(Almost) Automatic Conversion of the Venice Italian Treebank into the Merged Italian Dependency Treebank Format

Linda Alfieri
FICLIT, University of Bologna, Italy
lindalfieri1988@gmail.com

Fabio Tamburini
FICLIT, University of Bologna, Italy
fabio.tamburini@unibo.it

Abstract

English. This paper describes the automatic procedure we developed to convert an Italian dependency treebank into a different format. We defined about 4,250 formal rules for rewriting dependencies and token tags as well as an algorithm for treebank rewriting able to avoid rule interference. At the end of this process a large portion of the whole treebank was automatically converted, with very few errors, leaving only a small amount of work to be done manually.

Italiano. *Questo contributo descrive la procedura automatica sviluppata per convertire un treebank italiano in un formato diverso. Abbiamo definito circa 4.250 regole formali di riscrittura per le strutture a dipendenza e i tag dei token e un algoritmo per la conversione del treebank in grado di evitare l'interferenza tra le regole. Al termine del processo una consistente sezione dell'intero treebank è stata automaticamente convertita, con un numero ridotto di errori, lasciando solo una piccola quantità di lavoro da svolgersi manualmente.*

1 Introduction

The availability of large annotated language resources is a prerequisite for the development of reliable automatic annotation tools using machine learning techniques.

Automatic tools able to enrich real texts with sentence syntactic structures are central instruments in Natural Language Processing (NLP) pipelines for a reliable annotation of text corpora. Modern NLP parsers heavily depend on complex training phases performed by examining manually annotated treebanks. Data sparsity, especially for low-resourced languages, seriously affect parsers performances, forcing scholars to annotate more and more data.

Since 2012 the state-of-the-art for Italian treebanks were not so satisfactory: three different projects and institutions produced three treebanks using different background theories, different formats and also different syntactic structures. They were the Italian Syntactic Semantic Treebank - ISST (Montemagni and Simi, 2007), the Turin University Treebank - TUT (Bosco et al., 2000) and the Venice Italian Treebank - VIT (Delmonte et al., 2007). Table 1 outlines the main characteristics of these treebanks at that time.

	ISST	**TUT**	**VIT**
Size (approx.)			
tokens	80,000	104,000	320,000
sentences	3,100	3,400	10,200
Type	Depend.	Depend.	Phr. Str.

Table 1: Italian treebanks in 2012.

ISST and TUT were used as gold standards in various evaluation campaigns (CoNLL2007 and EVALITA series), but only in 2012 the research groups developing such treebanks started to integrate them into a unique resource. In 2012 the Merged Italian Dependency Treebank - MIDT - was created and released by fusing the two resources (Bosco et al., 2012) and in the following years this project evolved such resource inserting it into the big Universal Dependency - UD - project (Nivre, 2015; Attardi et al., 2015), through another intermediate step, the Italian Stanford Dependency Treebank - ISDT (Bosco et al., 2013). During this process some other annotated texts were added to the treebank leveraging its size to around 315,000 tokens and 12,700 sentences (UD_Italian, v1.3).

This paper describes the latest effort for the Italian treebank merging: the conversion, harmonisation and integration of the written sections of VIT, not previously included into ISST, with the other two resources for reaching a global amount of about 600,000 tokens and 23,000 sentences syntactically annotated. For practical issues we decided to convert VIT into the MIDT format and then use the set of already designed automatic procedures and checking programs to transform it into the final UD format.

There are other notable works aimed at treebank conversion in various languages, for example we can cite (Bos et al., 2009) for Italian.

2 The Venice Italian Treebank

The Venice Italian Treebank was created by the Laboratory of Computational Linguistics of the Department of Language Sciences, University of Venice (Delmonte et al., 2007). The theoretical framework behind VIT syntactic representation is the X-bar theory, thus the early version of the treebank expresses syntactic information as trees.

At a later time, one of the authors converted the treebank from phrase-structure to dependency structures (Delmonte, 2009), but this was not distributed. This version of VIT was the starting point for the conversion described in this paper.

3 The Merged Italian Dependency Treebank

The Merged Italian Dependency Treebank was created as a first attempt to merge two existing Italian resources, namely the TUT and a special version of the ISST treebank named ISST-TANL (Bosco et al., 2012) and represents the starting point for all subsequent attempts to convert and harmonise this resource to different standards, first the Stanford Dependencies[1] and last the Universal Dependencies[2].

4 VIT Conversion

The main part of the VIT conversion process was completely automatic. Using the Semgrex package[3] (Chambers et al., 2007) from the StanfordNLP group, we set up a set of procedures that, starting from the definition of conversion rules, automatically converted the VIT into the MIDT format. This procedure has been developed specifically for our conversion problem, but can be used, in principle, to convert any dependency treebank represented using the CoNLL format in a different format that does not require re-tokenisation steps.

4.1 The Semgrex language

Semgrex represents nodes in a dependency graph as a (non-recursive) attribute-value matrix. It then uses regular expressions for subsets of attribute values. For example, `{word:amo;tag:/N.*/}` refers to any node that has a value 'amo' for the attribute 'word' and a 'tag' starting with 'N', while '{}' refers to any node in the graph. The most important part of Semgrex is that it allows you to specify relations between nodes or group of nodes. For example, '`{}=1 <subj {}=2`' finds all the pairs of nodes connected by a directed 'subj' relation. Logical connectives can be used to form more complex patterns and node naming (the '=' assignments) can help retrieve matched nodes from the patterns.

Unfortunately Semgrex is simply a query language and, in its original form, cannot be used to rewrite dependency (sub)graphs. In order to extend the possibility of Semgrex, we then modified the original application to manage pairs of patterns: the first is used to search into the treebank for the required subgraphs, and the second is used to specify how the retrieved subsgraphs have to be rewritten. For example the pattern pair `{tag:det}=1 >arg {tag:noun}=2 --> {tag:ART}=1 <DET {tag:NN}=2`, what we called a 'Semgrex rule', changes the direction of the dependency and, at the same time, changes the words tags and relation label. The starting and final patterns have to contain the same number of nodes and dependency edges. Node naming has been the fundamental trick to introduce such extension allowing for node matching between patterns.

4.2 Conversion Procedure

For converting VIT into MIDT format, we manually defined about 4,050 Semgrex rules each capturing a specific syntactic configuration in VIT and transforming it into the MIDT schema and about 150 rules for residual tag rewriting. We spent about six months for writing the entire set of rules.

We have defined a set of new rewriting operations on a general dependency treebank:

- DEL_REL(graphID, depID, headID): deletes

[1]http://nlp.stanford.edu/software/stanford-dependencies.shtml

[2]http://universaldependencies.org/

[3]http://nlp.stanford.edu/software/tregex.shtml

a dependency edge between two graph nodes;

- INS_REL(graphID, depID, headID, label): inserts a new labelled dependency edge between two graph nodes;
- REN_TAG(graphID, nodeID, tag): replace the tag of a specific graph node.

The conversion task has been implemented as a three-steps process:

- first of all, each Semgrex rule is always applied to the original treebank producing a set of matching subgraphs that have to be rewritten;
- for each match, a set of specific operations for rewriting the subgraph corresponding to the processed matching are generated and stored;
- last, the whole set of operations produced processing the entire set of Semgrex rules, each applied to the original treebank, is sorted by graphID, duplicates are removed and every operation is applied graph by graph respecting the following order: first dependency deletions, second dependency insertions and lastly tag renaming.

This way of processing the original treebank and transforming it into the new format should guarantee that we do not experience rule interference during the conversion, because the generation of the rewriting operations due to the Semgrex rules application is decoupled from the real treebank rewriting.

5 Some Linguistic Issues

The set of rules manually written for converting VIT dependency structures can be subdivided into two macro-classes: (a) rules that do not modify the structures and (b) rules that need to modify the dependencies, both in term of edge direction and in term of different structuring between the involved nodes.

Regarding the rules that do not modify the dependency structures, they simply rename the dependency label using a 1:1 or an N:1 look-up table, as VIT, with respect to MIDT, typically involves more specific dependency types. Figure 1 outlines some simple examples of such kind of conversions.

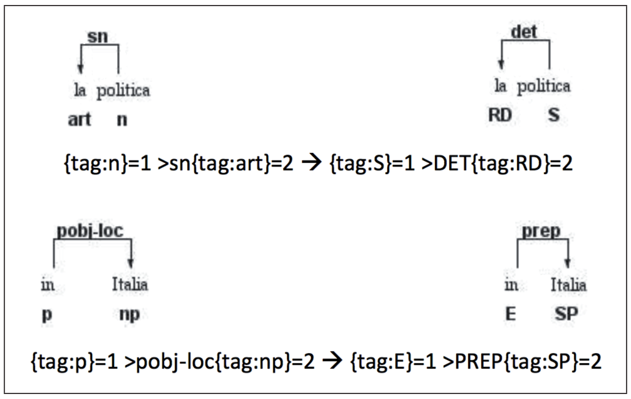
sn
la politica
art n
det
la politica
RD S
{tag:n}=1 >sn{tag:art}=2 → {tag:S}=1 >DET{tag:RD}=2
pobj-loc
in Italia
p np
prep
in Italia
E SP
{tag:p}=1 >pobj-loc{tag:np}=2 → {tag:E}=1 >PREP{tag:SP}=2

Figure 1: Some simple examples of rules that do not modify the dependency structures.

There are, of course, other kind of operations on subgraphs that require also the rewriting of the dependency structure. A good example concerns relative clauses in which the role of the relative pronoun and, as a consequence, the connections of the edge expressing the noun modification are completely different in the two formalisms. Figure 2 shows one example of this kind of rewriting.

Cases of coordination presented several problems: in VIT the head of the coordinated structure is linked to the connective and then the two (or possibly more) coordinated structures can be linked with a wide range of different dependency types (e.g. between phrases - *sn*, *sa*, *savv*, *sq*, *sp*, predicative complements - *acomp*, *ncomp*, adjuncts - *adj*, *adjt*, *adjm*, *adjv*, subjects - *subj*, objects - *obj*, etc.) leading to a large number of different combinations. Moreover, each dependency combination has to be further specified by the different token tags. MIDT represents coordinate structures in a different way: the connective and the second conjunct are both linked to the first conjunct that is connected to the head of the coordinated structure.

Figure 3 shows one example: the first formal rule represents an abstract rule pattern that has to be filled with all the real tag combinations found in VIT, generating a huge number of different rules, one of them outlined by the second complete formal rule. This process generated more than 2,800 different rules for handling all the coordinated structures in VIT.

There is also a need for a third kind of rules for rewriting single PoS-tags that might have remained unchanged during the main conversion process.

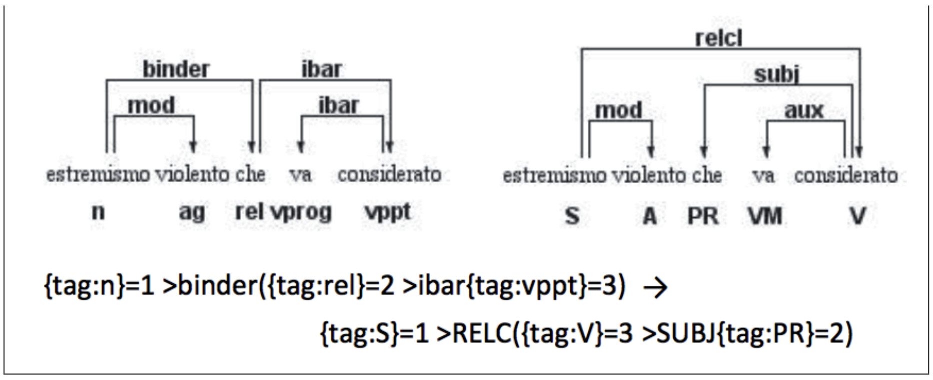
binder
ibar
mod
ibar
estremismo violento che va considerato
n ag rel vprog vppt
relcl
subj
mod
aux
estremismo violento che va considerato
S A PR VM V
{tag:n}=1 >binder({tag:rel}=2 >ibar{tag:vppt}=3) →
{tag:S}=1 >RELC({tag:V}=3 >SUBJ{tag:PR}=2)

Figure 2: An example of rule that rewrite the dependency structure.

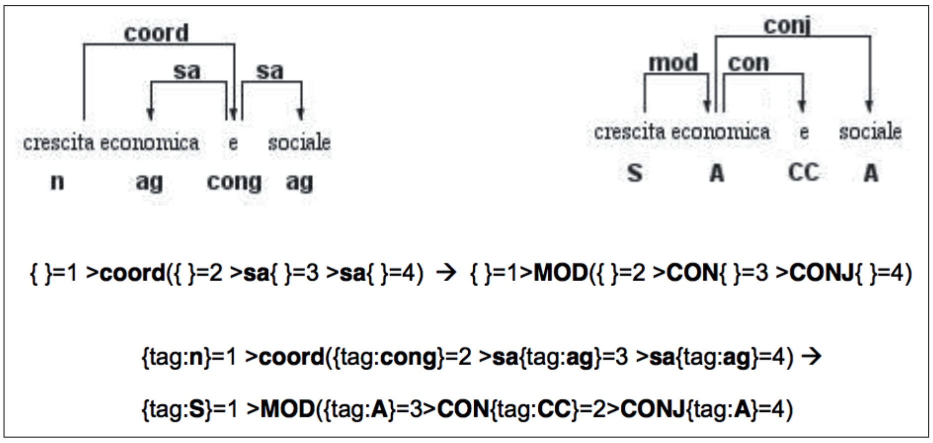
coord
sa
sa
crescita economica e sociale
n ag cong ag
conj
mod
con
crescita economica e sociale
S A CC A
{ }=1 >coord({ }=2 >sa{ }=3 >sa{ }=4) → { }=1>MOD({ }=2 >CON{ }=3 >CONJ{ }=4)
{tag:n}=1 >coord({tag:cong}=2 >sa{tag:ag}=3 >sa{tag:ag}=4) →
{tag:S}=1 >MOD({tag:A}=3>CON{tag:CC}=2>CONJ{tag:A}=4)

Figure 3: An example of coordination structures in VIT and MIDT and the conversion rule.

One further point deserves some discussion. In VIT, articulated prepositions are represented as two different tokens both linked with a common head: the preposition is tagged *part/partd/partda* and usually connected to the head with some kind of modification relations and the article is always tagged *art* and linked to the head with a *det* relation. In MIDT articulated prepositions are represented by a single token. As we said before, our process does not allow re-tokenisation rules. Given that MIDT is only an intermediate format and the goal is to convert VIT into the UD standard that requires two tokens for this phenomenon, we decided to avoid any re-tokenisation and to convert such structures linking the preposition to the head and the article to the preposition by introducing a new, dummy, relation label 'REL_EA'.

6 Evaluation

Applying all the 4,250 Semgrex rules, we obtained a converted treebank in which 228,534 out of 280,641 dependency relation were automatically converted, giving a global coverage of 81.4%.

To test the effectiveness of the conversion procedure and the conversion rules we randomly selected 100 sentences (2582 dependency relations to be converted) from the treebank and manually checked every newly created dependency relation, both in term of the connected nodes and the assigned label.

We obtained the following results: among the 2008 relations that have been automatically converted we found 125 wrongly converted dependency relations. So, on this sample, we obtained a coverage of 2008/2582 = 77.8%, slightly less than on the whole treebank, with a conversion error rate = 125/2008 = 6.2%.

7 Discussion and Conclusions

This paper presents the procedure we developed to convert VIT, one Italian treebank, into a different format. Most of the described conversion procedure rely on an automatic algorithm based on formal rules that is able to automatically convert the 81.4% of the treebank. This procedure can be, in principle, adaptable to any conversion between different dependency treebank formats.

The formal rules has been manually defined by using a well known dependency search procedure, Semgrex from StanfordNLP group, properly extended to handle rewriting rules and the final result was manually evaluated to test the effectiveness of the written rules obtaining a very small error rate.

To the best of our knowledge, there is no general purpose tool available to automatise this task for dependency graphs. We can find some powerful converters in literature but they are usually tied to specific pair of tagsets (often tailored to the Penn treebank) (Johansson and Nugues, 2007; Choi and Palmer, 2010), and cannot be easily adapted to general needs, or are devoted to tree manipulation, for example the tool 'Tregex' (Levy and Andrew, 2006).

Even if the described procedure can convert a large part of the treebank automatically with a very small quantity of errors, the conversion certainly needs a careful manual analysis to complete the task and check the new treebank for remaining mistakes. The VIT treebank contains a lot of specific and peculiar dependency subgraph for representing phenomena in a very detailed way. Trying to capture all these different variations into formal rules can result in a very large rule set mostly composed of rule that handle single cases. We stopped the production of new rules when this situation arose.

Acknowledgments

We wish to thank Rodolfo Delmonte and Maria Simi for their precious suggestions and explanations for the analysis of linguistic phenomena and for defining the conversion process.

References

Giuseppe Attardi, Simone Saletti, and Maria Simi. 2015. Converting Italian Treebanks: Towards an Italian Stanford Dependency Treebank. In *Proc. of 2nd Italian Conference on Computational Linguistics - CLiC-it 2015*, pages 25–30, Trento.

Johan Bos, Cristina Bosco, and Alessandro Mazzei. 2009. Converting a dependency treebank to a categorial grammar treebank for Italian. In *Proc. of 8th International Workshop on Treebanks and Linguistic Theories - TLT8*, Milano.

Cristina Bosco, Vincenzo Lombardo, Daniela Vassallo, and Leonardo Lesmo. 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.

Cristina Bosco, Simonetta Montemagni, and Maria Simi. 2012. Harmonization and Merging of two Italian Dependency Treebanks. In *Proc. of LREC 2012, Workshop on Language Resource Merging*, pages 23–30, Istanbul.

Cristina Bosco, Simonetta Montemagni, and Maria Simi. 2013. Converting Italian Treebanks: Towards an Italian Stanford Dependency Treebank. In *Proc. of ACL Linguistic Annotation Workshop & Interoperability with Discourse*, Sofia.

Nathanael Chambers, Daniel Cer, Trond Grenager, David Hall, Chloe Kiddon, Bill MacCartney, Marie-Catherine de Marneffe, Daniel Ramage, Eric Yeh, and Christopher Manning. 2007. Learning Alignments and Leveraging Natural Logic. In *Proc. of the Workshop on Textual Entailment and Paraphrasing*, pages 165–170.

Jinho Choi and Martha Palmer. 2010. Robust Constituent-to-Dependency Conversion for English. In *Proc. of 9th International Workshop on Treebanks and Linguistic Theories - TLT9*, Tartu, Estonia.

Rodolfo Delmonte, Antonella Bristot, and Sara Tonelli. 2007. VIT - Venice Italian Treebank: Syntactic and Quantitative Features. In *Proc. Sixth International Workshop on Treebanks and Linguistic Theories*.

Rodolfo Delmonte. 2009. Treebanking in VIT: from Phrase Structure to Dependency Representation. In Sergei Nirenburg, editor, *Language Engineering for Lesser-Studied Languages*, pages 51–81. IOS Press, Amsterdam, The Netherlands.

Richard Johansson and Pierre Nugues. 2007. Extended Constituent-to-dependency Conversion for English. In *Proc. of NODALIDA 2007*, Tartu, Estonia.

Roger Levy and Galen Andrew. 2006. Tregex and Tsurgeon: tools for querying and manipulating tree data structures. In *Proc. of 5th International Conference on Language Resources and Evaluation - LREC 2006*, Genoa, Italy.

Simonetta Montemagni and Maria Simi. 2007. The italian dependency annotated corpus developed for the conll-2007 shared task. Tech. report, ILC-CNR.

Joakim Nivre. 2015. Towards a Universal Grammar for Natural Language Processing. In *Proc. of 16th International Conference Computational Linguistics and Intelligent Text Processing - CICLing 2015*, pages 3–16, Cairo, Egypt.