Evolution of Italian Treebank and Dependency Parsing towards Universal Dependencies

Giuseppe Attardi
Dipartimento di Informatica
Università di Pisa
Largo B. Pontecorvo 3
56127 Pisa
attardi@di.unipi.it

Simone Saletti
Dipartimento di Informatica
Università di Pisa
Largo B. Pontecorvo 3
56127 Pisa
saletti@di.unipi.it

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

Abstract

English. We highlight the main changes recently undergone by the Italian Dependency Treebank in the transition to an extended and revised edition, compliant with the annotation schema of Universal Dependencies. We explore how these changes affect the accuracy of dependency parsers, performing comparative tests on various versions of the treebank. Despite significant changes in the annotation style, statistical parsers seem to cope well and mostly improve.

Italiano. *Illustriamo i principali cambiamenti effettuati sulla treebank a dipendenze per l'italiano nel passaggio a una versione estesa e rivista secondo lo stile di annotazione delle Universal Dependencies. Esploriamo come questi cambiamenti influenzano l'accuratezza dei parser a dipendenze, eseguendo test comparativi su diverse versioni della treebank. Nonostante i cambiamenti rilevanti nello stile di annotazione, i parser statistici sono in grado di adeguarsi e migliorare in accuratezza.*

1 Introduction

Universal Dependencies (UD) is a recent initiative to develop cross-linguistically consistent treebank annotations for several languages that aims to facilitate multilingual parser development and cross-language parsing (Nivre, 2015). An Italian corpus annotated according to the UD annotation scheme was recently released, as part of version 1.1 of the UD guidelines and resources. The UD-it v1.1 Italian treebank is the result of conversion from the ISDT (Italian Stanford Dependency Treebank), released for the shared task on dependency parsing of Evalita-2014 (Bosco et al., 2013 and 2014). ISDT is a resource annotated according to the Stanford dependencies scheme (de Marneffe et al. 2008, 2013a, 2013b), obtained through a semi-automatic conversion process starting from MIDT (the Merged Italian Dependency Treebank) (Bosco, Montemagni, Simi, 2012 and 2014). MIDT in turn was obtained by merging two existing Italian treebanks, differing both in corpus composition and adopted annotation schemes: TUT, the Turin University Treebank (Bosco et al. 2000), and ISST-TANL, first released as ISST-CoNLL for the CoNLL-2007 shared task (Montemagni and Simi, 2007).

UD can be considered as an evolution of the Stanford Dependencies into a multi-language framework and introduce significant annotation style novelties (deMarneffe et al., 2014). The UD schema is still evolving with many critical issues still under discussion, hence it is worthwhile to explore the impact of the proposed standard on parser performance, for example to assess whether alternative annotation choices might make parsing easier for statistically trained parsers.

For Italian we are in the position to compare results obtained in the Evalita 2014 DP parsing tasks with the performance of state-of-the-art parsers on UD, since both treebanks share a large subset of sentences.

Moreover, since UD is a larger resource than ISDT, we can also evaluate the impact of increasing the training set size on parser performance.

Our aim is to verify how differences in annotation schemes and in the corresponding training resources affect the accuracy of individual state-of-the-art parsers. Parser combinations, either

stacking or voting, can be quite effective in improving accuracy of individual parsers, as proved in the Evalita 2014 shared task and confirmed by our own experiments also on the UD. However our focus here lies in exploring the most effective single parser techniques for UD with respect to both accuracy and efficiency.

2 From ISDT to UD-it

In this section we highlight the changes in annotation guidelines and corpus composition between ISDT and UD-it.

2.1 Differences in annotation guidelines

The evolution of the Stanford Dependencies into a multi-language framework introduces two major changes (deMarneffe et al., 2014), concerning: (i) the treatment of copulas and (ii) the treatment of prepositions with case marking.

SD already recommended a treatment of the copula "to be" ("*essere*" in Italian) as dependent of a lexical predicate. In UD this becomes prescriptive and is motivated by the fact that many languages often lack an overt copula. This entails that the predicate complement is linked directly to its subject argument and the copula becomes a dependent of the predicate.

The second major change is the decision to fully adhere to the design principle of directly linking content words, and to abandon treating prepositions as a mediator between a modified word and its object: prepositions (but also other case-marking elements) are treated as dependents of the noun with specific *case* or *mark* labels.

The combined effect of these two decisions leads to parse trees with substantially different structure. Figure 1 and 2 show for instance the different parse trees, in passing from ISDT to UD annotations, for the sentence "È stata la giornata del doppio oro italiano ai Mondiali di atletica." [*It was the day of the Italian double gold at World Athletics Championships*.].

In fact exceptions to the general rule are still being discussed within the UD consortium, since the issue of copula inversion is somewhat controversial. In particular there are cases of prepositional predicates where the analysis with copula inversion leads to apparently counterintuitive situations. UD-it version 1.1 in particular does not implement copula inversion when the copula is followed by a prepositional predicate.

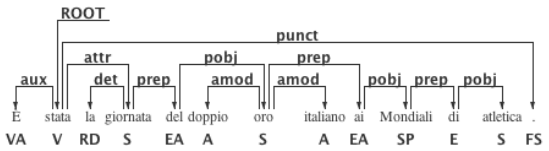

Figure 1. Example parse tree in ISDT

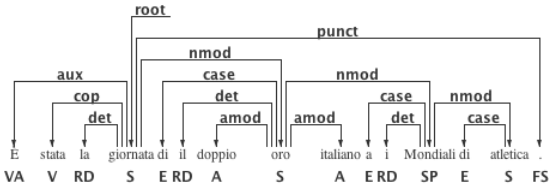

Figure 2. Example parse tree in UD1.1

Figure 3 illustrates the treatment advocated by strictly adhering to the UD guidelines, which is being considered for adoption in UD-it version 1.2. Notice that a quite different structure would be obtained for a very similar sentence like "La scultura appartiene al pachistano Hamad Butt" [*The sculpture belongs to the Pakistan Hamad Butt*].

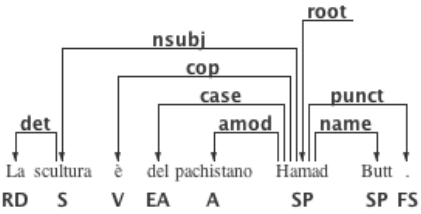

Figure 3. Example parse tree contemplated in UD 1.2

For the purpose of this presentation, we will call this version of the resource UD-it 1.2.[1]

Other changes in the annotation guidelines moving from ISDT and UD are less relevant for this discussion and involve the renaming of dependency labels, the introduction of special constructs for dealing with texts of a conversational nature (*discourse*, *vocative*) and the standardization of part-of-speech and morphological features.

2.2 Change of format

UD 1.1 also introduces an extension of the classical CoNLL-X tab separated format, called CoNNL-U. The main difference is the introduction of a notation for representing aggregated words (e.g. verbs with clitics or articulated prepositions): these can be split into their constituents and given as ID the range of the ID's of the constituents. An example from the guidelines is the following: "vámonos al mar" [*let's go to the sea*]:

1 By this we do not mean to imply that version 1.2 of UD-it, due in November 2015, will match exactly this conventions.

```
1-2     vámonos     _
1       vamos       ir
2       nos         nosotros
3-4     al          _
3       a           a
4       el          el
5       mar         mar
```

2.3 Corpus extension

The ISDT corpus released for Evalita 2014 consists of 97,500 tokens derived from the TUT and 81,000 tokens derived from the ISST-TANL. Moreover a gold test dataset of 9,442 tokens was produced for the shared task. UD-it is a larger resource including the previous texts (with converted annotations), a new corpus of questions, and data obtained from ParTUT[2] (the Multilingual Turin University Treebank) for a total of 324,406 tokens (13,079 sentences). For release 1.1, UD-it was randomly split into train, development and test data sets. Both development and test include 500 sentences each (~13,300 tokens).

3 Dependency parsers

We provide a short description of the state-of-the-art parsers chosen for our experiments.

DeSR was chosen as a representative of transition-based parsers for two main reasons, besides our own interest in developing this technology: given its declarative configuration mechanism it allows to experiment with different feature sets; other parsers in this category, in particular Maltparser (Nivre et al.), were consistently reported to provide inferior results in all Evalita evaluation campaigns for Italian.

3.1 DeSR

DESR MLP is a transition-based parser that uses a Multi-Layer Perceptron (Attardi 2006, Attardi et al., 2009a and 2009b). We trained it on 300 hidden variables, with a learning rate of 0.01, and early stopping when validation accuracy reaches 99.5%. The basic feature model used in the experiments on the Evalita training set is reported in Table 1.

The match expression indicates a feature to be extracted when a value matches a regular expression. Conditional features are used for representing linguistic constraints that apply to long distance dependencies. The feature used in the model takes into account a prepositional phrase (indicated by a dependent token with coarse POS of "E"), and it extracts a feature consisting of the pair: $b_0.l$ and the lemma of last preceding verb (a token whose POS is "V").

Single word features
$s_0.f\ b_0.f\ b_1.f$
$s_0.l\ b_0.l\ b_1.l\ b_0^{-1}.l\ lc(s_0).l\ rc(b_0).l$
$s_0.p\ b_0.p\ b_1.p\ rc(s_0).p\ rc(rc(b_0)).p$
$s_0.c\ s_0.c\ b_0.c\ b_1.c\ b_2.c\ b_3.c\ b_0^{-1}.c\ lc(s_0).c\ rc(b_0).c$
$s_0.m\ b_0.m\ b_1.m$
$lc(s_0).d\ lc(b_0).d\ rc(s_0).d$
$match(lc(b_0).m$, "Number=.")
$match(lc(b_0).m$, "Number=.")
Word pair features
$s_0.c\ b_0.c$
$b_0.c\ b_1.c$
$s_0.c\ b_1.c$
$s_0.c\ b_2.c$
$s_0.c\ b_3.c$
$rc(s_0).c\ b_0.c$
Conditional features
$if(lc(b_0).p$ = "E", $b_0.l)\ last$(POSTAG, "V")).l

Table 1. Feature templates: s_i represents tokens on the stack, b_i tokens on the input buffer. $lc(t)$ and $rc(t)$ denote the leftmost and rightmost child of token t, f denotes the form, l denotes the lemma, p and c the POS and coarse POS tag, m the morphology, d the dependency label. An exponent indicates a relative position in the input sentence.

Furthermore, an experimental feature was introduced, for adding a contribution from the score of the graph to the function of the MLP network. Besides the score computed by multiplying the probabilities of the transitions leading to a certain state, the score for the state reached for sentence x, after the sequence of transitions t, given the model parameters θ, is given by:

$$s(x, t, \theta) = \prod_{i=1}^{n} f_\theta(t_i) + E\left(x, t_1^i\right)$$

where $f_\theta(t)$ is the output computed by the neural network with parameters θ, and $E(x, t)$ is the score for the graph obtained after applying the sequence of transitions t to x. The graph score is computed from the following features:

Graph features
$b_0.l\ rc(b_0).p$
$b_0.l\ lc(b_0).p$
$b_0.l\ rc(b_0).p\ lc(rc(b_0)).p$
$b_0.l\ rc(b_0).p\ rc(rc(b_0)).p$
$b_0.l\ rc(b_0).p\ ls(rc(b_0)).p$
$lc(b_0).p\ b_0.l\ rc(b_0).p$
$b_0.l\ lc(b_0).p\ rc(lc(b_0)).p$
$b_0.l\ rc(b_0).p\ lc(lc(b_0)).p$
$b_0.l\ rc(b_0).p\ rs(lc(b_0)).p$
$rc(b_0).p\ b_0.l\ lc(b_0).p$

Table 2. A graph score is computed from these features. *ls* denotes the left sibling, *rs* the right sibling.

[2] http://www.di.unito.it/~tutreeb/partut.html

For the experiments on the UD corpus, the base feature model was used with 28 additional 3rd order features, of which we show a few in Table 3.

3rd order features
$s_0^{+1}.f\ b_0^{+2}.f\ b_0.p$
$s_0^{+2}.f\ b_0^{+3}.f\ b_0.p$
$s_0^{+2}.f\ b_0.f\ b_0.p$
$s_0^{+3}.f\ b_0^{+2}.f\ s_0.p$ …

Table 3. Sample of 3rd order features used for UD corpus.

3.2 Turbo Parser

TurboParser (Martins et al., 2013) is a graph-based parser that uses third-order feature models and a specialized accelerated dual decomposition algorithm for making non-projective parsing computationally feasible (cite). TurboParser was used in configuration "full", enabling all third-order features.

3.3 MATE Parser

The Mate parser is a graph-based parser that uses passive aggressive perceptron and exploits reach features (Bohnet, 2010). The only configurable parameter is the number of iterations (set to 25).

The Mate tools also include a variant that is a combination of transition-based and graph-based dependency parsing (Bohnet and Kuhn, 2012). We tested also this version, which achieved, as expected, accuracies that are half way between a pure graph-based and a transition-based parser and therefore they are not reported in the following sections.

4 Experiments

4.1 Evalita results on ISDT

The table below lists the best results obtained by the three parsers considered, on the Evalita 2014 treebank. Training was done on the train plus development data set and testing on the official test data set.

Parser	LAS	UAS
DeSR	84.79	87.37
Turbo Parser	86.45	88.98
Mate	86.82	89.18

Table 4. Evalita 2014 ISDT dataset

The best official results were obtained using a preprocessing step of tree restructuring and performing parser combination: 87.89 LAS, 90.16 UAS (Attardi and Simi, 2014).

4.2 Evalita dataset in UD 1.1

Our first experiment is performed on the same dataset from Evalita 2014, present also in the official UD-it 1.1 resource. We report in Table 5 the performance of the same parsers.

Parser	LAS	UAS	Diff
DeSR	85.57	88.68	+0.78
Turbo Parser	87.07	90.06	+0.62
Mate	88.01	90.43	+1.19

Table 5. Evalita 2014 dataset, UD-it 1.1 conventions

Using the resource converted in UD, the LAS of all the three parsers improved, as shown in the Diff column. This was somehow not expected since the tree structure is characterized by longer distance dependencies.

In fact a basic tree combination of these three parsers achieves 89.18 LAS and 91.28 UAS, an improvement of +1.29 LAS over the best Evalita results on ISDT.

5 Training with additional data

As a next step we repeated the experiment using the additional data available in UD-it 1.1 for training (about 71,000 additional tokens).

Parser	LAS	UAS	Diff
DeSR	85.19	88.18	-0.38
Turbo Parser	87.42	90.25	0.35
Mate	88.25	90.54	0.24

Table 6. Evalita 2014 dataset with additional training data, UD-it 1.1 conventions

The added training data do not appear to produce a significant improvement (Table 6). This may be due to the fact that the new data were not fully compliant with the resource at the time of release of UD1.1. Column Diff shows the difference with respect to the LAS scores reported in 4.2.

5.1 Evalita dataset in UD 1.2

The experiment in section 4.2 was repeated with UD-it 1.2, the version where copula inversion is performed also in the case of prepositional arguments. Table 7 also reports the difference with the LAS scores in 4.2.

Parser	LAS	UAS	Diff
DeSR	85.97	88.52	0.40
Turbo Parser	87.93	90.64	0.86
Mate	88.55	90.66	0.54

Table 7. Evalita 2014 dataset, UD-it 1.2 conventions

5.2 UD-it 1.1 dataset

The next set of experiments was performed with official release of the UD-it 1.1. Tuning of DeSR was done on the development data and the best parser was used to obtain the following results on the test data (Table 8).

Parser	Devel		Test	
	LAS	UAS	LAS	UAS
DeSR	88.28	91.13	87.93	90.78
Turbo Parser	89.99	92.48	89.77	92.46
Mate	91.24	93.05	90.53	92.59

Table 8. UD-it 1.1 dataset, partial copula inversion

5.3 UD-it 1.2 dataset

For completeness, we repeated the experiments with the UD-it 1.2 dataset (same data of UD-it 1.1, but complete copula inversion), obtaining even better results (Table 9).

Parser	Devel		Test	
	LAS	UAS	LAS	UAS
DeSR	89.09	91.40	89.02	90.39
Turbo Parser	89.54	92.10	89.40	92.17
Mate	90.81	92.70	90.22	92.47

Table 9. UD-it 1.1 dataset, complete copula inversion

5.4 Parser efficiency

Concerning parser efficiency, we measured the average parsing time to analyze the test set (500 sentences), employed by the three parsers under the same conditions. This also means that for MATE we deactivated the multicore option and used only one core. The results are as follows:

- DeSR: 18 seconds
- TurboParser: 47 seconds
- Mate: 2 minutes and 53 seconds

6 Conclusions

We have analyzed the effects on parsing accuracy throughout the evolution of the Italian treebank, from the version used in Evalita 2014 to the new extended and revised version released according to the UD framework.

General improvements have been noted with all parsers we tested: all of them seem to cope well with the inversion of direction of prepositional complements and copulas in the UD annotation. Improvements may be due as well to the harmonization effort at the level of PoS and morpho-features carried out in the process.

Graph based parsers still achieve higher accuracy, but the difference with respect to a transition based parser drops when third order features are used. A transition-based parser still has an advantage in raw parsing speed (i.e. disregarding speed-ups due to multithreading) and is competitive for large scale applications.

References

Giuseppe Attardi. 2006. Experiments with a Multilanguage Non-Projective Dependency Parser, Proc. of the Tenth Conference on Natural Language Learning, New York, (NY).

Giuseppe Attardi, Felice Dell'Orletta. 2009. Reverse Revision and Linear Tree Combination for Dependency Parsing. In: *Proc. of Human Language Technologies: The 2009 Annual Conference of the NAACL, Companion Volume: Short Papers*, 261–264. ACL, Stroudsburg, PA, USA.

Giuseppe Attardi, Felice Dell'Orletta, Maria Simi, Joseph Turian. 2009. Accurate Dependency Parsing with a Stacked Multilayer Perceptron. In: *Proc. of Workshop Evalita 2009*, ISBN 978-88-903581-1-1.

Giuseppe Attardi, Maria Simi, 2014. Dependency Parsing Techniques for Information Extraction, Proceedings of Evalita 2014.

Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *Proc. of Coling 2010*, pp. 89–97, Beijing, China. Coling 2010 Organizing Committee.

Bernd Bohnet and Jonas Kuhn. 2012. The Best of BothWorlds -- A Graph-based Completion Model for Transition-based Parsers. Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL), pages 77–87.

Cristina Bosco, Vincenzo Lombardo, Leonardo Lesmo, Daniela Vassallo. 2000. Building a treebank for Italian: a data-driven annotation schema. In Proceedings of LREC 2000, Athens, Greece.

Cristina Bosco, Simonetta Montemagni, Maria Simi. 2012. Harmonization and Merging of two Italian Dependency Treebanks, Workshop on Merging of Language Resources, in Proceedings of LREC 2012, Workshop on Language Resource Merging, Instanbul, May 2012, ELRA, pp. 23–30.

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

Cristina Bosco, Felice Dell'Orletta, Simonetta Montemagni, Manuela Sanguinetti, Maria Simi. 2014. The Evalita 2014 Dependency Parsing task, CLiC-it 2014 and EVALITA 2014 Proceedings, Pisa University Press, ISBN/EAN: 978-886741-472-7, 1–8.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. The Stanford typed dependencies representation. In COLING Workshop on Cross-framework and Cross-domain Parser Evaluation.

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

Marie-Catherine de Marneffe and Christopher D. Manning. 2013. Stanford typed dependencies manual, September 2008, Revised for the Stanford Parser v. 3.3 in December 2013.

Marie-Catherine De Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, Christopher D. Manning. 2014. Universal Stanford Dependencies: a Cross-Linguistic Typology. In: *Proc. LREC 2014*, Reykjavik, Iceland, ELRA.

Andre Martins, Miguel Almeida, and Noah A. Smith. 2013. Turning on the turbo: Fast third-order non-projective turbo parsers. In: *Proc. of the 51st Annual Meeting of the ACL* (Volume 2: Short Papers), 617–622, Sofia, Bulgaria. ACL.

Simonetta Montemagni, Maria Simi. 2007. The Italian dependency annotated corpus developed for the CoNLL–2007 shared task. Technical report, ILC–CNR.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2006. Maltparser: a data-driven parser-generator for dependency parsing. In Proceedings of LREC-2006, volume 2216–2219.

Joakim Nivre. 2015. Towards a Universal Grammar for Natural Language Processing, CICLing (1) 2015: 3–16

Maria Simi, Cristina Bosco, Simonetta Montemagni. 2008. Less is More? Towards a Reduced Inventory of Categories for Training a Parser for the Italian Stanford Dependencies. In: *Proc. LREC 2014*, 26–31, May, Reykjavik, Iceland, ELRA.