Comparing the Influence of Different Treebank Annotations on Dependency Parsing

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$, J. Nivre+

* Università di Torino, Italy, {bosco,mazzei,vincenzo,lesmo}@di.unito.it
† Istituto di Linguistica Computazionale - Pisa, {simonetta.montemagni, felice.dellorletta, alessandro.lenci}@ilc.cnr.it
◇Università di Pisa, {attardi,simi}@unipi.it
•FBK-irst - Trento, lavelli@fbk.eu
+Uppsala University, Sweden, {johan.hall,joakim.nivre}@lingfil.uu.se
$Vaxjo University, Sweden, jens.nilsson@vxu.se

Abstract

As the interest of the NLP community grows to develop several treebanks also for languages other than English, we observe efforts towards evaluating the impact of different annotation strategies used to represent particular languages or with reference to particular tasks. This paper contributes to the debate on the influence of resources used for the training and development on the performance of parsing systems.It presents a comparative analysis of the results achieved by three different dependency parsers developed and tested with respect to two treebanks for the Italian language, namely TUT and ISST–TANL, which differ significantly at the level of both corpus composition and adopted dependency representations.

1. Introduction

As the interest of the NLP community grows to develop several treebanks also for languages other than English, we observe efforts towards evaluating the impact of different annotation strategies used to represent particular languages or with reference to particular tasks. For instance, a recent line of research focuses on the question of whether and to what extent parsers developed with respect to different syntactic resources differ in their performance; this issue is tackled from different perspectives by, among others, (Nivre et al., 2007c), (Boyd and Meurers, 2008) and (Kübler et al., 2009). A comparison of results obtained by the same parsing system with respect to different treebanks for the same language can, in fact, help to assess the impact of different training resources following different annotation strategies at the parsing level. Nevertheless, the comparison among the results of systems developed on the basis of different resources is a very difficult task, first of all because of the number of variables usually involved, e.g. corpus composition and size, or different granularity in the representation of specific information.

The main goal of this paper is to contribute to the debate on the influence of training resources on the performance of parsing systems. Our methodology is based on a controlled experiment with different treebanks and parsers, and some common data for testing. In particular, we focus on the analysis of the results of three parsers which have been applied to two different treebanks. The resources involved are two treebanks developed for Italian, namely the Turin University Treebank[1] developed by the Natural Language Porcessing group of the University of Torino, and the ISST–TANL[2], an annotated corpus originating as a revision of the ISST–CoNLL corpus (Montemagni and Simi, 2007), in turn derived from the Italian Syntactic–Semantic Treebank or ISST (Montemagni et al., 2003). In spite of the fact that both treebanks feature a dependency–based annotation, they differ significantly at the level of both corpus composition and dependency annotation schemes, thus providing an interesting testbed to start evaluating the influence of treebanks on the parsing performance.

As a starting point, we assume the results achieved within Evalita'09[3], an evaluation campaign carried out for Italian, which included a dependency parsing track (Bosco et al., 2009) articulated into two subtasks differing at the level of treebanks: TUT was used as the development set in the Main Subtask, and ISST–TANL represented the development set for the Pilot Subtask. There have been five parsing systems which participated in both subtasks: two rule–based parsers (by Lesmo (2009) and by Testa et al. (2009)), and three statistical parsers, following different models (by Attardi et al. (2009), Lavelli et al. (2009) and Søgaard and Rishøj (2009)).

In this paper, we focus on the results obtained by the three systems which turned out to have achieved the best scores in the two subtasks, namely two statistical parsers (DeSR by Attardi et al., MaltParser by Lavelli et al.) and one rule–based parser (TULE by Lesmo). The performance of these systems appears to be in line with the state of the art dependency parsing technology for Italian (see tables 3 and 4 below)[4] and for English [5]. In particular, our aim is to in-

[1] http://i777777o o7 e-tut7 fb 7 .os ar.com/

[2] http://i d c 2o o7 e 70 o 7 .os ar.com/wiki/SemaWiki

[3] http://i 7 c 7 o 2 bo 7 .os ar.com/index

[4] The best results previously published for Italian are LAS 84.40, UAS 87.91, according to (Nivre et al., 2007b), were LAS (Labelled Attachment Score) represents the percentage of dependencies which are both correct and correctly labelled, and UAS (Unlabelled Attachment Score) the percentage of correct dependencies.

[5] The reported results for English are LAS 88.11 and UAS

vestigate the influence of the design of both treebanks and to evaluate the effectiveness of the assumed representations by testing these parsers on a common set of test data which has been annotated following both annotation schemes.
The paper is organized as follows. After a short description of TUT and ISST–TANL resources and of the selected parsing systems, we present the results achieved with respect to the two treebanks. A final section presents an across treebanks analysis of the influence of the different features of the used resources on the performance of the three selected parsing systems.

2. The data sets

The TUT and ISST–TANL resources differ under different respects, at the level of both corpus composition and adopted dependency representations, all having a potential impact on the parsing performance.

2.1. Size and composition of corpora

TUT currently includes 2,400 sentences (72,149 tokens in TUT native format, corresponding to 66,055 tokens in CoNLL format[6]) that represent various written text genres. They are organized in the following three sub–corpora: newspaper, i.e. texts from Italian newspapers and journals (1,100 sentences and 30,561 tokens); civill law, i.e. legal texts from the Italian Civil Law Code (1,100 sentences and 28,048 tokens); JRC-Passage, i.e. legal texts of the European Community extracted from the Italian section of the JRC-Acquis Multilingual Parallel Corpus[7] (200 sentences and 7,446 tokens) shared with the evaluation for French parsing Passage[8] that exploits texts from the corresponding French section of the same multilingual corpus.
ISST–TANL includes instead 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 and consisting of articles from newspapers and periodicals, selected to cover a high variety of topics (politics, economy, culture, science, health, sport, leisure, etc.).

2.2. Dependency annotation schemes

Although both TUT and ISST–TANL adopt a dependency scheme, they assume different inventories of dependency types characterised by different degrees of granularity in the representation of specific relations. Even when the dependency type appears to be the same, its coverage can differ significantly, due to different annotation criteria.
The different degree of granularity of the annotation schemes is testified by the size of the adopted dependency tagsets, including 72 dependency types in the case of TUT and 29 in the case of ISST–TANL. A difference in terms of granularity refers e.g. to the annotation of appositive (or unrestrictive) modifiers, which in TUT are annotated by resorting to a specific relation (`APPOSITION`), and which in ISST–TANL are not distinguished from other kinds of modifiers (`mod`). Similarly, TUT partitions predicative complements into two classes, i.e. subject and object predicative complements (`PREDCOMPL+SUBJ` and `PREDCOMPL+OBJ` respectively), depending on whether the complement refers to the subject or the object of the sentence. In ISST–TANL the same dependency type (`pred`) is used to annotate both cases since, at least as far as Italian is concerned, the subject–object predicative distinction can be inferred from contextual information: if the head of the predicative complement also includes among its dependents an object, then the predicative complement has to be interpreted as an object predicative complement; otherwise, it is a subject predicative complement. There are also cases in which ISST–TANL adopts finer–grained distinctions with respect to TUT: for instance, ISST–TANL envisages two different relation types for determiner–noun and preposition–noun constructions (`det` and `prep` respectively), whereas TUT represents both cases in terms of the same relation type (`ARG`). This latter example follows from another important dimension of variation between the two schemes, concerning head selection (see below).

Another interesting example can be found for what concerns the partitioning of the space of prepositional complements, be they modifiers or subcategorized arguments. TUT distinguishes between `MODIFIER`(s) on the one hand and subcategorised arguments on the other hand; the latter are further distinguished between indirect objects (`INDOBJ`) and all other types of indirect complements (`INDCOMPL`). ISST–TANL does not make an a priori distinction between subcategorised arguments and modifiers, which are subsumed under the same `comp` (mnemonic for complement) relation, thus allowing for the possibility of leaving the dependency type underspecified in those cases where the distinction is difficult to draw in practice. On the other hand, `comp`(lements) are further subdivided into semantically oriented categories, such as temporal, locative or indirect complements (`comp_temp`, `comp_loc` and `comp_ind`). In this case, the difference between TUT and ISST–TANL is not a matter of different degree of granularity at the level of representation but rather of orthogonal distinctions.

However, even when – at first glance – the two schemes show common dependency types, they can diverge at the level of their interpretation. This is the case, for instance, of the "obj" relation which in the TUT annotation scheme refers to the direct argument (either in the nominal or clausal form) occurring at least and most once and expressing the subcategorized object, and in ISST–TANL is meant to denote the relation holding between a verbal head and its non–clausal direct object (other dependency types are foreseen to mark clausal complements).

Another important dimension of variation between the TUT and ISST–TANL schemes concerns head selection: following word grammar (Hudson, 1984), TUT always assigns heads on the basis of syntactic criteria, i.e. in construc-

90.13 as in (Nivre et al., 2007a).

[6] In order to both enabling the application of standard evaluation measures, and increasing comparability with other resources, the CoNLL format (Buchholz and Marsi, 2006) has been applied to TUT data. With respect to the native TUT resource, that in CoNLL format mainly differs because it exploits only part of the rich set of the native TUT relations and does not include null elements.

[7] `http://i c e 77 o a72 o 7 .os ar. cam/JRC-Acquis.html`

[8] `http://i 7 f c co e72 o 72 .os ar.com/passage/index.html`

tions involving one function and one content word (e.g. determiner–noun and preposition–noun) the head role is played by the function word (the determiner and the preposition respectively). By contrast, in ISST–TANL head selection follows from a combination of syntactic and semantic criteria: i.e. whereas in the determiner–noun construction the head role is assigned to the semantic head (i.e. the noun), in the preposition–noun case, the head role is played by the preposition. This different strategy in head selection explains the asymmetric treatment of determiner–noun constructions with respect to preposition–noun ones in ISST–TANL and the fact the for TUT one dependency type (`ARG`) is sufficient.

Moreover, whereas TUT assumes the projectivity constraint[9], ISST–TANL corpus recognizes the need for non–projective representations due to the free word order property of the Italian language.

Further important differences between TUT and ISST–TANL concern the treatment of coordination and punctuation, phenomena which are particularly problematic to deal with in the dependency framework. In both resources, coordinated constructions are considered as asymmetric structures, but while in ISST–TANL the conjunction and the subsequent conjuncts are all linked to the first conjunct, in TUT the conjuncts starting from the second one are linked to the immediately preceding conjunction. In both treebanks punctuation is annotated: the main difference lies at the level of dependency types and head selection criteria. Whereas ISST-TANL has just one dependency type for punctuation tokens, TUT has many (4): for example, in TUT an explicit notion of parenthetical is marked, like e.g. in the Penn Treebank, while in ISST–TANL it is not.

Last but not least, distinct tokenization and sentence splitting criteria are assumed in the two resources with repercussions at different levels; e.g. TUT annotated sentences conform to the single root constraint, but in ISST–TANL there may be multiple–rooted sentences.

2.3. An annotation example

In order to give the reader the flavour of how and to what extent the two annotations differ, in tables 1 and 2 respectively we report the TUT (in CoNLL format) and ISST–TANL annotations for the same sentence *La coppia, residente a Milano anche se di origini siciliane, stava trascorrendo un periodo di vacanza*, 'The couple, living in Milan although of Sicilian origin, was having a period of holiday'.

By comparing tables 1 and 2, it can be noticed that differences lie at the level of both morpho–syntactic tagging and dependency annotation. If we focus on dependency annotation, we can observe to what extent the inventory of assumed dependency types represents an important dimension of variation. Consider the relation holding between the words *coppia* 'couple' and *residente* 'living': in TUT *residente* is interpreted as a modifier which is the head of a relative clause whereas in ISST–TANL it is treated as a modifier. Instead, the case of the object relation holding between the verb *trascorrere* and *periodo* in the ISST–TANL case and *un* in the TUT case, highlights the variation of the head assignment criteria between the two treebanks, since in TUT articles govern nouns, whereas in ISST–TANL the reverse holds. Annotations in tables 1 and 2 also reveal important differences in the treatment of punctuation. In the example, TUT recognizes a parenthetical structure between the two occurring commas and marks it with specific dependency types (`OPEN+PARENTHETICAL` and `CLOSE+PARENTHETICAL`); the head of the punctuation tokens coincides with the governing head of the subtree covering to the parenthetical structure (i.e. 2). In ISST–TANL, the same relation type is used in both cases, which is `punc`, and the two paired commas are both connected to the head of the delimited phrase (4).

2.4. TUT and ISST–TANL at Evalita'09

TUT and ISST–TANL, as described above, have been used in Evalita'09 as training/development sets in the two subtasks. For what concerns the test sets, in both cases they have been constructed to reflect the same balancing of text genres in the respective training corpora. The TUT and the ISST–TANL test sets were constituted respectively by 240 sentences (corresponding to 5,287 tokens) and by 260 sentences (5,011 tokens). Both test sets share a common set of 100 sentences (henceforth referred to as *shared test set*) extracted from newspapers (in particular from the balanced partition of ISST), which were newly annotated in TUT format for Evalita'09.

3. The parsing systems

The comparative analysis across treebanks has been carried out with respect to the three best performing parsing systems in Evalita'09. In the following section you can find a brief description of these systems.

3.1. DeSR

Attardi et al. (2009) used DeSR, a transition–based statistical parser that is trained on a treebank and learns which rules to apply for carrying out a Shift/Reduce algorithm. DeSR uses specific reduction rules that allow direct handling of non–projective dependencies, without a preprocessing step. Several algorithms can be used for training DeSR: in Evalita'09 both SVM and Multilayer Perceptron were used. For improving accuracy, a beam search strategy was used as well as parser combination. Three different parser configurations were used – namely a left to right DeSR, right to left DeSR, and a stacked Reverse Revision system. The latter uses hints extracted from the trees produced by a first parse in one direction while parsing the same sentence in the opposite direction. This significantly reduces errors due to long distance dependencies. The outputs of the three parsers were then combined using a greedy linear algorithm.

3.2. MaltParser

Lavelli et al. (2009) participated to the Dependency Parsing Task of Evalita'09 with a version of MaltParser[10], a system

[9] This constraint is maintained both in TUT native format, where non-projective constructions are reduced to the corresponding projective structures by using null elements, and in CoNLL one, which doesnt admit traces.

[10] http://www.maltparser.org/

1	La	IL	ART	ART	DEF—F—SING	14	SUBJ	_	_
2	coppia	COPPIA	NOUN	NOUN	COMMON—F—SING	1	ARG	_	_
3	,	#	PUNCT	PUNCT	_	2	OPEN+ PARENTHETICAL	_	_
4	residente	RISIEDERE	VERB	VERB	MAIN—PARTICIPLE—PAST—INTRANS—SING—ALLVAL	2	RMOD+ RELCL+REDUC	_	_
5	a	A	PREP	PREP	MONO	4	INDCOMPL	_	_
6	Milano	MILANO	NOUN	NOUN	PROPER—F—SING—CITY	5	ARG	_	_
7	anche	ANCHE	ADV	ADV	CONCESS	8	RMOD	_	_
8	se	SE	CONJ	CONJ	SUBORD—COND	4	RMOD	_	_
9	di	DI	PREP	PREP	MONO	8	ARG	_	_
10	origini	ORIGINE	NOUN	NOUN	COMMON—F—PL	9	ARG	_	_
11	siciliane	SICILIANO	ADJ	ADJ	QUALIF—F—PL	10	RMOD	_	_
12	,	#	PUNCT	PUNCT	_	2	CLOSE+ PARENTHETICAL	_	_
13	stava	STARE	VERB	VERB	AUX—IND—IMPERF—INTRANS—3—SING	14	AUX+ PROGRESSIVE	_	_
14	trascorrendo	TRASCORRERE	VERB	VERB	MAIN—GERUND—PRES—TRANS—SING	0	TOP	_	_
15	un	UN	ART	ART	INDEF—M—SING	14	OBJ	_	_
16	periodo	PERIODO	NOUN	NOUN	COMMON—M—SING	15	ARG	_	_
17	di	DI	PREP	PREP	MONO	16	RMOD	_	_
18	vacanza	VACANZA	NOUN	NOUN	COMMON—F—SING	17	ARG	_	_
19	.	#.	PUNCT	PUNCT	_				
14	END	_	_						

Table 1: An example of TUT annotation in CoNLL format.

1	La	lo	R	RD	num=s—gen=f	2	det	_	_
2	coppia	coppia	S	S	num=s—gen=f	13	subj	_	_
3	,	,	F	FF	_	4	punc	_	_
4	residente	residente	A	A	num=s—gen=n	2	mod	_	_
5	a	a	E	E	_	4	comp_loc	_	_
6	Milano	milano	S	SP	_	5	prep	_	_
7	anche_se	anche_se	C	CS	_	4	con	_	_
8	di	di	E	E	_	4	conj	_	_
9	origini	origine	S	S	num=p—gen=f	8	prep	_	_
10	siciliane	siciliano	A	A	num=p—gen=f	9	mod	_	_
11	,	,	F	FF	_	4	punc	_	_
12	stava	stare	V	VA	num=s—per=3—mod=i—ten=i	13	modal	_	_
13	trascorrendo	trascorrere	V	V	mod=g	0	ROOT	_	_
14	un	un	R	RI	num=s—gen=m	15	det	_	_
15	periodo	periodo	S	S	num=s—gen=m	13	obj	_	_
16	di	di	E	E	_	15	comp	_	_
17	vacanza	vacanza	S	S	num=s—gen=f	16	prep	_	_
18	.	.	F	FS	_	13	punc	_	_

Table 2: An example of ISST–TANL annotation in CoNLL format.

for data–driven dependency parsing that can be used to induce a parsing model from treebank data and to parse new data using the induced model. MaltParser implements the transition–based approach to dependency parsing, which has two essential components: (i) a nondeterministic transition system for mapping sentences to dependency trees; (ii) a classifier that predicts the next transition for every possible system conguration. Given these two components, dependency parsing can be performed as greedy deterministic search through the transition system, guided by the classier. With this technique, it is possible to perform parsing in linear time for projective dependency trees and quadratic time for arbitrary (non-projective) trees. Feature models developed in the CoNLL 2007 shared task were reused and four different parsing algorithms (Nivres arc-eager, Nivres arc-standard, Covingtons projective, and Covingtons non-projective) were evaluated. The best results were achieved with Covington's non-projective parsing algorithm (Covington, 2001).

3.3. TULE

TULE (Turin University Linguistic Environment) by Lesmo is a wide coverage rule-based parser, which has been applied to various domains and which has been the starting point for the development of the treebank TUT. Moreover, it has been the best scored parser in Evalita 2007 (Lesmo, 2007). The parsing process is based on two main steps: chunking and analysis of verbal dependents. Chunking consists in extracting relevant portions of a sentence (chunks) on the basis of highly reliable rules. The analysis of verbal dependents collect instead the chunks and attach them to verbs in order to build complete connected structures. This requires a selection of clause boundaries (accomplished via heuristic rules) and the decision about the role of the dependents, made via verbal subcategorization and a flexible representation of verbal case frames.

4. Results and evaluation

A comparative analysis of the results obtained by parsers developed with respect to different resources is a very difficult task mainly because of the number of involved variables. In our case, the task is made easier due to the availability of the shared test set of 100 sentences annotated in both TUT and ISST–TANL formats. In this case, the evaluation is free from the effects of variables that can crucially influence the evaluation based on test sets that are different, e.g., in terms of text genre or in frequency of less/more hard to parse phenomena. Nevertheless, because of the limited

size of our shared test set (100 sentences), we have to carefully balance the evaluation based on it with that based on the whole larger test set.
In what follows, we report the results of the three parsing systems on the full and shared test sets (see section 4.1.) of both treebanks, followed by a dependency–based analysis of the performance of the parsing systems with respect to TUT and in ISST–TANL annotations.

4.1. Overall performance of parsers

System evaluation, carried out in terms of LAS and UAS measures, is presented separately for each subtask, respectively in tables 3 and 4. In particular, the tables show achieved results with respect to the whole and shared test sets.

participant	**whole**		**shared**	
	LAS	UAS	LAS	UAS
TULE[11]	88.73	92.28	84.68	89.73
DeSR	88.67	92.72	82.60	89.17
MaltParser	86.5	90.88	79.91	87.15
average	87.97	91.96	82.40	88.69

Table 3: Performance results for TUT, whole test set and shared test set only.

participant	**whole**		**shared**	
	LAS	UAS	LAS	UAS
DeSR	83.38	87.71	84.67	88.99
MaltParser	80.54	84.85	81.12	85.02
TULE	73.44	80.80	75.12	82.58
average	79.12	84.45	80.30	85.53

Table 4: Performance results for ISST–TANL, whole test set and shared test set only.

Let us compare the results obtained in the two subtasks. If we focus on the performance achieved with respect to the whole TUT and ISST–TANL test sets, it can be noticed that the best results refer to TUT for both the best score (i.e. LAS +5.35 wrt ISST–TANL) and the average score (i.e. LAS +8.85 wrt ISST–TANL). But if the comparison is circumscribed to the shared test sets, no significant difference can be noticed between the best LAS scores in the two subtasks (TUT 84.68 vs ISST–TANL 84.67); for what concerns average results, the difference is much lower with respect to the whole test set, with TUT having +2.10 for LAS. By comparing the results obtained with respect to the whole and shared test sets, it should be pointed out that in the case of TUT the performance achieved on the shared test set is worst with respect to the whole test set, while in the case of ISST–TANL the reverse holds.
Since different parsing models can be differently influenced by the features of the annotations and text genres, it is important to also consider the performance of individual parsers with respect to the two treebanks. Focusing on the statistical systems only, i.e. DeSR and MaltParser, and comparing the performance achieved in the whole and shared test sets, we see that the LAS scores are higher in the shared test set for ISST–TANL, but they are much lower in the case of TUT. The reverse holds for the rule–based parser TULE which is the top parser wrt TUT but whose performance is significanly lower wrt ISST–TANL (both whole and shared test sets). This can be mainly motivated by the fact that the parser has been developed in parallel with TUT and may be not enough tuned on the other resource.
In the analysis of these results, it appears that various elements should be taken into account. such as the dependency annotation schemes used in the two resources and the composition of the training corpora. In principle, both issues can play a significant role in the parsers performance. In what follows we will focus on the influence of annotation schemes.

4.2. Dependency–based performance of parsers

With the aim of assessing the impact of annotation schemes on parsing results, we performed a dependency–based analysis of the performance of parsers. For each relation in the TUT and ISST–TANL dependency tagsets, we analyzed the performance of the three parsers in terms of Precision (P), Recall (R) and related f-score[12]. The analysis has been circumscribed to relations occurring at least 20 times[13] within the whole test sets.
In order to identify problematic areas of parsing, both TUT and ISST–TANL selected dependency–relations were partitioned into three classes with respect to the associated f-score, which could be taken to reflect their parsing difficulty. The three classes were defined as follows: we started from the results achieved by the best performing system in the two subtasks, i.e. respectively TULE for TUT and DeSR for ISST–TANL. We calculated the average of f-scores (av-f) obtained with respect to individual relations, which is 86.97 for TULE and 78.05 for ISST–TANL. We then found the thresholds for discriminating high, medium and low f-scores by averaging the scores respectively above and below the av-f value.

	TULE	DeSR
Low scored DR:	$\leq$77.53	$\leq$60.31
Medium scored DR:	77.54–94.22	60.32–89.44
Best scored DR:	$\geq$94.23	$\geq$89.45

Table 5: Thresholds for low, medium and high f-scores.

4.2.1. TUT: dependency–based performance

In the TUT test set, the low scored relations for all parsers are `APPOSITION` (which annotates unrestrictive modifiers and juxtappositions) and `INDOBJ` (indirect object), while for both the statistical parsers, i.e. MaltParser and

[12]The f-score formula we used is 2*(P*R)/(P+R).

[13]The average occurences for relation is around 106 in TUT test set and 179 in ISST–TANL test set.

DeSR, relations for the annotation of punctuation, such as SEPARATOR (which is used in cases where comma plays the role of disambiguating mark and an ambiguity could result if the mark were not there), OPEN+PARENTHETICAL and CLOSE+PARENTHETICAL (used for the annotation of paired punctuation of parenthetical clauses) also have to be included in the low scored relations' set. Instead, for the rule–based parser TULE, the low scores refer also to the PREDCOMPL+SUBJ (predicative complement of the subject) and COORD2ND+BASE (which introduces the second conjunct in coordinations).

The higher scored relations for all the parsers are instead very "local" dependencies such as CONTIN+LOCUT (used to linkparts of idiomatic expressions), END (which links the final punct to the sentence head), ARG (which annotates the arguments of preposition, articles and adjectives), and SUBJ/INDCOMPL (subject of passive verbs). The statistical systems did not achieved high scores with respect to other relations, while TULE shows a wider set of high scored relations, which includes relations for the annotation of auxiliary verbs (AUX+PASSIVE, which links the auxiliary to the main verb in passive clauses, and AUX+TENSE, which does the same in case of active clauses), TOP (which marks the root of the sentence), RMOD (which annotates the restrictive modifiers) and the relations used for the annotation of modifiers which are relative clauses (RMOD+RELCL and RMOD+RELCL+REDUC, respectively for full or reduced relatives).

These trends are generally confirmed in the shared test set for TUT, but, as figure 1 shows, in both statistical and rule–based approaches the performance decreases, in particular, with respect to the whole test set, in the case of the relations used for relative clauses and coordination. Moreover, for statistical systems it decreases also for predicative complements, and for the rule–based one it decreases for the relations exploited for punctuation, thus showing a more similar but negative result for all parsers. For what concerns the higher scored relations, the results in the shared test set are similar for all systems.

4.2.2. ISST–TANL: dependency–based performance

For what concerns ISST–TANL, the low scored relations shared by all parsers in the whole test set are just two, namely locative and temporal complements (comp_loc and comp_temp). There are three more relations, namely indirect complements (comp_ind, denoting the affected participant of an event), and con/conj (used to deal with coordinate structures) which belong to the class of hard to parse relation only in the case of TULE. However, they also show values very close to the low threshold value with DeSR and MaltParser. Whereas it is a widely acknowledged fact that coordinate structure analysis still represents a difficult area of parsing, the problems raised by the analysis of prepositional complements originate at a different level: this suggests that at the level of dependency parsing we do not have enough information for dealing with semantically oriented distinctions, or more simply that the dimension of the training corpus is not sufficient to reliably deal with them. We tried to neutralise the distinction among the different complement types by collapsing all of them into a unique class comp. The result in this case changed significantly: with an underspecified comp relation the f-score of the different systems increased to 83.94 for DeSR, 82.49 for MaltParser and 76.58 for TULE. For TULE, the rule–based parser, there are other hard to parse relations, involving the treatment of punctuation, clausal arguments, and finer–grained distinctions at the level of modifiers and subjects.

As already observed for TUT, the best scored relations for all parsers include very "local dependencies": i.e. those linking auxiliaries to the verbal head (aux), determiners to the nominal head (det), modal verbs to the verbal head (modal), nouns to prepositions (prep), negative modifiers to their head (neg). For the statistical parsers only, the best scored relations include also: the relation linking complementizers to the verbal head they introduce (sub) and clitic pronouns to their verbal head (clit).

The general trend depicted above is confirmed at the level of the shared test set. As it can be observed in figure 2, besides a few exceptions the f–scores by all parsers are higher: this can be explained with the fact that the shared test set is easier to parse with respect to the remaining test sentences (due to the constraint for which the sentence length in the shared test set could not exceed 40 tokens). If we take the sentence length to be indicative – at least to some extent – of the linguistic complexity of the corpus, we observe an average sentence length 17.16 tokens in the shared test set against 20.59 in the remaining sentences.

5. Discussion

We have seen that the differences between the two resources mainly lie at two different levels, namely the composition of the training corpora and the adopted annotation schemes. The discussion will be therefore organised around these two different issues.

For what concerns the former, it emerges clearly that the composition of the corpora has some impact on the parsing performance. In particular, the lower results on the TUT shared portion of the test set have to be interpreted in this sense, i.e. as a consequence of the text genre of the sentences included in the shared test set. While the training for ISST–TANL is based on sentences belonging exactly to the same text genre as those included in the shared test set, TUT training corpus does not provide enough evidence to tackle some of the linguistic constructions occurring in the shared test set.

For what concerns instead the annotation schemes, the impact of projectivity cannot be considered as significant mainly because of the very low frequency of non–projective constructions (i.e. 3 in the shared and 16 in the whole test sets), whereas more relevant effects seem to be caused by the different dependency types and annotation strategies adopted in TUT and ISST–TANL thus confirming that the analysis has to be referred to both single relations and complex linguistic constructions, like Kübler et al. (2009). The latter represents a crucial but too often underestimated issue: annotation schemes are not all equal, when they are used to create data for the training of statistical parsers and also for the development of rule–based ones. It could perhaps be the case that some syntactic distinctions encoded

in one annotation scheme can not be easily learned by the parser, or simply that they are too sparse in the training data, which therefore should be enlarged in a significant way. Whatever the specific reason of the different performances of the systems in the shared test set, the results suggest the need for some deeper reflections on parsing annotation schemes, showing that the improvement of parsing technology should proceed hand in hand with the development of more suitable representations for annotated syntactic data.

Interestingly enough, the dependency–based analysis reported in section 4.2. shows that similar trends can be observed in the performance of parsers against TUT and ISST–TALN. First, in both cases hard to parse relations include "semantically loaded" relations such as `comp_temp`, `comp_loc` and `comp_ind` for ISST–TALN and `APPOSITION` and `INDOBJ` for TUT. Moreover, relations involving punctuation appear to be difficult to parse for statistical parsers in the case of TUT, whereas TULE has problems dealing with coordinate structures in ISST–TALN; it should be noted however that ISST–TALN `con`/`conj` relations show values very close to the low threshold value also in the case of DeSR and MaltParser. Our contrastive analysis confirms a widely acknowledged claim, i.e. that coordination and punctuation phenomena still represent particularly challenging areas for parsing (Cheung and Penn, 2009): to improve their treatment in both treebanks further investigation is needed. The problems raised by the analysis of "semantically loaded" relations in the case of both treebanks suggest that the parsers do not appear to have sufficient evidence to deal reliably with them; the solutions to the problem range from increasing the size of the training corpus, to postponing their treatment to further processing levels. Again, further analysis is needed to identify an appropriate solution to the problem.

Concerning the best scored relations, it came out that in both cases they mainly refer to "local" relations. Interestingly to note, there is a significant overlapping between the two sets: e.g. the TUT `ARG` and the ISST–TANL `det`/`prep` together have the same coverage; the same holds for the TUT `AUX+PASSIVE`/ `AUX+TENSE` relations with respect to the ISST–TANL `aux` relation.

6. Conclusions and future work

The paper contributes to the debate about the influence of training resources and their annotation schemes on the performance of parsing systems. Starting from the results of a set of parsers for Italian with results close to the state of the art, we developed a comparative analysis of two Italian dependency-based treebanks, i.e TUT and ISST–TANL. Our analysis reveals various factors of the training corpora which influenced the results of parsing systems, e.g. corpus composition and peculiarities of the annotation schemes. In particular, we performed a fine–grained observation of the relations in the treebanks by distinguishing them in three score classes, i.e. low scores that identify hard to parse relations, high scores that identify easy to parse relations and the medium scores that identify the remaining relations.

We are well aware that there are many issues left open by our analysis. Further analysis should be performed in order to find the missing answers. An important contribution will come from the development of a larger shared set of data. Moreover, a deeper linguistic comparison between the two resources can be based e.g. on the development of tools for the conversion among the involved formats. A more detailed analysis of the errors of each single parser can also produce interesting data for the development of parsing methodologies.

7. References

G. Attardi, F. Dell'Orletta, M. Simi, and J. Turian. 2009. Accurate dependency parsing with a stacked multilayer perceptron. In *Proceedings of Evalita'09*, Reggio Emilia.

C. Bosco, S. Montemagni, A. Mazzei, V. Lombardo, F. Dell'Orletta, and A. Lenci. 2009. Evalita'09 parsing task: comparing dependency parsers and treebanks. In *Proceedings of Evalita'09*, Reggio Emilia.

A. Boyd and D. Meurers. 2008. Revisiting the impact of different annotation schemes on PCFG parsing: a grammatical dependency evaluation. In *Proceedings of the ACL Workshop on Parsing German - PaGe '08*, Morristown, NJ, USA.

S. Buchholz and E. Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of CoNLL-X*.

J.C.K. Cheung and G. Penn. 2009. Topological field parsing of German. In *Proceedings of ACL-IJCNLP'09*.

M. A. Covington. 2001. A fundamental algorithm for dependency parsing. In *Proceedings of the 39th Annual ACM Southeast Conference*.

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

S. Kübler, I. Rehbein, and J. van Genabith. 2009. TePaCoC - a corpus for testing parser performance on complex German grammatical constructions. In *Proceedings of the Seventh International Workshop on Treebanks and Linguistic Theories*, Groningen, The Netherlands.

A. Lavelli, J. Hall, J. Nilsson, and J. Nivre. 2009. MaltParser at the EVALITA 2009 dependency parsing task. In *Proceedings of Evalita'09*, Reggio Emilia.

L. Lesmo. 2007. The rule-based parser of the NLP group of the University of Torino. *Intelligenza Artificiale*, 12.

L. Lesmo. 2009. The Turin University parser at Evalita 2009. In *Proceedings of Evalita'09*, Reggio Emilia.

S. Montemagni and M. Simi. 2007. The Italian dependency annotated corpus developed for the CoNLL–2007 shared task. Technical report, ILC–CNR. `http://i777777o c o e72o 7 .os ar.com/tressi\CoNNL2007/ISST/ISST@CoNNL2007.pdf`.

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 Anne Abeillé, editor, *Building and Using syntactically annotated corpora*. Kluwer, Dordrecht.

J. Nivre, J.H. Hall, and A. Chanev. 2007a. MaltParser:

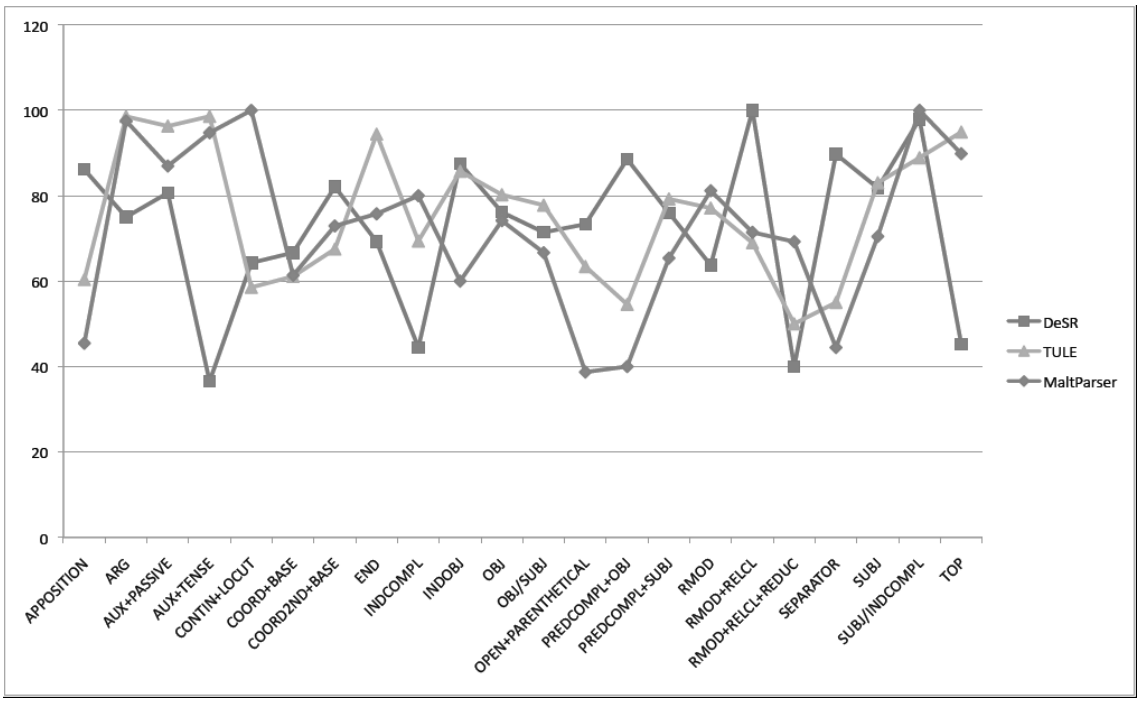

Figure 1: Dependency–based performance of parsers wrt TUT: F-scores obtained in the shared test set.

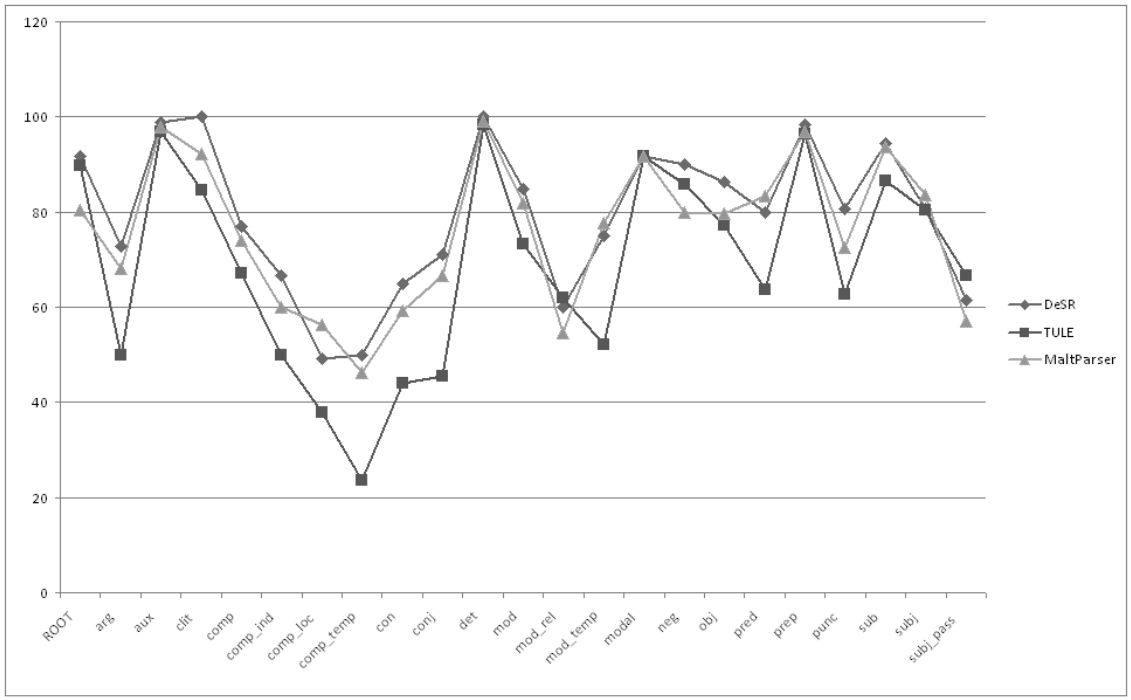

Figure 2: Dependency–based performance of parsers wrt ISST–TANL: F-scores obtained in the shared test set.

a language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2).

J. Nivre, J. Hall, S. Kübler, R. McDonald, J. Nilsson, S. Riedel, and D. Yuret. 2007b. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the EMNLP-CoNLL*.

J. Nivre, J. Nilsson, and J. Hall. 2007c. Generalizing tree transformations for inductive dependency parsing. In *Proceedings of the ACL*.

A. Søgaard and C. Rishøj. 2009. Vine parsing augmented treebanks. In *Proceedings of Evalita'09*, Reggio Emilia.

M. Testa, A. Bolioli, L. Dini, and G. Mazzini. 2009. Evaluation of a semantically oriented dependency grammar for Italian at Evalita 2009. In *Proceedings of Evalita'09*, Reggio Emilia.