 0, n_2 \geq 0$ and $n_1 \leq n_2$;

F is a bijection between subcategories and subcategorization frames, $F : T \rightarrow Q$;

$\leq_T$ is an ordering relation among subcategories.

In order to define $\leq_T$, we need some notation:

$N_q(d)$, where $q \in Q$ and $d \in D$, is the first element of $q(d)$, i.e. the *number restrictions* associated with the relation d in the subcategorization frame q.

$V_q(d)$, where $q \in Q$ and $d \in D$, is the second element of $q(d)$, i.e. the *value restrictions* associated with the relation d in the subcategorization frame q.

Intuitively, $N_q(d)$ is the number of times the dependency relation d can be instantiated according to the subcategorization frame q; $V_q(d)$ is the set of subcategories that can be in relation d with a subcategory having q as a subcategorization frame.

Let $\leq_{R_N}$ be an order relation of number restrictions; given two pairs of natural numbers R_1 and R_2,

$$R_1 \leq_{R_N} R_2 \textit{ iff } min(R_1) \geq min(R_2) \wedge max(R_1) \leq max(R_2)$$

namely, the range R_1 is inside the range R_2.

Let $\leq_{R_V}$ be an order relation of value restrictions; given two sets of subcategories V_1 and V_2,

$$V_1 \leq_{R_V} V_2 \textit{ iff } V_1 \subseteq V_2$$

Now, we can say that, for each $t_1, t_2 \in T$:

$$t_1 \leq_T t_2 \textit{ iff } \forall d \in D \; (N_{F(t_1)}(d) \leq_{R_N} N_{F(t_2)}(d) \wedge (V_{F(t_1)}(d) \leq_{R_V} V_{F(t_2)}(d))$$

The relation $\leq_T$ is a partial order on T. If we assume the existence of a most general element TOP, it can act as the root of a hierarchy defined on $\leq_T$. In the definitions above, each subcategory in the hierarchy defined by $\leq_T$ is associated, through F, with a *subcategorization frame*. So, through L and F, each word in the lexicon is associated with one or more subcategorization frames. Actually, lexical ambiguity is due to L since F is a bijection.

In the rest of this section we show that each subcategorization frame q defines a set of *dependency rules*, in the sense used in section 2 for the formal definition of the grammar. In this way, we get that the hierarchy specifies a correspondence between words and rules. Moreover, we show that the hierarchy acts as a taxonomy: given that $rules(t_1) \subseteq H$ is the set of dependency rules whose head is the syntactic category t_1, we have that

[3] In this paper we focus our attention to verbal subcategorization frames.

$$\begin{array}{l} \forall t_1, t_2 \in T \; \forall dr \in H \\ \quad (t_1 \leq_T t_2 \wedge dr \in rules(t_1) \rightarrow dr \in rules(t_2)) \end{array}$$

In order to specify the correspondence between subcategorization frames and dependency rules, we first define

$$\begin{array}{l} Dep_q(d) = \{m \mid m = [< d,t > \mid t \in V_q(d)] \wedge \\ \qquad\qquad minN_q(d) \leq Card(m) \leq maxN_q(d)\} \end{array}$$

Given a subcategorization frame q and a relation d, $Dep_q(d)$ is the set of all multisets of pairs $< d,t >$, where t is a subcategory $\in V_q(d)$. The multisets come from the fact that the same relation can be instantiated many times (depending on the range). In order to compute the sets of dependency relations that the subcategorization frame includes, we form the cartesian product of the various $Dep_q(d)$:

$$Cart_q = \prod_{d \in D} Dep_q(d)$$

and we evaluate the union of each member of $Cart_q$; each of them is extended by including the special symbol $\#$:

$$DepSet_q = \{m \mid m = (\cup_{s \in S, S \in Cart_q} s) \cup \{\#\}\}$$

where the union is a multiset union, preserving duplications. Finally, by picking all the permutations of each member of $DepSet_q$, we get the set of rules (also called *subcategorization patterns*):

$$Rules_q = \{r \mid r \in Permute(m) \wedge m \in DepSet_q\}$$

An example should make clear how the above definitions work. Let's assume that

$$\begin{array}{l} D = \{sogg, ogg, compl\} \\ q = \{<sogg,<<1,1>,\{N\}>>, \\ \quad <ogg,<<0,1>,\{N,C\}>>, \\ \quad <compl,<<0,2>,\{P\}>>\} \end{array}$$

(where C is short for CHESUB - subordinating conjunction - and P for PREP).
Then we have:

$$\begin{array}{l} Dep_q(sogg) = \{\{<sogg,N>\}\} \\ Dep_q(ogg) = \{\{\},\{<ogg,N>\},\{<ogg,C>\}\} \\ Dep_q(compl) = \{\{\},\ \{<compl,P>\}, \\ \qquad\qquad\qquad \{<compl,P>,<compl,P>\}\} \end{array}$$

$$\begin{array}{l} Cart_q = \\ \{\ <\{<sogg,N>\},\{\},\{\}>, \\ \quad <\{<sogg,N>\},\{\},\{<compl,P>\}>, \\ \quad <\{<sogg,N>\},\{\},\{<compl,P>,<compl,P>\}>, \\ \quad <\{<sogg,N>\},\{<ogg,N>\},\{\}>, \\ \quad <\{<sogg,N>\},\{<ogg,N>\},\{<compl,P>\}>, \\ \quad <\{<sogg,N>\},\{<ogg,N>\},\{\ <compl,P>, \\ \qquad\qquad\qquad\qquad\qquad\qquad <compl,P>\}>, \\ \quad <\{<sogg,N>\},\{<ogg,C>\},\{\}>, \\ \quad <\{<sogg,N>\},\{<ogg,C>\},\{<compl,P>\}>, \\ \quad <\{<sogg,N>\},\{<ogg,C>\},\{<compl,P>, \\ \qquad\qquad\qquad\qquad\qquad\qquad <compl,P>\}>\} \end{array}$$

$$\begin{array}{l} DepSet_q = \\ \{\ \{<sogg,N>,\#\}, \\ \quad \{<sogg,N>,<compl,P>,\#\}, \\ \quad \{<sogg,N>,<compl,P>,<compl,P>,\#\}, \\ \quad \{<sogg,N>,<ogg,N>,\#\}, \\ \quad \{<sogg,N>,<ogg,N>,<compl,P>,\#\}, \\ \quad \{<sogg,N>,<ogg,N>,<compl,P>,\ <compl,P>,\ \#\}, \\ \quad \{<sogg,N>,<ogg,C>,\#\}, \\ \quad \{<sogg,N>,<ogg,C>,<compl,P>,\#\}, \\ \quad \{<sogg,N>,<ogg,C>,<compl,P>,<compl,P>,\ \#\} \end{array}$$

If we take all the permutations of the various subsets, we finally obtain the rules. So that if we have

$$\begin{array}{l} L(\text{“}to\ sprong\text{”}) = \{t_{137}\} \\ F(t_{137}) = q \end{array}$$

we obtain dependency rules of the form in the previous section:

$$\begin{array}{l} to\ sprong : t_{137}(<sogg,N>\ \#) \\ to\ sprong : t_{137}(\#\ <sogg,N>) \\ to\ sprong : t_{137}(<sogg,N><compl,PREP>\ \#) \\ to\ sprong : t_{137}(<sogg,N>\ \#\ <compl,PREP>) \\ \ldots \end{array}$$

This procedure has the goal of mapping the subcategorization frames onto the dependency rules. In the actual practice, the frames are not multiplied out before processing (for instance, exactly 200 rules would be generated for our very simple example). Processing issues will be sketched in section 4.

3.2 Ordering among dependents

The hierarchy, and in particular the subcategorization frames, does not enforce a specific ordering among dependents of the same head. We propose an extension of the formalism that prevents some permutations of the rules from being generated. The definition of subcategorization frame is modified in the following way:

Q is a set of ordered *subcategorization frames*. Each of them is a pair consisting of a subcategorization frame and a set of *ordering constraints*.
$\forall q \in Q\ [q :<<D \rightarrow R \times 2^T> \times 2^O>]$, where R is as before and O is a set of pairs $<d_1,d_2>$ where $d_1,d_2 \in D \cup \{\#\}$.

The pairs in O define a partial order on the relative positions of the dependency relations and the head. If both d_1 and d_2 are members of D, the constraint specifies that the dependent whose grammatical relation is d_1 (if any) must precede linearly the dependent whose grammatical relation is d_2 (if any). If the first (second) member of the constraint is $\#$, it is specified that the dependent whose grammatical relation is d_2 (d_1 respectively), if any, must follow (precede) the head. The "if any" clauses say that in all cases where one of the two elements is optionally present (minimum of the range equal to

0), the constraint is assumed to be respected in case the number of actual instantiations is 0.
The ordering relation is transitive, namely:

if $<e_1, e_2> \in O_x \wedge <e_2, e_3> \in O_x$ *then*
$<e_1, e_3> \in O_x$

We require that the set of ordering constraints O_x associated with any subcategorization frame be consistent:

a) for all $e_i \in D \cup \{\#\}$, $<e_i, e_i> \notin O_x$
b) for all $e_i, e_j \in D \cup \{\#\}$, *if* $<e_i, e_j> \in O_x$
then $<e_j, e_i> \notin O_x$

Finally, we modify the $\leq_T$ relation (which defines the hierarchy):

for each $t_1, t_2 \in T$:

$$
\begin{aligned}
& t_1 \leq_T t_2 \textit{ iff} \\
& \quad (O_{F(t_1)} \supseteq O_{F(t_2)}) \wedge \\
& \quad \forall d \in D \\
& \qquad (N_{F(t_1)}(d) \leq_{R_N} N_{F(t_2)}(d) \wedge \\
& \qquad V_{F(t_1)}(d) \leq_{R_V} V_{F(t_2)}(d))
\end{aligned}
$$

This corresponds to the requirement that a subcategory t_1, which is more specific than t_2, does not have looser constraints on linear order than t_2 has. If we refer to our previous example, a possible O_q is $\{<sogg, \#>, <\#, ogg>\}$, specifying that the subject must precede the verbal head, which, in turn, must precede the direct object. If each permutation in $Rules_q$ is checked to verify if it satisfies the constraints, then only 40 rules are left, corresponding to the possible (free) positions of the (0 to 2) complements.

3.3 Inheritance

We briefly mention here a notational convention which is useful to simplify the description of the subcategorization frames; this convention is widespread in almost all taxonomic hierarchies. For details about inheritance we remind to the extensive literature on semantic networks, frames and description logics (Nebel, 1990).

We define:

$t_1 <_T t_2$ *iff* $t_1 \leq_T t_2 \wedge \neg(t_2 \leq_T t_1)$

If we define in the same way $<_{R_N}$ and $<_{R_V}$, it is easy to verify that:

$$
\begin{aligned}
& t_1 <_T t_2 \textit{ iff } t_1 \leq_T t_2 \wedge \\
& \quad (O_{F(t_1)} \supset O_{F(t_2)} \vee \\
& \quad \exists d \in D \\
& \qquad (N_{F(t_1)}(d) <_{R_N} N_{F(t_2)}(d) \vee \\
& \qquad V_{F(t_1)}(d) <_{R_V} V_{F(t_2)}(d)))
\end{aligned}
$$

namely if $t_1 \leq_T t_2$ but they are not the same subcategory, there must be a *differentia* keeping them apart. This enables us to represent t_1 as $Ref(t_2) + Diff(t_1, t_2)$, where $Ref(t_2)$ is a way to

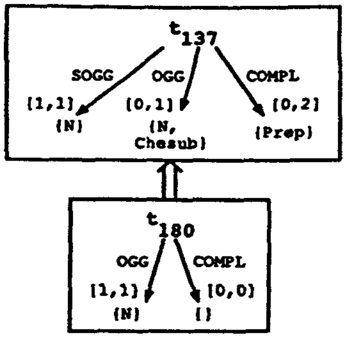

Figure 2: An example of subsumption between two sub categories.

identify t_2 from t_1, and $Diff(t_1, t_2)$ is a notatior for specifying the difference between the constraint associated with t_1, and the ones associated with t_2 So, we can say that the constraints associated witl t_1 are determined as the composition of the ones *inherited* from t_2 and the ones specified locally (the differentia) for t_1.
Graphically, an arc from t_2 to t_1 represents the subsumption relation ($Ref(t_2)$ in previous terms), parsimoniously represented by the immediate ancestor We show in figure 2 an example of subsumption between two subcategories, t_{137} – corresponding to the subcategorization frame q shown in the example o paragraph 3.1 – and t_{180}.
For the sake of clarity, we show the subcategorization frame associated with t_{137} with a graph. In t_{180} (subsumed by t_{137}), we specify the local constraint restrictions: the number restrictions of OGG become [1, 1], and those of COMPL become [0, 0]. Moreover the value restrictions of OGG become {N} (CHESUB is ruled out). By inheriting the constraints of t_{137} and restricting them locally, we obtain that t_{180} requires an obligatory nominal subject and an obligatory nominal object, and cannot have any complement. The order constraints – not shown in the figure – are also inherited in the obvious way.
A more significative example is in figure 4, that we will describe in section 5.

4 Parsing issues

Computational desiderata point towards a processing model that is input-driven, predictive, and abl to prune the parsing space as early as possible In this section, we propose an Earley-type parsing model with left-corner filtering [4]. The parser goe left-to-right and builds a structure that is alway connected, by hypothesizing templates for the lexical items which are predicted but not yet encountered in the input. It uses the information in th

[4] The basis of our work is (Lombardo and Lesmo, 1996) where the authors present an Earley-type recognizer for dependency grammar, and propose the compilation of dependency rules into parse tables.

hierarchy, by descending from the top class towards more specific classes. The descent is motivated by the fact that lower subcategories provide stronger constraints. It is possible to specify a procedure - described in (Barbero, 1998) - that consults the hierarchy just one time, in a compilation phase (during parsing it would be very time-consuming), and builds a parse table that guides the parser moves. In the following we give an intuitive description of the algorithm by assuming the dependency tree as data structure instead of the sets of items that characterize Earley's parsing style.

Initially, the parser guesses the presence of a node of a root category in the dependency tree. Then, given a node n associated to the subcategory t and a word w, the parser can perform three types of action: PREDICTION, SCANNING and COMPLETION.

1. **Prediction:** the parser guesses the presence of the dependents of n (by using left-corner information), given the constraints of the subcategory t of n. When the parser analyses a dependent which is distinctive for a possible specialization from the subcategory t to one of its children in the hierarchy, t_1 replaces t as the subcategory of n (for instance, if a direct object is hypothesized, we can directly descend from VERB to VERB-TRANS).
2. **Scanning:** the parser scans the head word of n (the word w in the input). The subcategory of w must be in the subtree rooted by t (including t itself). The left dependents of n that have been hypothesized in the prediction phase must fulfill the specific requirements imposed by the subcategory of the head (otherwise, the path is abandoned).
3. **Completion:** when the node n is "complete", namely all the dependents required by the subcategory t have been found, the next elements of the string can be analysed as dependents of the father node of n. If n has no father, i.e. it is the root of the dependency tree, and the end of the input string has been reached, the analysis ends successfully.

For example, the analysis of the sentence *Gli amici lo credevano un eroe*, "The friends considered him a hero", begins with the creation of a verbal root template (figure 3, "Initialization"). The first word in the input string is a determiner (*Gli*, "the"). A determiner can be the left-corner of a nominal group, so a prediction phase on the root node hypothesizes a left dependent of category NOUN labelled as subject (SOGG) [5]. The control goes to this node, from which a left dependent of category *Determ* is hypothesized. This last one is associated with the input word *Gli*, "the". The control returns to the node of category NOUN, that is associated with the next word *amici*, "friends". The node of category NOUN can be considered "complete" (no other dependent is required), and the control goes back to the root node.
At this point, the pronoun *lo*, "him", is read in input. A direct object is hypothesized and associated with it. A specialization from the top of the hierarchy to the subcategory of transitive verbs is possible: we know, in fact, that the root verb must be transitive, because a direct object has been hypothesized. The word *credevano* ("considered") is then read in input, and it is associated with the root node (scanning phase). Suppose that the verb *credere*, "consider", belongs to a class V-TR that requires a nominal subject (the hypothesis on the left dependent *amici* comes out to be correct), an object and a predicative complement.
The next input word, *un*, "a", is a determiner. Again, a nominal group is hypothesized, composed by a noun, playing the role of predicative complement, and a dependent of the noun, that is of category *Determ* and is associated with the word *un*.
The next input word, *eroe*, "hero", is associated with the node playing the role of predicative complement. The completion phase ends successfully the analysis of the sentence, as all the dependents required by the verb *credevano* (subject, object and predicative complement) have been found in the input sentence.

5 The classification of 101 Italian verbs

In investigating the empirical properties of a hierarchical grammar two issues must be addressed: the linguistic adequacy of the classification, and the parsimony of the hierarchy. We present some quantitative analyses of a corpus, showing that the proposed hierarchy reduces considerably the redundancy of a grammar for naturally occurring texts, while at the same time being sufficiently fine-grained to represent even very idiosyncratic items.
The hierarchy we propose encodes 101 Italian verbs taken from the grammar of Italian (Renzi, 1988) as the most representative of the main structures of Italian.

5.1 Materials and Method

The main sources of information used to carry out the classification are: (Renzi, 1988)'s Italian grammar, (Palazzi and Folena, 1992)'s Italian dictionary, and an Italian corpus of about 500 000 words. The corpus includes daily newspapers articles (367578 words), scientific dissertations (40013), young students compositions (27531), Verga's novels (12905), short news reports (6757), stories and various texts (5012). It is a varied corpus, representative of sev-

[5] In Italian, it could also be the direct object. We show here only one (non-deterministic) analysis path.

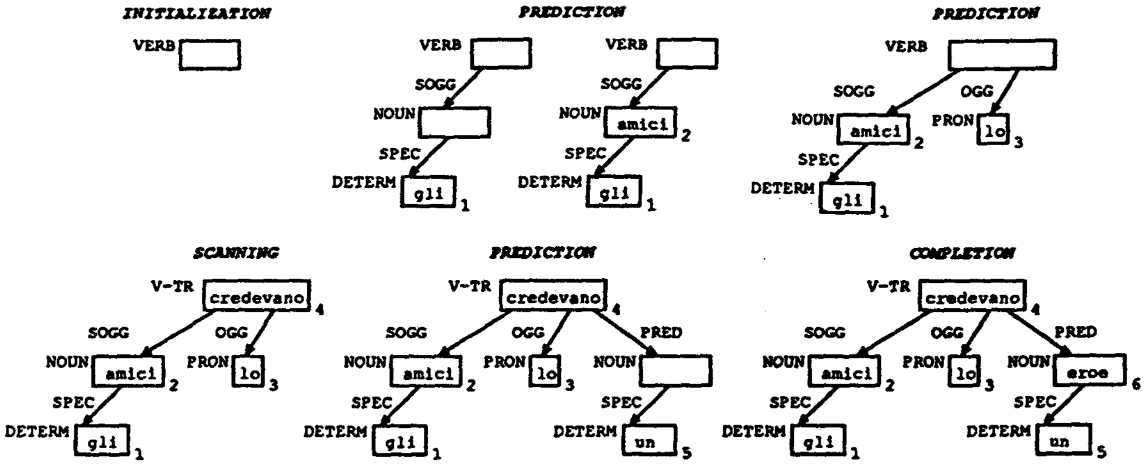

Figure 3: Analysis of the sentence *Gli amici lo credevano un eroe*, "The friends considered him a hero".

eral literary genres of written Italian.

The information required by our formalism — the grammatical relations associated to the dependents, their number ($N_q(d)$) and the set of categories ($V_q(d)$) that can realize them — was partly obtained by consulting Italian dictionaries, partly based on native speakers intuitions, and mostly from the analysis of the corpus.

All the sentences containing the verbs under analysis were automatically extracted from the corpus, and the subcategorization patterns (rules) exhibited by the verbs in those sentences were manually collected.

We represented the set of subcategorization patterns (rules) as subcategorization frames, by associating with each grammatical relation - according to the formalism - the related number ($N_q(d)$) and value ($V_q(d)$) restrictions computed on the corpus. In this test, we have kept the order between the dependents of a verb free, so there are no ordering constraints.

Each class t_1 is connected to its superclass t_2. $Diff(t_1, t_2)$, the difference between the constraints associated with t_1 and the ones associated with t_2, is expressed by specifying, for each relation that is restricted from t_2 to t_1, the relation itself with the new number and value restrictions.

5.2 Hierarchy

Figure 4 illustrates a small portion of the resulting hierarchy. This hierarchy is based on the dependency relations for a generic Italian verb summarized in Table 1 [6].

The whole hierarchy has 6 levels: the top level (class VERB) represents the general constraints for Italian verbs, the top+1 level distinguishes the constraints for impersonal (V1), intransitive (VERB-INTR) and transitive (VERB-TRANS) verbs, the top+2, top+3, etc. levels represent specific classes of verbs (from V2 to V50).

5.3 Results

The graph in figure 5 shows the distribution of verbs by type, namely how the number of verbs covered by the classes grows in relation to the number of classes. We can see that the first (more common) class covers 15 verbs, the first and second more common classes together covers 26 verbs, etcetera. With the first 9 classes we cover 63 verbs, giving rise to a reduction of 85.7% compared to having a distinct subcategorization frame for each verb. With the first 18 classes we cover 81 verbs (reduction of 77.7%). The whole set of verbs requires, however, 50 classes (reduction of 50.5%): in fact, we have found many verbs with very idiosyncratic behaviours.

Table 2 shows the distribution of verbs by token (sum of the occurrences, in the corpus, of all the verbs referring to each class), level by level. The fact that some rare classes occur is interesting if compared to the percentage of reduction in the representation. There is a compression of 55,7%, while still taking care of very low frequency patterns, where compression is almost 0%.

In Table 3, we show, for each level, the number of subcategorization patterns represented by all the classes of that level, namely the sum of the patterns of each class at that level. The number of patterns decreases rapidly by descending the hierarchy.

[6] Usually the adjuncts are not indicated as part of the subcategorization frames of the verbs: they are not obligatorily required by the verbs themselves. We have specified them anyway, as the hierarchy represents the grammar - which includes all the information about the dependents, adjuncts included. Moreover, by specifying the information about the adjuncts at the top level, we maintain the clarity of the representation and the mapping on the formal grammar. The representation of the syntactic knowledge concerning adjuncts is currently a research goal. Most authors tend to avoid it in the representation of subcategorization frames - see (Hudson, 1990) and the "adjoining" operation in LTAG (Joshi and Schabes, 1996).

GRAMMATICAL RELATION	SYNTACTIC CATEGORY (value restriction)	EXAMPLE + TRANSLATION
subject (SOGG)	nominal group, *N*	*Paolo ama Maria* "Paolo loves Maria"
	embedded clause headed by the complementizer *che*, "that", *Chesub*	*Mi diverte che tu dica ciò*, "It amuses me that you say this"
	preposition *di*, "of", *Prep[di]*	*Non mi interessa di venire*, "I am not interested in coming"
	infinitive verb, *Verb[inf]*	*Sciare è bello*, "Skiing is nice"
object (OGG)	nominal group, *N*	*Gianni mangia una mela*, "Gianni eats an apple"
	embedded clause headed by the complementizer *che*, "that", *Chesub*	*Credo che sia divertente*, "I think it is amusing"
	preposition *di*, "of", *Prep[di]*	*Aspetto di partire*, "I am waiting to leave"
predicative complement (PRED)	nominal group, *N*	*Considero Piero un amico*, "I consider Piero a friend"
	adjective group, *Adj*	*Luigi è gentile*, "Luigi is kind"
	prepositional group, *Prep*	*Tuo zio è senza ritegno*, "Your uncle is without reserve"
complements (COMPL), at most 3	prepositional group, *Prep*	*Metto il vaso sul tavolo* "I put the vase on the table"
	clitic, *Clitic*	*Gli ho dato un libro*, "I gave him a book"
adjunct (AGGIUNTO)	prepositional group, *Prep*	*Procedevo di buon passo*, "I was walking at a brisk pace"
	adverb, *Adv*	*Luigi corse velocemente*, "Luigi ran quickly"
	conjunction, *Conjdip*	*Telefonami quando puoi*, "Call me when you can"
	verb of non-finite mood, *Verb[non_fin]*	*Camminavo fischiettando*, "I was walking while whistling"

Table 1: The grammatical relations of a generic Italian verb, with their possible realizations and related examples.

For the patterns found in the texts, we observe a decrease similar but less marked than the grammatical patterns. Even the more specific classes describe a good portion of the patterns in the texts, so confirming the usefulness of very specific information in the analysis.

Table 2 show this point more clearly. The lower, more specific levels, while having fewer classes, still cover many occurrences of verbs in the text.

6 Conclusion

The paper has presented a hierarchical organization of a dependency formalism. The hierarchy is defined by the subsumption relation on subcategories, defined as a mapping between subcategories and subcategorization frames. Subcategorization frames, in turn, define the number of possible instantiations of a dependency relation and the subcategories that can realize it.

The hierarchical formalism has shown to be effective in representing parsimoniously – that is, without redundancy – the syntactic and lexical knowledge in an empirical test on 101 Italian verbs.

Moreover, we have sketched a left-to-right predictive parsing model that takes advantage of the hierarchical knowledge representation in order to make predictions on the structure of the input sentence.

In the next future we will address a massive empirical test of Italian corpora, and the formal specification of the parsing model, together with a complexity analysis.

7 Acknowledgements

This research was partly sponsored by the Swiss National Science Foundation, with fellowhip 8210-46569 to P. Merlo.

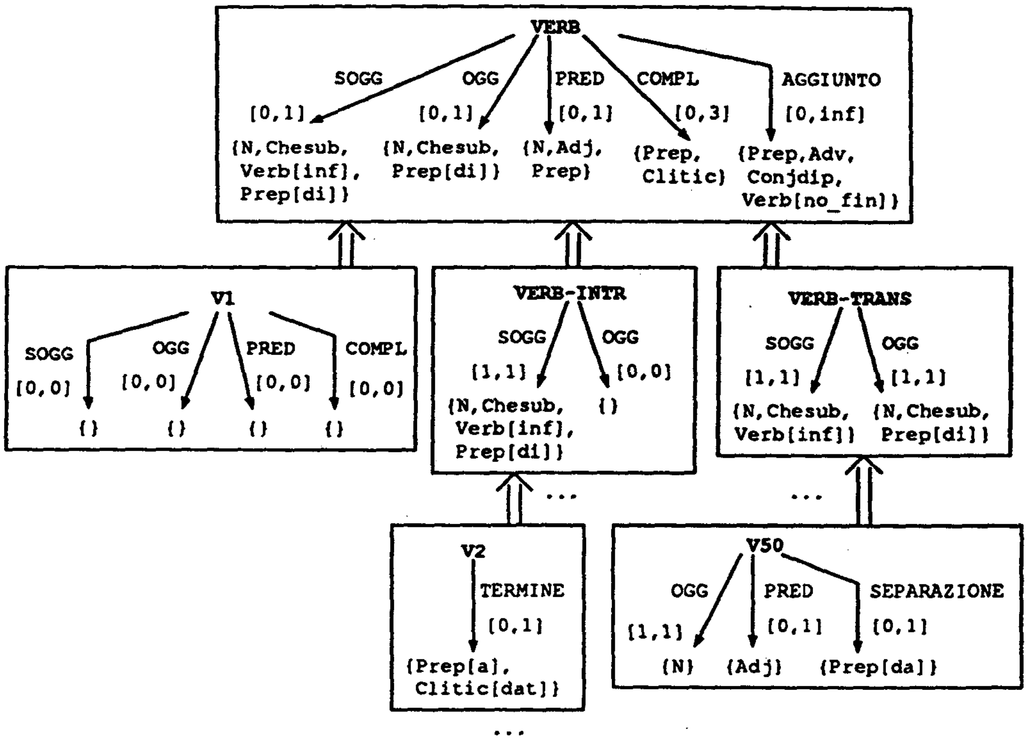

Figure 4: A portion of the hierarchy. Subclasses inherit and restrict the constraints at the top of the hierarchy. The top class, VERB, has three daughters. V1 is the class of impersonal verbs, that can only have adjuncts as dependents - the restriction is on the range, [0, 0], of the other relations. For example, we can say *Piove sui tetti della città*, "It rains on the roofs of the town". The classes VERB-INTR and VERB-TRANS correspond to intransitive and transitive verbs, respectively. VERB-INTR requires an obligatory subject ([1, 1]) and it cannot have a direct object ([0, 0]). VERB-TRANS requires an obligatory subject ([1, 1]), that can be headed by a nominal element, a conjunction *che* or an infinitive verb, and an obligatory object ([1, 1]). A subclass of VERB-INTR, V2, is shown: its only restriction is on the relation COMPL, which is specialized on the subrelation TERMINE, "Indirect Object", having a range [0, 1], and having *Prep[a]* and *Clitic[dat]* as associated categories (preposition lexically realized by *a*, "to", and dative clitic). For example, *sembrare*, "seem", is a verb belonging to this class: we can say *A Luigi Maria sembra bellissima*, "To Luigi Maria seems very beautiful". V50 is a subclass of VERB-TRANS: it restrics the sets of categories associated to the relations OGG and PRED, and specializes the relation COMPL on the subrelation SEPARAZIONE, "Separation" (realized by the preposition *da*, "from", *Prep[da]*). The verb *allontanare*, "distance", belongs to V50: *Luigi mi allontanò da te*, "Luigi distanced me from you".

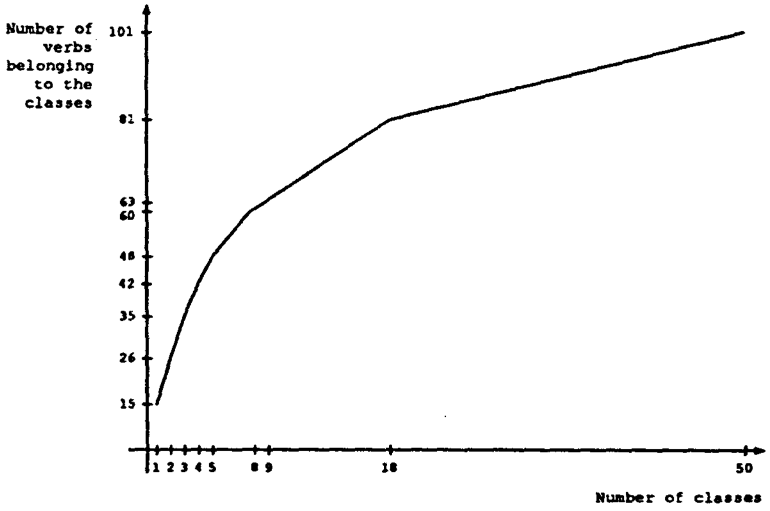

Figure 5: Distribution of verbs by type.

LEVEL	DISTRIBUTION OF CLASS SIZE
1	-
2	5
3	423 322 318 308 170 169 148 136 99 77 67 54 54 51 47 46 45 21 20 16 14 12 11 10 3 2
4	321 292 229 116 103 89 78 50 41 20 6 5
5	397 332 212 111 52 15
6	299 239 2

Table 2: Distribution of class size by level.

	PATTERNS	
LEVEL	GRAMM.	IN TEXT
1	484531	5674
2	244533	5674
3	102986	2643
4	20558	1351
5	1166	1135
6	134	540

Table 3: Number of possible and actual patterns at the levels of the hierarchy.

References

C. Barbero. 1998. *On the granularity of information in syntactic representation and processing: the use of a hierarchy of syntactic classes.* Ph.D. thesis, Università di Torino.

T. Becker. 1993. *HyTAG: a new type of Tree Adjoining Grammars for Hybrid Syntactic Representation of Free Order Languages.* Ph.D. thesis, University of Saarbruecken.

M.H. Candito. 1996. A principle-based hierarchical representation of LTAGs. In *Proceedings of COLING'96.*

C. Doran, B. Hockey, P. Hopely, J. Rosenzwieg, A. Sarkar, B. Srinivas, F. Xia, A. Nazr, and O. Ranbow. 1997. Maintaining the Forest and Burning out the Underbrush in XTAG. In *Computational Environments for Grammar Development and Language Engineering (ENVGRAM).*

R. Evans, G. Gazdar, and D. Weir. 1995. Encoding Lexicalized Tree Adjoining Grammar with a Nonmonotonic Inheritance Hierarchy. In *Proceedings of ACL'95.*

R. Hudson. 1990. *English Word Grammar.* Blackwell.

A. Joshi and Y. Schabes. 1996. Tree-Adjoining Grammars. In *Handbook of Formal Languages and Automata.* Springer-Verlag, Berlin.

V. Lombardo and L. Lesmo. 1996. An Earley-type recognizer for Dependency Grammar. In *Proceedings of COLING'96.*

V. Lombardo and L. Lesmo. 1998. Formal aspects and parsing issues of dependency theory. In *Proceedings of ACL-COLING'98.*

I. Mel'cuk. 1988. *Dependency Syntax: Theory and Practice.* SUNY Press, Albany.

W.D. Meurers. 1997. Using lexical principles in HPSG to generalize over valence properties. In *Proceedings of the Third Conference on Formal Grammar*, Aix-en-Provence, France.

B. Nebel. 1990. Reasoning and Revision in Hybrid Representation Systems. In *LNAI n. 422.* Springer-Verlag.

F. Palazzi and G. Folena. 1992. *Dizionario della lingua italiana.* Loescher.

C. Pollard and I. Sag. 1987. *Information-based syntax and semantics, vol. 1 Fundamentals.* CSLI.

L. Renzi. 1988. *Grande grammatica italiana di consultazione.* Il Mulino.

Y. Schabes. 1990. *Mathematical and Computational Aspects of Lexicalized Grammars.* Ph.D. thesis, University of Pennsylvania.

K. Vijay-Shanker and Y. Schabes. 1992. Structure sharing in lexicalized tree-adjoining grammars. In *Proceedings of COLING'92.*