Chapter 7

Italian Lexical-Classes Definition Using Automatic Methods

F. Tamburini, C. Seidenari, A. Bolognesi, R. Bernardi

1. Introduction

Following the pioneering work of Bloomfield (1933) on distributional patterns and word distributional equivalencies, most of the early approaches within American structuralism to the problem of deriving an empirically founded PoS classification were based on Harris' (1951) distributional hypothesis that if two words are syntactically and semantically different, they will appear in different contexts.

Within such theoretical framework, Fries (1952) induced a system of parts of speech via distributional analysis, using as raw data a 'corpus' of some fifty hours of recorded conversations. To minimize preconceptions, Fries avoided traditional labels for parts of speech, and assigned meaningless letters and numbers to the word-classes. Although *The structure of English* raised many relevant criticisms especially against its systematisations of English grammar, Fries' work for the employment of lexical distribution to induce parts of speech has been revalued by a large number of scholars, and it seems to have inspired many of the recent natural language processing studies on that matter.

As a matter of fact, based on essentially the same distributional grounds, there is a number of recent studies 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; Clark 2000; Gobet and Pine 1997; Pereira, Tishby and Lee 1993; Redington, Chater and Finch 1998; Schütze 1993). These works examine the distributional behaviour of target words by comparing the lexical distribution of their respective collocates and by using quantitative measures of distributional similarity.

On a theoretical opposite perspective, recent studies in the field of language typology (Hagège 2004, Ramat 1999) attempting a viable 'epistemological' definition of categories seem to support a radically alternative position on parts of speech categorisation. Ramat in particular maintains that for an accurate functional definition of categories a formal approach (meaning a morpho-syntactic one) needs to be complemented by taking into account the "semantic aspects of categories", i.e. universal semantic functions, like predication and nomination, irrespective of their language-specific implementation. Such an approach, supporting a multi-criterion (both formal and semantic) categorisation, seems more in line with the traditional distinction of lexical classes.

The distributional approach on the one hand, and the traditional or 'multi-criterion' one on the other, as mutually excluding perspective on word-class categorisation, have been subject to opposed criticisms. Strictly distributional approaches have been criticised for being not enough delicate and underestimating the complexity of the lexicon. On the other hand, traditional approaches have been criticised for not resorting to a robust and consistent set of defining criteria.

On the field of natural-language-processing research, 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). Moreover, several research groups have worked on PoS annotation to develop syntactically annotated corpora (treebanks), such as VIT (Venice Italian Treebank) (Delmonte 2004) and TUT (Turin University Treebank) (Bosco 2003; Bosco, Lombardo, Vassallo and Lesmo 2000) or NLP commercial devices, e.g. the morpho-syntactic analyser by XEROX.

A first comparison of the set of PoS classes (Tagset) devised by these groups with EAGLES guidelines reveals that though there is general agreement on the main parts-of-speech to be adopted, considerable divergence exists about the criteria (semantic, morpho-syntactic, pragmatic . . . ?) for subdividing EAGLES' main classes into the PoS sub-classes actually employed for tagging purposes. For instance, VIT is quite fine-grained in specifying the noun class: as opposed to the straightforward EAGLES distinction into *proper* and *common*, VIT hypothesizes semantically motivated noun sub-classes, viz. *colour* (NC), *factive* (NF), *temporal* (NT), *human* (NH). Both VIT and XEROX use several singleton PoS sub-classes: VIT has word specific tags (*PD*, *PDA*) for prepositions 'di' and 'da' respectively; similarly, XEROX uses a *CONNCHE* tag for 'che' both when used as relative pronoun and as conjunction.

Moreover, at a further in-depth comparison, the systems under examination turned out to disagree about the lexical assignments for each PoS class, i.e. about which words need to be classified into which PoS classes. For example, a very frequent lexical form like 'molti' (many), was classified as *indefinite determiner* by EAGLES, as *indefinite adjective* by TUT, and as plural/indefinite *quantifier* by XEROX and VIT respectively. In VIT 'stesso', in its adjectival usage, is grouped with words like 'quello' (that) or 'questo' (this) within *demonstrative adjectives* (DIM). In XEROX, while 'quello' and 'questo' are tagged as *determiners* (DETSG, DETPL), 'stesso' is tagged as *adjective* (ADJSG, ADJPL) together with the large lexical class of qualifying adjectives.

Unfortunately the classification mismatches between the presented tagsets do not boil down to simply terminological differences resolvable by a mere one-to-one relabeling or by mapping different classes into a greater one. Such mismatches may easily influence the kind of conclusions one can draw from the annotated corpus.

Our purpose to automatically induce an empirically founded PoS classification is an attempt to make up for such inconvenient mismatches. The present work is founded basically on an empirical approach to word-class definition based on distributional information. However, the kind of contextual information employed for the present work is more structured and richer than the raw lexical/distributional data exploited in previous studies. Simple lexical co-occurrences in the immediate context will be replaced by distribution-sensitive dependency relations between words in sentences, defined along broadly accepted syntactic categorisations as *Head/Dependent* and

Argument/Adjunct.

In this sense, the kind of distribution-sensitive structural information we rely on could be said to represent a step forward from a pure distributional approach toward a semi-functional one.

2. Tagset induction

2.1. The lexical approach

Our first experiments on this field (described in detail in Tamburini, De Santis and Zamuner 2002) were based on the well-known Brill's algorithm (Brill and Marcus 1992). The main drawback of this as well as of similar approaches is the limited context of analysis. Lexical information is collected from a context of, for instance, ±3 words: such restricted word-span can conceal syntactic relations ranging over stretches of text longer than the context interval. Not surprisingly our results suffered from the same shortcomings.

Even so, three main uncontroversial lexical classes emerged from this broad range process: nouns (N), verbs (V) and an undifferentiated portion of the lexicon, which we referred to as X. This Noun-Verb empirical distinction is supported by widely accepted theoretic assumptions about most of the western languages including Italian.

In this contribution we will focus on such X class, suggesting a feasible methodology to further break down X by automatically grouping words that share similar syntactic behaviours. Our approach to solve this problem is to use basic syntactic relations together with distributional information to induce an empirically motivated set of lexical classes.

2.2. Adding Dependency Information

The algorithm we propose exploits loosely labeled dependency structures derived from a treebank. Such structures may be termed as "loosely labeled" in that they a) feature the N, V, X tags derived from the lexical clustering cited before; b) encode the basic and broadly accepted *Head/Dependent* relation; c) identify each dependent either as *Argument* or *Adjunct*.

Our structures are derived from TUT (Bosco 2003, Bosco et al. 2000).

The treebank currently includes about 1800 sentences organized in different sub-corpora from which we converted the dependency structures maintaining only the basic syntactic information outlined before. Words are marked as N (nouns), V (verbs) or X (all others). We use the symbols $<$ $>$ to mark Head-Argument relation and $\ll$ and $\gg$ to mark Head-Adjunct relation where the arrows point to the Head.

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

3. Algorithm Outline

The algorithm devised for the automatic induction of word classes consists essentially of three phases:

- STEP 1: each word is assigned the complete set of syntactic types extracted from the loosely labeled dependency structures collected in TUT;

- STEP 2: a first approximation of relevant word classes is obtained by grouping words that exhibit similar behaviours building a large structure defined as *inclusion graph*. This is obtained by creating sets of words showing the same type at least once in STEP 1, and by pairing these sets of words with their shared set of types.

- STEP 3: the inclusion graph obtained is *pruned* 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 lexical classes.

Let us examine each step in detail, showing all the algorithm operations.

3.1. STEP 1—Type Resolution

From the TUT 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.

Let $W = \langle w_1, ..., w_n \rangle$ stand for an ordered sequence of words in a given sentence and let $w_j = \left\langle orth_j, bl_j, t_j \right\rangle$ 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,

```
$-\forall w_j \in W, \;\; t_j = Lex$
$-$ foreach $\left\langle R, w_i, w_j \right\rangle \in E$
    if $R = '>'$   $\left\langle w_j, bl_j, t_j \right\rangle \rightsquigarrow \left\langle w_j, bl_j, bl_i > t_j \right\rangle$   $(\dagger)$
                   $\langle w_i, bl_i, t_i \rangle \rightsquigarrow \left\langle w_i, bl_i, t_i >^* bl_j \right\rangle$   $(\diamond)$
    if $R = '<'$   $\langle w_i, bl_i, t_i \rangle \rightsquigarrow \left\langle w_i, bl_i, t_i < bl_j \right\rangle$   $(\dagger)$
                   $\left\langle w_j, bl_j, t_j \right\rangle \rightsquigarrow \left\langle w_j, bl_j, bl_i <^* t_j \right\rangle$   $(\diamond)$
    if $R = '\ll'$   $\left\langle w_j, bl_j, t_j \right\rangle \rightsquigarrow \left\langle w_j, bl_j, bl_i \ll t_j \right\rangle$   $(\dagger)$
                   $\langle w_i, bl_i, t_i \rangle \rightsquigarrow \left\langle w_i, bl_i, t_i \ll^* bl_j \right\rangle$   $(\diamond)$
    if $R = '\gg'$   $\langle w_i, bl_i, t_i \rangle \rightsquigarrow \left\langle w_i, bl_i, t_i \gg bl_j \right\rangle$   $(\dagger)$
                   $\left\langle w_j, bl_j, t_j \right\rangle \rightsquigarrow \left\langle w_j, bl_j, bl_i \gg^* t_j \right\rangle$   $(\diamond)$
```

where the operator $\rightsquigarrow$ replaces the first item with the second in W. Each rule above is composed by two $\rightsquigarrow$ operations: if we apply the $(\dagger)$ ones we will obtain the 'nuclear types' only, while if we apply both $(\dagger)$ and $(\diamond)$ rules we will obtain what we call 'extended types'. Figure 7.1 shows a type resolution example for two simple dependency graphs outlining both nuclear and extended types.

The type resolution procedure, creating a set of word-type pairs, transforms the dependency treebank into a lexicon in which every word contained in the treebank exhibit all the syntactic types emerged from the type resolution process.

The algorithm proposed creates pairs of words and syntactic types, by

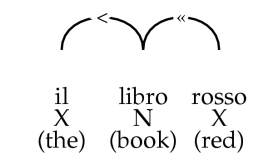
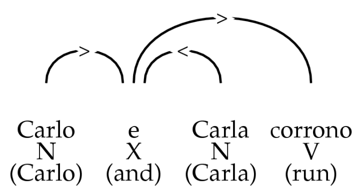
Initial dependency structure	Final type resolution	
	il:	$Lex<N$
		(-)
	libro:	Lex
		$(X<^{*}Lex\ll^{*}X)$
	rosso:	$N\ll Lex$
		(-)
 > > < Carlo N (Carlo) e X (and) Carla N (Carla) corrono V (run)	Carlo:	Lex
		$(Lex>^{*}X)$
	e:	$N>Lex<N$
		$(N>Lex<N>^{*}V)$
	Carla:	Lex
		$(X<^{*}Lex)$
	corrono:	$X>Lex$
		(-)

Figure 7.1: Type resolution examples. Nuclear types and extended types (in parenthesis).

means of the type resolution algorithm outlined above. A large number of localized syntactic descriptions per word are produced to identify differences in the syntactic behaviour. 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).

3.2. STEP 2—Inclusion Graph

The whole process of building up our lexical classes from scratch could be conveniently outlined by the following questions:

- Is it a viable solution trying to classify lexical entries looking at their respective syntactic behaviours (types)?
- Is there a way to group together in the same lexical class words having similar syntactic types?
- Can we do that automatically?

A strategy to answer affirmatively to the questions above could be to start building up *mechanically* all possible sets of words following this criterion: "create a new set every time two words exhibit at least one syntactic type in common". The naive idea behind this criterion is that if two words receive the same syntactic type in our treebank (in other words they behave syntactically in a similar way) then there is a chance that they should be clustered in the same lexical class.

To state our main background assumption the other way around: two similar lexical entries X and Y, that has to be assigned, presumably, to the same class, should exhibit the same types more frequently than two dissimilar lexical entries, say X and Z, that, on the contrary, should be assigned to different classes.

Therefore in this step lexicon entries are gathered together by connecting words which have received the same types in step 1. 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 7.2. 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 set of types carried by the corresponding connection arrow.

For instance, from the example in Figure 7.2 we obtain the following pairs:

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

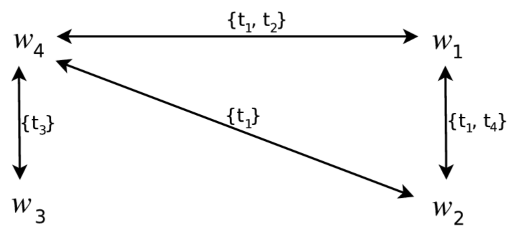


Figure 7.2: An example of connection structure.

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

From the given dependency structures 215 pairs are obtained providing us with a first word class approximation. Every class P includes a set of lexical entries plus the set of their syntactic behaviours, corresponding to the syntactic types exhibited by the lexical entries grouped into class P. In other words, lexical entries in P are grouped together according to how close they are from a syntactic point of view, i.e. on the basis of the syntactic types they share.

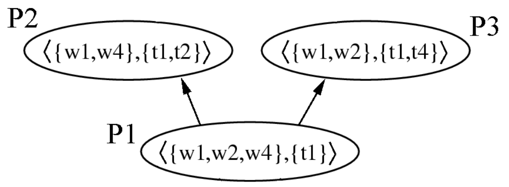
Given this very large set of PPoS, how can we identify which classes, or set of classes, are to be chosen to build up the lexical classes (PoS) we are trying to induce? A solution would be trying to find a way to connect every PPoS X to the ones that, from a syntactic point of view, are the most similar to X. Automatically, by exploiting the syntactic types associated to each PPoS, we can easily compute such similarity as a type set inclusion. We automatically build a network of PPoS, which we call *Inclusion Graph*, in such a way that inclusion relations between PPoS in term of syntactic behaviours are made explicit.

In this sense, each inclusions between PPoS represents a relation of syntactic similarity. Therefore the *Inclusion Graph* is a schematic representation of all possible syntactic-similarity links between all our PPoS. As a consequence of this the graph resulted is extremely intricate.

Let us illustrate it more formally.

7.0.1. Definition. [Inclusion Graph] The nodes of the graph are pairs $\langle W, T \rangle$. Given two nodes $n_i = \langle W_i, T_i \rangle$ and $n_j = \langle W_j, T_j \rangle$ there is an inclusion relation between n_i and n_j ($n_i \sqsubset n_j$) iff $W_i \supset W_j$ and $T_i \subset T_j$. Two nodes n_i, n_j are **connected** ($n_i \rightarrow n_j$) iff $n_i \sqsubset n_j$ and $\neg\exists\, n_k$ such that $n_i \sqsubset n_k$ and $n_k \sqsubset n_j$.

Consider the following example: $P_1 = \langle\{w_1, w_2, w_4\},\ \{t_1\}\rangle$ is included in $P_2 = \langle\{w_1, w_4\},\ \{t_1, t_2\}\rangle$ and $P_3 = \langle\{w_1, w_2\},\ \{t_1, t_4\}\rangle$. Both P_2 and P_3 increase PPoS P_1 by one syntactic type (see the picture below).



The inclusion graph displays *all possible inclusion relations* between all the pairs. It contains all the linguistically relevant information we are actually looking for, namely the syntactic similarities between words which lead to their PoS classification.

Figure 7.3 shows a simplified fragment of the inclusion graph extracted from our treebank.

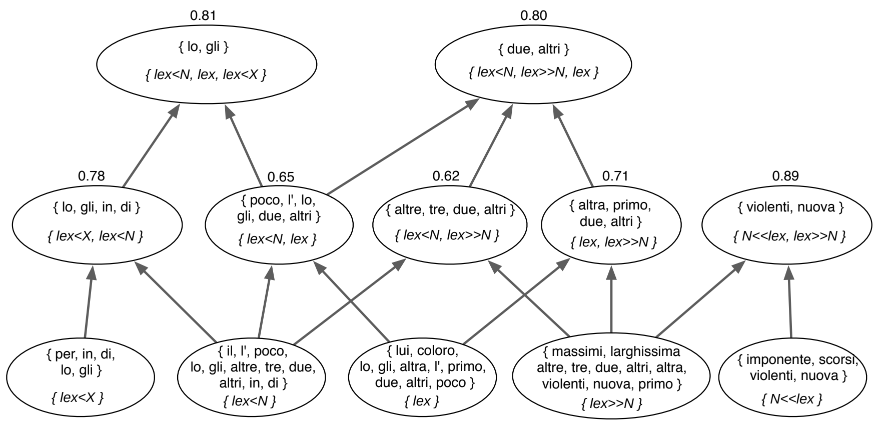

0.81
{ lo, gli }
{ lex<N, lex, lex<X }
0.80
{ due, altri }
{ lex<N, lex>>N, lex }
0.78
{ lo, gli, in, di }
{ lex<X, lex<N }
0.65
{ poco, l', lo, gli, due, altri }
{ lex<N, lex }
0.62
{ altre, tre, due, altri }
{ lex<N, lex>>N }
0.71
{ altra, primo, due, altri }
{ lex, lex>>N }
0.89
{ violenti, nuova }
{ N<<lex, lex>>N }
{ per, in, di, lo, gli }
{ lex<X }
{ il, l', poco, lo, gli, altre, tre, due, altri, in, di }
{ lex<N }
{ lui, coloro, lo, gli, altra, l', primo, due, altri, poco }
{ lex }
{ massimi, larghissima altre, tre, due, altri, altra, violenti, nuova, primo }
{ lex>>N }
{ imponente, scorsi, violenti, nuova }
{ N<<lex }

Figure 7.3: A simplified fragment of the Inclusion Graph.

3.3. STEP 3—Inclusion Graph Pruning

In order to extract a suitable PoS classification from the *Inclusion Graph*, this must be pruned by discarding less relevant nodes. Starting from such intricate Inclusion Graph, how can we identify for each class P all those classes that are most significantly similar, syntactically, to class P? In other words, for each class P we need to identify recursively, inside the set of classes that include P, which class Q is syntactically the most similar to P. What we obtain is a chain (where each PPoS could be described as a ring) in which every PPoS is connected only to the PPoS most similar to it. At this point, including all the 'rings' of the chain in the same lexical class (PoS) seems a plausible solution.

In general, the whole process of building a PoS class in our system means to isolate relatively simple word-sets exhibiting few, specialized and uniform syntactic behaviours (low part of the inclusion graph in Figure 7.3), and find a way to enlarge these word-set with those syntactically comparable but exhibiting many and less uniform syntactic behaviours.

Consider the example used before in section 3.2 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. Which is the best way to extend P_1 selecting one the two syntactic behaviours represented by t_2 and t_4? The idea is that if P_1 has to be extended, we have to choose the node that exhibits the maximal 'coherence' between the words that contains and the syntactic behaviours they have presented.

In order to evaluate the coherence of a class, we introduce a *cohesion measure* on Inclusion Graph nodes.

3.3.1. Cohesion Measure

Given a generic pair $\langle W, T\rangle$, we evaluate the internal cohesion of its members by introducing a cohesion measure based on the word and type frequency definitions.

7.0.2. Definition. [Word Frequency]
Let Ω be the set of all words, Ψ the set of all types. Let $o : \Omega \times \Psi \rightarrow \mathbb{N}$

returns the number of occurrences of word per type and $\eta : \Omega \rightarrow \mathbb{N}$ the total number of occurrences of a given word.

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

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

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

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.

7.0.3. Definition. [Type Frequency]
Let $\xi : \Psi \rightarrow \mathbb{N}$ returns the total number of occurrences of a given type. We call **type frequency** of $\langle W, T \rangle$ the function $F_{type} : \mathcal{P}(\Omega) \times \mathcal{P}(\Psi) \rightarrow \mathbb{R}$ 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, T, Ω and Ψ are the same as in Definition 7.0.2.

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:

$$weight_i = \frac{F_{words}(\langle W_i, T_i \rangle) + F_{types}(\langle W_i, T_i \rangle)}{2}$$

For each node in Figure 7.3, the weight measuring the cohesion of each node pair is showed above the nodes.

3.3.2. Pruning Algorithm

Let P be the set of all pairs of the *Inclusion Graph* 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 7.4 shows the pruned portion of the Inclusion Graph given in Figure 7.3. Notice that each node is weighted except the leaf nodes, because weighting leaves is not necessary for the algorithm proposed. The graph is then transformed into a *Forest of trees*.

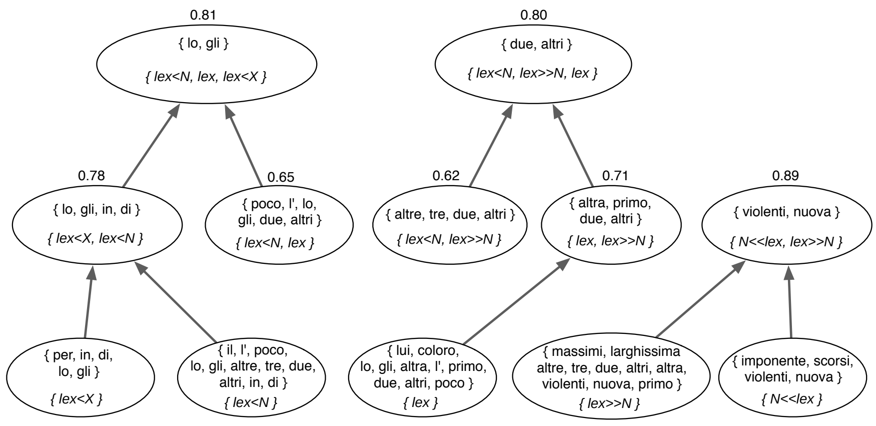

0.81
{ lo, gli }
{ lex<N, lex, lex<X }
0.80
{ due, altri }
{ lex<N, lex>>N, lex }
0.78
{ lo, gli, in, di }
{ lex<X, lex<N }
0.65
{ poco, l', lo, gli, due, altri }
{ lex<N, lex }
0.62
{ altre, tre, due, altri }
{ lex<N, lex>>N }
0.71
{ altra, primo, due, altri }
{ lex, lex>>N }
0.89
{ violenti, nuova }
{ N<<lex, lex>>N }
{ per, in, di, lo, gli }
{ lex<X }
{ il, l', poco, lo, gli, altre, tre, due, altri, in, di }
{ lex<N }
{ lui, coloro, lo, gli, altra, l', primo, due, altri, poco }
{ lex }
{ massimi, larghissima altre, tre, due, altri, altra, violenti, nuova, primo }
{ lex>>N }
{ imponente, scorsi, violenti, nuova }
{ N<<lex }

Figure 7.4: Pruned version of the simplified Inclusion Graph in Figure 7.3.

3.3.3. Core extraction method

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**[1] those PPoS with singleton type set not including any other and **root nodes**[2] the 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 specialised, prototypical syntactic patterns. We assume those patterns to be the syntactic core of a given tree, i.e. the representative syntactic component of the corresponding PPoS root node. Thus, starting from the root nodes of a tree, we identify its *core types* as the subset of types obtained by the union of all type sets from the leaves of that tree. Accordingly, we identify the corresponding *core words* as those lexical entries showing *exclusively* types belonging to the core types set.

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 simplified inclusion graph proposed in Figure 7.4. The third tree has the following two leaves:

⟨ {massimi, larghissima, altre, tre, due, altri, violenti, nuova, primo} , {$Lex \gg N$} ⟩
⟨ {imponente, scorsi, violenti, nuova} , {$N \ll Lex$} ⟩,

[1]Shown at the bottom of the trees in Figure 7.4.

[2]Shown at the top of the trees in Figure 7.4.

thus its *core types* are $\{Lex \gg N, N \ll Lex\}$, and the *core words* are

$$\{\text{massimi, larghissima, violenti, nuova, imponente, scorsi}\}.$$

The algorithm then produces a set of words and their shared syntactic types as outlined below:

$$\langle\{\text{massimi, larghissima, violenti, nuova, imponente, scorsi}\}, \{Lex \gg N, N \ll Lex\}\rangle$$

The set of associated core words and types is the final output of our algorithm. Core words and core types constitute respectively the sample word set and the syntactic outline supporting our PoS class hypothesis, which in the next section we refer to as PoS prototype.

4. The induced PoS Tagset for Italian

The tagset induction procedure has been configured as a multi-phase process. As a first phase, the algorithm described in the previous section has been applied to the lexicon derived from the entire dependency treebank with the aim of further subdividing the X class in more categories. Six categories emerged quite clearly, added to our original classes noun (N) and verb (V): adjectival (ADJ), adverbial (ADV), ENTITY, relative (REL), sentential adjunct subordinator (SUB), coordinator (COORD)[3]. This subdivision has been obtained by exploiting the nuclear types only. However, a substantial portion of our lexicon remained ill-classified.

Therefore, in a second phase, we removed from the lexicon all the word-type pairs consistently identified by the five categories just mentioned, and re-applied STEP 2 and 3 onto the remaining word-type pairs. This time, unlike in the first phase, we employed the extended instead of the nuclear types. Five more classes emerged as the output of the second phase (italicized in Table 7.1 below): argument subordinator (SUB_ARG), ARG, polysyllabic prepositional (PREP_POLI), noun adjunct prepositional (PREP_NA) and verb adjunct prepositional (PREP_VA)[4].

Table 7.1 shows the results obtained after this two-phase process.

[3]A detailed description of the induced classes will be provided in the next section.

[4]See footnote above.

Prototype		Syntactic outline	Lexical core
Noun		N	nuvola, finestra, tv
Verb		V	stupire, raggiunto, concludendo
X	Adverbial	V≪Lex, Lex≫V, Lex≫X	allora, appena, decisamente, ieri molto, persino, rapidamente, presto
	Adjectival	Lex≫N, N≪Lex, X≪Lex	economici, elettorale, forti, giovane idrica, importanti, nuove, piccolo, positiva suo, terzo, ufficiale, ultima, vicino
	Coordinator	V>Lex<V, N>Lex<N, X>Lex<X, N>Lex<X, X>Lex<N, V>Lex<X, X>Lex<V, V≪X>Lex<X, X>Lex<X≫V, N≪X>Lex<X	e, ed, ma, mentre, o, ovvero, oppure, sia
	Entity	Lex	ci, io, noi, ti
	Relative	N>Lex	che, cui, dove, quale
	Subordinator ADJ	Lex<V≫V, V≪Lex<V	affinché, come, dopo, mentre, perché
	Subordinator ARG	V<*Lex<V, X<*Lex<V, Lex<V>*X, V<*Lex<X, X<*Lex<X, N≪Lex<V, X≪Lex<V	a, che, da, di, per, se
	ARG	Lex<N, Lex<X>*V, X<*Lex<N, X<*Lex<N, V<*Lex<N, X<*Lex<V ...	*ARG Det*: alcuni, gli, il, l', le, questa, qualche, quattro, un *ARG Prep*: a, alla, con, da, della, di, nei
	Prepositional POLI	V≪Lex<X, Lex<X≫V, N≪Lex<X, Lex<X≫X	contro, dopo, durante, secondo, verso
	Prepositional NA	N≪Lex<N, X≪Lex<N, N<*Lex<N, N<*Lex<X	agli, degli, del, della, di, nei, nelle, sui
	Prepositional VA	V≪Lex<N, Lex<N≫V, V≪X≫*Lex<N	alla, all', con, dagli, nella, nell', sulle, tra

Table 7.1: The PoS classification automatically obtained from the two-phase class induction process.

The two-phase induction process yielded a set of word-classes devised as *lexico-syntactic prototypes*. As the consistent output of our induction algorithm, each PoS prototype comprises a *lexical core* (the 'core words' in Section 3.3.3), i.e. a sample word set, and a *syntactic outline* (the 'core types' again in Section 3.3.3), i.e. the set of syntactic patterns encoded as types linked to the word set. As stated above, core words and types are assumed to be prototypical samples (in that they are extracted as representative items) of a PoS class hypothesis: hence the head 'prototype' for the output of our algorithm.

A PoS prototype might be thought of as a two-sided entity built out of the interaction of two sets of related items. The functional side of the prototype (its syntactic outline) and the lexical side of it (the lexical core) were regarded to as mutually defining sets of objects: in view of this we consider their interaction as a PoS *class hypothesis* supported by our data.

As hypotheses automatically suggested by our system, we decided to regard them cautiously as work-in-progress entities capable of further analysis, more delicate classification[5] as well as post-processing improvement as explained in

[5]In principle, since the system is fully automatized, further processing may be carried out to refine the prototypes in order to obtain finer-grained distinctions, if desired.

the next section.

4.1. Analysis of the induced PoS

The two PoS classes induced following Brill's method (noun and verb) and two PoS prototypes, *adverbial* and *adjectival*[6], are easily comparable to the corresponding PoS classes proposed by EAGLES guidelines: *noun*, *verb*, *adverb*, *adjective*. Significant difference, as will be discussed below, arise with respect to *entity*, *rel*, *subord*, *coord*, *arg* and *prep*.

Following below is a brief description of the proposed PoS prototypes. For reasons of space the descriptions are indicative: by no means do the syntactic features reported exhaust the phenomena connected to the induced categories. Keeping in line with their lexico-syntactic composition, each prototype is explained resorting to 1) its leading syntactic arrangements, 2) a partial comparison (if possible) with traditional lexical classes and 3) some real-word examples from our working corpus.

ENTITY: ENTITY prototype encloses non-functional elements engaged in Argument relation with a verbal Head. Typical lexical items identified as ENTITIES[7] by our algorithm are pronominal expressions, for instance 'coloro' (those) in the following example

- ... tutti **coloro** che offrono aiuto ...

REL: RELATIVE prototype encompasses lexical items typically exploited as operators of relative adjunctions. As a matter of fact the induced prototype

[6]Our algorithm originally produced two separate classes of adjectivals according to their distribution with respect to the noun Head. Considering the sparseness of our starting data, such a distinction on a distributional basis seemed too sharp, so we decided to merge the two sets into a bigger one.

[7]The algorithm occasionally misclassified as ENTITIES also adjectives, when syntactically behaving as predicative components of copulative structures. As a matter of fact, due to our simplified notation system, such predicative components ended being represented as engaged in the same dependency relation with the verb. This misrepresentation would have been avoided had we imposed a-priori distinctions among verbs, which was not in line with the empirical proposal of the project.

resulted in a merging of two sets of lexical items, a) pronominals and b) adverbials, traditionally conceived as neatly divided from each other.

a) . . . ai terreni su **cui** esistevano . . .

b) . . . vicino all'università **dove** nel '90 scoppiò la rivolta . . .

COORD: COORD prototype includes items behaving as Head, syntactically specialized as operators bridging two or more structures and connecting them in a non-hierarchical fashion. Typical examples are straightforward coordinators such as 'e' (and), 'o' (or), 'ma' (but), etc.

SUB: SUBORDINATOR prototype includes expressions, syntactically behaving as Head, connecting mainly sentential and verbal structures in a hierarchical fashion. In fact the induction algorithm detected two different prototypes: subordinators typically performing as Head in sentential Adjunct (SUB_ADJ) for example 'quando' (when), 'perché' (because), and subordinators mainly performing as Head in sentential or verbal Argument structures e.g. 'di' (to), 'che' (that) as illustrated by the following examples:

a) . . . si applicano anche **quando** si tratta di togliere un ingombro . . .

b) . . . salvo che esigenze tecniche impongano **di** costruirlo . . .

c) . . . ammisero **che** si era trattato di un errore . . .

ARG: ARGUMENT OPERATOR prototype includes expressions syntactically performing as Head in Argument structures depending on, typically, a verbal or a prepositional Head. Therefore ARG encompasses lexical items distributionally and syntactically close to Italian articles like 'il', 'la' (the) 'un' (a), but including as well expressions morphologically different like 'mio' (my) and 'di' (of) when occurring in syntactic arrangements as shown in a) and b), respectively.

a) . . . l'unica volta che **mio** padre mi portò al cinema . . .

b) . . . si parla **di** 250-300 milioni di dollari . . .

As illustrated by the examples above, the induction of ARG prototype partially resulted in a fusion between word classes, such as determiners (articles) and prepositions, which instead are usually thought of as being neatly divided. Due to the fact that they statistically tend to overlap from a syntactical and distributional point of view, our algorithm ended up grouping them together.

As a consequence ARG prototype did not seem enough user oriented for the purpose of tagging a corpus like CORIS/CODIS, which is intended as a reference resource for Italian language. Splitting ARG prototype into two subsets following morphological criteria seemed a viable solution, both respecting the automatic output of the algorithm and the practical facet of the tagging. The two subclasses, including mainly determiners and prepositions respectively, were coded as ARG_Det and ARG_Prep.

PREP: PREPOSITIONAL prototype includes expressions which, governing noun, determiner and prepositional structures, yield mainly verb or noun adjuncts. As the label of the prototype may suggest, PREP encompasses lexical items distributionally close to prepositions like, for instance, 'attraverso' (through), 'secondo' (roughly: according to), 'con' (with), 'sul' (on + sing. masculine determiner), 'nel' (in + sing. masculine det., 'di' (of), 'degli' (of + pl. masculine det.), etc. These expressions, as explained in the previous section, having been found in different syntactic constructions as well, pertain also to ARG prototype.

Actually, our algorithm detected three different prepositional prototype: a) *PREP_POLI*[8] prototype, for lexical items (e.g. 'attraverso', 'secondo', 'contro' (against), etc.) chiefly governing a determiner or a prepositional structure and forming verb adjuncts; b) *PREP_NA* prototype, for expressions (e.g. 'del', 'degli' etc.) mainly governing bare nouns and yielding noun adjuncts; c) *PREP_VA* prototype, for items (e.g. 'nella' (in + sing. feminine determiner), 'sul' etc.) mostly governing a bare noun and forming verb adjuncts. The three prepositional patterns are exemplified below:

a) . . . protestare **contro** il Governo . . .

b) . . . proporzione **del** vantaggio . . .

[8]POLI stands for polysyllabic, as our algorithm proved to single out very clearly a set of expression traditionally identified within grammatic Italian tradition as polysyllabic preposition.

c) . . . provvedere **in** tempo . . .

Processing prepositional items proved to be a challenge for the algorithm. As a set of items typically involved in a wide range of highly specific syntactic constructions, prepositional instances were translated in our system into a huge mass of syntactic formulas (see Section 3.1). Not surprisingly, as a consequence of this, a relatively high number of PoS prototypes were induced by the algorithm (ARG_Prep, PREP_NA, PREP_VA, PREP_POLI) whose syntactic outlines encoded different and very specialized prepositional patterns. As a matter of fact, such fine-grained distinctions were not complemented on the lexical side with word sets as clearly defined as the type sets on the syntactic side. In other words, the three[9] prepositional syntactic outlines were not reliable enough as a clue to well defined and stable lexical sets.

Moreover, from a practical point of view, such lexically blurred prepositional prototypes would have dramatically increased the overall lexical ambiguity of our tagset. In practice, during a hypothetical tagging test, almost every preposition could have fit each of the three prototypes.

Hence the need was perceived for these PoS prototypes to be post-processed. Our aim, in line with the theoretical grounds of our work, was to identify more reliable lexical sets resting exclusively on the syntactic outlines induced by the algorithms. To obtain this, a statistical value was computed to represent the distribution of each preposition among the three prototypes. In practice, an overall frequency value was devised computing, for each prepositional item x, how many instances of x were to be traced back to the syntactic outlines represented in ARG_Prep, PREP_NA and PREP_VA respectively.

Then, in order to automatically derive statistically relevant lexical clusters, a standard hierarchical bottom-up clustering algorithm was applied to the whole set of prepositional items by processing the frequency value outlined above (see Figure 7.5). As a result three lexical clusters were detected, where prepositional items are clustered together with respect to their statistically dominant syntactic outline(s) (see Table 7.2).

In the next section we describe our first experiments for the evaluation

[9]i.e. ARG_Prep, PREP_NA, PREP_VA; the lexical core induced for PREP_POLI prototype turned out to be steady enough both for tagset induction and tagging purposes.

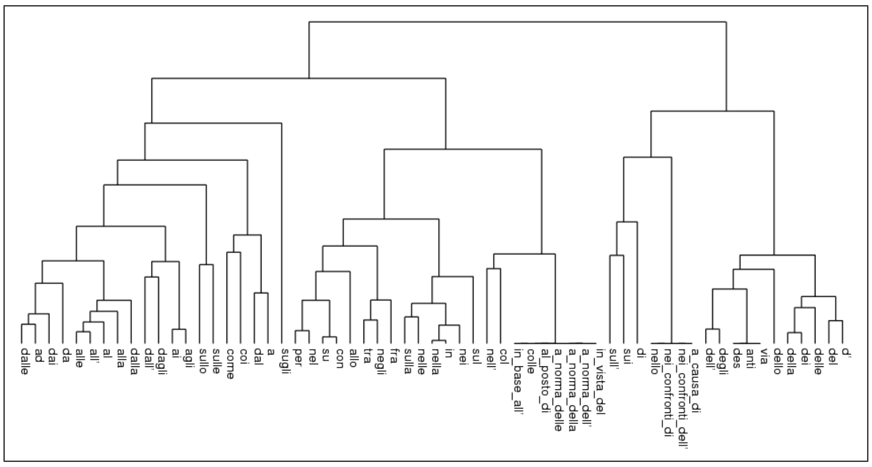


Figure 7.5: Clustering dendrogram of prepositional items.

Prepositional cluster	Lexical sample
PREP_VA dominant	con, col, fra, in, nelle, nei, per, su, sulla, sul, tra ...
PREP_NA dominant	di, degli, dell', dei ...
ARG_Prep – *PREP_VA* dominant	a, alle, al, da, dai, dall' ...

Table 7.2: Prepositional lexical clusters.

of the effects of the induced tagsets on the performances of automatic PoS-taggers.

5. Tagging experiments

In order to evaluate the effectiveness of the proposed PoS tagsets a number of experiments have been carried out. In particular we report on the large experimentation of the presented tagset inside an international evaluation campaign of NLP products for the Italian language EVALITA 2007[10].

This section briefly presents an evolution of CORISTagger (Tamburini

[10] http://evalita.itc.it/

2000), an high-performance PoS-tagger for Italian that paricipated to the evaluation campaign. The system is composed of an Hidden Markov Model tagger followed by a Transfomation Based tagger. The use of such a stacked structure paired with a powerful morphological analyser based on a 120.000-lemma lexicon, allowed the tagger to obtain very good performances in the EVALITA 2007 PoS-Tagging Task.

In EVALITA 2007 the tagset presented in the previous sections was one of the two tagsets considered for the evaluation. The other one is a traditional tagset similar to those proposed by the EAGLES project. We refer to the task guidelines (Tamburini and Seidenari 2007) for a complete description of the two tagsets as well as other task procedures and evaluation metrics.

The data sets provided by the organisation were composed of various documents belonging mainly to journalistic and narrative genres, with small sections containing academic and legal/administrative prose. Two separate data sets were provided: the Development Set (DS), composed of 133,756 tokens, was used for system development and for the training phase, while a Test Set (TS), composed of 17,313 tokens, was used as a gold standard for systems evaluation. The ratio between DS and TS is 8/1.

These data have been manually annotated assigning to each token its lexical category (PoS-tag) with respect to two different tagsets producing two different subtasks.

Table 7.3 shows the evaluation results for CORISTagger with respect to the two evaluation metrics. The performances are very high, both as absolute value when compared to the state-of-the-art tagging results for English and when compared to the other participants of EVALITA 2007 campaign.

Tagset	**TA**	**UWTA**
EAGLES-Like	97.59	92.16
DISTRIB	97.31	92.99

Table 7.3: CORISTagger results with respect to Tagging Accuracy (TA) and Unknown Words Tagging Accuracy (UWTA).

A careful evaluation of the tagging errors is showed in Table 7.4.

EAGLES-like		Proposed tagset	
86	ADJ - NN	96	ADJ - N
43	ADJ - V_PP	57	ADJ - V
37	CONJ_S - PRON_REL	41	N - V
17	NN - V_PP	33	REL - SUB_ARG
15	NN - NN_P	32	ADJ - ENTITIES
14	ADV - PRON_PER	28	ARG_DET - ENTITIES
14	ADJ_IND - PRON_IND	26	ADV - ENTITIES
11	NN_P - V_GVRB	21	ADJ - ARG_DET

Table 7.4: CORISTagger's most frequent errors in EVALITA 2007.

6. Conclusions

An automatically induced and syntactically motivated set of PoS classes for Italian has been presented as a complete tagset for automatic parts-of-speech annotation purposes. Little, if any, language-specific knowledge was employed in order to devise the final set of lexical classes: hence the method outlined is, in principle, *applicable to any language*.

Encouragingly, the 14 automatically induced PoS classes have not been found *in marked contrast* with traditional lexical classes nor with widely accepted guidelines such as EAGLES' ones. Interesting differences were nonetheless reported.

In general, for sections of the lexicon characterized by high morphological and lexical complexity, the proposed classification turned out to be under-specifying with respect to the received tagsets. *ARG_Det* alone, for instance, includes lexical items elsewhere classified as **Article**, **Determiner**, **Numeral**, **Quantifier**, **Indefinite Adjective**.

On the other hand, portions of the lexicon traditionally thought of as uniform parts-of-speech but involved in a wide range of highly specific syntactic constructions (this is the case, for example, of prepositions), produced a more detailed classification in our system.

References

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

Bloomfield, L. (1933). *Language.* New York: Henry Holt and Co.

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

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

Brill, E. and Marcus, M. (1992). "Tagging an unfamiliar text with minimal human supervision". In *Proc. of Fall Symposium on Probabilistic Approaches to Natural Language*, Cambridge, pp. 10–16.

Clark, A. (2000). "Inducing Syntactic Categories by Context Distribution Clustering". In *Proc. of CoNLL-2000 and LLL-2000 Conference*, Lisbon, pp. 91–94.

Delmonte, R. (2004). "Strutture sintattiche dall'analisi computazionale di corpora di italiano". In Cardinaletti, A. and Frasnedi, F. (Eds.), *Intorno all'italiano contemporaneo. Tra linguistica e didattica*, Milano: F. Angeli, pp. 187–220.

Fries, C. C. (1952). *The structure of English.* New York: Harcourt Brace.

Gobet, F. and Pine, J. (1997). "Modelling the acquisition of syntactic categories". In *Proc. of 19th Annual Meeting of the Cognitive Science Society*, pp. 265–270.

Hagège, C. (2004). "On categeries, rules and interfaces in linguistics". *The Linguistic Review*, 21, pp. 257–276.

Harris, Z. (1951). *Methods in Structural Linguistics*. Chicago: University of Chicago Press.

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

Pereira, F., Tishby, T. and Lee, L. (1993). "Distributional clustering of English words". In *Proc. of the 31st ACL*, Columbus, pp. 183–190.

Ramat, P. (1999). "Linguistic categories and linguists' categorizations". *Linguistics*, 37 (1), pp. 157–180.

Redington, M., Chater, N. and Finch, S. (1998). "Distributional Information: a Powerful Cue for Acquiring Syntactic Categories". *Cognitive Science*, 22 (4), pp. 425–469.

Schütze, H. (1993). "Part-of-speech induction from scratch". In *Proc. of the 31st ACL*, Columbus, pp. 251–258.

Tamburini, F. (2000). "Annotazione grammaticale e lemmatizzazione di corpora in italiano". In Rossini Favretti, R. (Ed.), *Linguistica e informatica: multimedialità, corpora e percorsi di apprendimento*, Roma: Bulzoni, pp. 57–73.

Tamburini, F., De Santis, C. and Zamuner, E. (2002). "Identifying phrasal connectives in Italian using quantitative methods". In Nuccorini, S. (Ed.), *Phrases and Phraseology—Data and Description*, Berlin: Peter Lang, pp. 45–64.

Tamburini, F. and Seidenari, C. (2007). "EVALITA 2007. The Italian Part-of-Speech Tagging Evaluation—Task Guidelines. http://evalita.itc.it/tasks/pos.html".