Converting Italian Treebanks: Towards an Italian Stanford Dependency Treebank

Cristina Bosco
Dipartimento di Informatica
Università di Torino
Corso Svizzera 185
10149 Torino (Italy)
cristina.bosco@unito.it

Simonetta Montemagni
Istituto di Linguistica
Computazionale
"Antonio Zampolli" (ILC–CNR)
Via G. Moruzzi 1
56124 Pisa Pisa (Italy)
simonetta.montemagni@ilc.cnr.it

Maria Simi
Dipartimento di Informatica
Università di Pisa
Largo B. Pontecorvo 3
56127 Pisa (Italy)
simi@unipi.it

Abstract

The paper addresses the challenge of converting MIDT, an existing dependency–based Italian treebank resulting from the harmonization and merging of smaller resources, into the Stanford Dependencies annotation formalism, with the final aim of constructing a standard–compliant resource for the Italian language. Achieved results include a methodology for converting treebank annotations belonging to the same dependency–based family, the *Italian Stanford Dependency Treebank* (ISDT), and an Italian localization of the Stanford Dependency scheme.

1 Introduction

The limited availability of training resources is a widely acknowledged bottleneck for machine learning approaches for Natural Language Processing (NLP). This is also the case of dependency treebanks within statistical dependency parsing. Moreover, the availability of a treebank in a standard format strongly improves its usefulness, increasing the number of tasks for which it can be exploited and allowing the application of a larger variety of tools. It also has an impact on the reliability of achieved results, and, last but not least, it permits comparability with other resources.

This motivated a variety of initiatives devoted to the definition of standards for the linguistic annotation of corpora. Since the early 1990s, different initiatives have been devoted to the definition of standards for the linguistic annotation of corpora with a specific view to re–using and merging existing treebanks. The starting point is represented by the EAGLES (Expert Advisory Groups on Language Engineering Standards) initiative, which ended up with providing provisional standard guidelines (Leech et al., 1996), operating at the level of both content (i.e. the linguistic categories) and encoding format. More recent initiatives, e.g. LAF/GrAF (Ide and Romary, 2006; Ide and Suderman, 2007) and SynAF (Declerck, 2008) representing on–going ISO TC37/SC4 standardization activities[1], rather focused on the definition of a pivot format capable of representing diverse annotation types of varying complexity without providing specifications for the annotation of content categories (i.e., the labels describing the associated linguistic phenomena), for which standardization appeared since the beginning to be a much trickier matter. Recently, other standardization efforts such as ISOCat (Kemps-Snijders et al., 2009) tackled this latter issue by providing a set of data categories at various levels of granularity, each accompanied by a precise definition of its linguistic meaning. Unfortunately, the set of dependency categories within ISOCat is still basic and restricted. We can thus conclude that as far as content categories are concerned *de jure* standards are not suitable at the moment for being used in the harmonization and merging of real dependency treebanks.

The alternative to *de jure* standards is represented by *de facto* standards. For what concerns dependency–based annotation, which in the recent past has been increasingly exploited for a wide range of NLP–based information extraction tasks, the Stanford Dependency (SD) scheme (de Marneffe et al., 2006) is gaining popularity as a *de facto* standard. Among the contexts where SD has been applied, we can observe e.g. parsers and corpora exploited in biomedical information extraction, where it has been suggested to be a suitable unifying syntax formalism for several incompatible syntactic annotation schemes (Pyysalo et al., 2007). SD has already been applied to different languages, e.g. Finnish in the Turku treebank (Haverinen et al., 2010), Swedish in the Talbanken

[1] http://www.tc37sc4.org/

treebank[2], Chinese in the Classical Chinese Literature treebank (Seraji et al., 2012) or Persian in the Uppsala Persian Dependency Treebank (Lee and Kong, 2012).

In this paper, we describe the conversion of an existing Italian resource into the SD annotation scheme, with the final aim of developing a standard–compliant treebank, the *Italian Stanford Dependency Treebank* (ISDT). The reference resource, called *Merged Italian Dependency Treebank* (MIDT)[3] (Bosco et al., 2012), is the result of a previous effort in the direction of improving interoperability of data sets available for Italian by harmonizing and merging two existing dependency–based resources, i.e. TUT and ISST–TANL, adopting incompatible annotation schemes. The two conversion steps are visualized in Figure 1: note that in both of them the focus is on the conversion and merging of the content of linguistic annotation; for what concerns the representation format, all involved treebanks follow the CoNLL tab–separated format (Buchholz and Marsi, 2006) which nowadays represents a *de facto* standard within the international dependency parsing community. In this paper, we deal with the second step, focusing on the MIDT to ISDT conversion.

Starting from a comparative analysis of the MIDT and SD annotation schemes, we developed a methodology for converting treebank annotations belonging to the same dependency–based family based on:

- a comparative analysis of the source and target annotation schemes, carried out with respect to different dimensions of variation, ranging from head selection criteria, dependency tagset granularity to defined annotation criteria;
- the analysis of the performance of a state–of–the–art dependency parser by using as training the source and the target treebanks;
- the mapping of the MIDT annotation scheme onto the SD data categories.

[2] http://stp.lingfil.uu.se/~nivre/swedish_treebank/talbanken-stanford-1.2.tar.gz

[3] MIDT was developed within the project PARLI (http://parli.di.unito.it/project_en.html) partially funded in 2008-2012 by the Italian Ministry for University and Research, for fostering the development of new resources and tools that can operate together, and the harmonization of existing ones. MIDT is documented at http://medialab.di.unipi.it/wiki/MIDT/.

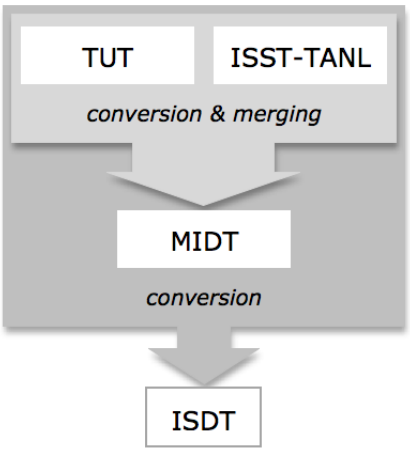

Figure 1: Merging and conversion process from TUT and ISST–TANL to MIDT and ISDT.

In this conversion process, we had to deal with the peculiarities of the Italian language: the tackled issues range from morphological richness, presence of clitic pronouns to relatively free word order and pro–drop, all properties requiring specific annotation strategies to be dealt with. Therefore, a by product of this conversion process is represented by the specialization of the SD annotation scheme with respect to Italian.

In the following sections, after briefly describing the methodology applied for the development of the MIDT resource (Section 2), we focus on a comparative analysis of the MIDT and SD annotation schemes (Section 3) followed by a description of the implemented conversion process (Section 4). Finally, we present the results obtained by training a parsing system on the newly developed resource (Section 5).

2 The starting point: MIDT

ISDT originates from the conversion towards the SD standard of the MIDT resource, whose origins and development are summarised below (for more details on this harmonization and merging step the interested reader is referred to Bosco et al. (2012)).

2.1 The ancestors: TUT and ISST–TANL

The TUT and ISST–TANL resources differ under different respects, at the level of both corpus composition and adopted annotation schemes.

For what concerns size and composition, TUT (Bosco et al., 2000)[4] currently includes 3,452 Italian sentences (i.e. 102,150 tokens in TUT native,

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

and 93,987 in CoNLL) and represents five different text genres (newspapers, Italian Civil Law Code, JRC-Acquis Corpus[5], Wikipedia and the Costituzione Italiana), while ISST–TANL includes 3,109 sentences (71,285 tokens in CoNLL format), which were extracted from the "balanced" ISST partition (Montemagni et al., 2003) exemplifying general language usage as testified in articles from newspapers and periodicals, selected to cover a high variety of topics (politics, economy, culture, science, health, sport, leisure, etc.).

As far as the annotation scheme is concerned, TUT applies the major principles of the Word Grammar theoretical framework (Hudson, 1984) using a rich set of dependency relations, but it includes *null* elements to deal with non–projective structures, long distance dependencies, equi phenomena, pro–drop and elliptical structures[6]. The ISST–TANL annotation scheme originates from FAME (Lenci et al., 2008), an annotation scheme which was developed starting from *de facto* standards and which was specifically conceived for complying with the basic requirements of parsing evaluation, and – later – for the annotation of unrestricted Italian texts.

2.2 Creating the merged MIDT resource

The challenge we tackled in the development of MIDT was to translate between different annotation schemes and merging them. We focused on the harmonization and merging of content categories. To this specific end, we defined a set of linguistic categories to be used as a "bridge" between the specific TUT and ISST–TANL schemes.

First of all, we analyzed similarities and differences of the underlying schemes, which led to identify a core of syntactic constructions for which the annotations agreed, but also to highlight variations in head selection criteria, inventory of dependency types and their linguistic interpretation, projectivity constraint and analysis of specific syntactic constructions. For instance, TUT always assigns heads on the basis of syntactic criteria, i.e. the head role is played by the function word in all constructions where one function word and one content word are involved (e.g. determiner–noun, verb–auxiliary), while in ISST–TANL head selection follows from a combination of syntactic and semantic criteria (e.g. in determiner–noun and auxiliary–verb relations the head role is played by the content word). Both schemes assume different inventories of dependency types and degrees of granularity in the representation of specific relations. Moreover, whereas ISST–TANL allows for non–projective representations, TUT assumes the projectivity constraint. Further differences are concerned with the treatment of coordination and punctuation, which are particularly problematic to deal with in the dependency framework.

As a second step, we defined a bridge annotation, i.e. the MIDT dependency tagset, following practical considerations: bridge categories should be automatically reconstructed by exploiting morpho–syntactic and dependency information contained in the original resources; for some constructions, the MIDT representation is parameterizable, i.e. the tagset provides two different options, corresponding to the TUT and ISST–TANL annotation styles (e.g. for determiner–noun or preposition–noun relations).

The final MIDT tagset contains 21 dependency tags (as opposed to the 72 tags of TUT and the 29 of ISST–TANL), including the different options provided for the same type of construction. CoNLL is used as encoding format.

3 Comparing the MIDT and SD schemes

The MIDT and SD annotation schemes are both dependency–based and therefore fall within the same broader family. This fact, however, does not guarantee *per se* an easy and linear conversion process from one to the other: as pointed out in Bosco et al. (2012), harmonizing and converting annotation schemes can be quite a challenging task, even when this process is carried out within a same paradigm and with respect to the same language. In the case at hand, this task is made easier thanks to the fact that the MIDT and SD schemes share similar design principles: for instance, in both cases preference is given a) to relations which are semantically contentful and useful to applications, or b) to relations linking content words rather than being indirectly mediated via function words (see design principles 2 and 5 respectively in de Marneffe and Manning (2008a)). Another peculiarity shared by MIDT and SD consists in the fact that they both neutralize the argument/adjunct distinction for what concerns prepositional complements, which is taken to be "largely useless

[5] http://langtech.jrc.it/JRC-Acquis.html

[6] The CoNLL format does not include null elements, but the projectivity constraint is maintained at the cost of a loss of information with respect to native TUT in some cases.

in practice" as de Marneffe and Manning (2008a) claim. In spite of their sharing similar design principles, there are also important differences concerning the inventory of dependency types and their linguistic interpretation, the head selection criteria as well as the treatment of specific syntactic constructions. In what follows, we summarize the main dimensions of variation between the MIDT and SD annotation schemes, with a specific view to the conversion issues they arise.

3.1 Granularity and inventory of dependency types

MIDT and SD annotation schemes assume different inventories of dependency types characterized by different degrees of granularity in the representation of specific relations: the adopted dependency tagset includes 21 dependency types in the case of MIDT and 48 in the case of SD. Interestingly however, it is not always the case that the finer grained annotation scheme – i.e. SD – is the one providing more granular distinctions: whereas this is typically the case, there are also cases in which more granular distinction are adopted in the MIDT annotation scheme.

Consider first SD relational distinctions which are neutralized at the level of the MIDT annotation. As reported in de Marneffe and Manning (2008a), so–called NP–internal relations are critical in real world applications: the SD scheme therefore includes many relations of this kind, e.g. `appos` (appositive modifier), `nn` (noun compound), `num` (numeric modifier), `number` (element of compound number) and `abbrev` (abbreviation). In MIDT all these relation types are lumped together under the general heading of `mod` (modifier). To deal with these cases, the MIDT to SD conversion has to simultaneously combine dependency and morpho–syntactic information (e.g. the morpho–syntactic category of the nodes involved in the relation), which however is not always sufficient as in the case of appositive modifiers for which further evidence is needed.

Let us consider now the reverse case, i.e. in which MIDT adopts finer–grained distinctions with respect to SD. For instance, MIDT envisages different relation types for auxiliary–verb and preposition–verb (within infinitive clauses, be they modifiers or subcategorized arguments) constructions, which are `aux` and `prep` respectively. By contrast, SD represents both cases in terms of the same relation type, i.e. `aux`. Significant differences between English and Italian justify the different strategies adopted in SD and MIDT respectively: in English, open clausal complements are always introduced by the particle 'to', whereas in Italian different prepositions can introduce them (i.e. 'a', 'di', 'da'), which are selected by the governing head. The SD representation of the element introducing infinitival complements and modifiers in terms of `aux` might not be appropriate as far as Italian is concerned and it would be preferable to have a specific relation for dealing with introducers of infinitival complements (like `complm` in the case of finite clausal complements): as reported in Section 4, we are currently evaluating different representational options with a specific view to the syntactic peculiarities of the Italian language.

Another interesting and more complex example can be found for what concerns the partitioning of the space of sentential complements. MIDT distinguishes between `mod`(ifiers) on the one hand and subcategorised `arg`(uments) on the other hand: note that whereas `arg` is restricted to clausal complements subcategorized for by the governing head, the `mod` relation covers different types of modifiers (nominal, adjectival, clausal, adverbial, etc.). By contrast, SD resorts to specific relations for dealing with sentential complements: in particular, distinct relation types are envisaged depending on e.g. whether the clause is a subcategorized complement or a modifier (see e.g. `ccomp` vs `advcl`), or whether the governor is a verb or a noun (see e.g. `xcomp` vs `infmod`), or whether the clausal complement is headed by a finite or non–finite verb (see e.g. `ccomp` vs `xcomp`). Starting from MIDT, the finer–grained distinctions adopted by SD for dealing with clausal complements can be recovered by combining dependency information with morpho-syntactic one (e.g. the mood of the verbal head of the clausal complements or the morpho–syntactic category of the governing head).

3.2 Head selection

Criteria for distinguishing the head and the dependent within relations have been widely discussed in the linguistic literature in all frameworks where the notion of syntactic head plays an important role. Unfortunately, different criteria have been proposed, some syntactic and some semantic, which do not lead to a single coherent notion

of dependency (Kübler et al., 2009). Head selection thus represents an important and unavoidable dimension of variation among dependency annotation schemes, especially for what concerns constructions involving grammatical function words. MIDT and SD agree on the treatment of tricky cases such as the determiner–noun relation within nominal groups, the preposition–noun relation within prepositional phrases as well as the auxiliary–main verb relation in complex verbal groups. In both schemes, head selection follows from a combination of syntactic and semantic criteria: i.e. whereas in the determiner–noun and auxiliary–verb constructions the head role is assigned to the semantic head (noun/verb), in preposition–noun constructions the head role is played by the element which is subcategorized for by the governing head, i.e. the preposition which is the syntactic head but can also be seen as as a kind of role marker. In this area, the only but not negligible difference is concerned with subordinate clauses whose head in SD is assumed to be the verb, rather than the introducing element (whether a preposition or a subordinating conjunction) as in MIDT: in this case, the MIDT to SD conversion requires restructuring of the dependency tree.

3.3 Coordination and punctuation

In both MIDT and SD schemes, coordinate constructions are considered as asymmetric structures with a main difference: while in MIDT both the conjunction and conjuncts starting from the second one are linked to the immediately preceding conjunct, in SD the conjunction(s) and the subsequent conjunct(s) are all linked to the first one. Also the treatment of punctuation is quite problematic in the framework of a dependency annotation scheme, although this has not been specifically dealt with in the linguistic literature. Whereas MIDT has its own linguistically–motivated strategy to deal with punctuation, SD does not appear to provide explicit and detailed annotation guidelines in this respect.

3.4 MIDT– or SD–only relations

It is not always the case that a dependency type belonging to the MIDT or SD annotation scheme has a counterpart in the other. Let us start from SD relation types which are not explicitly encoded in the MIDT source annotation, due to constraints of the CoNLL representation format. This is the case of the `ref` dependency linking the relative word introducing the relative clause and its antecedent, or of the `xsubj` relation which in spite of its being part of the original TUT and ISST resources have been omitted from the most recent and CoNLL–compliant versions, which represent the starting point of in MIDT: in both cases, the "one head per dependent" constraint of the CoNLL representation format is violated. From this, it follows that ISDT won't include these dependency types. Other SD relations which were part of the MIDT's ancestors but were neutralized in MIDT are concerned with semantically–oriented distinctions which turned out to be problematic to be reliably identified in parsing in spite of their being explicitly encoded in both source annotation schemes (Bosco et al., 2012). This is the case of the indirect object relation (`iobj`) or of temporal modifiers (`tmod`).

The MIDT relation types which instead do not have a corresponding relation in SD are those that typically represent Italian–specific peculiarities. This is the case of the `clit`(ic) dependency, linking clitic pronouns to the verbal head they refer to. In MIDT, whenever appropriate clitic pronouns are assigned a label that reflects their grammatical function (e.g. "dobj" or "iobj"): this is the case of reflexive constructions (*Maria si lava* lit. 'Maria her washes' meaning that 'Maria washes herself') or of complements overtly realized as clitic pronouns (*Giovanni mi ha dato un libro* lit. 'Giovanni to–me has given a book' meaning that 'Giovanni gave me a book'). With pronominal verbs, in which the clitic can be seen as part of the verbal inflection, a specific dependency relation (`clit`) is resorted to link the clitic pronoun to the verbal head: for instance, in a sentence like *la sedia si è rotta* lit. 'the chair it is broken' meaning that 'the chair broke', the dependency linking the clitic *si* to the verbal head is `clit`.

4 The MIDT to SD conversion

The conversion process followed to generate the *Italian Stanford Dependency Treebank* (ISDT) starting from MIDT is based on the results of the comparative analysis reported in the previous section. It is organized in two different steps: the first one aimed at generating an enriched version of the MIDT resource, henceforth referred to as MIDT++, including SD–relevant distinctions neutralized in MIDT, and the second one in charge of converting the MIDT++ annotation in terms

of the Stanford Dependencies as described in de Marneffe and Manning (2008b) specialized with respect to the Italian language syntactic peculiarities. Note that also the resulting ISDT resource adheres to the CoNLL tabular format.

The first step relied on previous harmonization work leading to the construction of the MIDT resource starting from the CoNLL–compliant TUT and ISST–TANL treebanks (described in Bosco et al. (2012)). During this step, we recovered from the native resources relevant distinctions that have been neutralized in MIDT, because of choices made in the design of the MIDT annotation scheme (e.g. indirect objects or temporal modifiers which are assigned an underspecified representation in MIDT, see Section 3) or simply because the harmonization of the source annotation schemes was not possible without manual revision (this is the case of appositions, explicitly annotated only in TUT).

Other issues tackled during this first pre–processing step include the treatment of coordination and multi–word expressions. Since in SD conjunctions and conjuncts, after the first one, are all linked to the first conjunct, exactly as it was in ISST–TANL, the intermediate MIDT++ is generated according to this scheme, with no conversion for ISST–TANL and by restructuring the different cascading coordination style of TUT. For what concerns multi–word expressions, we unified the multi–word repertoires of the two resources. Another area that required some pre–processing with manual revision is concerned with the annotation of the parataxis relation. The augmented resource resulting from this pre–processing step, i.e. MIDT++, is used as a "bridge" towards the SD representation format.

Starting from the results of the comparative analysis detailed in Section 3, we defined conversion patterns which can be grouped into two main classes according to whether they refer to individual dependencies (case A) or they involve dependency subtrees due to head reassignment (case B).

A) ***Structure–preserving mapping rules*** involving dependency retyping without restructuring of the tree:

A.1) ***1:1 mapping*** requiring dependency retyping only (e.g. MIDT `prep` > SD `pobj`, or MIDT `subj` > SD `nsubj`);

A.2) ***1:n mapping*** requiring finer–grained dependency retyping (e.g. MIDT `mod` > SD `abbrev` | `amod` | `appos` | `nn` | `nnp` | `npadvmod` | `num` | `number` | `partmod` | `poss` | `preconj` | `predet` | `purplcl` | `quantmod` | `tmod`);

B) ***Tree restructuring mapping rules*** involving head reassignment and dependency retyping. Focusing on dependency retyping we distinguish the following cases:

B.1) head reassignment with ***1:1 dependency mapping*** (e.g. MIDT `subj` > SD `csubj` in the case of clausal subjects);

B.2) head reassignment with ***1:n dependency mapping*** based on finer–grained distinctions (e.g. MIDT `arg` > SD `xcomp` — `ccomp`, or MIDT `mod` (with verbal head) > SD `advcl` | `infmod` | `prepc` | `purpcl`).

In what follows, we will exemplify how the abstract patterns described above have been translated into MIDT_to_SD conversion rules. The conversion of the MIDT `arg` relation, referring to clausal complements subcategorized for by the governing head, represents an interesting example of 1:n dependency mapping with tree restructuring (case B.2 above). 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. The main conversion rules to SD can be summarised as follows, 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. $\$1[S|V|A] \rightarrow_{arg} \$2[E] \rightarrow_{prep} \$3[V_{infinitive}] \Rightarrow \$1 \rightarrow_{xcomp} \$3; \$3 \rightarrow_{aux} \$2$

2. $\$1[S|V|A] \rightarrow_{arg} \$2[CS] \rightarrow_{sub} \$3[V_{finite}] \Rightarrow \$1 \rightarrow_{ccomp} \$3; \$3 \rightarrow_{complm} \$2$

In the rules, the \$ followed by a number is a variable indentifying a given dependency node. Constraints on tokens in the left–hand side of the rule

(a) MIDT representation

(b) SD representation

Figure 2: MIDT vs SD annotation of the same sentence

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. Consider as an example the MIDT dependency tree in Figure 2(a) 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' whose SD conversion is reported in Figure 2(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 whereas in SD it is the verb; the relation linking the governing head and the head of the infinitival clause is `arg` in MIDT and `xcomp` in SD.

Currently, the conversion script implements over 100 rules which are still being tested with the final aim of finding the most appropriate representation with respect to the Italian syntactic peculiarities. The problematic area of sentential complements is still being explored to find out adequate representational solutions. Consider as an example the case of the word introducing infinitival complements: Figure 2(b) above, reporting the result of the SD conversion, shows that the same `aux` relation is used to link the preposition to the verb heading the infinitival complement as well as the auxiliary *avere* 'to have' to the main verb. This solution might not be so appropriate given the peculiarities of the Italian language, where different prepositions (lexically selected by the governing head) can introduce infinitival complements.

During the conversion step, the SD scheme has been specialized with respect to the Italian language. There are SD dependency relations which were 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 relations `ref` or `xsubj`). The SD tagset 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 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.

5 Using ISDT as training corpus

In this section, we report the results achieved by using ISDT for training a dependency parser, namely DeSR (Dependency Shift Reduce), a transition–based statistical parser (Attardi, 2006), where it is possible to specify, through a configuration file, the set of features to use (e.g. POS tag, lemma, morphological features) and the classification algorithm (e.g. Multi-Layer Perceptron (Attardi and Dell'Orletta, 2009), Support Vector Machine, Maximum Entropy). DeSR has been trained on TUT and ISST–TANL in the framework of the evaluation campaigns Evalita, for the last time in 2011 (Bosco and Mazzei, 2012; Dell'Orletta et al., 2012). More recently DeSR has been trained and tested on MIDT: the results ob-

Table 1: Parsing results with ISDT resources

TRAINING	TEST	PARSER	LAS	LAS no punct
TUT–SDT_train	TUT–SDT_test	DeSR MLP	84.14%	85.57%
ISST–TANL–SDT_train	ISST–TANL–SDT_test	DeSR MLP	80.55%	82.11%
TUT+ISST–TANL–SDT_train	TUT+ISST–TANL–SDT_test	DeSR MLP	83.34%	84.16%
TUT+ISST–TANL–SDT_train	TUT–SDT_test	DeSR MLP	84.14%	85.79%
TUT+ISST–TANL–SDT_train	ISST–TANL–SDT_test	DeSR MLP	79.94%	81.86%

tained on both the MIDT version of the individual TUT and ISST–TANL resources and the merged resource are reported in (Bosco et al., 2012): the best scores, achieved applying a parser combination strategy and training on TUT in MIDT format, are LAS 90.11% and LAS 91.58% without punctuation.

For the experiments on the ISDT resource we used a basic and fast variant of the DeSR parser based on Multi-Layer Perceptron (MLP). In fact, the purpose of the experiment was not to optimize the parser for the new resource but to compare relative performances of the same parser on different versions of the same resources. As a result, the substantial drop in performance observed with respect to the MIDT resource is in part due to this factor, and cannot be totally attributed to the greater complexity of the SD scheme or quality of the conversion output.

Table 1 reports, in the first two rows, the values of Labeled Attachment Score (LAS, with and without punctuation) obtained against the TUT–ISDT and ISST–TANL–ISDT datasets. The different performance of the parser on the two converted datasets (TUT–ISDT and ISST–TANL–ISDT) is in line with what was observed in previous experiments with native resources and MIDT (Bosco et al., 2010; Bosco et al., 2012); therefore, the composition of the training and test corpora can still be identified as possible causes for such a difference. The results reported in rows 3–5 have been obtained by training DeSR with the larger resource including both TUT–ISDT and ISST–TANL–ISDT. As test set, we used a combination of the two test sets (row 3) and test sets from the two data sets separately (rows 4 and 5). The preliminary results achieved by using ISDT are encouraging, in line with what was obtained on the WSJ for English and reported in (Cer et al., 2010), where the best results in labeled attachment precision, achieved by a fast dependency parser (Nivre Eager feature Extract), is 81.7. For the time being, training with the larger combined resource does not seem to provide a substantial advantage, confirming results obtained with MIDT, despite the fact that in the conversion from MIDT to ISDT a substantial effort was spent to further harmonize the two resources.

6 Conclusion

In this paper, we addressed the challenge of converting MIDT, an existing dependency–based Italian treebank resulting from the harmonization and merging of smaller resources adopting incompatible annotation schemes, into the Stanford Dependencies annotation formalism, with the final aim of constructing a standard–compliant resource for the Italian language. SD, increasingly acknowledged within the international NLP community as a *de facto* standard, was selected for its being defined with a specific view to supporting information extraction tasks.

The outcome of this still ongoing effort is three–fold. Starting from a comparative analysis of the MIDT and SD annotation schemes, we developed a methodology for converting treebank annotations belonging to the same dependency–based family. Second, Italian has now a new standard–compliant treebank, i.e. the *Italian Stanford Dependency Treebank* (ISDT, 200,516 tokens)[7]: we believe that this conversion will significantly improve the usability of the resource. Third, but not least important, we specialized the Stanford Dependency annotation scheme to deal with the peculiarities of the Italian language.

7 Acknowledgements

This research was supported by a Google "gift". Giuseppe Attardi helped with the experiments with the DeSR parser, Roberta Montefusco produced the converter to the collapsed/propagated version of ISDT and in so doing helped us to reduce inconsistencies and errors in the resource.

[7]Both the MIDT and ISDT resources are released by the authors under the Creative Commons Attribution–NonCommercial-ShareAlike 3.0 Unported licence (`http://creativecommons.org/licenses/by-nc-sa/3.0/legalcode.txt`).

References

G. Attardi and F. Dell'Orletta. 2009. Reverse revision and linear tree combination for dependency parsing. In *Proceedings of of NAACL HLT (2009)*.

G. Attardi. 2006. Experiments with a multilanguage non–projective dependency parser. In *Proceedings of the CoNLL-X '06*, New York City, New York.

C. Bosco and A. Mazzei. 2012. The evalita 2011 parsing task: the dependency track. In *Working Notes of Evalita'11*, Roma, Italy.

C. Bosco, V. Lombardo, L. Lesmo, and D. Vassallo. 2000. Building a treebank for italian: a data-driven annotation schema. In *Proceedings of the LREC'00*, Athens, Greece.

C. Bosco, S. Montemagni, A. Mazzei, V. Lombardo, F. Dell'Orletta, A. Lenci, L. Lesmo, G. Attardi, M. Simi, A. Lavelli, J. Hall, J. Nilsson, and J. Nivre. 2010. Comparing the influence of different treebank annotations on dependency parsing. In *Proceedings of the LREC'10*, Valletta, Malta.

C. Bosco, M. Simi, and S. Montemagni. 2012. Harmonization and merging of two italian dependency treebanks. In *Proceedings of the LREC 2012 Workshop on Language Resource Merging*, Istanbul, Turkey.

Sabine Buchholz and Erwin Marsi. 2006. Conll-x shared task on multilingual dependency parsing. In *In Proc. of CoNLL*, pages 149–164.

D. Cer, M.C. de Marneffe, D. Jurafsky, and C.D. Manning. 2010. Parsing to stanford dependencies: Trade-offs between speed and accuracy. In *Proceedings of the LREC'10)*, Valletta, Malta.

M.C. de Marneffe and C. Manning. 2008a. The stanford typed dependencies representation. In *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.

M.C. de Marneffe and C.D. Manning. 2008b. Stanford typed dependencies manual. Technical report, Stanford University.

M.C. de Marneffe, B. MacCartney, and C.D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of 5th International Conference on Language Resources and Evaluation (LREC 2006)*.

T. Declerck. 2008. A framework for standardized syntactic annotation. In *Proceedings of the LREC'08*, Marrakech, Morocco.

F. Dell'Orletta, S. Marchi, S. Montemagni, G. Venturi, T. Agnoloni, and E. Francesconi. 2012. Domain adaptation for dependency parsing at evalita 2011. In *Working Notes of Evalita'11*, Roma, Italy.

K. Haverinen, T. Viljanen, V. Laippala, S. Kohonen, F. Ginter, and T. Salakoski. 2010. Treebanking Finnish. In *Proceedings of the 9th Workshop on Treebanks and Linguistic Theories (TLT-9)*, pages 79–90, Tartu, Estonia.

R. Hudson. 1984. *Word Grammar*. Basil Blackwell, Oxford and New York.

N. Ide and L. Romary. 2006. Representing linguistic corpora and their annotations. In *Proceedings of the LREC'06*, Genova, Italy.

N. Ide and K. Suderman. 2007. GrAF: A graph-based format for linguistic annotations. In *Proceedings of the Linguistic Annotation Workshop*, Prague, Czech Republic.

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

S. Kübler, R.T. McDonald, and J. Nivre. 2009. *Dependency Parsing*. Morgan & Claypool Publishers, Oxford and New York.

John Lee and Yin Hei Kong. 2012. A dependency treebank of classical chinese poems. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 191–199, Montréal, Canada, June. Association for Computational Linguistics.

G. Leech, R. Barnett, and P. Kahrel. 1996. Eagles recommendations for the syntactic annotation of corpora. Technical report, EAG-TCWG-SASG1.8.

A. Lenci, S. Montemagni, V. Pirrelli, and C. Soria. 2008. A syntactic meta–scheme for corpus annotation and parsing evaluation. In *Proceedings of the LREC'00*, Athens, Greece.

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

S. Pyysalo, F. Ginter, K. Haverinen, J. Heimonen, T. Salakoski, and V. Laippala. 2007. On the unification of syntactic annotations under the Stanford dependency scheme: A case study on Bioinfer and GENIA. In *BioNLP 2007: Biological, translational, and clinical language processing*, pages 25–32, Prague.

M. Seraji, B. Megyesi, and J. Nivre. 2012. Bootstrapping a persian dependency treebank. *Special Issue of Linguistic Issues in Language Technology (LiLT) on Treebanks and Linguistic Theories*, 7.