Automatic induction of a POS tagset for Italian

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

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

Abstract

In this paper we present work in progress on the PoS annotation of an Italian Corpus (CORIS) developed at CILTA (University of Bologna). We aim to automatically induce the PoS tagset by analysing the distributional behaviour of Italian words by relying only on theory-neutral linguistic knowledge. To this end, we propose an algorithm that derives a possible tagset to be further interpreted and defined by the linguist. The algorithm extracts information from loosely labelled dependency structures that encode only basic and broadly accepted syntactic relations, namely Head/Dependent, and the distinction of dependents into Argument vs. Adjunct.

1 Introduction

The work presented in this paper is part of a project aiming to annotate CORIS/CODIS (Rossini Favretti et al., 2002), a 100-million-word synchronic corpus of contemporary written Italian, with part-of-speech (PoS) tags.

Italian is one of the languages for which a set of annotation guidelines has been developed in the context of the EAGLES project (Monachini, 1995). Several research groups have worked on PoS annotation in practice (for example, Torino University, Xerox and Venice University), but comparing the tag sets used by these groups with Monachini's guidelines reveals that though there is a general agreement on the main parts of speech to be used[1], 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* (many) have been classified as "indefinite determiners" by Monachini, "plural quantifiers" by Xerox, "indefinite adjectives" by the Venice and Turin groups. It is not simply a matter of different terminological options resolvable by a mere one-to-one relabelling, nor a matter of simply mapping different classes into a greater one. Crossings between tagsets are complex mostly because of the different theoretical points of view used in categorizing words. For instance, the single tag DET "determiner" in the Xerox tagset matches with DIM "demonstrative adjective" or ART "article" in the Venice group (and with DET "determiner" or ART "article" in Monachini) whereas, viceversa, the single tag DEIT "deictic pronoun" by the Venice group matches alternatively with DEM "demonstrative" or PRON "personal pronoun" in Xerox.

These simple examples show that the choice of PoS tag is already influenced by the underlying linguistic theory adopted. This theoretical bias will then influence the kind of conclusions one can draw from the annotated corpus.

Our aim is to automatically derive an empirically founded PoS classification making no *a priori* assumptions about the PoS classes to be distinguished.

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 based on this hypothesis in the fields of both computational linguistics and cognitive science aiming at building automatic or semi-automatic procedures for clustering words (Brill and Marcus, 1992; Pereira et al., 1993; Schütze, 1993; Clark, 2000; Redington et al., 1998).These papers examine the distributional behaviour of some target words by comparing the lexical distribution of their respective collocates and by using quantitative measures of distributional similarity.

The main drawback of these techniques is the

[1]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.

Proceedings of the Australasian Language Technology Workshop 2005, pages 176–183,
Sydney, Australia, December 2005.

limited context of analysis. Information is collected from a restricted context, of for instance 3 words, which can conceal syntactic dependencies longer than the context interval.

Our approach to solve this problem is to use basic syntactic relations together with distributional and morphological information. The system we have developed consists of three phases: (1) a first basic distinction of word classes is induced by means of Brill's algorithm (Brill and Marcus, 1992); (2) in the second phase, this distinction is further specified by means of minimal syntactic information; and (3) in the third phase, the ultimate PoS tagset is obtained by using distributional and morphological knowledge. Little, if any, language-specific knowledge is used, hence the method is in principle applicable to any language.

A large number of localized syntactic descriptions per word are exploited to identify differences in the syntactic behaviour of words. Associating rich descriptions to lexical items, our approach is, to some extent, related to supertags (Bangalore and Joshi, 1999).

The outcome is a *hierarchy* of PoS tags that is expected to help annotators and enhance the search interface of the annotated corpus.

Section 2 gives an outline of our work; Section 3 describes in details the algorithm; Section 4 analyses the results of the work, listing the PoS tags obtained with this method; section 5 briefly outlines further work.

2 Proposal

The present paper focuses on the second phase of the system describing how syntactic information can be exploited to induce the PoS tagset. It builds on the results obtained in (Tamburini et al., 2002) where it is shown that Brill's algorithm identifies three main word classes, namely noun (N), verbs (V) and all the others (X).

In this article we will focus on the X class, describing how this can be further broken down by automatically grouping words that share similar syntactic behaviours. The algorithm uses the tags obtained in the first phase and dependency structures carrying only basic syntactic information about Head/Dependent relations and Argument/Adjunct distinctions among the Dependents.

Starting from these loosely labelled dependency structures, the type resolution algorithm obtains type assignments for each word. The syntactic type assignments obtained encode the different syntactic behaviour exhibited by each word. Examples of the labelled dependency structures and the obtained assignments are given in Figure 2. An information lossless simplification algorithm is used to automatically derive a first tagset approximation (see Section 3).

At the end of the second phase, the X class is divided into 9 PoS tags that are sets of syntactic behaviours. In the third phase, we plan to further divide the classes obtained by means of distributional and morphological information.

3 The Algorithm

The algorithm consists essentially of three components: (i) in the first, each word is assigned the complete set of syntactic types extracted from loosely labelled dependency structures; (ii) in the second, we obtain a first approximation of relevant classes by grouping words that display similar behaviours, and we build their inclusion chart. This is obtained by creating the sets of those words that in (i) showed the same type at least once, and by pairing these sets of words with their shared set of types. In the following sections we will refer to such pairs as Potential PoS (PPoS); (iii) finally, we prune the obtained inclusion chart by highlighting those paths that relate pairs which are significantly similar, where the similarity is measured in terms of frequency of types and words. The pruning results in a forest of trees whose leaves form sets identifying the induced PoS tags.

Figure 1 shows a flow chart which summarizes the three phases of our algorithm.

3.1 Dependency Structures

Our dependency structures are derived from a sub-treebank of TUT, The Turin University Treebank (Bosco et al., 2000; Bosco, 2003). The treebank currently includes 1500 sentences organized in different sub-corpora from which we converted 441 dependency trees, maintaining only the basic syntactic information required for this study. More specifically, we mantained information on Head-Dependent relations by distinguishing each dependent either as an Argument or as an Adjunct.

Moreover, words are marked as N (nouns), V (verbs) or X (all others) according to the results obtained in (Tamburini et al., 2002). We use $< >$ to mark Head-Argument relation and $\ll$ and $\gg$ to mark Head-Adjunct relation where the arrows point to the Head. From

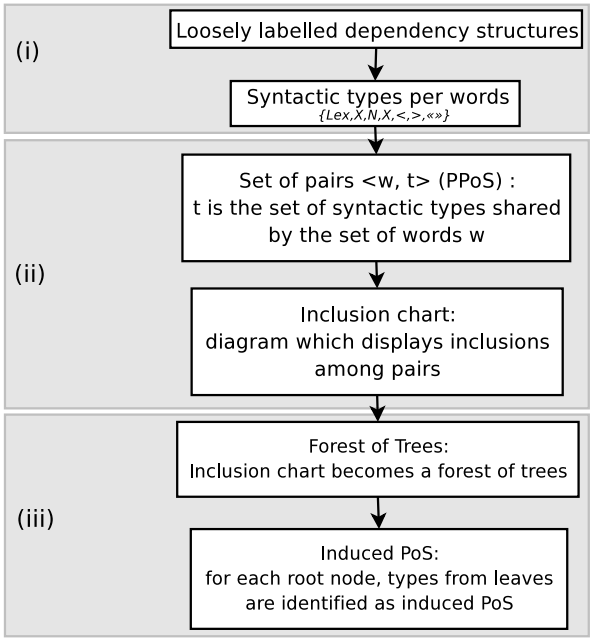


Figure 1: Algorithm Architecture

these dependency structures we extract syntactic type assignments by projecting dependency links onto formulas. Formulas are built out of $\{<, >, \ll, \gg, N, X, V, lex\}$ where the symbol lex stands for the word the formula has been assigned to. The formal details of the type resolution algorithm are provided below.

Type Resolution Let $W = \langle w_1, ..., w_n \rangle$ stand for an ordered sequence of words in a given sentence and let $w_j = \langle orth_j, bl_j, t_j \rangle$ stand for a word in the sentence, where $orth_j$, $bl_j \in \{N, V, X\}$ and t_j represent the orthographic transcription, the basic label and the type of the j-th word respectively. Let $E = \{\langle R, w_i, w_k \rangle\}$ be the set of edges where $R \in \{<, >, \ll, \gg\}$ is ordered by $|k - i|$ in ascending order. Given a dependency structure represented by means of W and E,

$-\forall w_j \in W, \quad t_j = lex$

$-$ foreach $\langle R, w_i, w_j \rangle \in E$

if $R = '>'$ $\quad \langle w_j, bl_j, t_j \rangle \rightsquigarrow \langle w_j, bl_j, bl_i > t_j \rangle$

if $R = '<'$ $\quad \langle w_i, bl_i, t_i \rangle \rightsquigarrow \langle w_i, bl_i, t_i < bl_j \rangle$

if $R = '\ll'$ $\quad \langle w_j, bl_j, t_j \rangle \rightsquigarrow \langle w_j, bl_j, bl_i \ll t_j \rangle$

if $R = '\gg'$ $\quad \langle w_i, bl_i, t_i \rangle \rightsquigarrow \langle w_i, bl_i, t_i \gg bl_j \rangle$

where the operator $\rightsquigarrow$ replaces the first item with the second in W.

For the sake of simplicity in Figure 2 for each word w_j only $orth_j$ and bl_j are displayed.

After applying the type resolution algorithm to all the given dependency structures, a lexicon is built with sets of types assigned to all words except nouns and verbs, which are discarded as

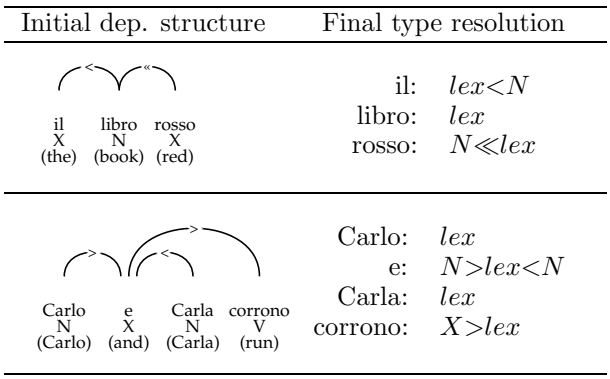


Figure 2: Type resolution example

they are not the subject of the present investigation.

For instance, the lexicon entry for the word "e" (*and*) is as below.

$$\text{e} : \begin{cases} X{>}lex{<}X \\ V{>}lex{<}V \\ N{>}lex{<}N \\ N{\ll}X{>}lex{<}X \\ V{\ll}X{>}lex{<}X \\ N{\ll}V{>}lex{<}V \\ N{>}lex{<}X \\ X{>}lex{<}N \\ X{>}lex{<}X \gg N \end{cases}$$

3.2 Inclusion chart

Lexicon entries are gathered together by connecting words which have received the same types. This results in a set of pairs $\langle W, T \rangle$ comprising a set of words W and their shared set of types T.

A consequence of this is that sets of words are composed of at least two occurrence words. In doing this we are assuming that a set of syntactic types represented by a single word does not have a linguistic significance.

Consider for example the following sample words with the corresponding types:

$$w_1 : \begin{cases} t_1 \\ t_2 \\ t_4 \end{cases} \quad w_2 : \begin{cases} t_1 \\ t_4 \end{cases} \quad w_3 : \begin{cases} t_3 \\ t_5 \end{cases} \quad w_4 : \begin{cases} t_1 \\ t_2 \\ t_3 \end{cases}$$

where $w_1, w_2, ..., w_n$, $n \in \mathbb{N}$ is the lexicon of our example, and t_i, $i \in \mathbb{N}$ stands for types. w_1 is connected both to w_4 and w_2 since they have $\{t_1, t_2\}$ and $\{t_1, t_4\}$ types in common respectively; furthermore, w_4 is connected both to w_2 and w_3 since they have $\{t_1\}$ and $\{t_3\}$ in common, as shown in Figure 3.

From the connection structure built as described above, we obtain the pairs $\langle W, T \rangle$ where W is the set of connected words and T is the

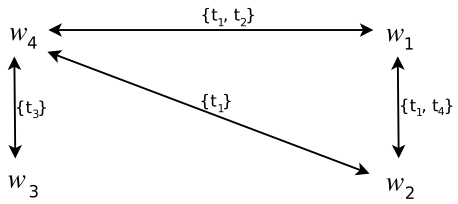


Figure 3: Example of connection structures

set of types carried by the corresponding connection arrow.

For instance, from the example in Figure 3 we obtain the following pairs:

$$\langle \{w_1, w_4\},\ \{t_1, t_2\} \rangle,$$
$$\langle \{w_1, w_2\},\ \{t_1, t_4\} \rangle,$$
$$\langle \{w_1, w_2, w_4\},\ \{t_1\} \rangle,$$
$$\langle \{w_3, w_4\},\ \{t_3\} \rangle$$

We will refer to each pair $\langle W, T \rangle$ as *Potential* PoS (PPoS).

From the given dependency structures we have obtained 215 pairs. They provide us with a first word class approximation with their associated syntactic behaviours.

In order to interpret the classification obtained and to further refine it, we first organize the pairs into an *Inclusion chart* based on subset relations among the PPoS and then we prune it as described below.

Our basic assumption is that type-set inclusions are due to syntactic similarities between words.

Definition 1 (Inclusion Chart) *The nodes of the Inclusion chart are pairs $\langle W, T \rangle$ where W and T are sets of words and sets of types respectively. Given two nodes $n_i = \langle W_i, T_i \rangle$ and $n_j = \langle W_j, T_j \rangle$ of the Inclusion chart, there is an inclusion relation between n_i and n_j, and we write $n_i \sqsubset n_j$, iff $W_i \supset W_j$ and $T_i \subset T_j$. Two nodes n_i, n_j of the Inclusions chart are **connected**, and we write $n_i \rightarrow n_j$, iff $n_i \sqsubset n_j$ and $\neg\exists\ n_k$ such that $n_i \sqsubset n_k$ and $n_k \sqsubset n_j$.*

To illustrate this, let us consider the lexicon entries "e" (*and*), "o" (*or*) and "p_com" (*comma separator*). The set of types assigned to "e" is shown above, those for "o" and "p_com" are as below.

$$\text{o} : \begin{cases} X{>}lex{<}X \\ X{>}lex{<}X{\gg}V \\ N{>}lex{<}N \\ V{>}lex{<}V \\ N{\ll}X{>}lex{<}X \\ N{\ll}N{>}lex{<}N \end{cases} \qquad \text{p\_com} : \begin{cases} X{>}lex{<}X \\ V{>}lex{<}V \\ N{>}lex{<}N \\ N{\ll}X{>}lex{<}X \\ N{>}lex{<}X \\ N{\ll}V{>}lex{<}V \\ N{>}lex{<}X \\ V{>}lex{<}X \end{cases}$$

The set of words

$$W_1 = \{\ \text{p\_com},\ \text{e},\ \text{o}\ \}$$

with the shared set of types

$$T_1 = \{V{>}lex{<}V, X{>}lex{<}X, N{>}lex{<}N, N{\ll}X{>}lex{<}X\}$$

constitute the pair $\langle W_1, T_1 \rangle$.

Once we have obtained the set of all pairs out of the lexicon entries, we build the *Inclusion chart*. Figure 4 shows a portion of this, which contains the pair $\langle W_1, T_1 \rangle$ discussed above.

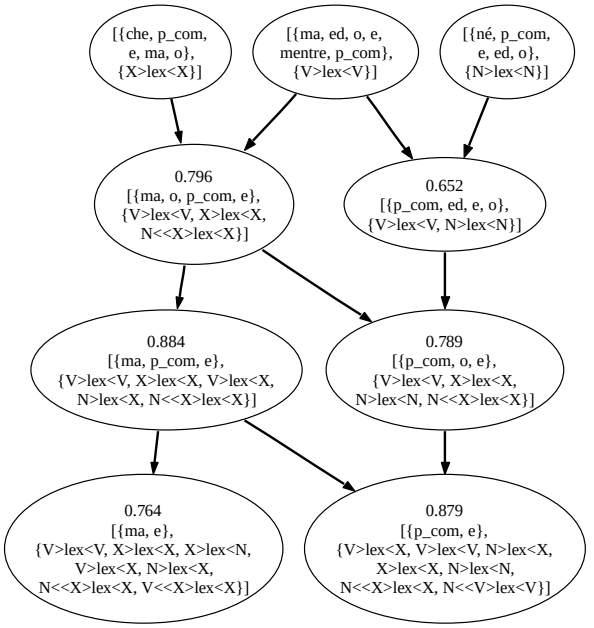


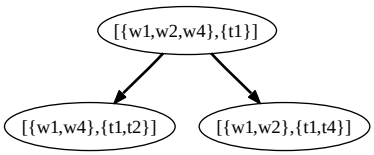
Figure 4: Example of *Inclusion chart*.

Since the *Inclusion chart* obtained displays all possible subset relations between all the pairs, it is rather complex and it conceals the linguistically relevant information we are actually looking for, namely the syntactic similarities between words which lead to their PoS classification.

It is our belief that by identifying the closest connections we can establish the correct PPoS links, i.e. induce a PoS hypothesis.

Consider the example at the beginning of this section, where $P_1 = \langle \{w_1, w_2, w_4\},\ \{t_1\} \rangle$

is included in $P_2 = \langle\{w_1, w_4\},\ \{t_1, t_2\}\rangle$ and $P_3 = \langle\{w_1, w_2\},\ \{t_1, t_4\}\rangle$. This means that both PPoS P_2 and P_3 increase PPoS P_1 by one syntactic type. The following *Inclusion chart* represents the connections between these pairs:



At this point, it is necessary to establish which is the better way to extend P_1, i.e. which of the two syntactic behaviours represented by t_2 and t_4 has to be selected to make the PPoS P_1 closer to a correct PoS.

In order to extract a suitable PoS classification from the *Inclusion chart*, this must be pruned by discarding less relevant nodes; hence, we need to introduce a relevance criterion.

3.3 Forest of Trees

The pruning phase is handled by means of a distance measure between PPoS which helps to highlight the closest pairs.

Before formally defining the distance measure and explaining its role in depth, we present the pruning algorithm.

Pruning Algorithm Let P be the set of all pairs of the *Inclusion chart* and let $e = \langle p_i, p_j, weight_j\rangle$ be an edge, where p_i is connected to p_j and $weight_j$ is a cohesion measure of p_j. For all $p_i \in P$ we indicate with E_{p_i} the set of all edges leaving p_i.

Given P:
$\forall p_i \in P$
  $\forall \langle p_i, p_j, weight_j\rangle \in E_{p_i}$
    if $weight_j$ differs from $\max_j\{weight_j\}$
      then remove $\langle p_i, p_j, weight_j\rangle$ from E_{p_i}

For each pair p_i only the edge connecting it to a pair p_j exhibiting the maximal cohesion measure is maintained.

Figure 5 shows the pruned portion of the *Inclusion chart* given in Figure 4. Notice that each node is weighted apart from the leaf node, because weighting leaves is not necessary for the algorithm proposed. The graph is then transformed into a *Forest of trees*.

We can now move on to explain how linguistically relevant similarities are automatically identified by means of the distance measure. First of all, we need to measure the relevance of a PPoS in terms of how representative its members are with respect to each other.

Definition 2 (Word Frequency)
Let Ω be the set of all words, Ψ the set of all types, and $o : \Omega \times \Psi \to \mathbb{N}$ the function which returns the number of occurrences of word per type. Let $\eta : \Omega \to \mathbb{N}$ be a function which returns the total number of occurrences of a given word.

*We call **word frequency** of $\langle W, T\rangle$ the function $F_{words} : \mathcal{P}(\Omega) \times \mathcal{P}(\Psi) \to \mathbb{N}$ defined as follows:*

$$F_{words}(\langle W, T\rangle) = \frac{1}{|W|} \cdot \sum_{i=1}^{k} \sum_{j=1}^{m} \frac{o(\langle w_i, t_j\rangle)}{\eta(w_i)}$$

where $W = \{w_1, w_2, ..., w_k\}$ is a set of words and $T = \{t_1, t_2, ...t_m\}$ is a set of types.

Definition 3 (Type Frequency)
Let $\xi : \Psi \to \mathbb{N}$ be a function which returns the total number of occurrences of a given type.

*We call **type frequency** of $\langle W, T\rangle$ the function $F_{type} : \mathcal{P}(\Omega) \times \mathcal{P}(\Psi) \to \mathbb{N}$ defined as follows:*

$$F_{types}(\langle W, T\rangle) = \frac{1}{|T|} \cdot \sum_{i=1}^{k} \sum_{j=1}^{m} \frac{o(\langle w_i, t_j\rangle)}{\xi(t_j)}$$

where W and T are as in Definition 2.

Given a pair $\langle W, T\rangle$, we evaluate the internal cohesion of its members as follows. The *word frequency* focuses on the similarity between words in W by rating how far words agree in their syntactic behaviour. Roughly, if the word frequency returns a high value for a pair then we can conclude that words within that pair have a close syntactic resemblance. On the other hand, the *type frequency* rates the similarity between types in T according to the number of times the words to which they have been assigned in the lexicon have shown that syntactic behavior in the dependency structures.

The evaluation of the pair $p_i = \langle W_i, T_i\rangle$ is given by the average of the two cohesion evaluations. We indicate this value by means of the symbol $\mathcal{C}_i$:

$$\mathcal{C}_i = \frac{F_{words}(\langle W_i, T_i\rangle) + F_{types}(\langle W_i, T_i\rangle)}{2}.$$

For each node of the example seen so far Figure 5 displays a weight which measures the cohesion of each node pair.

At first sight, C_1 may appear simplistic, with words and types being equally weighted. However other measures had been tried before C_1

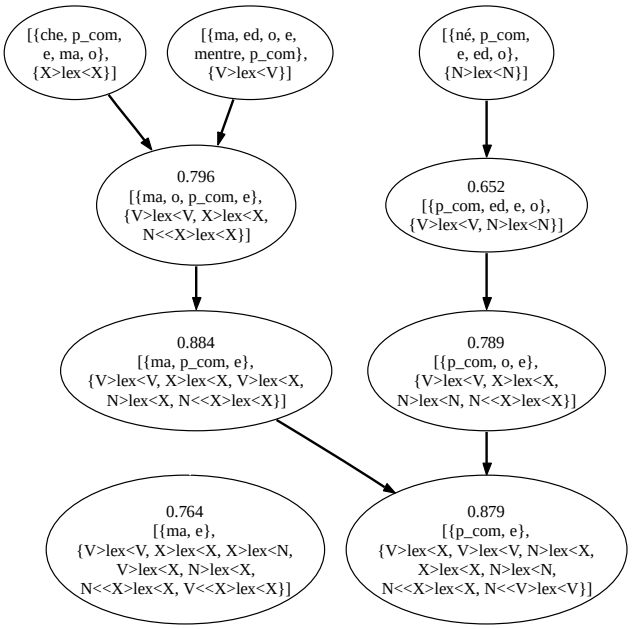


Figure 5: Example of *Forest of trees*.

was decided on, as giving the same importance to a set of words and a set of syntactic behaviours showed itself to be effective.

New kind of measures are currently being carried out. For instance, we are testing how the system works by varying the weight for each edge on the basis of the words added and the frequency with which they demonstrated the syntactic types of the augmented initial PPoS.

3.4 Induced PoS

Each tree in the Forest marks off complex groups of syntactic types. However, the same types occur in more than one tree, therefore we need to identify all and only those belonging to a given tree.

To this end, let us call **leaf nodes**[2] those PPoS with singleton type set not including any other; **root nodes**[3] PPoS not included by any other.

Leaves of each tree are grouped together; such groups constitute the whole type set partition. Clearly each group corresponds to a unique root node.

Syntactic types from leaf nodes encode few specialized syntactic patterns. We assume those patterns to be the syntactic core of a given tree, i.e. the relevant syntactic component of the corresponding PPoS root node.

Once a syntactic core is defined, the corresponding lexical core is automatically derived by identifying word sets showing exclusively sets of types belonging to that syntactic core.

Syntactic core extraction algorithm The following algorithm extracts syntactic cores from root nodes: for all type sets belonging to root nodes we identify the syntactic core as the subset of types obtained by the union of all type sets from the leaves of the corresponding tree. Given R, sets of root nodes:

$\forall \langle W_i, T_i \rangle = p_i \in R$
  $\forall t_k \in T_i$
    $N = \bigcup_j T_j$,where p_j leaf node of p_i tree
    if $t_k \in N$ then
      let $t_k \in T_i$ into the syntactic core

Consider the example proposed in Figure 5, which displays a portion of the Inclusion chart. Here we have the following two PPoS root nodes:

$$\langle \{\text{ma, e}\}, \{V{>}Lex{<}V, X{>}Lex{<}X, X{>}Lex{<}N, V{>}Lex{<}X, N{>}Lex{<}X, V{\ll}X{>}Lex{<}X, N{\ll}X{>}Lex{<}X\} \rangle,$$
$$\langle \{\text{p\_com, e}\}, \{V{>}Lex{<}X, V{>}Lex{<}V, N{>}Lex{<}X, X{>}Lex{<}X, N{>}Lex{<}N, N{\ll}X{>}Lex{<}X, N{\ll}V{>}Lex{<}V\} \rangle$$

The first root node has no leaf, being a root without branches, so it contains no syntactic core. On the other hand, the second has the following three leaves:

$$\langle \{\text{che, p\_com, e, ma, o}\}, \{X{>}Lex{<}X\} \rangle$$
$$\langle \{\text{ma, ed, o, e, mentre, p\_com}\}, \{V{>}Lex{<}V\} \rangle$$
$$\langle \{\text{nep\_apo, p\_com, e, ed, o}\}, \{N{>}Lex{<}N\} \rangle$$

Thus its type set contains the syntactic core

$$\{X{>}Lex{<}X, V{>}Lex{<}V, N{>}Lex{<}N\}$$

In order to associate it with its lexical core a visit to the tree rooted by this node is needed to collect those words $w \in W$ which show only types belonging to the syntactic core, for a given pair $\langle W, T \rangle$.

For example, the word "o" has shown $X{>}Lex{<}X$, $V{>}Lex{<}V$, $N{>}Lex{<}N$, but also $N{\ll}X{>}Lex{<}X$ which belongs to both root nodes so the word "o" cannot be part of the lexical entries the syntactic core is represented by.

The second root node is then associated with the lexical core consisting of { ed, mentre, né, che }. Hence the algorithm

[2] shown at the top of the tree in Figure 5

[3] shown at the bottom of the tree in Figure 5

concludes the existence of the following PoS prototype:

$$\langle \quad \{\text{ed, mentre, né, che}\}, \quad \{X{>}Lex{<}X, V{>}Lex{<}V, N{>}Lex{<}N\} \quad \rangle$$

Notice that this PoS corresponds to the *Coordinators* PoS depicted in Table 1, but here it is simpler because of the simplification of the *Inclusion chart* taken as an example.

The syntactic and lexical core is the output of our algorithm. We assume the core to be the syntactic (and lexical) prototype to be used for PoS classification.

4 Results and Evaluation

The proposed automatic method leads to the subdivision of the first level within the X class (see Section 2) as shown in Table 1.

The sets of automatically extracted syntactic types represent the prototypical syntactic behaviours of the corresponding words summarised by the explanatory PoS labels.

This classification is not fine-grained enough to be used by a tagger to reach an informative and useful annotation and should be intended as a first step through the empirical construction of a hierarchical tagset, e.g. following the parameters for taxonomic classification shown in (Kawata, 2005). Further analysis for each class must be carried out to increase the granularity of the tagset, for instance by exploiting morphological information.

The present study was carried out on a limited quantity of data; the sparseness of primary information we used to derive the proposed tagset might affect the conclusions we have drawn. The results will need to be checked with more data and with different treebanks to avoid biases introduced by the treebank used (TUT) from which the initial dependency structures were extracted.

Despite this, and the fact that further results of the third phase are currently being induced and remain to be investigated, it is promising that the 9 parts of speech induced in this second phase are not in marked contrast with traditional ones nor with widely accepted guidelines, such as (Monachini, 1995).

However, employing dependency structures as described in section 3.1, which means minimal syntactic information, leads to some ambiguities between word classes which may disagree with the linguist's intuitions.

From this point of view, the overlapping of determiners and prepositions within the same PoS is noteworthy. The lack of accuracy this classification results in is due, on the one hand, to the wide range of highly specific syntactic constructions involving determiners and prepositions that share the same loosely labelled dependency structures. Moreover, Italian monosyllabic (or 'proper') prepositions may be morphologically joined with the definite article (for example *di* ('of') + *il* ('the') = *del* ('of the')), performing sintactically both as a preposition and a determiner. Clearly this class will be further specialized by exploiting morphological information.

Polysyllabic (or 'not proper') prepositions, as opposed to monosyllabic ones, tend to occur in a lower number of syntactic patterns and, more crucially, cannot be fused with the article. In this case our system performs more accurately as it is able to correctly detect the syntactic similarities between such prepositions. As they typically tend to carry the function of the head (together with prepositional locutions) in verb-modifying structures they have been classified as 'Verb-Modifing Prepositionals' as shown in Table 1.

The 4 word classes grouping words commonly classified as adjectives and conjunctions may be considered an interesting result of the syntactically motivated induction algorithm presented here. As for adjectives[4] they have been divided into 2 separate classes depending on predicative or attributive distribution with respect to the noun they modify ('Left/Right Adjectivals' in Table 1). As far as conjunctions (and conjunctional locutions) are concerned, again, their syntactic patterning enforced a very clear split between 'Coordinators' and 'Subordinators'.

By contrast a relatively strong syntactic resemblance has been automatically recognised between words (and locutions) traditionally described as adverbs (and adverbial locutions): hence, the single 'Adverbials' word class is derived. Again, further anlysis exploiting distributional and morphological data may be useful in obtaining a finer-grained classification if necessary.

A final point to make is about copulative structures: our system proved not to prop-

[4]We refer to qualifying adjectives; other items traditionally classified as adjectives, for example 'determinative adjectives' as proposed by (Serianni, 1989), in our system are grouped together with determiners

PoS Label		Associated types	Prototypical words
Nouns		N	nuvola, finestra, tv
Verbs		V	stupire, raggiunto, concludendo, abbiamo
X	Prepositionals & Determiners	Lex<N, Lex<X, N≪Lex<N, N≪Lex<X, N≪Lex<V, X≪Lex<N, X≪Lex<V, X≪Lex<X	alcuna, della, dieci, diversi, le, molti, negli, numerose, quegli, questi, sei, sull'
	Verb-Modif. Prepositionals	V≪Lex<N, Lex<N≫V, V≪Lex<X, Lex<X≫V	a_causa_del, attraverso, contro, davanti_al, secondo, senza
	Left Adjectivals	Lex≫N	forti, giovane, grande, nuove, piccolo, suo,
	Right Adjectivals	N≪Lex, X≪Lex	economici, elettorale, idrica, importanti, positiva, ufficiale
	Adverbials	V≪Lex, Lex≫V, Lex≫X	allora, appena, decisamente, ieri, mai, molto, persino, rapidamente, presto, troppo
	Coordinators	V>Lex<V, N>Lex<N, X>Lex<X, N>Lex<X, X>Lex<N, V>Lex<X, V≪X>Lex<X, N≪V>Lex<V, N≪X>Lex<X	e, ed, ma, mentre, o, sia
	Subordinators	Lex<V, Lex<V≫V, V≪Lex<V	in_modo_da, oltre_a, quando, perché, se
	Relatives	N>Lex	che, cui, dove, quale
	Entities	Lex	ci, di_più, in_salvo, io, inferocito, noi, ti, sprovveduto, una

Table 1: Resulting PoS classification

erly process them in general, as shown by the fact that their predicative components ended up classified under either 'Entities' or 'Prepositionals & Determiners'.

5 Conclusions and Further Research

The final output of the three phase system will be a *hierarchy* of PoS tags. Such structured organization is expected to help the linguist during the annotation phase as well as when searching the annotated corpus.

On the one hand, the linguist can browse the graph for a given word to get a sense of its syntactic distribution or to improve the proposed classification (e.g. by splitting an induced category that is too coarse.)

On the other hand, since the resulted PoS classification is organized as a hierarchy with inclusion relations, a more intelligent search interface can be constructed to help the user extract the relevant information from the annotated corpus.

References

S. Bangalore and A. Joshi. 1999. Supertagging: An approach to Almost Parsing. *Computational Linguistics*, 25(2):237–265.

C. Bosco, V. Lombardo, Vassallo D., and Lesmo L. 2000. Building a treebank for Italian: a data-driven annotation schema. In *Proc. 2nd International Conference on Language Resources and Evaluation - LREC 2000*, pages 99–105, Athens.

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.

A. Clark. 2000. Inducing Syntactic Categories by Context Distribution Clustering. In *Proceedings of CoNLL-2000 and LLL-2000 Conference*, pages 94–91, Lisbon, Portugal.

Y. Kawata. 2005. *Tagsets for Morphosyntactic Corpus Annotation: the idea of a 'reference tagset' for Japanese*. Ph.D. thesis, University of Essex, Colchester, UK.

M. Monachini. 1995. ELM-IT: An Italian Incarnation of the EAGLES-TS. Definition of Lexicon Specification and Classification Guidelines. Technical report, Pisa.

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.

R. Rossini Favretti, F. Tamburini, and C. De Santis. 2002. CORIS/CODIS: A corpus of written Italian based on a defined and a dynamic model. In A. Wilson, P. Rayson, and T. McEnery, editors, *A Rainbow of Corpora: Corpus Linguistics and the Languages of the World*. Munich: Lincom-Europa.

H. Schütze. 1993. Part-of-speech induction from scratch. In *Proceedings of the 31st ACL*, pages 251–258, Columbus, Ohio.

L. Serianni. 1989. *Grammatica italiana. Italiano comune e lingua letteraria*. UTET, Torino.

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 Lang.