Less is More? Towards a Reduced Inventory of Categories for Training a Parser for the Italian Stanford Dependencies

Maria Simi[1], Cristina Bosco[2], Simonetta Montemagni[3]

[1] Dipartimento di Informatica, Università di Pisa, Italy

[2] Dipartimento di Informatica, Università di Torino, Italy

[3] Istituto di Linguistica Computazionale "Antonio Zampolli" (ILC–CNR), Italy

E-mail: simi@unipi.it, bosco@di.unito.it, simonetta.montemagni@ilc.cnr.it

Abstract

Stanford Dependencies (SD) represent nowadays a *de facto* standard as far as dependency annotation is concerned. The goal of this paper is to explore pros and cons of different strategies for generating SD annotated Italian texts to enrich the existing *Italian Stanford Dependency Treebank* (ISDT). This is done by comparing the performance of a statistical parser (DeSR) trained on a simpler resource (the augmented version of the *Merged Italian Dependency Treebank* or MIDT+) and whose output was automatically converted to SD, with the results of the parser directly trained on ISDT. Experiments carried out to test reliability and effectiveness of the two strategies show that the performance of a parser trained on the reduced dependencies repertoire, whose output can be easily converted to SD, is slightly higher than the performance of a parser directly trained on ISDT. A non-negligible advantage of the first strategy for generating SD annotated texts is that semi-automatic extensions of the training resource are more easily and consistently carried out with respect to a reduced dependency tag set. Preliminary experiments carried out for generating the collapsed and propagated SD representation are also reported.

Keywords: Italian Treebank, Harmonization and Merging of Resources, Stanford Dependencies

1. Introduction

The limited availability of training resources is a widely acknowledged bottleneck for machine learning approaches for Natural Language Processing (NLP) and motivates the current trend towards the harmonization and merging of existing data sets, possibly converting them into *de facto* standards.

Within this general picture, we have been involved in the harmonization of two CoNLL–compliant dependency–annotated Italian Treebanks: the *Turin University Treebank* (TUT)[1] developed by the NLP group of the University of Turin (Bosco, Lombardo and Lesmo, 2000) and the treebank called ISST–TANL, which was developed as a joint effort by the Istituto di Linguistica Computazionale (ILC–CNR) and the University of Pisa and originating from the *Italian Syntactic–Semantic Treebank* or ISST (Montemagni et al., 2003).

In combining existing resources we defined a methodology for translating between different annotation schemes and merging them, articulated into the following steps: i) analysis of similarities and differences of considered source and target dependency annotation schemes; ii) analysis of the performance of state of the art dependency parsers trained on source and target treebanks; iii) mapping of the source annotation scheme(s) onto a set of target (possibly underspecified) data categories. This methodology was tested in two case studies.

The first case study was carried out within the national project "Portal for the Access to the Linguistic Resources for Italian" (PARLI), where an annotation scheme to be used as a "bridge" between the native schemes was defined and used for the harmonization and merging of the TUT and ISST–TANL resources. This resulted in the construction of the *Merged Italian Dependency Treebank* (MIDT) (Bosco, Montemagni and Simi, 2012).

The second case study, performed in the framework of a collaboration with Google, consisted in the conversion of the resource resulting from the first case study, i.e. MIDT, into the Stanford Dependencies (SD) *de facto* standard. The MIDT to SD conversion process produced a new standard–compliant resource, i.e. the *Italian Stanford Dependency Treebank* (or ISDT) (Bosco, Montemagni and Simi, 2013).

The process of harmonization of TUT and ISST-TANL into MIDT entailed a reduction of the tag set to a small number of shared categories (from 72 and 29 respectively to 19) that could be preserved in the conversion, thus compromising on the richness of both native annotation schemes and raising the question of whether the resulting annotation is rich enough for applications. Table 1 reports the dependencies types in MIDT.

Despite this "simplification", in the second conversion step leading to ISDT, it was possible to recover, through an automatic conversion process, *most* of the complexity and distinctions accounted for by the Stanford scheme. Different factors contributed to make this conversion possible, ranging from the fact that the MIDT and SD schemes share important design principles (see design principles 2 and 5 in de Marneffe and Manning, 2008), to the rich PoS tag set used in MIDT also including morph-syntactic features complementing the part of speech information (such as number, gender, etc.), and –last but not least– the possibility to extend the apparently poor annotation scheme of MIDT (which passed from 19 to 24 tags) with information recovered from the original resources. These facts, combined together, proved adequate to allow for automatic conversion into the more elaborate SD scheme, which in the current version includes about 50 relation types (de Marneffe and Manning, 2013).

[1] http://www.di.unito.it/~tutreeb

MIDT	DESCRIPTION
arg	clausal argument
aux	auxiliary verb
clit	relation between a clitic pronoun and a verbal head used in pronominal form
comp	relation between a clitic pronoun and a verbal head used in pronominal form
con	relation between a copulative conjunction and the first conjunct in coordinate construction
concat	concatenation
conj	relation between a copulative conjunction and the first conjunct in coordinate construction
det	determiner
mod	modifier (adjectival, adverbial, nominal or clausal)
neg	negation
obj	direct object
pred	predicative complement
prep	relation between a prepositional head and its complement, whether clausal or not clausal
punc	punctuation
relcl	relative clause modifier
root	sentence root
sub	relation between a subordinative conjunction and the verbal head of its clausal complement
subj	subject
subj_pass	subject of a passive verb

Table 1. MIDT dependencies types

Despite this "simplification", in the second conversion step leading to ISDT, it was possible to recover, through an automatic conversion process, *most* of the complexity and distinctions accounted for by the Stanford scheme. Different factors contributed to make this conversion possible, ranging from the fact that the MIDT and SD schemes share important design principles (see design principles 2 and 5 in de Marneffe and Manning, 2008), to the rich PoS tag set used in MIDT also including morph-syntactic features complementing the part of speech information (such as number, gender, etc.), and – last but not least– the possibility to extend the apparently poor annotation scheme of MIDT (which passed from 19 to 24 tags) with information recovered from the original resources. These facts, combined together, proved adequate to allow for automatic conversion into the more elaborate SD scheme, which in the current version includes about 50 relation types (de Marneffe and Manning, 2013).

The state of affairs just described raises interesting research questions, whose answer has a potential impact on which is the most appropriate annotation scheme to be used for extending the Italian Dependency Treebank. Assuming that our final goal is annotating texts according to the Stanford Dependencies, which is gaining popularity for many languages as a suitable formalism for further semantic processing and information extraction, the questions which need to be answered are: shall we aim at producing an Italian Treebank using the SD or the MIDT annotation scheme, considering that the simpler MIDT scheme would be easier for annotators to apply? And, would it be viable to train a dependency-parsing model on a MIDT resource and convert algorithmically to SD or shall we train a model directly on the SD representation? Finally, which solution will give us a better parsing accuracy?

Given the focus on the definition of a core set of relations to be accounted for in a shared annotation scheme, we believe that an answer to these questions will also contribute to ongoing efforts towards universal (McDonald et al., 2013) or standard annotation schemes (Declerck, 2008; Kemps-Snijders et al., 2009).

This paper is organized as follows. After describing the semi-automatic conversion processes leading to MIDT and then to ISDT, we present an extension of MIDT, MIDT+, which allows for fully automatic conversion to the *basic* variant of the Stanford Dependencies. We then present the results of comparing the output of a parser trained on MIDT+, and automatically converted, with the performance of a parser trained on a resource annotated with Stanford Dependencies. A further extension to MIDT+, i.e. MIDT++, is introduced with the specific goal of providing all the information needed for obtaining the collapsed version of the SD, including relations such as agents and external subjects (xsubj).

2. MIDT+ to basic Stanford Dependencies

The harmonization process leading to the construction of the MIDT resource starting from the CoNLL compliant TUT and ISST-TANL treebanks informed the design of the MIDT annotation scheme. On the one hand, practical considerations played a key role in the definition of the MIDT scheme which was conceived as the lowest common ground between the TUT and ISST–TANL annotations: the harmonization of the source schemes was carried out by exploiting morpho–syntactic and dependency information contained in the original resources and by reducing as much as possible the need for manual revision (which in some cases was unavoidable, see e.g. appositions, explicitly annotated only in TUT). On the other hand, the desire to improve on the performance of existing parsers suggested to drop dependency types which proved to be difficult to predict by syntactic parsers, in spite of their being explicitly encoded in both annotation schemes: e.g. indirect objects or temporal modifiers were assigned an underspecified representation in MIDT on the basis of the results reported in (Bosco et al., 2010). For more details on the creation of the MIDT resource we refer to (Bosco, Montemagni and Simi, 2012).

The conversion process followed to generate the *Italian Stanford Dependency Treebank* (ISDT) starting from MIDT is organized in two different steps:

- the first step aims at generating an enriched version of the MIDT resource, including SD–relevant distinctions which were originally neutralized. In what follows, we will refer to this augmented version of MIDT as MIDT+;

- the second step is in charge of converting this intermediate annotation (resulting from the previous step) in terms of the Stanford Dependencies as described in (de Marneffe and Manning, 2008 and 2013) which were specialized with respect to the Italian language syntactic peculiarities. A preliminary account of this process as well as the Italian localization of the Stanford Dependency scheme are described in detail in (Bosco, Montemagni and Simi, 2013).

2.1 Extending MIDT

The first conversion step generates MIDT+, an intermediate Treebank to be used as a bridge towards the SD representation. MIDT+ is an extended version of MIDT, recovering information from the native TUT and ISS-TANL resources with the aim to produce a richer resource, easily convertible to the basic Stanford Dependencies annotation scheme. In particular, MIDT+ includes the additional dependency types listed below, with examples (where dependents are marked in bold and heads in italiac):

aux_pass: a specialization of *aux* in passive constructions.
Example:
I due cittadini svizzeri **furono** *liberati* a Beirut poco dopo, nella metà dell'agosto 1990.
The two Swiss citizens were released in Beirut soon after, in the middle of August 1990.

comp_ind: indirect object.
Example:
Anche Don Battista, il parroco che **a** Mondo Giusto *dà* una mano con offerte e qualche viaggio.
Even Don Battista, the priest that lends a hand to Mondo Giusto with offerings and some travel.

mod_appos: appositional modifier (also covering abbreviations).
Example:
È Kurt *Kola*, **presidente** del *forum*, l'**organizzazione** che riunisce i partiti dell'opposizione.
Is Kurt Kola, President of the Forum, the umbrella organization of opposition parties.

mod_parataxis: relation linking two sentences placed side by side without any explicit coordination or subordination.
Example:
Sì *è* vero - **ammette** il ministro albanese -.
Yes it's true - the Minister of Albania admits -.

mod_temp: temporal modifier.
Example:
Il sequestro era *avvenuto* lo scorso **sabato**.
The seizure had taken place last Saturday

The dependency type *aux_pass* could be automatically recovered by exploiting the simultaneous presence of a passive subject (*subj_pass*) and the auxiliary verb 'essere' (be) (*aux*) in the MIDT annotation. Indirect complements (*comp_ind*) were originally annotated in both the TUT and ISST-TANL resources and for this reason could be easily recovered. Temporal modifiers, together with temporal complements, were part of the ISST-TANL tag set but were only partially marked in TUT (as a specialization of the modifier relation): recovering temporal modifiers, in the restricted sense of the SD, entailed the use of heuristics and manual checking. This was also the case for appositions (*mod_appos* in MIDT+) and parataxis (*mod_parataxis*). The introduction of parataxis was especially tricky since it required in most cases a substantial restructuring of the parse tree.
Thanks to the enriched dependency annotation scheme (consisting now of 24 dependency types) and detailed PoS information, MIDT+ provides now enough information for the conversion into the basic variant of the SD scheme, i.e. the one preserving the tree structure as described in (de Marneffe and Manning, 2013).

2.2 Conversion to SD

MIDT+, resulting from the pre-processing step described above, constitutes the starting point of the real conversion process, targeting the SD representation format. As a reference for the conversion we used the version of the Stanford manual released in June 2013 (v. 3.2), which already includes some of the new constructions discussed in the DepLing 2013 workshop (deMarneff, Connor et al. 2013).
Conversion patterns were defined which can be grouped into two main classes according to whether they refer to individual dependencies or involve dependency sub-trees: whereas the former are structure–preserving rules simply involving dependency retyping (with both 1:1 and 1:n mapping), the latter represents the most problematic case, corresponding to tree restructuring rules simultaneously involving head reassignment and dependency retyping.
Consider the conversion of the MIDT+ *arg* relation, referring to clausal complements subcategorized for by the governing head: this represents an interesting example of the conversion issues that had to be addressed in the latter and more complex case. In MIDT, clausal complements, either finite or non–finite clauses, are linked to the governing head (which can be a verb, a noun or an adjective) as *arg*(uments), with a main difference with respect to SD, i.e. that the head of the clausal complement is the word introducing it (be it a preposition or a subordinating conjunction) rather than the verb of the clausal complement. Depending on the mood (finite vs non-finite) of the clausal complement, the assigned SD label is different: *ccomp* is used for clausal complements headed by finite verbs, *xcomp* for complements headed by verbs used in the infinitival form. Rules 1. and 2. below summarize the conversion steps to be followed, where the $\Rightarrow$ separates the left from the right hand side of the rule, the notation $x \rightarrow_{dep\ label} y$ denotes that token *y* is governed by token *x* with the dependency label specifying the relation holding between the two (a MIDT+ tag is found on the left side of the rule, whereas an SD one occurs on the right side):

1. $\$1[S|V|A] \rightarrow_{arg} \$2[E] \rightarrow_{prep} \$3[Vinfinitive] \Rightarrow$
   $\$1 \rightarrow_{xcomp} \$3$; $\$3 \rightarrow$aux $\$2$
2. $\$1[SjV\ jA] \rightarrow_{arg} \$2[CS] \rightarrow_{sub} \$3[Vfinite] \Rightarrow$
   $\$1 \rightarrow_{ccomp} \$3$; $\$3 \rightarrow_{aux} \$2$

In the rules, the $ followed by a number is a variable identifying a given dependency node. Constraints on tokens in the left–hand side of the rule are reported within square brackets: they are typically concerned with the grammatical category of the token (CS stands for subordinative conjunction, E for preposition, S for noun, V for verb). Rule 1 above handles the transformation of the infinitival clause from the MIDT+ representation to SD.

Figure 1 exemplifies input and output of rule 1 for the sentence "Giovanni ha dichiarato ai giudici di avere pagato i terroristi", lit. '*Giovanni told to–the judges to have paid the terrorists*' '*Giovanni told the judges that he has paid the terrorists*': the MIDT+ dependency tree is reported in Figure 1(a) whereas its SD counterpart is in Figure 1(b). By comparing the trees, we see that head restructuring and dependency retyping have both been performed in the conversion of the infinitival clause representation: in MIDT+ the head of the infinitival clause is the preposition ("di") whereas in SD it is the verb ("pagato"); the relation linking the governing head and the head of the infinitival clause is *arg* in MIDT+ and *xcomp* in SD.

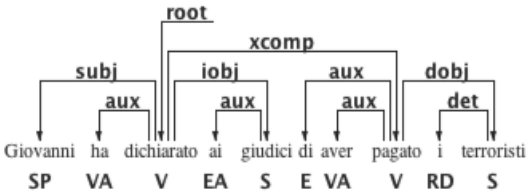

Figure 1(a). MIDT representation

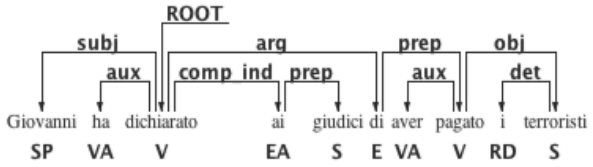

Figure 1(b) STD representation

Currently, the conversion script implements over 100 rules which are still being tested with the final aim of reducing annotation inconsistencies originating from the native resources and finding the most appropriate representation with respect to the Italian syntactic peculiarities.

A byproduct of the conversion to ISDT is represented by the specialization of the SD scheme with respect to the Italian language. There are SD dependency relations excluded from the Italian localization of the standard scheme, either because not appropriate given the syntactic peculiarities of this language (this is the case e.g. of the *prt* relation) or because they could not be recovered from the CoNLL–compliant versions of the resources we started from (see e.g. the relation *xsubj*). The SD tag set was also extended with new dependency types: this is the case of the *clit* relation used for dealing with clitics in pronominal verbs, or of the *nnp* relation specifically defined for compound proper nouns. Other specializations are concerned with the use of underspecified intermediate categories: rather than resorting to the most generic relation, i.e. *dep* used when it is impossible to determine a more precise dependency relation, we exploited the hierarchical organization of SD typed dependencies, i.e. we used the *comp* and *mod* relations when we could not find an appropriate relation within the set of their dependency subtypes.

This conversion step lead to the construction of the basic tree structure preserving version of ISDT.

2.3 Evaluation

Having generated the MIDT+ and ISDT resources, we are now in the position to answer the questions we started with, i.e. whether it would be preferable to train a dependency parser on the MIDT+ resource and to convert algorithmically the output to SD, or to train a parser directly on the SD representation. In order to investigate which is the most appropriate strategy to generate a SD-compliant dependency representation for the Italian language, we conducted different experiments whose results are discussed below. Figure 2 sketches the strategy we defined for the comparison: i.e. we compared the performance of a statistical parser trained on MIDT+, whose output was automatically converted to SD, with the results of a parser directly trained on the ISDT resource. For these experiments we used the DeSR parser, with Multi-Layer Perceptron as classification algorithm (Attardi, Dell'Orletta, 2009).

Achieved results are reported in Table 2 where: each row refers to the experiments carried out by using for training the converted version of the individual resources (TUT and ISST-TANL) and the combined resource (MERGED); each column reports the results obtained with respect to a

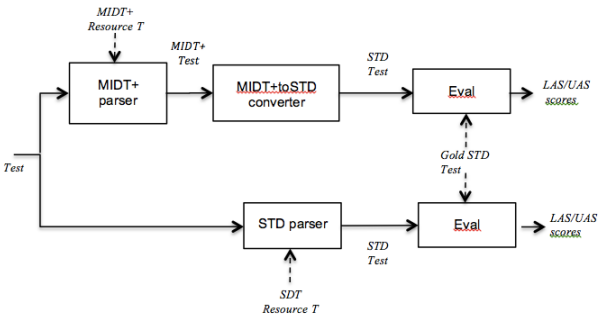

Figure 2 Comparison strategy

specific annotation scheme, namely MIDT, MIDT+, MIDT+→SD (the SD representation obtained from conversion of the MIDT+ output) and ISDT. The last column (Diff) reports the difference in performance between the experiments targeting an SD representation, i.e. MIDT+→SD and ISDT. In all reported experiments, the split between training and test is fixed in order to guarantee comparability of achieved results: the proportion between the test and training sets is about 1/14 (overall 12,700 tokens corresponding to 580 sentences). The results in Table 2 are summarized in terms of Labeled Attachment Score (LAS) and Unlabeled Attachment Score (UAS), without considering punctuation.

It is interesting to observe that using the MIDT resource for training leads to more accurate results with respect to MIDT+: this is not surprising since the MIDT+ tag set includes new semantically-oriented relations.

Table 3 shows recall and precision recorded with respect to these new tags, which are very likely responsible for the limited loss in performance (on average, less than one point).

Resource	MIDT	MIDT+	MIDT+→SD	ISDT	Diff
TUT 97,558 tokens	LAS: 89.47 UAS: 92.43	LAS: 88.86 UAS: 91.77	LAS: 87.90 UAS: 90.41	LAS: 86.42 UAS: 88.85	+1.48 +1.56
ISST-TANL 80,990 tokens	LAS: 84.99 UAS: 87.91	LAS: 83.55 UAS: 86.91	LAS: 83.10 UAS: 85.88	LAS: 82.39 UAS: 85.47	+0.71 +0.41
MERGED 178,548 tokens	LAS: 86.53 UAS: 90.95	LAS: 85.35 UAS: 90.42	LAS: 86.36 UAS: 89.23	LAS: 85.19 UAS: 88.18	+1.17 +1.05

Table 2. Comparison of results

deprel	recall	precision
aux_pass	84.50%	90.00%
comp_ind	39.02%	80.00%
mod_appos	22.73%	33.33%
mod_parataxis	26.32%	41.67%
mod_temp	56.67%	71.83%

Table 3. Recall and precision of MIDT+-only tags

We should consider, however, that this limited loss in performance is repaid by the fact that the MIDT+ representation contains all the information needed for a reliable conversion into the SD basic scheme. In fact, by comparing the converted SD output with the output produced by a parser directly trained on the SD resource, we observe that the output obtained by conversion (MIDT+→SD) shows higher LAS and UAS scores, with more than one point improvement on average.

In the wake of these encouraging but preliminary results and in order to confirm them, we performed another experiment following a 10-fold cross-validation strategy operating on the merged resource whose final results are shown in Table 4. The conversion strategy turned out to achieve consistently better results in all the ten runs, so we can state the preliminary results of Table 2 are confirmed; however, the recorded improvement is, on average, less than what our preliminary results had shown, i.e. +0.68 for LAS and +0.53 for UAS.

MIDT	MIDT+	MIDT+→SD	ISDT
LAS: 86.20 UAS: 88.95	LAS: 85.11 UAS: 88.75	LAS: 84.47 UAS: 87.40	LAS: 83.79 UAS: 86.87

Table 4. 10-fold cross-validation on the merged resource

By comparing the results of the two sets of experiments (reported in Table 2 and 4 respectively), there is a common trend worth being emphasizing here: the improvement is almost always higher at the level of LAS (as opposed to UAS). This is not surprising if we consider that the main difference between the MIDT+ and SD (basic version) schemes is at the level of the tag set granularity (24 vs 45 tags). The only exception is represented by TUT, where the improvement is slightly higher for UAS.

On the basis of the results achieved so far, we are in the position to answer the questions we started with. The experiments show that better results are achieved by training a dependency parser on the MIDT+ resource and by converting algorithmically the output to the SD representation, rather than by directly training a parser on the SD representation. These results have interesting implications at the level of the strategy to be pursued in the extension of the Treebank resource for the Italian language: the MIDT+ scheme, which is much simpler than the basic SD scheme, is a better candidate for manually extending the Treebank annotation. The SD representation, which is our target, can be easily and consistently obtained by conversion from the MIDT+ representation. In fact, the manual annotation with a more granular tag set seems to introduce potential annotation inconsistencies, which are reduced if the SD representation is generated automatically.

3. MIDT++ to collapsed Stanford Dependencies

Continuing along this line of reasoning one might wonder whether MIDT+ contains enough information to obtain the *collapsed and propagated* variants of the Stanford Dependencies, which are meant to provide a semantic-oriented representation.

In the *collapsed* representation, dependencies involving prepositions, multi-words, conjuncts, and the referent of relative clauses (i.e. the relative word introducing them) are collapsed to get direct dependencies between content words. Additional dependencies are also added, including the *agent* of a verb. Some of these cannot be accommodated in a tree structure (which is assumed by a CoNLL format), and turn the dependency structure into a directed graph, with the possibility of local cycles. The output representation of the collapsed and propagated SD variant is usually in terms of a set of binary relations.

Conjunct propagation aims at distributing the dependencies involving one conjunct to the other conjuncts, but only when it is appropriate to do so without altering the meaning of the sentence. As discussed in (Nyblom et al. 2013) this is not an easy task as it involves resolving, at least partially, coordination scope ambiguities that arise in most languages, including Italian. For addressing these problems, the authors propose the use of machine learning techniques, and show that they can reach a high level of accuracy in this task.

In the current version of the Stanford parser, conjunct propagation is handled only partially by focusing on a limited set of cases. Our conversion to collapsed and propagated Italian SD is based on the Stanford approach. In particular, conjunct propagation is limited to the following general cases[2]:

1. dependencies of the first conjunct are propagated to the other conjuncts (as illustrated in Figure 3, where B is the first conjunct and C … D are the other conjuncts);
2. the subject of the first conjunct is propagated to the other conjuncts (as illustrated in Figure 4, where A is

[2] There are a few exceptions to these general rules, discussed in the project thesis report of Roberta Montefusco.

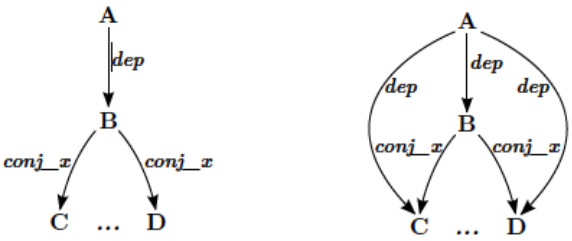

Figure 3. Propagation of head over conjuncts

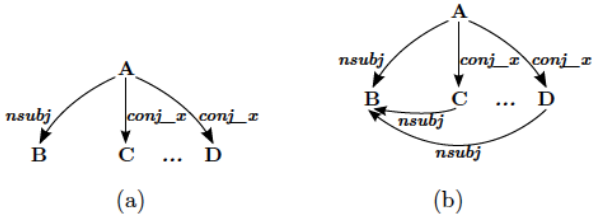

Figure 4. Propagation of nsubj relation over conjuncts

the first conjunct and C … D are the other conjuncts). For the collapsed representation, there are at least two pieces of information missing from the MIDT+ annotation scheme.

The first one is the explicit marking of the *agent* in passive constructions, which cannot be automatically reconstructed due to the ambiguity of the preposition introducing the agent, i.e. "da": in Italian "da" can also be used to introduce origin as in '*La lettera è stata spedita* ***da*** *Roma*' – '*The letter has been sent from Rome*' or temporal complements as in "*Sono stato licenziato* ***dalla*** *settimana prossima*" – "*I was fired since next week*".

To the specific end of generating the collapsed version of SD, the MIDT+ scheme was extended with the *comp_agent* relation (which is a specialization of the dependency *comp* specifically introduced to mark agents): the extended MIDT+ scheme is henceforth referred to as MIDT++.

The second type of missing information concerns relation types not explicitly encoded in the MIDT source annotation due to the constraints of the CoNLL representation format. This is the case of the *ref* dependency, linking the relative pronoun to its antecedent, or of the *xsubj* relation holding between the head of an open clausal complement (*xcomp*) and its external subject. In spite of their being part of the original TUT and ISST resources, these relations were omitted in the CoNLL–compliant versions used as a starting point for MIDT: in both cases, the "one head per dependent" constraint of the CoNLL representation format is violated. This was in a sense the price to pay in the merging strategy leading to a shared resource for Italian.

Different strategies are being implemented to cope with the variety of cases illustrated above. The antecedent of a relative pronoun (*ref*) can be easily reconstructed from the internal structure of the dependency sub-tree headed by it, also thanks to the explicit marking of relative pronouns with a specific part of speech. Figure 1 shows an example of the result of this type of collapsing: the sentence is "Il valore di soglia al di sopra del quale si vedono danni fogliari visibili sulle piante sensibili, è di 700 ppb/ora" (*The threshold value above which visible foliar damage on susceptible plants are seen, is 700 ppb / hour*). Here the *ref* relation is implicitly used to introduce the additional appropriate relation linking back token "`vedono-11`" to "`valore-2`" (i danni si vedono *al di sopra dei* valori – *damages are seen above the threshold value*).

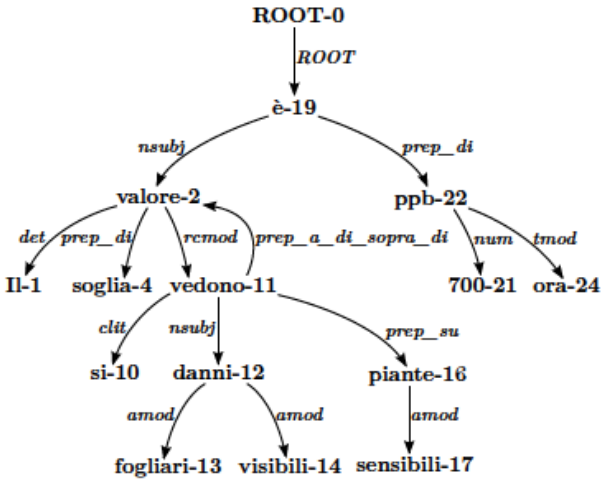

Figure 5 An example of collapsing in relative clauses

The situation for controlled subjects (*xsubj*) is different. The *xsubj* relation holds between the head of an open clausal complement (*xcomp*) and its external subject. Following Lexical-Functional Grammar, the SD scheme uses the *xcomp* and *xsubj* relations (the latter being part of the collapsed SD scheme only) to account for control constructions. Depending on the type of verb (control vs raising) governing the open clausal complement, different strategies can be adopted for reconstructing the *xsubj* relation. With control verbs, knowledge about the lexical properties of the verbal head governing the open clausal complement is needed: for this reason, the *xsubj* relation with control verbs (e.g. "promettere" 'promise', "ordinare" 'order') is not being dealt with for the time being due to the inherent difficulty of recovering this information type. Different is the case of raising verbs such as "volere" 'want', which is only the one currently being dealt with and explicitly marked in the MIDT++ resource to allow for its conversion towards SD.

With this extra information, a text annotated according to MIDT++, can be converted by means of a graph transformation algorithm, into the collapsed version of the Stanford Dependencies.

Preliminary experiments show that the drop in performance in moving from MIDT+ to MIDT++ (enriched with *comp-agent* and a subset of *xsubj* cases) is not significant: LAS is -0,21 and UAS is -0,12 on the basis of a 10-fold cross-validation experiment.

The extension of the MIDT+ resource is still in progress. We plan to address the new extensions discussed in the DepLing 2013 workshop (deMarneff, Connor et al. 2013), which would be useful when extending the treebank to different genres and for a more linguistically sound treatment of some problematic constructions.

Alternative directions to recover the information missing from the MIDT resource but present in the native formats of TUT and ISST are being explored: i.e. a direct conversion e.g. from TUT to the extended version of the SD format is expected to allow a principled analysis of a

larger variety of constructions and relations that cannot be extracted from MIDT or MIDT+, like e.g. the representation of free relatives and comparative expressions in terms of graphs rather than dependency trees. This alternative direction, however, goes beyond the scope of this paper.

4. Conclusion

Stanford Dependencies represent nowadays a *de facto* standard as far as dependency annotation is concerned: this representation scheme is gaining popularity for many languages for its being defined with a specific view to supporting information extraction tasks. Starting from this widely acknowledged assumption, our goal is to extend the existing *Italian Stanford Dependency Treebank* (ISDT) with new texts, possibly representing different registers, textual genres and sublanguages. In this paper we explored pros and cons of different strategies for generating SD annotated Italian texts. This was done by comparing the performance of a statistical parser (DeSR) trained on a simpler resource (the augmented version of the *Merged Italian Dependency Treebank* or MIDT+) and whose output was automatically converted to SD, with the results of the parser directly trained on ISDT. Experiments carried out to test reliability and effectiveness of the two strategies show that the performance of a parser trained on the reduced dependencies repertoire, whose output can be easily converted to SD, is slightly higher than the performance of a parser directly trained on ISDT. A non-negligible advantage of this strategy for generating SD annotated texts is that semi-automatic extensions of the training resource are more easily and consistently carried out with respect to a reduced dependency tag set. Preliminary experiments carried out along the same lines for generating the collapsed and propagated SD representation have also been reported.

We believe that achieved results also contribute to shed light on important open issues of the linguistic annotation literature. First, we showed that a set of reduced relation types (corresponding to the MIDT+ dependency tag set) combined with a rich morpho-syntactic annotation is sufficient to automatically reconstruct the full complexity of the much richer SD annotation scheme. Second, we also contribute to the debate on whether and to what extent the granularity of an annotation scheme influences the performance of a dependency parser. Our results are in line with those reported by Mille et al. (2012) who compare the results of state-of-the-art parsers trained on different versions of a Spanish corpus annotated following schemes of different granularity (ranging from 15 to 44 tags). Although looked at from different perspectives, the reported trend is similar, i.e. achieved results show that different levels of granularity in the dependency tag set do not imply a too significant variation in accuracy of the parser performance: in both cases, the less granular tag set shows a better performance.

5. Acknowledgements

Roberta Montefusco worked, as part or her final undergraduate thesis project, to the conversion from the SD to the collapsed and propagated dependency representation; she also contributed the carefully drawn graphics in this paper. Giuseppe Attardi advised in the parsing experiments using DeSR. This work was partially supported by a Google gift.

6. References

Attardi, G., Dell'Orletta, F.: Reverse revision and linear tree combination for dependency parsing. In: Proceedings of the NAACL HLT'09, Boulder, Colorado (2009) 261–264.

Bosco, C., Lombardo, V., Lesmo, L., Vassallo, D.: Building a treebank for Italian: a data-driven annotation schema. In: Proceedings of the 2nd Language Resources and Evaluation Conference (LREC'00), Athens, Greece, ELRA (2000) 99–105.

Bosco, C., Montemagni, S., Mazzei, A., Lombardo, V., Dell'Orletta, F., Lenci, A., Lesmo, L., Attardi, G., Simi, M., Lavelli, A., Hall, J., Nilsson, J., Nivre, J.: Comparing the influence of different treebank annotations on dependency parsing. In: Proceedings of LREC'10, Valletta, Malta.

Bosco, C., Montemagni, S., Simi, M.: Harmonization and merging of two Italian dependency treebanks. In: Proceedings of the LREC Workshop on Language Resource Merging, Istanbul, Turkey, ELRA (2012) 23–30.

Bosco C., Montemagni, S., Simi, M.: Converting Italian Treebanks: Towards an Italian Stanford Dependency Treebank. In: ACL Linguistic Annotation Workshop & Interoperability with Discourse, Sofia, Bulgaria (2013).

Declerck, T.: SynAF: Towards a standard for syntactic annotation. In: Proceedings of the 6th Language Resources and Evaluation Conference (LREC'08), Marrakech, Morocco, ELRA (2008), pp. 229–232.

de Marneffe, M., Manning, C.: The Stanford typed dependencies representation. In: Proceedings of the COLING Workshop on Cross-Framework and Cross-Domain Parser Evaluation, Stroudsburg, PA, USA, Association for Computational Linguistics (2008), pp. 1–8.

de Marneffe, M., Manning, C.: Stanford typed dependencies manual. Technical report, rev. August 2013, Stanford University (2013)

de Marneffe, M., Connor M., Silveira N., Bowman S. R., Dozat T., Manning C. D.: More constructions, more genres: Extending Stanford Dependencies. Proc. of the Second International Conference on Dependency Linguistics (DepLing 2013), Prague, August 27–30, 2013, Charles University in Prague, Matfyzpress, Prague, pp. 187–196.

Kemps-Snijders, M., Windhouwer, M., Wittenburg, P., Wright, S.: ISOcat: remodelling metadata for language resources. IJMSO **4**(4) (2009), pp. 261–276.

McDonald R., Nivre, J., Quirmbach-Brundage, Y., Goldberg, Y., Das, D., Ganchev, K., Hall, K., Petrov, S., Zhang, H., Täckström, O., Bedini, C., Bertomeu Castelló, N., Lee, J.: Universal Dependency Annotation for Multilingual Parsing. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria (2013).

Mille S., Burga A., Ferraro G., Wanner L.: How Does the granularity of an Annotation Scheme Influence Dependency Parsing Performance?. In: Proceedings of the 24th International Conference on Computational Linguistics (COLING), Mumbai, India (2012).

Montemagni, S., Barsotti, F., Battista, M., Calzolari, N., Corazzari, O., Lenci, A., Zampolli, A., Fanciulli, F., Massetani, M., Raffaelli, R., Basili, R., Pazienza, M.T., Saracino, D., Zanzotto, F., Mana, N., Pianesi, F., Delmonte, R.: Building the Italian Syntactic-Semantic Treebank. In Abeillé, A., ed.: Building and Using syntactically annotated corpora. Kluwer, Dordrecht (2003) pp. 189–210.

Nyblom J., Kohonen S., Haverinen K., Salakoski T., Ginter F.: Predicting conjunct propagation and other extended Stanford Dependencies. Proc. of the Second International Conference on Dependency Linguistics (DepLing 2013), Prague, August 27–30, 2013, Charles University in Prague, Matfyzpress, Prague, pp. 252–261.