Categorial Type Logic meets Dependency Grammar to annotate an Italian Corpus

R. Bernardi
KRDB,
Free University of Bolzano Bozen,
P.zza Domenicani, 3
39100 Bolzano Bozen,
Italy,
bernardi@inf.unibz.it

A. Bolognesi and **F. Tamburini**
CILTA,
University of Bologna,
P.zza San Giovanni in Monte, 4,
I-40124, Bologna,
Italy,
{bolognesi,tamburini}@cilta.unibo.it

M. Moortgat
UiL OTS,
Utrecht University,
Trans 10,
3512 JK, Utrecht,
The Netherlands
Moortgat@phil.uu.nl

Abstract

In this paper we present work in progress on the annotation of an Italian Corpus (CORIS) developed at CILTA (University of Bologna). We induce categorial type assignments from a dependency treebank (Torino University treebank, TUT) and use the obtained categories with annotated dependency relations to study the distributional behavior of Italian words and reach an empirically founded part-of-speech classification.

1 Introduction

The work reported on here is part of a project[1] aimed at annotating the CORIS/CODIS 100-million-words synchronic corpus of contemporary Italian with linguistic information: first part-of-speech tagging for the complete corpus, and in a later stage syntactic analysis for a subcorpus.

We have been investigating existing Italian treebanks in order to assess their potential usefulness for bootstrapping the CORIS/CODIS annotation tasks. We are aware of two such treebanks that are relevant to our purposes: the TUT corpus developed at Torino University, and ISST (Italian Syntactic-Semantic treebank) developed under the national program SI-TAL by a consortium of companies and research centers coordinated by the "Consorzio Pisa Ricerche" (CPR)[2].

ISST is a multi-layered corpus, annotated at the syntactic and lexico-semantic levels. A user interface is provided to explore the corpus. The ISST corpus is rather competitive in terms of size: it counts 305,547 word tokens (Montemagni et al., 2003). A drawback is that the corpus is not publicly available yet. The TUT corpus is rather small, consisting only of 38,653 words. There is no user interface for TUT but the corpus is downloadable (`http://www.di.unito.it/~tutreeb/`). Despite its small size, TUT can serve as a training corpus for creating larger annotated resources.

Our goal is to annotate CORIS with part-of-speech (PoS) tags and semi-automatically build a treebank for a fragment of it. To achieve this two-fold task we start from TUT exploiting its information on dependency relations. We are still in a preliminary phase of the project and so far attention has been focused on the first task. However, the work done in this phase is expected to play a role in our second task too.

In Section 2, we will describe in more detail the problems which arise when working out the first task. In Section 3, we briefly introduce the formalisms we work with. In Section 4, we explain how we encode dependency relations into categorial type assignments (CTAs), and how we automatically induce these types from TUT dependency structures. Finally, in Section 5 we draw some preliminary conclusions and briefly describe our action list.

2 PoS tagging for Italian

Before embarking on our first task, we have studied the current situation with respect to PoS tagging for Italian. Italian is one of the languages for which a set of annotation guidelines has been developed in the context of the EAGLES project (Expert Advisory Group on Language Engineering Standards (Monachini, 1995)). Several research groups have worked on PoS annotation in practice (for example, Torino University, Xerox and Venice University).

We have compared the tag sets used by these groups with Monachini's guidelines. From this comparison, it results that though there is a general agreement on the main parts of speech

[1] The project is funded under the FIRB 2001 action.

[2] The parnters of the consortium were: ILC-CNR/CPR, Venice University/CVR, ITC-IRST, "Tor Vergata" University/CERTIA, and Synthema.

to be used[3], considerable divergence exists when it comes to the actual classification of Italian words with respect to these main PoS classes. The classes for which differences of opinion are most evident are adjectives, determiners and adverbs. For instance, words like *molti* (tr. many) have been classified as "indefinite determiners" by Monachini, "plural quantifiers" by Xerox, "indefinite adjectives" by the Venice and Turin groups. This simple example shows that the choice of PoS tags is already influenced by the linguistic theory adopted in the background. This theoretical bias will then influence the kind of conclusions one can draw from the annotated corpus.

Our aim is to derive an empirically founded PoS classification, making no *a priori* assumptions about the PoS classes to be distinguished. Our background assumptions are minimal and, we hope, uncontroversial: we assume that we have access to head-dependent (H-D) and functor-argument (F-A) relations in our material. We encode the H-D and F-A information into categorial type formulas. These formulas then serve as "labels/tags" from which we obtain the desired empirically founded PoS classification by means of a clustering algorithm.

To bootstrap the process of type induction, we transform the TUT corpus into a simplified dependency treebank. The transformation keeps the bare dependency relations but removes the more theory-laden annotation. In Section 4, we describe how we use the simplified dependency treebank for our distributional study of Italian PoS classification. First, we briefly look at H-D and F-A relations as they occur in the TUT corpus and in Categorial Type Logic (CTL).

3 Dependency and functor-argument relations

3.1 Dependency structures in TUT

The Turin University Treebank (TUT) is a corpus of Italian sentences annotated by specifying relational structures augmented with morphosyntactic information and semantic role (henceforth ARS) in a monostratal dependency-based representation. The treebank in its current release includes 38,653 words and 1,500 sentences from the Italian civil law code, the national newspapers *La Stampa* and *La Repubblica*, and from various reviews, newspapers, novels, and academic papers.

The ARS schema consists of i) morpho-syntactic, ii) functional-syntactic and iii) semantic components, specifying part-of-speech, grammatical relations, and thematic role information, respectively. The reader is referred to (Bosco, 2003) for a detailed description of the TUT annotation schema. An example is given below (tr. "The first steps have not been encouraging"). In this example, the node TOP-VERB is the root of the whole structure[4].

```
**************  FRASE ALB-71  **************
1  I (IL ART DEF M PL)
                [6;VERB-SUBJ]
2  primi (PRIMO ADJ ORDIN M PL)
                [3;ADJC+ORDIN-RMOD]
3  approcci (APPROCCIO NOUN COMMON M PL)
                [1;DET+DEF-ARG]
4  non (NON ADV NEG)
                [6;ADVB-RMOD]
5  sono (ESSERE VERB AUX IND PRES INTR 3 PL)
                [6;AUX+TENSE]
6  stati (ESSERE VERB MAIN PART PAST INTR PL M)
                [0;TOP-VERB]
7  esaltanti (ESALTANTE ADJ QUALIF ALLVAL PL)
                [6;VERB-PREDCOMPL+SUBJ]
8  . (#\. PUNCT) [6;END]
```

Because we are interested in extracting dependency relations, we can focus on the functional-syntactic component of the TUT annotation, where information relating to grammatical relations (heads and dependents) is encoded.

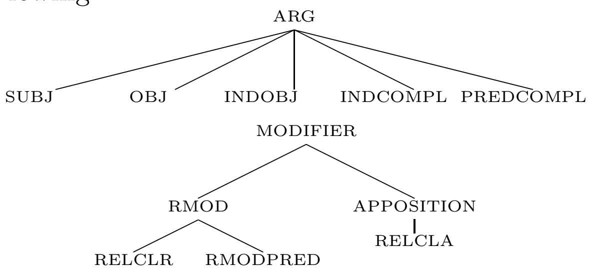
The TUT annotation schema for dependents makes a primary distinction between (a) functional and (b) non-functional tags, for dependents that can and that cannot be assigned thematic roles, respectively. These two classes are further divided into (a') arguments (ARG) and modifiers (MOD) and (b'), AUX, COORDINATOR, INTERJECTION, CONTIN, EMPTYCOMPL, EMPTYLOC, SEPARATOR and VISITOR[5]; and furthermore, the arguments

[3] The standard classification consists of nouns, verbs, adjectives, determiners, articles, adverbs, prepositions, conjunctions, numerals, interjections, punctuation and a class of residual items which differs from project to project.

[4] The top nodes used in TUT are TOP-VERB, TOP-NOUN, TOP-CONJ, TOP-ART, TOP-NUM, TOP-PRON, TOP-PHRAS and TOP-PREP

[5] The labels that require some explanation are: (i) CONTIN, (ii) EMPTYCOMPL, (iii) EMPTYLOC and (iv) VISITOR. They are used for expressions that (i) introduce a part of an expression with a non-compositional interpretation (e.g. locative or idiomatic expressions and denominative structures: "Arrivò [prima]$_H$ [de]$_D$ ll'alba", lit. tr. "(She) arrived ahead of the daybreak"); (ii) link a re-

(ARG) and modifiers (MOD) are sub-divided as following



3.2 Categorial functor-argument structures

Categorial Type Logic (CTL) (Moortgat, 1997) is a logic-based formalism belonging to the family of Categorial Grammars (CG). In CTL, the type-forming operations of CG are viewed as logical connectives. As the slogan "Parsing-as-Deduction" suggests, such a view makes it possible to do away with combinatory syntactic rules altogether; establishing the well-formedness of an expression becomes a process of deduction in the logic of the type-forming connectives.

The basic distinction expressed by the categorial type formulas is the Fregean opposition between complete and incomplete expressions. Complete expressions are categorized by means of *atomic* type formulas; grammaticality judgements for expressions with an atomic type do not require further contextual information. Typical examples of atomic types would be 'sentence' (s) and 'noun' (n). Incomplete expressions are categorized by means of fractional type formulas; the denominators of these fractions indicate the material that has to be found in the context in order to obtain a complete expression of the type of the numerator.

Definition 3.1 (Fractional type formulas) *Given a set of basic types ATOM, the set of types TYPE is the smallest set such that:*

i. *if $A \in$ ATOM, then $A \in$ TYPE;*
ii. *if A and $B \in$ TYPE, then A/B and $B\backslash A \in$ TYPE.*

There are different ways of presenting valid type computations. In a Natural Deduction format, we write $\Gamma \vdash A$ for a demonstration that the structure Γ has the type A. The statement $A \vdash A$ is axiomatic. Each of the connectives / and \ has an Elimination rule and an Introduction rule. Below, we give these inference rules for / (incompleteness to the right). The cases for \ (incompleteness to the left) are symmetric. Given structures Γ and Δ of types A/B and B respectively, the Elimination rule builds a compound structure $\Gamma \circ \Delta$ of type A. The Introduction rule allows one to take apart a compound structure $\Gamma \circ B$ into its immediate substructures.

$$\frac{\Gamma \vdash A/B \quad \Delta \vdash B}{\Gamma \circ \Delta \vdash A}\,/E \qquad \frac{\Gamma \circ B \vdash A}{\Gamma \vdash A/B}\,/I$$

Notice that the language of fractional types is essentially higher-order: the denominator of a fraction does not have to be atomic, but can itself be a fraction. The Introduction rules are indispensable if one is interested in capturing the full set of theorems of the type calculus. Classical CG (in the style of Ajdukiewicz and Bar-Hillel) uses only the Elimination rules, and hence has restricted inferential capacities. It is impossible in classical CG to obtain the validity $A \vdash B/(A\backslash B)$, for example. Still, the classical CG perspective will be useful to realize our aim of automatically inducing type assignments from structured data obtained from the TUT corpus thanks to the type resolution algorithm explained below.

Type inference algorithms for classical CG have been studied by (Buszkowski and Penn, 1990). The structured data needed by their type inference algorithms are so-called *functor-argument structures* (*fa*-structures). An *fa*-structure for an expression is a binary branching tree; the leaf nodes are labeled by lexical expressions (words), the internal nodes by one of the symbols ◀ (for structures with the functor as the left daughter) or ▶ (for structures with the functor as the right daughter).

To assign types to the leaf nodes of an *fa*-structure, one proceeds in a top-down fashion. The type of the root of the structure is fixed (for example: s). Compound structures are typed as follows:

- to type a structure $\Gamma \blacktriangleleft \Delta$ as A, type Γ as A/B and Δ as B;
- to type a structure $\Gamma \blacktriangleright \Delta$ as A, type Γ as B and Δ as $B\backslash A$.

If a word occurs in different structural environments, the typing algorithm will produce dis-

flexive personal pronoun with particular verbal head (e.g. "La porta [si]$_D$ [apre]$_H$", lit. tr. "the door (it) opens"); (iii) link a pronoun with a verbal head introducing a sort of metaphorical location of the head (e.g. "In Albania [ci]$_D$ [sono]$_H$ molti problemi", lit. tr. "In Albany there are many problems".); (iv) mark the extraction of a part of a structure (e.g. "Così devi vedere questo argomento", lit. tr. "This way (you) must see this topic").

tinct types. The set of type assignments to a word can be reduced by *factoring*: one identifies type assignments that can be unified. For an example, compare the structured input below:

a. Claudia ▶ parla
b. Claudia ▶ (parla ▶ bene)

Assuming a goal type s, from (a) we obtain the assignments

$$\text{Claudia} : A, \text{parla} : A\backslash s$$

and from (b)

$$\text{Claudia} : C, \text{parla} : B, \text{bene} : B\backslash(C\backslash s)$$

Factoring leads to the identifications $A = C$, $B = (A\backslash s)$, producing for "bene" the modifier type $(A\backslash s)\backslash(A\backslash s)$.

3.3 From TUT dependency structures to categorial types

To accomplish our aims, we will have an occasion to use two extensions of the basic categorial machinery outlined in the section above: a generalization of the type language to multiple modes of composition, and the addition of structural rules of inference to the logical rules of slash Elimination and Introduction.

Multimodal composition The intuitions underlying the distinction between heads and dependents in Dependency Grammars (DG) and between functors and arguments in CG often coincide, but there are also cases where they diverge (Venneman, 1977). In the particular case of the TUT annotation schema, we see that for all instances of dependents labeled as ARG (or one of its sublabels), the DG head/dependent articulation coincides with the CG functor/argument asymmetry. But for DG modifiers, or dependents without thematic roles of the class AUX (auxiliary)[6] there is a mismatch between dependency structure and functor-argument structure. Modifiers would be functors in terms of their categorial type: functors where the numerator and the denominator are identical. This makes them into 'identities' for the fractional multiplication, which explains their optionality and the possibility of iteration. AUX elements in DG would count as morphological modifiers of the head verbs. From the CG point of view, they would be typed as functors with non-identical numerator and denominator, distinguishing them that way from optional modifiers, and capturing the fact that they are indispensable to build a complete grammatical structure.

To reconcile the competing demands of the head-dependent and functor-argument classification, we make use of the type calculus proposed in (Moortgat and Morrill, 1991), which treats dependency and functor-argument relations as two orthogonal dimensions of linguistic organization. Instead of one composition operation $\circ$, the system of (Moortgat and Morrill, 1991) has two: $\circ_l$ for structures where the left daughter is the head, and $\circ_r$ for right-headed structures. The two composition operations each have their slash and backslash operations for the typing of incomplete expressions:

- $A/_lB$: a functor looking for a B to the right to form an A; the functor is the head, the argument the dependent;
- $A/_rB$: a functor looking for a B to the right to form an A; the argument is the head, the functor the dependent;
- $B\backslash_l A$: a functor looking for a B to the left to form an A; the argument is the head, the functor the dependent;
- $B\backslash_r A$: a functor looking for a B to the left to form an A; the functor is the head, the argument the dependent.

The type inference algorithm of (Buszkowski and Penn, 1990) can be straightforwardly adapted to the multimodal situation. The internal nodes of the *fa*-structures now are labeled with a fourfold distinction: as before, the triangle points to the functor daughter of a constituent; in the case of the black triangle, the functor daughter is the head constituent, in the case of the white triangle, the functor daughter is the dependent.

	ad	ah	fd	fh
fh	◀			
fd		◁		
ah			▷	
ad				▶

The type-inference clauses can be adapted accordingly.

- to type a structure Γ ◀ Δ as A, type Γ as $A/_lB$ and Δ as B;

[6] And also COORDINATOR, INTERJECTION.

- to type a structure $\Gamma \lhd \Delta$ as A, type Γ as $A/_r B$ and Δ as B.
- to type a structure $\Gamma \blacktriangleright \Delta$ as A, type Δ as $B\backslash_r A$ and Γ as B;
- to type a structure $\Gamma \rhd \Delta$ as A, type Δ as $B\backslash_l A$ and Γ as B.

Structural reasoning The dependency relations in the TUT corpus abstract from surface word order. When we induce categorial type formulas from these dependency relations, as we will see in Section 4.1, the linear order imposed by '/' and '\' in the obtained formulas will not always be compatible with the observable surface order. Incompatibilities will arise, specifically, in the case of non-projective dependencies. Where such mismatches occur, the induced types will not be immediately useful for parsing — the longer term subtask of the project discussed here.

To address this issue, we can extend the inference rules of our categorial logic with *structural* rules. The general pattern of these rules is: infer $\Gamma' \vdash A$ from $\Gamma \vdash A$, where Γ' is some rearrangement of the constituents of Γ. These rules, in other words, characterize the structural deformations under which type assignment is preserved. Structural rules can be employed in two ways in CTL (see (Moortgat, 2001) for discussion). In an *on-line* use, they actually manipulate structural configurations during the parsing process. Such on-line use can be very expensive computationally. Used *off-line*, they play a role complementary to the factoring operation, producing a number of derived lexical type-assignments from some canonical assignment. With the derived assignments, parsing can then proceed without altering the surface structure.

As indicated in the introduction, the use of CTL in the construction of a treebank for a part of the CILTA corpus belongs to a future phase of our project. For the purposes of this paper we must leave the exact nature of the required structural rules, and the trade-off between offline and on-line uses, as a subject for further research.

4 A distributional study of Italian part-of-speech tagging

In order to annotate the CORIS corpus with a theory-neutral set of PoS tags, we plan to carry out a distributional study of its lexicon.

Early approaches to this problem were based on the hypothesis that if two words are syntactically and semantically different, they will appear in different contexts. There are a number of studies that, starting from this hypothesis, have built automatic or semi-automatic procedures for clustering words (Brill and Marcus, 1992; Pereira et al., 1993; Martin et al., 1998), especially in the field of cognitive sciences (Redington et al., 1998; Gobet and Pine, 1997; Clark, 2000). They examine the distributional behaviour of some target words, comparing the lexical distribution of their respective collocates using quantitative measures of distributional similarity (Lee, 1999).

In (Brill and Marcus, 1992) it is given a semi-automatic procedure that, starting from lexical statistical data collected from a large corpus, aims to arrange target words in a tree (more precisely a dendrogram), instead of clustering them automatically. This procedure requires a linguistic examination of the resulting tree, in order to identify the word classes that are most appropriate to describe the phenomenon under investigation. In this sense, they use a semi-automatic word-class generator method.

A similar procedure has been applied on Italian in (Tamburini et al., 2002). The novelty of this work is that it derives the distributional information on words from a very basic set of PoS tags, namely nouns, verbs and adjectives. This method, completely avoiding the sparseness of the data affecting Brill and Marcus' method, uses general information about the distribution of lexical words to study the internal subdivisions of the set of grammatical words, and results more stable than the method based only on lexical co-occurrence.

The main drawback of these techniques is the limited context of analysis. Collecting information from a defined context, typically two or three words will invariably miss syntactic dependencies longer than the context interval. To overcome this problem we propose to exploit the expressivity of CTAs (with encoded core dependency relations, as we saw in the section above) by applying the clustering algorithms on them. Below we sketch how we intend to induce CTAs from the TUT dependency treebank, and the clustering method we plan to implement. The whole procedure can be summarized by the picture below.

$$\begin{tikzcd} \text{Treebank} \arrow[r, "conversion"] & \text{CTL structures} \arrow[d, "type\ \ resolution"] \\ \text{PoS tagset} & \text{Categorial Types} \arrow[l, "clustering"'] \end{tikzcd}$$

4.1 Inducing categorial types from TUT

The first step is to reduce the distinctions encoded in the TUT treebank to bare head-dependent relations: the ARG type on the one hand, and the MOD and AUX types on the other. These relations are converted into *fa*-structures built by means of the dependency-sensitive operators $\blacktriangleleft$, $\blacktriangleright$, $\lhd$, $\rhd$.

By means of example, we consider some simple sentences exemplifying the different relations.

Figure 1 shows a head-dependent structure in which edges represent head-dependent relations and each edge points to the dependent of each relation. In this example, each H-D relation agrees with the F-A relation, i.e. each head corresponds to a functor and the dependents are all labeled as arguments (or sub-tags of it).[7]

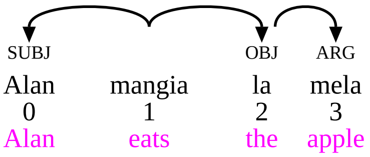


Figure 1: ARG: Functor and Head coincide

Figure 2 adds to the example from figure 1 the use of qualifying adjectives, which is an example of a modifier, and past tense auxiliaries. Considering the relation between “mela”(apple) and “rossa” (red), and between “ha” (has) and “mangiato” (eaten), we have the dependency trees in Figure 2.

In the first case, the noun is the head and the adjective is the dependent, but from the functor-argument perspective, the adjective (in general, the modifier) is the incomplete functor component. A similar discrepancy is observed for the auxiliary and the main verb, where the auxiliary should be classified as the incomplete functor, but as the dependent element with respect to the main verb. In this case the absence of the auxiliary would result in an ungrammatical sentence. The relations of MOD and AUX exhibit a different behavior than ARG, and hence are depicted with different arcs.

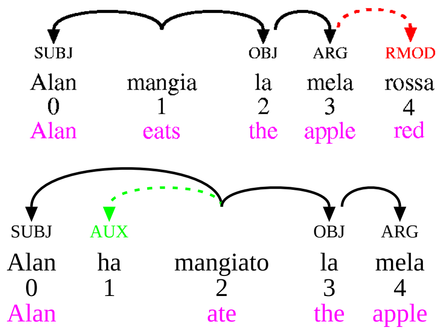


Figure 2: MOD and AUX: Functors as Dependents

Our simple example sentences could be converted into the following *fa*-structures:

- Allen $\blacktriangleright$ (mangia $\blacktriangleleft$ (la $\blacktriangleleft$ mela)
- Allen $\blacktriangleright$ (mangia $\blacktriangleleft$ (la $\blacktriangleleft$ (mela $\rhd$ rossa))
- Allen $\blacktriangleright$ ((ha $\lhd$ mangiato) $\blacktriangleleft$ (la $\blacktriangleleft$ mela))

The second step is to run the Buszkowski-Penn type-inference algorithm (in its extended form, discussed above) on the *fa*-structures obtained from TUT, and to reduce the lexicon by factoring (identification of unifiable assignments) and (in a later phase) structural derivability. Fixing the goal type for these examples as s, we obtain the following type assignments from the *fa*-structures given above:

Allan	A
mangia	$(A\backslash_r s)/_l B$
la	$B/_l C$
mela	C
rossa	$C\backslash_l C$
ha	$((A\backslash_r s)/_l B)/_r D$
mangiato	D

Notice that from the output in our tiny sample, we have no information allowing us to identify the argument assignments A and B. Notice also that from an *fa*-structure which takes together “ha mangiato” in a constituent, we obtain a type assignment for “mangiato” that does not express its incompleteness anymore — instead, the combination with the auxiliary expresses this. This is already an example where structural reasoning can play a role: compare the above analysis with the type solution one would obtain by starting from an

[7] The example follows TUT practice in designating the determiner as the head of the noun phrases. We are aware of the fact that this is far from controversial in the dependency community. In preprocessing TUT before type inference, we have the occasion to adjust such debatable decisions, and representational issues such as the use of empty categories, for which there is no need in a CTL framework.

fa-structure which takes "mangiato la mela" as a constituent, which yields a type solution $(A\backslash_r s)/_r E$ for the auxiliary, and $E/_l B$ for the head verb. We are currently experimenting with the effect of different constituent groupings on the size of the induced type lexicon.

4.2 Clustering Algorithms

Once we have induced the categorial type assignments for the TUT lexicon, the last step of our first task is to divide it into clusters so to study the distributional behavior of the corresponding lexical entries. The advantage of using categorial types as objects of the clustering algorithm is that they represent long distance dependencies as well as limited distributional information. Thus the categorial types become the basic elements of syntactic information associated with lexical entries and the basic "distributional fingerprints" used in the clustering process.

Every clustering process is based on a notion of "distance" between the objects involved in the process. We should define an appropriate metric among categorial types. We believe that a crucial role will be played by the dependency relation encoded into the types by means of compositional modes.

Currently, we are studying the application of proper distance measures considering types as trees and adapting the theoretical results on tree metrics to our problem. The algorithm for computing the tree-edit distance (Shasha and Zhang, 1997), designed for generic trees, appears to be a good candidate for clustering in categorial-type domain. What remains to be done is to experiment the algorithm and finetune the metrics to our purpose.

5 Conclusions and Further Research

In this paper we have presented work in progress devoted to the syntactic annotation of a large Italian corpus. We have just started working in this direction and the biggest part of the work has still to be done. We are currently evaluating the TUT encoding of dependency information, and identifying areas that allow optimization from the point of view of CTL type induction. A case in point is the heavy reliance of TUT on empty elements and/or traces, which conflicts with our desire for an empirically-based and theory-neutral representation of linguistic dependencies. It seems that the trace artifact can be avoided if one properly exploits the more expressive category concept of CTL, allowing product types for asyndetic constructions, and higher-order types for multiple dependencies. In parallel, we are looking for other sources of dependency information for Italian, in order to complement the rather small TUT database we have at our disposal now.

6 Acknowledgments

Our thanks go to FIRB 2001 project RBNE01H8RS coordinated by prof. R. Rossini Favretti for the funding supports. Thanks to L. Surace and C. Seidenari for the detailed comparison on Italian PoS classifications.

References

C. Bosco. 2003. *A grammatical relation system for treebank annotation.* Ph.D. thesis, Computer Science Department, Turin University.

E. Brill and M. Marcus. 1992. Tagging an unfamiliar text with minimal human supervision. In *Proceedings of the Fall Symposium on Probabilistic Approaches to Natural Language*, pages 10–16, Cambridge. MA: American Association for Artificial Intelligence.

W. Buszkowski and G. Penn. 1990. Categorial grammars determined from linguistic data by unification. *Studia Logica*, 29:431–454.

A. Clark. 2000. Inducing syntactic categories by context distribution clustering. In *Proceedings of CoNLL-2000 and LLL-2000 Conference*, pages 94–91, Lisbon, Portugal.

F. Gobet and J. Pine. 1997. Modelling the acquisition of syntactic categories. In *Proceedings of the 19th Annual Meeting of the Cognitive Science Society*, pages 265–270.

L. Lee. 1999. Measures of distributional similarity. In *Proceedings of the 37th ACL*, pages 25–32, College Park, MD.

S. Martin, J. Liermann, and H. Ney. 1998. Algorithms for bigram and trigram word clustering. *Speech Communication*, 24:19–37.

M. Monachini. 1995. ELM-IT: An Italian incarnation of the EAGLES-TS. definition of lexicon specification and classification guidelines. Technical report, Pisa.

S. Montemagni, F. Barsotti, M. Battista, N. Calzolari, O. Corazzari, A. Lenci, A. Zampolli, F. Fanciulli, M. Massetani, R. Raffaelli, R. Basili, M. T. Pazienza, D. Saracino, F. Zanzotto, N. Mana, F. Pianesi, and R. Delmonte, 2003. *Building and using parsed corpora*, chapter Building the Italian Syntactic-Semantic Treebank, pages 189–210. Language and Speech series. Kluwer, Dordrecht.

M. Moortgat and G. Morrill. 1991. Heads and phrases. Type calculus for dependency and constituent structure. Technical report, Utrecht.

M. Moortgat. 1997. Categorial type logics. In J. van Benthem and A. ter Meulen, editors, *Handbook of Logic and Language*, pages 93–178. The MIT Press, Cambridge, Massachusetts.

Michael Moortgat. 2001. Structural equations in language learning. In P. de Groote, G. Morrill, and C. Retoré, editors, *Logical Aspects of Computational Linguistics*, volume 2099 of *Lecture Notes in Artificial Intelligence*, pages 1–16, Berlin. Springer.

F. Pereira, T. Tishby, and L. Lee. 1993. Distributional clustering of English words. In *Proceedings of the 31st ACL*, pages 183–190, Columbus, Ohio.

M. Redington, N. Chater, and S. Finch. 1998. Distributional information: a powerful cue for acquiring syntactic categories. *Cognitive Science*, 22(4):425–469.

D. Shasha and D. Zhang. 1997. Approximate tree pattern matching. In A. Apostolico and Z. Galig, editors, *Pattern matching algorithms.* Oxford University Press.

F. Tamburini, C. De Santis, and Zamuner E. 2002. Identifying phrasal connectives in Italian using quantitative methods. In S. Nuccorini, editor, *Phrases and Phraseology -Data and Description.* Berlin: Peter Land.

T. Venneman. 1977. Konstituenz und Dependenz in einigen neueren Grammatiktheorien. *Sprachwissenschaft*, 2:259–301.