Challenges in Enhancing the *Index Thomisticus* Treebank with Semantic and Pragmatic Annotation

Berta González Saavedra and Marco Passarotti

Università Cattolica del Sacro Cuore, Milan, Italy
E-mail: {berta.gonzalezsaavedra; marco.passarotti}@unicatt.it

Abstract

Building treebanks for ancient languages, like Ancient Greek and Latin, raises a number of challenges that have restricted so far the enhancement of the available treebanks for Classical languages with higher levels of analysis, like semantics and pragmatics. By detailing the semi-automatic annotation procedures and the treatment of two specific constructions of Latin, this paper presents the first steps towards the semantic and pragmatic annotation of a Medieval Latin treebank, the *Index Thomisticus* Treebank.

1 Introduction

When working with ancient/dead languages, like Ancient Greek and Latin, a number of specific aspects must be considered that affect the construction of Language Resources (LRs) like treebanks. First, there are not native speakers (and, actually, no speakers at all), which is not a trivial matter, since more than one interpretation of the same text is often possible, stemming from two millennia of philological work. Interpretation can be difficult also because most of the extant texts belong to a high register, which in turn makes the corpora for Classical languages poorly representative. Finally, building a LR for a Classical language requires a close collaboration between scholars from (often conservative areas in) the Humanities and computational linguists, which is not yet widespread in the research community.

These features raise a number of challenges for those scholars who want to build new LRs for Classical languages, especially when higher levels of analysis (like semantics and pragmatics) are concerned, since they depend heavily on deep textual interpretation. So far, this has restricted the enhancement of the available treebanks for Classical languages with such levels of annotation[1]. However, the times are mature enough also for such treebanks to get out of the cradle of surface syntactic analysis and to finally include semantic information. This paper presents the first steps towards the semantic and pragmatic annotation of a Medieval Latin treebank, the *Index Thomisticus* Treebank (IT-TB).

[1] Some semantic annotation of Classical languages is available in the PROIEL corpus [1].

2 From Analytical to Tectogrammatical Analysis

The IT-TB is a dependency-based treebank consisting of the texts of Thomas Aquinas and designed in accordance with the Prague Dependency Treebank (PDT) annotation style [3]. The PDT is based on Functional Generative Description (FGD), a theoretical framework developed in Prague, which motivates the three-layer analysis of sentences provided by the PDT [5]: (a) a morphological layer, consisting of lemmatization and morphological analysis; (b) a surface syntax layer (called "analytical"); (c) a semantic and pragmatic layer (called "tectogrammatical").

The development of each layer requires the availability of the previous one(s). Both the analytical and the tectogrammatical layers describe the sentence structure with dependency tree-graphs, respectively named Analytical Tree Structures (ATSs) and Tectogrammatical Tree Structures (TGTSs).

In ATSs every word and punctuation mark of the sentence is represented by a node of a rooted dependency tree. The edges of the tree correspond to dependency relations that are labelled with (surface) syntactic functions called "analytical functions" (like Subject, Object etc.).

TGTSs describe the underlying syntactic structure of the sentence, conceived as the semantically relevant counterpart of the grammatical means of expression (described by ATSs). The nodes of TGTSs represent autosemantic words only, while function words and punctuation marks are left out. The nodes are labelled with semantic role tags called "functors". These are divided into two classes according to valency: (a) arguments, called "inner participants", i.e. obligatory complementations of verbs, nouns, adjectives and adverbs: Actor, Patient, Addressee, Effect and Origin; (b) adjuncts, called "free modifications": different kinds of adverbials, like Place, Time, Manner etc. TGTSs feature two dimensions that represent respectively the syntactic structure of the sentence (the vertical dimension) and its information structure ("topic-focus articulation"), based on the underlying word order (the horizontal dimension). Also ellipsis resolution and coreferential analysis are performed at the tectogrammatical layer and are represented in TGTSs through newly added nodes (ellipsis) and arrows (coreference).

The first two layers of annotation are already available for the IT-TB, while the tectogrammatical annotation of data has just been started. The present size of the IT-TB is 249,271 nodes, in 14,447 sentences. So far, the first 600 sentences of *Summa Contra Gentiles* (SCG) have been annotated at tectogrammatical layer (8,910 nodes). The annotation guidelines used are those for the tectogrammatical layer of the PDT [2].

2.1 Annotation Procedures

The workflow for tectogrammatical annotation in the IT-TB is based on TGTSs automatically converted from ATSs. The TGTSs that result from the conversion are then checked and refined manually by two annotators. The conversion is performed by adapting to Latin a number of ATS-to-TGTS conversion modules provided by

the NLP framework *Treex* [4]. Relying on ATSs, the basic functions of these modules are: (a) to collapse ATSs nodes of function words and punctuation marks, as they no longer receive a node for themselves in TGTSs, but are included into the nodes for autosemantic words; (b) to assign "grammatemes", i.e. semantic counterparts of morphological categories (for instance, *pluralia tantum* are tagged with the number grammateme "singular"); (c) to resolve grammatical coreferences, i.e. coreferences in which it is possible to pinpoint the coreferred expression on the basis of grammatical rules (mostly with relative pronouns); (d) to assign functors.

Tasks (a) and (b) are quite simple and the application of the modules that are responsible for them results in good accuracy on average.

Collapsing nodes for not autosemantic words and punctuations relies on the structure of the ATSs given in input: in this respect, Latin does not feature any specific property to require for modifications of the ATS-to-TGTS conversion procedures already available in *Treex* and already applied to other languages.

Assigning grammates is a task strictly related with the lexical properties of the nodes in TGTSs. Thus, we are in the process of populating the modules that assign grammatemes with lists of words (lemmas) that are regularly assigned the same grammatemes.

The automatic processing of task (c) is just at the beginning. So far, the modules are able to resolve only those grammatical coreferences that show the simplest possible construction occurring in ATSs, i.e. that featuring an occurrence of a relative pronoun (*qui* in Latin) direcly depending on the main predicate of the relative clause. However, this construction is the most frequent for relative clauses in the IT-TB: among the 326 occurrences of *qui* in our data, 176 present this construction and are correctly assigned their grammatical coreference by the conversion modules. The remaining 150 occurrences either lack grammatical coreference or do occur in more complex constructions.

In order to assign functors automatically (task (d)), we rely both on analytical functions and on lexical properties of the ATSs nodes. For instance, all the nodes with analytical function Sb (Subject) that depend on an active verb are assigned functor ACT (Actor), and all the main predicates of subclauses introduced by the subordinating conjunction *si* (*if*) are assigned functor COND (Condition). Table 1 reports the number of nodes occurring in the TGTSs of the first 600 sentences of SCG automatically produced by the modules (column "Parsed") and in the same ones manually checked and modified (column "Gold"). The column "Correct" reports the number of nodes that are assigned the correct functor in the automatically parsed data[2]. Precision, recall and F-score of automatic functor assignment are provided [6].

The overall accuracy of the automatic assignment of functors (provided by the F-score) is around 66%. However, since the accuracy varies heavily by functor,

[2]The nodes that are newly added in TGTSs (for ellipsis resolution purposes) are not considered in table 1, since no reconstructed node is supplied in the TGTSs built automatically by the conversion modules. The automatically parsed data include 101 nodes more than the gold standard; these nodes are those that were manually collapsed and included into others.

Parsed	Gold	Correct	Precision	Recall	F-Score
6620	6519	4318	65.23	66.24	65.73

Table 1: Evaluation of automatic functor assignment

table 2 reports the evaluation of the automatic assignment for the ten most frequent functors in the gold standard that occur at least once also in the automatically parsed data[3]. Precision, recall and F-score are reported for each functor.

Functor	Parsed	Gold	Correct	Precision	Recall	F-Score
PAT	1249	1307	964	77.18	73.76	75.43
RSTR	2752	1124	1052	38.23	93.59	54.28
ACT	774	858	628	81.14	73.19	76.96
PRED	515	503	447	86.8	88.87	87.82
PREC	220	266	215	97.73	80.83	88.48
CONJ	256	255	238	92.97	93.33	93.15
RHEM	231	239	221	95.67	92.47	94.04
MEANS	99	211	91	91.92	43.13	58.71
APP	65	208	62	95.38	29.81	45.42
MANN	82	207	54	65.85	26.09	37.37

Table 2: Evaluation of automatic functor assignment by single functors

5,785 out of the 6,519 not newly added nodes in the gold standard are assigned a functor that is present at least once also in the automatically parsed data. The 734 nodes remaining are those that receive a functor that the modules for automatic conversion from ATSs to TGTSs have never assigned. Among these, the most frequent are the locative functors DIR1, DIR2, DIR3 and LOC (respectively, From, Which way, To and Where: 204 cases), REG (Regard: 101), CRIT (Criterion: 61), CPR (Comparison: 59) and ADDR (Addressee: 58).

The results reported in table 2 show that the modules for automatic conversion generally achieve high precision (always higher than 80% but for PAT and MANN), while recall shows lower values. In particular, recall is always lower than precision but for PRED and CONJ (where the two values are very close). The functor RSTR must be evaluated separately, since it is the functor that is assigned by default in those cases where no rule is available in the modules to assign a functor. This motivates its very low precision and, conversely, its high recall.

[3]ACT: Actor; APP: Appurtenance; CONJ: (paratactic) Conjunction; MANN: Manner; MEANS: Means; PAT: Patient; PREC: reference to Preceding text; PRED: Predicate of the main clause; RHEM: Rhematizer; RSTR: Restrictor. For more details about functors, see [2].

2.2 Modifications to the PDT Manual

Performing tectogrammatical annotation of Latin texts has required a number of modifications to the rules stated in the PDT manual. In the following, we discuss two of such modifications, one dealing with a typical Latin construction (passive periphrastics), the other with the semantics of one specific subordinating conjunction (*ne*).

The passive periphrastic construction in Latin expresses the idea of obligation. It consists of one form of the verb *sum* (*to be*) and of a gerundive, a mood for verbal adjectives (always bearing a passive meaning). In the analytical layer, the gerundive is treated as the predicate nominal depending on the node for *sum*.

In TGTSs, the node for a modal verb headings an infinitive (e.g. *debeo dicere*, *I must say*) is collapsed and included into the node for the infinitive and its meaning (e.g. obligation for *debeo*) is reported in a specific grammateme assigned to the infinitive ("deontmod": deontic modality). We treat the passive periphrastic construction in Latin consistently. Although the node for the verb *sum* heads this construction in ATSs, it still acts as an auxiliary verb for the gerundive; thus, in TGTSs the node for *sum* in passive periphrastics is collapsed and included into the node of the gerundive, which becomes the head of the construction. This implies that the values of all the grammatemes of *sum* are assigned to the gerundive. Among the grammatemes, deontmod is assigned the value for obligation ("hrt"). The functor of *sum* is assigned to the gerundive, and all the nodes depending on *sum* are made dependent on the gerundive. According to the passive meaning of the gerundive, the subject of *sum* in the ATS is assigned the functor PAT in the TGTS.

For instance, in the clause *quae de deo* [...] *consideranda sunt* (*those things about God that must be considered*; SCG, 1.9), the node for *sum* (lemma of *sunt*) is included into that for *considero* (lemma of *consideranda*), which is assigned the value "hrt" for the grammateme deontmod. All the nodes depending on *sum* in the ATS are made dependent on *considero* in the TGTS and the node for *qui* (lemma of *quae*), which is the subject of *sunt* in the ATS, is assigned the functor PAT.

For what concerns *ne* (*in order not to*), it is a subordinating conjunction that introduces clauses expressing a negative purpose, or a negative imperative. The meaning of *ne* is, thus, composite: negative + purpose/imperative. Like for all the subordinating conjunctions in TGTSs, the node for *ne* is collapsed and included into the node for the head-verb of the clause introduced by *ne*. Given the composite nature of the meaning carried by *ne*, this makes the semantic value of negation of *ne* to be lost in the TGTS. We solve this loss by adding in the TGTS a new node with the technical lemma "#Neg" depending on the head-verb of the clause.

For instance, in the clause *ne te inferas in illud secretum* (*do not get into that secret*; SCG, 1.8), the node for *ne* is included into that for *infero* (lemma of *inferas)* and a new node with lemma "#Neg" is added depending on *infero*.

3 Conclusions

Moving from analytical to tectogrammatical annotation concerns the long debated topic of the relations holding between syntax and semantics.

On one side, several aspects of tectogrammatical annotation can be automatically induced from ATSs. In our work, this is done by applying to Latin a number of ATS-to-TGTS conversion modules already used for other (modern) languages, thus opening research questions in diachronic comparative linguistics.

On the other side, starting the tectogrammatical annotation of a treebank that includes texts in a dead language, which lacks advanced NLP tools able to process semantics, demands a significant amount of manual work. In fact, so far ellipsis resolution, topic-focus articulation and textual coreference (i.e. coreference realized not only by grammatical means, but also via context, mostly with non-relative pronouns) are performed fully manually in the IT-TB.

In the near future, we have to both increase the recall of the already available rules for functor assignment and to build new ones for the automatic processing of both ellipsis resolution and textual coreference. Further, once a sufficient amount of annotated data will be available, we shall start to train stochastic NLP tools to perform semi-automatic annotation.

References

[1] Haug Dag and Jøhndal Marius. Creating a Parallel Treebank of the Old Indo-European Bible Translations. In *Proceedings of LaTeCH 2008*, pages 27-34, Marrakech, 2008.

[2] Mikulová Marie et alii. *Annotation on the Tectogrammatical Layer in the Prague Dependency Treebank*. Institute of Formal and Applied Linguistics, Prague, 2006. Available at `http://ufal.mff.cuni.cz/pdt2.0/doc/manuals/en/tlayer/html/index.html`.

[3] Passarotti Marco. Theory and Practice of Corpus Annotation in the *Index Thomisticus* Treebank. *Lexis*, 27, pages 5-23, 2009.

[4] Popel Martin and Žabokrtský Zdeněk. TectoMT: Modular NLP Framework. In *Proceedings of IceTAL, 7th International Conference on Natural Language Processing*, pages 293-304, Reykjavík, 2010.

[5] Sgall Petr, Hajicová Eva, and Panevová Jarmila. *The Meaning of the Sentence in its Semantic and Pragmatic Aspects*. Reidel, Dordrecht, 1986.

[6] Van Rijsbergen Cornelis Joost. *Information Retrieval*. Butterworths, London, 1979.