Enriching the Venice Italian Treebank with dependency and grammatical relations

Sara Tonelli (1), Rodolfo Delmonte (2), Antonella Bristot (2)

(1) FBK-IRST, via Sommarive 18, 38050 Povo (TN), Italy,
(2) Università Ca' Foscari, Ca' Bembo, Venezia, Italy
satonelli@fbk.eu
{delmont, bristot}@unive.it

Abstract

In this paper we propose a rule-based approach to extract dependency and grammatical relations from the *Venice Italian Treebank* (VIT) (Delmonte et al., 2007) with bracketed tree structure. To our knowledge, the only dependency annotated corpus for Italian available is the *Turin University Treebank* (Lesmo et al., 2002), which has 25,000 tokens and is about 1/10 of VIT. As manual corpus annotation is expensive and time-consuming, we decided to exploit an existing constituency-based treebank, the *VIT*, to derive dependency structures with lower effort. After describing the procedure to extract heads and dependents, based on a head percolation table for Italian, we introduce the rules adopted to add grammatical relation labels. To this purpose, we manually relabeled all non-canonical arguments, which are very frequent in Italian, then we automatically labeled the remaining complements or arguments following some syntactic restrictions based on the position of the constituents w.r.t to parent and sibling nodes. The final section of the paper describes evaluation results, carried out in two steps, one for dependency relations and one for grammatical roles. Since results are promising, we plan to use the dependency treebank to train a dependency-based parser and eventually a semantic role labelling system.

1. The source corpus

The starting point for our conversion is the Venice Italian Treebank (VIT), a treebank of written text created by the Laboratory of Computational Linguistics of University of Venice. The resource consists of 10,200 utterances with about 274,000 tokens and was syntactically annotated through a sequence of semi-automatic operations followed my manual validation. The first version of the treebank was created in the years 1985-88 and its rules were used to build a context-free parser for a speech synthesizer as described in (Delmonte and Dolci, 1989). VIT includes linguistic material of different nature extracted from five different types of text, i.e. news, bureaucratic genre, political genre, scientific genre and literary genre. The treebank has a bracketed tree structure with PoS and constituent labels, in the form:

f-[sn-[pron-noi], ibar-[vt-proponiamo], compt-[sn-[art-un, n-accordo, sp-[p-tra, sn-[n-gentiluomini]]]], punto-.]
We propose an agreement between gentlemen

The tagset comprises 102 PoS-labels and 31 constituent labels of three types: structural, functional and substantial. Structural constituents add structural information dependent on a previous head; for example, COMPT (Complement transitive) is used to indicate the presence of a (c-commanding) Transitive Verb in a verbal phrase and governs its complements and adjuncts. Functional constituents have a functional word as head, i.e. a preposition in a PP. Substantial constituents have a semantic word as head, e.g. a noun for an NP.
As shown below, having a rich inventory of PoS and constituents labels facilitates further conversion into dependency structure. For example, having a specialized node for tensed clauses, which is different from the one assigned to untensed ones, allows for better treatment of such constituents because it helps to detect some of its peculiar properties.

2. Conversion and information extraction process

2.1. From constituent-based structures to dependency relations

In the first step of our work, we induced dependency relations from the VIT following a rule-based procedure for head extraction. Unlike other conversion tasks like Lin's (Lin, 1995), our approach is based on a bottom-up algorithm to extract constituent heads and identify dependency relations. Besides, we applied Collins' procedure (Collins, 1999) to determine heads, following a head table which contains an entry for every non-terminal symbol in the grammar. During the conversion process, we basically followed three steps, namely *Sentence root identification*, *Head extraction* and *Dependency creation*.

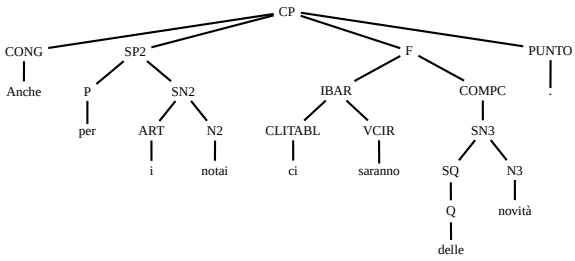
2.1.1. Sentence root identification

At first, we identify the main clause in every sentence and we extract its root following a small set of rules: for each sentence in the treebank we extract the main clause, then we look for the main verb, which usually corresponds to the tensed verb. If there is no tensed verb, we pick the first untensed verb, otherwise the head of the first NP - this latter case only applies to fragments. In case there are more coordinated verbs in the main sentence, we pick the first one.
A thorny problem for all dependency structure representations is coordination. In general, conjunction is a syntactic phenomenon that is treated differently in different theories, thus it has no generally accepted dependency structure. In our algorithm we decided that the phrase head should

be the coordinating conjunction, except for the sentence root.

2.1.2. Head extraction

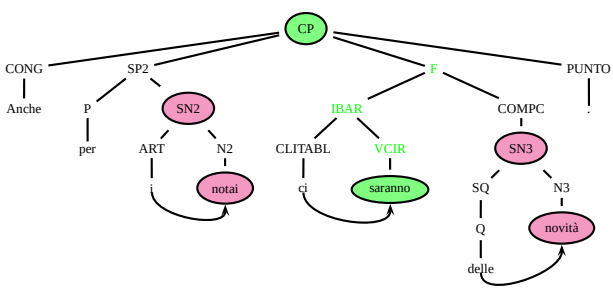
In the second step, we use a head table to identify bottom-up the heads of the local constituents, starting from the lexical head of every node directly dominating a list of terminals. Following Collins' model for English, the head table for Italian contains an entry for every non-terminal symbol in the grammar. In particular, *Direction* specifies whether search starts from the right or from the left end of the child list dominated by the node in the *Non-terminal* column. *Priority list* gives a priority ranking that decreases when moving down the list. An excerpt of the head table for Italian is given in Table 1.
Consider for example the following sentence:
cp-[cong-Anche, sp-[p-per, sn-[art-i, n-notai]], f-[ibar-[clitabl-ci, vcir-saranno], compc-[sn-[sq-[q-delle], n-novità]]], punto-.]
Also for solicitors there will be some novelties.

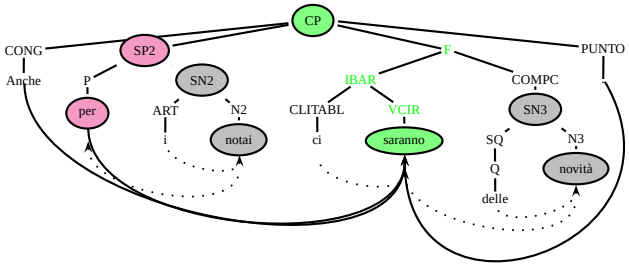


In the first step, we identify the tree root, i.e. the main verb *saranno (will be)*. Then we check for every terminal node if it is the head of the constituent it belongs to. If a node dominates one single terminal, the latter is taken as head regardless of its constituent/PoS label. Otherwise, we refer to the priority list. Differently from what some other treebankers have done, in case of a functional head like a conjunction (coordinate or subordinate), a relative pronoun or a complementizer, we treat the functional head as a governor and not as a dependent.

2.1.3. Dependency creation

For every node dominated by a head, we link all terminals to the latter. We proceed bottom-up and repeat iteratively the head identification step and the terminal connection until all terminals are linked to a head.





After conversion, the parenthesized version is mapped into a tabular structure, where every token is described through a word-id, a PoS and a constituent, as shown in Table 2. In the *Head* column, you can find the word-id of the token's head.

2.2. From Dependency relations to Grammatical functions

The final step in the overall conversion is the assignment of Grammatical Relation labels/roles to each head. While this kind of conversion is quite straightforward in languages like English, which imposes strict position for SUBJect NP and OBJect NP, in Italian, where constructions in non canonical positions are quite common, it is a problematic task. Beside marked constructions, which usually convey non thematic information, Italian also allows the omission of a SUBJect pronoun whenever it is a discourse topic, and has lexically empty non-semantic expletive SUBJects for impersonal constructions. This makes the automatic labelling of arguments and adjuncts a difficult task to achieve without any external additional (lexical) information. For this reason, we divided the assignment of grammatical functions into three steps. First, we manually listed all constituents in non canonical position, using different labels for preposed or postponed subjects and left dislocated complements. Secondly, we automatically labelled PP arguments in canonical position using a verb specialized lexicon with 17,000 verb entries. In this lexicon, each verb has been tagged with a specific subcategorization label and the list of prepositions in the verb valence, which allows to assign the OBL role to the prepositions heading an oblique constituent. A lexicon entry is in the form:

scegli: 2-ditr_prep_fra, 2-tr_prep_di
choose: 2-ditr_prep_between, 2-tr_prep_of

The first number describes the paradigmatic conjugation class of the verb, then the subcategorization type (transitive, ditransitive, etc.), then the preposition introducing the argument. As the example shows, a verb can belong to different subcategorization types, each having specific prepositions in its valence.
In the last step, we run a series of routines to assign a grammatical function to every head according to some syntactic restrictions. An excerpt of the assignment rules is displayed in Table 3. The first column contains the constituents whose head is the terminal word that should be assigned a function. The *Dependency* column lists the syntactic contraints ruling the assignment algorithm for the given constituent. The rules take into account the position of the constituent in relation to parent and sibling nodes. The third column shows the labels assigned if the constraints are fulfilled.

NON-TERMINAL	DIR.	PRIORITY LIST
AUXTOC	Right	AUSE, AUAG,AUEIR,AUSAI,VSUP
SN	Right	N,NPRO,NT,NH,NF,NP,NC,SECT,FW,RELQ,RELIN,RELOB,REL,PRON,PERCENTO INT,ABBR,NUM,DEIT,DATE,POSS,AGN,DOLL,SV2,F2,SA,COORD
SAVV	Left	PART,PARTD,AVVL,AVV,INT,REL,COORD,FW,NEG,F2
...	...	...

Table 1: Entries in the Head table

WORD-ID	TOKEN	POS	HEAD	CONSTITUENT
0	Anche	CONG (conjunction)	5	CP
1	per	P (preposition)	5	SP
2	i	ART (article)	3	SN
3	notai	N(noun)	1	SN
4	ci	CLITABL (clitic_pronoun_ablative/locative)	5	IBAR
5	saranno	VCIR (verb_copulative_mood_irrealis)	-	CL (main)
6	delle	Q (quantifier)	7	SQ
7	novità	N (noun)	5	SN
8	.	PUNTO (sentence_final)	5	CP

Table 2: Verticalized version of VIT Treebank with dependencies

The total number of labels for grammatical function is 24, including a.o. five types of adjuncts (normal, comparative, of manner, temporal and adverbial), direct and indirect objects, modifiers, arguments for passive verbs and four types of complements for copulative verbs (adjectival, nominal, prepositional and verbal).
After the assignment process, the verticalized version of the treebank with dependency relations is enriched with grammatical functions as shown in Table 4. Note that, in case of tokens which are not a lexical head, the constituent label is reported in place of the grammatical function.

3. Evaluation

3.1. Evaluation of dependency structures

In order to evaluate dependency, we created a gold standard with 500 sentences taken from all different types of text in the treebank, where heads and dependency relations are manually assigned.
Given the set of manually annotated sentences in the gold standard S_1 and the same sentences with automatically generated dependencies S_2, we took into account three different measures: precision of dependency arcs, precision of sentence roots and precision of dependency trees. The first is the ratio of all correct dependency arcs in S_2 to all dependency arcs in S_1, the second is the ratio of all correct sentence roots in S_2 to all sentence roots in S_1, while the third measure is the ratio of the sentences with every arc being correct in S_2 to the sentences in S_1.

Precision	
Dependency Arcs	97%
Dependency Trees	46%
Sentence roots	87%

Table 5: Dependency evaluation

The relatively low precision of dependency trees depends mainly upon the sentence length, wich is 50 words on the average. This means that one wrong dependency out of 50 is enough to dramatically drop this measure, including punctuation. Considering the single cases of wrong dependency, we noticed that there are no significant mistakes in the conversion algorithm, and that problematic cases often involve sentence fragments, where no verb is available. For instance, a part the treebank derives from legislative texts, and comprises sentences which introduce the object of a rule in the following way:

f3-[sn-[n-OGGETTO], sn-[n-agenti, spd-[partd-della, sn-[n-riscossione]], punto-.]].
OBJECT Revenue agents.

Although our conversion rule says that if no verb is available, the sentence head should be the first SN head, it does not apply to this sentence, because OBJECT only introduces the sentence topic, headed by *agenti*.

3.2. Evaluation of grammatical functions

As for grammatical relations, we concentrated on five main labels, i.e. SUBJECT, OBJECT (direct object for transitive verbs), ACOMP (adjectival complement for copulative verbs), OBLIQUE (arguments marking the semantic subcategorized preposition of ditransitive and intransitive verbs) and ADJUNCT. This time, we evaluated the whole treebank, not a gold standard. Results are reported in Table 6.

Evaluation of SUBJECT roles was carried out semiautomatically only on SUBJ labels in canonical position, since the others had been manually marked before conversion. First we checked agreement between (supposed) subjects and verbs, then we manually examined the cases of lack of agreement. In general, we noticed that SUBJ recognition task performs quite well and that only few errors may de-

CONSTITUENT	DEPENDENCY	GRAMM. FUNCTION
REL/RELQ/RELIN/RELOB	Parent: F2	BINDER
SV3	Parent: CP/F/FAC/FC/FS/FP/F3/F2/FINT/DIRSP	ADJ
	Parent: SN/SAVV/SQ	MOD
SAVV	Parent: COMP	ADJ
	Set between IBAR/IR_INFL and COMP	ADJT
	Simple AVV/IN placed inside IBAR/IR_INFL	ADJV
	Any other case	ADJM
...	...	...

Table 3: Example of Syntactic restrictions for Grammatical function identification

WORD-ID	TOKEN	POS	GR. FUNCTION	HEAD	CONSTITUENT
0	Anche	CONG	CONG	5	CP
1	per	P	ADJ	5	SP
2	i	ART	SN	3	SN
3	notai	N	POBJ	1	SN
4	ci	CLITABL	IBAR	5	IBAR
5	saranno	VCIR	IBAR	-	CL (main)
6	delle	Q	SQ	7	SQ
7	novità	N	S_FOC	5	SN
8	.	PUNTO	CP	5	CP

Table 4: Verticalized version of VIT Treebank with dependencies

Gr. funct.	Precision	Recall	F-measure
SUBJECT	0.99	0.96	0.97
OBJECT	0.98	0.99	0.98
ACOMP	0.96	0.97	0.96
OBLIQUE	0.93	0.63	0.75
ADJUNCT	0.93	0.99	0.96

Table 6: Evaluation of grammatical functions

pend upon mistakes in the original treebank, such as wrong constituent structures or missing dislocated labels. In most cases, wrong SUBJ assignment is due to bad head assignment. For instance, in the following sentence *Spoleto* was identified as head of *Spoleto festival orchestra* and bears the SUBJ role, whereas the correct head should be *orchestra*:

f-[sn-[art-la, np-Spoleto, sn-[n-festival], sn-[n-orchestra]], ir_infl-[aueir-sarà, vppt-diretta], comppas-[spda-[partda-dall_, sn-[sa-[ag-americano], mw-James, nh-Conlon, punto-.]]]]
The Spoleto festival orchestra will be directed by the American director James Conlon.

Having manually labeled all subjects in non-canonical position, we eliminated a potential source of errors, which corresponds to 32% of all subjects. A challenging task for the future would be to include in the conversion algorithm the detection of dislocated, topicalized and focalized subjects as well.

As for arguments/adjuncts, the most typical error is the exchange between the two labels, mostly depending on missing entries in the subcategorized verb lexicon we used for role assignment. Recall for obliques is significantly lower than that for adjuncts because the algorithm assigns by default an ADJ label to prepositional phrases depending on a verb, unless the verb and the preposition it heads are listed in the subcategorized verb lexicon.

F-measure for OBJECT and ACOMP, respectively 0.98 and 0.96, shows that the role extraction algorithm in this case performs quite well because the conversion step is very straightforward and relies on the verb subcategorization type (copulative vs. transitive verb) coded in the original treebank.

As shown by the evaluation data, the performance of our algorithm is in line with or above results given by similar procedures implemented for treebanks in other languages. (Gelbukh et al., 2005), for example, evaluated a transformation algorithm that maps constituency into dependency in the Spanish treebank Cast3LB. Although their gold standard only comprises 35 sentences, they infer that about 90% of dependency labels is correct. As reported in (Bohnet, 2003), also phrase structures in the NEGRA corpus were mapped to particular dependency structures called SSynt structures[1]. This experiment, which is quite similar to ours, achieved an overall accuracy of 74%. Our evaluation, though, takes into account different measures for dependency and grammatical function, while it does not consider lemmatization values, which on the contrary are computed in Bohnet's accuracy.

4. Conclusions and future work

In this paper, we described a rule-based approach for mapping phrase structures to dependency structures in the Venice Italian Treebank. This conversion task, which had been applied to treebanks in other languages such as En-

[1]SSynt structures (Surface Syntactic Structures) are dependency trees with nodes being labeled with the basic word form and edges being labeled with surface syntactic relations

glish and German, has proved to be suitable for Italian as well, despite some typical features of Italian which makes it more difficult to carry out automatic conversion without extra lexical information, such as subjects in non-canonical position, unexpressed subjects and dislocated constituents.
A direction for short-term investigation is to train and test a dependency based parser with memory based learning (i.e. Malt parser) on the dependency treebank. We could compare the results to those obtained by (Chanev, 2005) using Malt with the Turin University Treebank and to the performance of the same parser with other languages. Eventually, we could think of reducing the tagset and test the difference in parsing performance.
Secondly, we plan to use the dependency treebank to train a semantic role labelling system for Italian. In order to achieve satisfactory results, we need to complete the dependency treebank with all missing categories that are necessary to perform SRL, included empty subjects for untensed clauses and empty categories in relative clauses.

5. References

Bernd Bohnet. 2003. Mapping phrase structures to dependency structures in the case of (partially) free word order languages. In *Proceedings of the first international conference on Meaning-Text Theory*, Paris, France.

Atanas Chanev. 2005. Portability of dependency parsing algorithms. an application for italian. In *Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories (TLT2005)*, Barcelona, Spain.

M. Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

Rodolfo Delmonte and Roberto Dolci. 1989. Parsing italian with a context-free recognizer. *Annali di Ca' Foscari*, 28:123 – 161.

Rodolfo Delmonte, Antonella Bristot, and Sara Tonelli. 2007. VIT - Venice Italian Treebank: Syntactic and Quantitative features. In *Proceedings of the Sixth International Workshop on Treebanks and Linguistic Theories*, volume 1, pages 43–54.

Alexander Gelbukh, Hiram Calvo, and Sulema Torres. 2005. Transforming a constituency treebank into a dependency treebank. *Procesamiento del lenguaje natural*, (35):145 – 152.

L. Lesmo, V. Lombardo, and C. Bosco. 2002. Treebank development: the TUT approach. In *Proceedings of ICON*, Mumbasa, India.

Dekang Lin. 1995. A dependency-based method for evaluating broad-coverage parsers. In *IJCAI*, pages 1420–1427.