Improvements in Parsing the *Index Thomisticus* Treebank. Revision, Combination and a Feature Model for Medieval Latin

Marco Passarotti[1], Felice Dell'Orletta[2]
[1]Università Cattolica del Sacro Cuore
Largo A. Gemelli 1, 20123 Milan, Italy
[2]Istituto di Linguistica Computazionale, Consiglio Nazionale delle Ricerche
Via G. Moruzzi 1, 56124 Pisa, Italy
E-mail: marco.passarotti@unicatt.it, felice.dellorletta@ilc.cnr.it

Abstract

The creation of language resources for less-resourced languages like the historical ones benefits from the exploitation of language-independent tools and methods developed over the years by many projects for modern languages. Along these lines, a number of treebanks for historical languages started recently to arise, including treebanks for Latin. Among the Latin treebanks, the *Index Thomisticus* Treebank is a 68,000 token dependency treebank based on the *Index Thomisticus* by Roberto Busa SJ, which contains the *opera omnia* of Thomas Aquinas (118 texts) as well as 61 texts by other authors related to Thomas, for a total of approximately 11 million tokens. In this paper, we describe a number of modifications that we applied to the dependency parser DeSR, in order to improve the parsing accuracy rates on the *Index Thomisticus* Treebank. First, we adapted the parser to the specific processing of Medieval Latin, defining an ad-hoc configuration of its features. Then, in order to improve the accuracy rates provided by DeSR, we applied a revision parsing method and we combined the outputs produced by different algorithms. This allowed us to improve accuracy rates substantially, reaching results that are well beyond the state of the art of parsing for Latin.

1. Introduction

The creation of language resources for less-resourced languages like the historical ones benefits from the exploitation of language-independent tools and methods developed over the years by many projects for modern languages. Along these lines, a number of treebanks for historical languages started recently to arise, including treebanks for Classical languages like Latin and Greek.

The *Index Thomisticus* Treebank (IT-TB; http://itreebank.marginalia.it) is a 68,000 token Latin dependency treebank based on the *Index Thomisticus* (IT) by Roberto Busa SJ (1974-1980). The IT is a database containing the *opera omnia* of Thomas Aquinas (118 texts) as well as 61 texts by other authors related to Thomas, for a total of approximately 11 million tokens. The corpus is morphologically tagged.

In the context of the IT-TB project we have trained and tested a number of probabilistic dependency parsers, using IT-TB data as training and test sets. Once we identified the best performing parser, we adapted it to the specific processing of Medieval Latin, defining an ad-hoc configuration of its features. Then, in order to improve the accuracy rates provided by the parser, we applied a revision parsing method which learns how to reduce several errors of the parser by means of a second parser that analyses the sentence in reverse, employing additional features obtained from the output of the first parser. Finally, we combined the outputs produced by the different algorithms.

This allowed us to improve accuracy rates substantially, reaching results that are well beyond the state of the art of parsing for Latin.

2. Background

Despite its pioneering role in computational linguistics due to the IT itself, today Latin still lacks powerful NLP tools.

In particular, as far as syntactic tagging is concerned, we are aware of only two Latin parsers, both rule-based. The first is a dependency parser described by Koch (1993), who reports on the enhancement for Latin of an existing dependency parser (Covington, 1990), but no evaluation is provided. The second parser is reported by Koster (2005) as a rule-based top-down chart parser automatically generated from a grammar and a lexicon built according to the formalism of the two-level AGFL grammar (Affix Grammar over a Finite Lattice (Koster, 1991)). The parser is tested on the *Confessiones* of Augustine and on several texts of Caesar; the results are evaluated in terms of the number of words covered by the parser, reaching the following rates: 82.8 (Augustine) and 75.3 (Caesar). Recently, a hybridisation of this parser has been developed, extending the rule-based core parser with a probability-based ranking of dependency trees, through the statistics of dependency triplets generated by the parser itself.

However, the start of two projects for treebanking Latin texts in 2005 nowadays provides data that can be used to train data-driven NLP tools, such as parsers. These projects are the IT-TB (McGillivray et al., 2009) and the Latin Dependency Treebank (LDT; Bamman & Crane, 2007), the latter focused on texts of the Classical era and having a size of approximately 55,000 tokens. Both the treebanks are dependency-based and they share the same annotation guidelines (Bamman et al., 2007), which follow the annotation style developed for the 'analytical

layer' by the Prague Dependency Treebank of the Czech language (PDT; Hajič et al., 1999)[1].
Bamman and Crane (2008) report on the evaluation of the MST parser (McDonald & Pereira, 2006) trained on the LDT data. The training set size was approximately 47,000 tokens. They performed the evaluation on two different test sets, one provided with disambiguated morphological tags ('gold') and one for which the morphological tags were automatically assigned ('automatic') by a trained Part-of-Speech tagger (TreeTagger (Schmid, 1994)), showing accuracy rates of around 95% on part of speech (PoS). The accuracy rates they report are shown in Table 1 according to two evaluation measures, called respectively 'unlabeled' (correct head) and 'labeled' (correct head and syntactic label).

	Unlabeled	**Labeled**
Gold	64.99	54.34
Automatic	61.49	50

Table 1: Accuracy rates on LDT.

In our previous work (Passarotti & Ruffolo, forthcoming), we reported on the accuracy rates for four dependency parsers trained on the IT-TB data[2]. Data were PoS-tagged. The training set size was 44,195 tokens, while the test set size was 5,697 tokens. Table 2 shows their accuracy rates according to the evaluation measures adopted in the CoNLL-X Shared Task (Buchholz & Marsi, 2006):

- Labeled Attachment Score (LAS): the percentage of tokens with correct head and relation label;
- Unlabeled Attachment Score (UAS): the percentage of tokens with the correct head;
- Label Accuracy (LA): the percentage of tokens with the correct relation label.

Parser	**LAS**	**UAS**	**LA**
DeSR	71.26	78.35	81.07
Malt	69.85	75.87	81.74
ISBN	68.97	77.79	78.88
MST	68.79	79.43	79.35

Table 2: Accuracy rates on IT-TB.

3. Improving Parsing Performance

According to the results provided in Table 2, we decided to use the shift-reduce parser DeSR as the base parser to which we would apply modifications, in order to improve parsing performance on the IT-TB data.

3.1 Data Description

For training and evaluation purposes, we randomly partitioned the IT-TB data (all PoS-tagged) into a training set and a test set with a size ratio of approximately 9:1. Table 3 reports the size of this data in terms of sentences and tokens.

Data Sets	**Sentences**	**Tokens**
Training	2,820	61,024
Test	329	7,379

Table 3: Training and test sets.

3.2 A Feature Model for Medieval Latin

DeSR provides a feature model, which is used by the machine learning classifier in the training phase. This model is described in a configuration file. Each line of such file reports which feature has to be extracted from which token(s). The notation is the following:

$$\text{Feature type} \quad \text{token}_1 \ldots \text{token}_n$$

Here, type is one attribute of a token. The types are the ones adopted in the CoNLL-X format. The tokens are defined through path expressions, moving from one numbered token. Tokens in the input queue are positively numbered starting from 0, while tokens on the stack are negatively numbered. *LeftChild(x)* and *rightChild(x)* refer respectively to the leftmost and rightmost child of token *x*; *head(x)* to the head of *x*; and *prev(x)* to the token preceding *x* in the sentence (Attardi & Dell'Orletta, 2008).
The training of shift-reduce parsers depends heavily on the selection of the feature model. Since setting the best feature model for the processing of a specific language requires an ad-hoc investigation, in our previous work we just selected the best fitting model among the ones available from the CoNLL-X Shared Task. In particular, we tested the feature models for Czech, English and Italian. We found that the feature models for Czech and for Italian performed considerably better than the one for English[3]. After having achieved preliminary results through the exploitation of feature models designed for modern languages, we decided to create a specific model for Medieval Latin (and, particularly, for the Latin of Thomas Aquinas in the IT texts).
We tested 14 different feature models, which were created by changing the type-token combinations and the set of adopted features. For comparison purposes, the training phase with these models was performed on the same training set we used in our previous experiments (44,195 tokens). The best performing feature model was the

[1] A third Latin treebank is today available. This treebank is being developed by the PROIEL project at the University of Oslo (Pragmatic Resources in Old Indo-European Languages). The project is aimed at the dependency annotation of the oldest extant versions of the New Testament in Indo-European languages: Latin, Greek, Gothic, Armenian and Old Church Slavonic (Haug & Jøndal, 2008). The size of the Latin portion of the PROIEL corpus is approximately 100,000 annotated tokens. The PROIEL annotation style differs from IT-TB and LDT in some minor details, and conversion is possible.

[2] The parsers were the following: DeSR (Attardi, 2006), Malt (Nivre & Nilsson, 2005), ISBN (Titov & Henderson, 2007) and MST (McDonald & Pereira, 2006).

[3] The accuracy rates reported in Table 2 for the shift-reduce parsers (DeSR, Malt and ISBN) result from adopting the feature model for Italian.

following[4].

Feature	Tokens
LEMMA	-2 -1 0 1 2 3 *prev*(0) *next*(-1) *leftChild*(-1) *leftChild*(0) *rightChild*(-1) *rightChild*(0)
POSTAG	-2 -1 0 1 2 3 *prev*(0) *next*(-1) *leftChild*(-1) *leftChild*(0) *rightChild*(-1) *rightChild*(0)
CPOSTAG	-1 0 1 2
FEATS	-1 0 1 2
DEPREL	*rightChild*(-1)
HEAD	-1 0

Table 4: Feature model for Medieval Latin.

This feature model is quite similar to the ones for Czech and for Italian (which, in turn, resemble other models, such as those for Greek, Hungarian and Slovene). In comparison with those two models, we added the feature HEAD to the set of the features and we made some changes in the set of tokens, mostly collating the Czech and Italian models. The only setting that is not common with at least one of those two models is that we also considered the token numbered 2 for both the CPOSTAG and FEATS features, while only tokens numbered -1, 0 and 1 are considered for these features in the Czech and Italian models.

The application of DeSR trained using the best performing feature model (and tested on the previously adopted test set: 5,697 tokens) increased the accuracy rates by approximately 2% for all the evaluation metrics, as Table 5 reports.

Parser	Config.	FM	LAS	UAS	LA
DeSR	SVM-LR	Latin	73.73	79.90	83.10

Table 5: Accuracy rates using Medieval Latin feature model (old training set).

Once the best scoring feature model had been selected, this was used to train DeSR on the new training set (61,024 tokens). In addition to the default learning configuration of DeSR (Support Vector Machine (SVM) - left-to-right (LR)), we also used a right-to-left one (RL). This led to a further improvement of the accuracy rates (reached on the new test set: 7,379 tokens), as shown in Table 6[5].

Parser	Config.	FM	LAS	UAS	LA
DeSR	SVM-RL	Latin	78.26	83.90	86.75
DeSR	SVM-LR	Latin	76.31	82.38	85.09

Table 6: Accuracy rates using Medieval Latin feature model (new training set).

3.3 Revision and Combination

To further improve the accuracy rates by DeSR, we applied a parsing revision technique called 'Reverse Revision Parsing' (Attardi & Dell'Orletta, 2009), which learns how to correct the parsing errors, thereby producing a parse reviser.

This technique involves two steps. First, the sentence is parsed by a deterministic shift-reduce parser; then, a second deterministic shift-reduce parser analyses the sentence in reverse mode, using additional features extracted from the parse tree produced by the first parser.

DeSR provides two different reverse revision algorithms, named respectively 'rev2' and 'rev3'.

Both the parsing algorithms are trained on a training set extended with dependency information predicted by a lower accuracy parser (in this case, DeSR with the Maximum Entropy classification algorithm). Attardi and Dell'Orletta (2009) claim that the reason behind this is that the output of a low accuracy parser (with many errors) is a better source of learning for the stacked parser.

Both rev2 and rev3 use LR and RL parsers based on a SVM learning algorithm. The difference between rev2 and rev3 is that rev2 employs an LR shift-reduce parser to parse the sentence and, then, a second RL shift-reduce parser scans the sentence in reverse order using additional features, which are obtained from the prediction made by the first parser. On the contrary, rev3 involves a LR revision parser that uses the additional features obtained from the RL parser.

The additional features extracted from the first parsing step and used by rev2 and rev3 are the following:

- PHLEMMA: the lemma of the predicted head;
- PHPOS: the PoS of the predicted head;
- PDEP: the predicted dependency label of a token in relation to its predicted head;
- PHLOC: indicates whether a token is located before or after its predicted head;
- PHHLEMMA: the lemma of the predicted grandparent;
- PHDEP: the predicted dependency label of the predicted head of a token in relation to the predicted grandparent of the token.

As with the basic feature model, we tested a number of different feature models (8), changing the type-token combinations and the set of adopted features. The best performing feature model is reported in Table 7.

Feature	Tokens
PHLEMMA	0 1
PHPOS	0 1
PDEP	-1 0
PHLOC	0
PHHLEMMA	0 1
PHDEP	0 1

Table 7: Additional feature model used in training the revision parser.

A combined feature model resulting from the union of the basic and the additional feature models was used in training the reverse revision parser.

[4] CPOSTAG: coarse-grained PoS; FEATS: morphological features; DEPREL: dependency relation.

[5] It is remarkable that the RL configuration outperforms the LR configuration by approximately 2% (LAS).

This technique has already been applied to several language with different properties (Attardi & Dell'Orletta, 2009). In particular, we report in Table 8 the accuracy rates provided by the reverse revision parser on three selected languages: English, Italian and Czech.

Parser	Language	LAS	UAS
DeSR-LR	English	86.51	87.65
DeSR-RL	English	85.21	86.61
DeSR-rev2	English	88.27	89.45
DeSR-LR	Italian	81.4	85.38
DeSR-RL	Italian	82.89	86.95
DeSR-rev2	Italian	83.52	87.44
DeSR-LR	Czech	77.12	82.96
DeSR-RL	Czech	78.2	84.48
DeSR-rev2	Czech	79.95	85.18

Table 8: LAS and UAS for selected languages.

The training and test sets used to obtain such results are those supplied for the CoNLL 2007 Shared Task (Nivre et al., 2007). They are reported in Table 9[6].

Language	Training	Test
English	447K (18.6K)	5,003 (214)
Italian	71K (3.1K)	5,096 (249)
Czech	432K (25.4K)	4,724 (286)

Table 9: Training and test sets for selected languages.

These languages were selected in order to show the accuracy rates provided by the parser on languages showing different linguistic properties.

English is a language with minimal inflection and presents a fixed word-order. These two linguistic properties, together with the large dimension of the training corpus, make English the language that is best suited to the parser.

Italian is an inflected language showing a moderately free word-order. Unlike Czech, Italian does not have cases (with only the marginal exception of personal pronouns) and its phrase-order freedom is substantially lower (Dell'Orletta et al., 2006). Moreover, the CoNLL training set is much smaller than those of the other two selected languages.

Czech is a highly inflected language and presents a moderately free word-order. Despite a large training set, these features make Czech the language for which the lowest accuracy rates are provided.

Latin shares some relevant properties with Czech, such as being richly inflected, showing discontinuous phrases (non-projectivity) and a moderately free word-order, and having an high degree of synonymity and ambiguity of the endings. Both languages have three genders (masculine, feminine, neuter), cases with roughly the same meaning and no articles.

Just as with the other languages, it was also the case for Latin that the application of Reverse Revision Parsing to the IT-TB data improved the accuracy rates, as reported in Table 10.

Parser	Revision	LAS	UAS	LA
DeSR	SVM-rev3	79.27	84.63	87.72
DeSR	SVM-rev2	77.30	82.82	86.33

Table 10: Accuracy rates with Reverse Revision Parsing.

Finally, we produced a number of different combinations of the outputs of the four parsing models used: SVM-LR, SVM-RL, rev2 and rev3. Table 11 shows the results of these combinations. We applied a combination method called 'Linear Tree Combination' (Attardi & Dell'Orletta, 2008). This method combines the parse tree in a top-down fashion with linear time complexity. Experiments described by Attardi and Dell'Orletta (2008) show that the application of such method produces results that outperform those achieved by adopting the usual Maximum Spanning Tree algorithm[7].

Parser	Combination	LAS	UAS	LA
DeSR	SVM-rev3 + SVM-RL + SVM-LR	80.02	85.23	87.79
DeSR	SVM-rev2 + SVM-rev3 + SVM-RL	79.82	85.08	87.76
DeSR	SVM-rev2 + SVM-rev3 + SVM-LR	79.21	84.67	87.23
DeSR	SVM-rev2 + SVM-RL + SVM-LR	78.91	84.36	86.69

Table 11: Accuracy rates with Linear Tree Combination.

3.4 Compared Evaluation

In order to evaluate in greater detail how the above modifications and the use of a wider training set improved the accuracy rates, we tested — on the test set reported in Table 3 — the best performing version of DeSR (see Table 11) and three other parsers (ISBN, Malt, MST), which were trained on the training set shown in Table 3.

We performed an in-depth evaluation of the results using MaltEval (Nilsson & Nivre, 2008). In particular, we focussed the evaluation on some relevant dependency relations and coarse-grained PoS.

Tables 12 and 13 respectively report the accuracy rates on

[6] The number of sentences is reported in square brackets.

[7] The combinations were performed involving three parser outputs at a time. Since Linear Tree Combination is based on a voting scheme aimed at preserving the dependency tree representation, this only makes sense if at least three parser outputs are combined. Conversely, we did not combine four or more parser outputs because, as shown also by Nivre et al. (2007), combination methods significantly increase the accuracy rates when just three systems are combined. This holds true especially when parsers based on one common approach are used.

the subject and object dependency relations[8]. Among the others, the subject and object relations were chosen for evaluation because they are used to tag the arguments of verbs and adjectives (vs. the adjuncts) and, thus, they are involved in the annotation of valency[9]. This information is needed for the creation of the IT-TB valency lexicon, which is produced by induction from treebank data (McGillivray & Passarotti, 2009). If the parsing of such information is more accurate, the parser can be run on the entire IT corpus and the noise caused by the automatic processing of data can be reduced by considering only the most common arguments in the creation of the valency lexicon (see the approach followed by Bamman and Crane (2008)).

Parser	Precision	Recall	F-score	DepRel
DeSR	88.9	87.6	88.2	Sb
ISBN	86.2	88.9	87.5	Sb
Malt	85.3	88	86.7	Sb
MST	81.6	79.4	80.5	Sb

Table 12: Evaluation by subject relation.

Parser	Precision	Recall	F-score	DepRel
DeSR	83.5	82	82.8	Obj
Malt	82.4	80.8	81.6	Obj
ISBN	81.7	79.4	80.6	Obj
MST	71.8	68.8	70.3	Obj

Table 13: Evaluation by object relation.

Table 14 shows the accuracy rates concerning the predicate dependency relation. This is an important relation since the tag Pred (Predicate) is assigned to the predicate of the main clause (or clauses, in case of coordination or apposition) of a sentence and represents the root of the dependency tree (apart from one technical root node).

Parser	Precision	Recall	F-score	DepRel
DeSR	82.8	94.4	88.2	Pred
Malt	78.6	86	82.1	Pred
MST	74.6	87.9	80.7	Pred
ISBN	65.7	60.7	63.1	Pred

Table 14: Evaluation by predicate relation.

[8] Precision is defined here as the percentage of times a tag X is correctly assigned to the correct head with respect to the number of occurrences of that tag in the automatically parsed data; recall is the percentage of times a tag X is correctly assigned to the correct head with respect to the number of occurrences of that tag in the gold standard. F-score (or F-measure) is the weighted harmonic mean of precision and recall, calculated as follows (van Rijsbergen, 1979): F = 2*(precision*recall) / (precision+recall).

[9] Two other relations are involved in valency. They are tagged with Pnom (nominal predicate: determining complement of the subject) and OComp (determining complement of the object). For more details see McGillivray et al. (2009).

As with the PDT analytical layer annotation style, in the IT-TB all of the tags can be appended with a suffix (_Co) in the event that the given node is a member of a coordinated construction. Thus, for completeness, we ran the evaluation also on coordinated subject, object and predicate relations. The results are provided in Tables 15, 16 and 17.

Parser	Precision	Recall	F-score	DepRel
DeSR	69.6	52.5	59.8	Sb_Co
ISBN	53.7	47.5	50.4	Sb_Co
Malt	50	36.1	41.9	Sb_Co
MST	33.8	36.1	34.9	Sb_Co

Table 15: Evaluation by coordinated subject relation.

Parser	Precision	Recall	F-score	DepRel
ISBN	42.5	41.5	42	Obj_Co
Malt	33.3	41.5	37	Obj_Co
DeSR	43.5	24.4	31.2	Obj_Co
MST	29.3	29.3	29.3	Obj_Co

Table 16: Evaluation by coordinated object relation.

Parser	Precision	Recall	F-score	DepRel
DeSR	85	86.3	85.6	Pred_Co
Malt	82.1	74.6	78.2	Pred_Co
ISBN	73.9	74.6	74.2	Pred_Co
MST	68.2	77.2	72.4	Pred_Co

Table 17: Evaluation by coordinated predicate relation.

Comparing DeSR with the best performing parser on each single dependency relation, we notice that the F-score of the predicate relation is the one showing the highest improvement: +6.1% for Pred and +7.4% for Pred_Co. On the other hand, the subject and object relations show varied results: +0.7% for Sb, +9.4% for Sb_Co, +1.2% for Obj and -10.8% for Obj_Co.

A further evaluation was performed on coarse-grained PoS. In the IT tagset coarse-grained PoS consist of the following inflectional classes:

- 1: nominal inflection: nouns, pronouns and adjectives;
- 2: nominal inflection of verbs: verbal forms with case, number and/or gender, but without person (participles, gerunds, gerundives);
- 3: verbal inflection: verbal forms with person and number, but without case and/or gender;
- 4: invariable: prepositions, conjunctions, adverbs, interjections;
- punctuations.

Inflection	DeSR	ISBN	Malt	MST
1	81.7	78.7	76.5	73.3
2	68.3	70.5	68.6	65.7
3	75.6	66.6	66.6	65.9
4	76.4	71	68.4	72.8
Punc	89.7	84.2	78.7	88.3

Table 18: Evaluation by inflectional class.

Table 18 reports the accuracy rates on the IT inflectional classes (evaluation measure: LAS), showing that DeSR performs significantly better than the other parsers on the verbal inflection class (+9%). Invariable items (+3.6%), nouns, pronouns and adjectives (+3%) and punctuations (+1.4%) show lower improvements, while no improvement is reported for the class of nominal inflection of verbs (-2.2%).

As an example, Figure 1 reports one sentence of the IT-TB test set (Gold) compared to the different analyses provided by DeSR and by the other three parsers involved in the evaluation.

The sentence is excerpted from the book *Super Sententiis Petri Lombardi* of Thomas Aquinas (Liber IV, Distinctio 5, Quaestio 1, Articulus 3, Argumentum 3, 2-1, 3-2). The interlinear translation of this sentence is the following: "but [*sed*] to some creatures [*aliquibus creaturis*] it was conferred [*collatum est*] that [*ut*] they could [*possint*] produce [*producere*] forms [*formas*]".

In Figure 1, wrong dependency relations are circled and wrong attachments are represented by dotted arcs. The gold standard analysis is provided at the top of the figure, while the analysis by the four different parsers are reported below.

Similarly to PDT, our annotation style assigns the tag Pred to the predicate of the main clause of a sentence: in this case, the main predicate is the compound verb *collatum est*, formed by the participle *collatum* (Pred_Co: the tag is appended with the suffix _Co, since *collatum* depends on the coordination *sed*) and by the auxiliary verb *est* (tag: AuxV).

The main verb of a subordinate clause is annotated according to the clause's role in the sentence and made dependent on the verb of the governing clause: in this case, the clause *ut possint producere formas* is a declarative clause acting as subject. Thus, its main verb (*possint*) is tagged Sb (Subject), while the tag assigned to the conjunction *ut* is AuxC. The final punctuation is always assigned the tag AuxK.

The tag Atr (Attributive) is given to those sentence members that specify a noun in some respect; typical attributives are adjectives and nouns in the genitive case. In the example sentence, the word *aliquibus* is tagged as an attribute of the noun *creaturis*.

The object relation (tag: Obj) is given to *creaturis* (argument of *collatum*), to *producere* (argument of *possint*) and to *formas* (argument of *producere*).

While no mistakes were found in the DeSR analysis, some wrong assignment was produced by the other parsers.

ISBN does not recognise that *collatum est* is a compound verb in which *est* is the auxiliary verb. Indeed, ISBN tags *est* as the main predicate of the sentence (Pred_Co) and *collatum* as the nominal predicate (Pnom), which, thus, is made dependent on *est*. This implies that the declarative clause headed by the conjunction *ut* depends on *est* and not on *collatum*. Following such an analysis, the translation of the sentence would be the following: "but [*sed*] to some creatures [*aliquibus creaturis*] that [*ut*] they could [*possint*] produce [*producere*] forms [*formas*] is [*est*] conferred [*collatum*]".

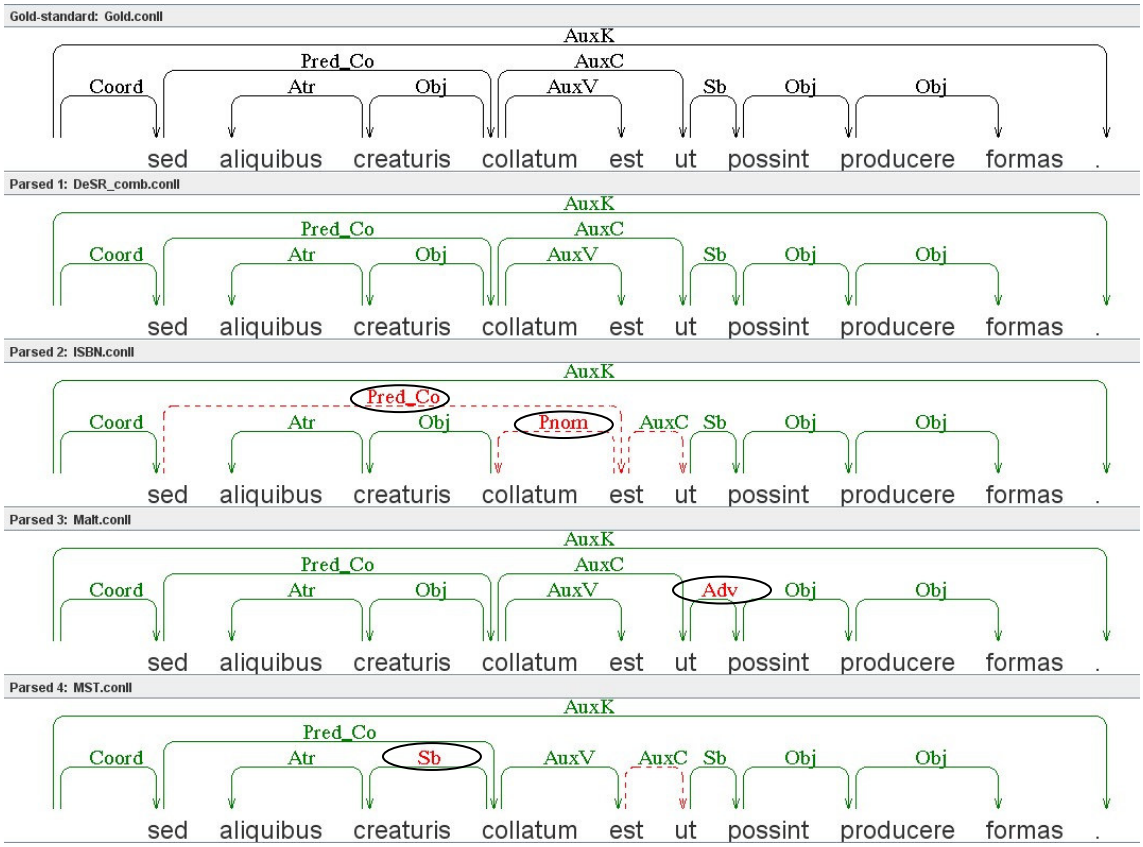

Figure 1: One sentence of the *Index Thomisticus* Treebank parsed by four different parsers.

The analysis provided by Malt is almost perfect, except for one wrong dependency relation which is assigned to the main verb of the subordinate clause headed by *ut*. Malt assigns the tag Adv (Adverbial) to *possint*, while the subject is missing. According to the different possible values of the dependent clauses headed by *ut* (final, concessive, consecutive, etc.), the translation of the sentence in this analysis could look like the following: "but [*sed*] to some creatures [*aliquibus creaturis*] (it) was conferred [*collatum est*] so that (in order that) [*ut*] they could [*possint*] produce [*producere*] forms [*formas*]".
Finally, like ISBN, MST also makes the conjunction *ut* dependent on *est* and not on *collatum*, while *collatum* is correctly assigned the Pred_Co tag and *est* is tagged with AuxV. Although *possint* is correctly assigned the tag Sb, this analysis results in another mistake, since no subject can be dependent on an auxiliary verb. Furthermore, MST makes *creaturis* the subject of the main predicate *collatum*.

4. Conclusion

We described a number of modifications that we applied to a base parser, in order to improve the accuracy rates on a Latin dependency treebank. The results we reached are well beyond the state of the art of parsing for Latin. An accuracy rate of approximately 80% for the LAS evaluation metric is quite high for a very small training set (especially if a richly inflected language, like Latin, is involved) and is around 9% higher that the results reported in previous work. The improvement of the accuracy rates is also due to the availability and use of a training set which is more than one-third larger than the one adopted in our previous experiments.
Since Latin is a language used over a long timespan (of more than two thousand years), the syntax shown by Latin texts of different eras and styles can be very changeable. This must be carefully considered if our parser, trained on Medieval Latin, is applied to texts of other eras, since it is well known that probabilistic parsers tend to perform best when both trained and used on texts of the same genre or era.
In our previous work (Passarotti & Ruffolo, forthcoming) we demonstrated that, although IT-TB and LDT share a common manual of annotation, the difference between the syntax of the texts in the IT-TB and LDT data sets is so great that the data from one treebank cannot be used to train parsers to be applied on data from the other treebank. Indeed, using LDT data to enlarge the training set did not provide significantly better results in parsing the IT-TB (less than +1%). Furthermore, using only LDT data as training set to parse IT-TB data led to very low results (approximately 13% of LAS). And the same holds for the opposite experiment: parsing LDT data using a training set formed by only IT-TB data.
Nonetheless, while Classical and Medieval Latin syntax are so different that combining data from IT-TB and LDT does not improve the parsing performances, we are confident that the accuracy rates will be increased by the exploitation of data from the PROIEL corpus.
We are now planning to develop a combination involving parsers which follow a graph-based approach (like, for instance, MST). Indeed, while shift-reduce parsers process each sentence-token in a linear order using a stack, graph-based parsers analyse each sentence as a whole, thus resulting in quite different analyses, which makes such a combination a promising attempt to further improve the accuracy rates.
Finally, after a further in-depth evaluation of our results, another way to reach better results will be to apply hand-crafted, intuition-based rules on the parser output, in order to correct certain predictable mistakes.

5. Acknowledgments

Many thanks to Erik Norvelle for his help.

6. References

Attardi, G. (2006). Experiments with a Multilanguage Non-Projective Dependency Parser. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pp. 166--170.

Attardi, G. & Dell'Orletta, F. (2008). Chunking and Dependency Parsing. In *Proceedings of LREC Workshop on Partial Parsing: Between Chunking and Deep Parsing. Marrakech, Morocco*, pp. 27--32.

Attardi, G. & Dell'Orletta, F. (2009). Reverse Revision and Linear Tree Combination for Dependency Parsing. In *Proceedings of the North American Chapter of the Association for Computational Linguistic (NAACL) 2009. Boulder, Colorado*, pp. 261--264.

Bamman, D. & Crane, G. (2007). The Latin Dependency Treebank in a cultural heritage digital library. In *Proceedings of theWorkshop on Language Technology for Cultural Heritage Data (LaTeCH 2007). Prague, Czech Republic*, pp. 33--40.

Bamman, D. & Crane, G. (2008). Building a Dynamic Lexicon from a Digital Library. In *Proceedings of the 8th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2008)*.

Bamman, D., Crane, G., Passarotti, M. & Raynaud, S. (2007). *Guidelines for the Syntactic Annotation of Latin Treebanks*. Technical report, Tufts Digital Library, Boston.

Buchholz, S. & Marsi, E. (2006). CoNLL-X Shared Task on Multilingual Dependency Parsing. In *CoNLLX, SIGNLL, 2006*.

Busa, R. (1974-1980). *Index Thomisticus: sancti Thomae Aquinatis operum omnium indices et concordantiae, in quibus verborum omnium et singulorum formae et lemmata cum suis frequentiis et contextibus variis modis referuntur quaeque / consociata plurium opera atque electronico IBM automato usus digessit Robertus Busa SI*. Stuttgart-Bad Cannstatt: Frommann-Holzboog.

Covington, M.A. (1990). *A dependency parser for variable-word-order languages*. Technical report, Artificial Intelligence Programs, University of Georgia.

Dell'Orletta, F., Lenci, A., Montemagni, S., & Pirrelli, F. (2006). Probing the Space of Grammatical Variation:

Induction of Cross-Lingual Grammatical Constraints from Treebanks. In *Proceedings of the Workshop on Frontiers in Linguistically Annotated Corpora, Association for Computational Linguistics (ACL), Sydney, Australia*.

Hajič J., Panevová, J., Buráňová, E., Urešová, Z. & Bémová, A. (1999). *Annotations at analytical level: Instructions for annotators* (English translation by Z. Kirschner). Technical report, ÚFAL MFF UK, Prague, Czech Republic.

Haug, D. & Jøhndal, M. (2008). Creating a Parallel Treebank of the Old Indo-European Bible Translations. In *Proceedings of the Language Technology for Cultural Heritage Data Workshop (LaTeCH 2008). Marrakech, Morocco*, pp. 27--34.

Koch, U. (1993). *The Enhancement of a Dependency Parser for Latin*. Technical Report n° AI-1993-03, Artificial Intelligence Programs, University of Georgia.

Koster, C.H.A. (1991). Affix Grammars for natural languages. In *Attribute Grammars, Applications and Systems, International Summer School SAGA, Lectures Notes in Computer Science vol. 545*, Prague, Czech Republic.

Koster, C.H.A. (2005). Constructing a Parser for Latin. In *Computational Linguistics and Intelligent Text Processing, Lectures Notes in Computer Science,* Berlin - Heidelberg, pp. 48--59.

McDonald, R. & Pereira, F. (2006). Online Learning of Approximate Dependency Parsing Algorithms. In *Proceedings of EACL 2006*, pp. 81--88.

McGillivray, B. & Passarotti, M. (2009). The Development of the *Index Thomisticus* Treebank Valency Lexicon. In *Proceedings of the Workshop on Language Technology and Resources for Cultural Heritage, Social Sciences, Humanities, and Education (LaTeCH-SHELT&R 2009), Athens, Greece.*

McGillivray, B., Passarotti, M. & Ruffolo, P. (2009). The *Index Thomisticus* Treebank Project: Annotation, Parsing and Valency Lexicon. In J. Denooz & S. Rosmorduc (Eds.), *Natural Language Processing for Ancient Languages*. Special issue of *Traitement Automatique des Langues*, 50 (2), pp. 103--127.

Nilsson, J. & Nivre, J. (2008). MaltEval: an Evaluation and Visualization Tool for Dependency Parsing. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08). Marrakech, Morocco*.

Nivre, J., Hall, J., Kübler, S., McDonald, R., Nilsson, J., Riedel, S. & Yuret, D. (2007). The CoNLL 2007 Shared Task on Dependency Parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007. Prague, Czech Republic*, pp. 915--932.

Nivre, J. & Nilsson, J. (2005). Pseudo-projective dependency parsing. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pp. 99--106.

Passarotti, M. & Ruffolo, P. (forthcoming). Parsing the *Index Thomisticus* Treebank. Some Preliminary Results. In P. Anreiter. & M. Kienpointner (Eds.), *Proceedings of the 15th International Colloquium on Latin Linguistics, 4-9 April 2009. Innsbruck, Austria*.

Schmid, G. (1994). *TreeTagger - a language independent part-of-speech tagger*. Available at http://www.ims.uni-stuttgart.de/Tools/DecisionTreeTagger.html.

Titov, I. & Henderson, J. (2007). A Latent Variable Model for Generative Dependency Parsing. In *Proceedings of the Tenth International Conference on Parsing Technologies, Association for Computational Linguistics. Prague, Czech Republic*, pp. 144--155.

Van Rijsbergen, C.J. (1979). *Information Retrieval*. London: Butterworths.