Building the *Syntactic Reference Corpus of Medieval French* Using *NotaBene RDF Annotation Tool*

Nicolas Mazziotta
Universität Stuttgart, Institut für Linguistik/Romanistik
nicolas.mazziotta@ulg.ac.be

Abstract

In this paper, we introduce the *NotaBene RDF Annotation Tool* free software used to build the *Syntactic Reference Corpus of Medieval French*. It relies on a dependency-based model to manually annotate Old French texts from the *Base de Français Médiéval* and the *Nouveau Corpus d'Amsterdam*.

NotaBene uses OWL ontologies to frame the terminology used in the annotation, which is displayed in a tree-like view of the annotation. This tree widget allows easy grouping and tagging of words and structures. To increase the quality of the annotation, two annotators work independently on the same texts at the same time and NotaBene can also generate automatic comparisons between both analyses. The RDF format can be used to export the data to several other formats: namely, TigerXML (for querying the data and extracting structures) and graphviz dot format (for quoting syntactic description in research papers).

First, we will present the *Syntactic Reference Corpus of Medieval French* project (SRCMF) (1). Then, we will show how the *NotaBene RDF Annotation Tool* software is used within the project (2). In our conclusion, we will stress further developments of the tool (3).

1 Introducing the SRCMF Project

1.1 Main goals

There currently exists no widely available syntactically annotated corpus for Medieval French. Several syntactic corpora are available for Latin[1] or Old Portuguese.[2] Research for automatic annotation of Medieval French is being carried out by the *Modéliser le changement: les voies du français* project.[3]

SRCMF is an international initiative, gathering French (dir. Sophie Prévost, CNRS, Paris) and German (dir. Achim Stein, Institut für Linguistik/Romanistik, University of Stuttgart) resources and teams. The aim of this project is to provide selected excerpts[4] of the two biggest Medieval French corpora – the *Base de Français Médiéval* (Guillot et al., 2007), and the *Nouveau Corpus d'Amsterdam* (Kunstmann and Stein, 2007a) with a syntactic annotation layer that is meant to follow the same guidelines in both corpora.

It was decided at the very beginning of the project that, at first, the syntactic analysis would be manually added to the corpus by experts, rather than automatically inserted by an automaton.[5]. Accordingly, annotation layers that previously exist are not used to elaborate the new layer. This choice leads to several consequences, when one considers the mistakes that could be made during the annotation procedure: 1/ errors are less systematic than those introduced by an automaton; 2/ the annotation model does not need to be formalised at first; 3/ proofreading is very important. While the first point might be a major advantage in a further statistical exploration of the data (because of the "better" randomness of the errors), the third is a major problem: proofreading is very time-consuming. But as previous automatic POS annotation is provided in both corpora, this tagging can be used *a posteriori*. We plan to perform mutual validation between the POS and the syn-

[1] The *Latin Dependency Treebank* and the *Index Thomisticus Treebank* (Bamman et al., 2008).

[2] *Tycho Brahe* project http://www.tycho.iel.unicamp.br/~tycho/.

[3] Which provide syntactic annotation for 19 texts dating from the 11th to the end of the 13th C. (Martineau, 2008).

[4] There are still legal and technical issues that interfere with the final size of the corpus.

[5] Automatic annotation will be investigated later on.

Proceedings of the Fourth Linguistic Annotation Workshop, ACL 2010, pages 142–146,
Uppsala, Sweden, 15-16 July 2010.

tactic annotations: this procedure is allowed by the independency of their elaborations.

At the time this paper was submitted, the sample annotation of *Le Roman de Tristan* (Defourques and Muret, 1947) (ca 28.000 words, ca 54.000 annotations)[6] has been completed and will be made available soon.

1.2 Syntactic Annotation Model

We will not give an in-depth description of the model here: we limit ourselves to a general presentation that will make the rest of the paper more easily understandable.

The deficient nominal flexion in Medieval French makes the task of identifying the head of NPs very difficult, and there is considerable ambiguity. Therefore, the basic annotation we provide only concerns the structure of the clause, and relations at phrase- or word-level (Lazard, 1984) are not described, except by a basic identification of prepositions and conjunctions, and by delimitation, when necessary (e.g., relative clauses occur at phrase-level: we mark their boundaries in order to describe their structure).

It is to be stressed that the added annotations are as genuinely syntactic as possible. This means that neither semantic, nor enunciative analyses are encoded –following the *Théorie des trois points de vue* (Hagège, 1999). On the formal part, as far as morphological features are concerned, only verbal inflexion is taken into account, since it has obvious effects on the syntax of the clause. It is also important to distinguish between *syntactic structures*, which occur at deep level, and *word order*, which is considered as an expression of these structures and does not receive any annotation.

The model is dependency-based (Polguère and Mel'čuk, 2009; Kahane, 2001), and relations are centered on verb forms, which are the main governor nodes of the clauses. Everything in the clause depends on this central verb –including the subject, which is not compulsory in Medieval French, and is therefore described as a complement. The model gives higher priority to morphosyntactic criteria than to semantic ones, and the relation linking it to its satellites can be *qualified* by checking precise criteria. E.g., subjects are identified by verb-subject agreement, objects become subjects in a passive transformation, etc.

[6] We do not provide exact figures, for they are subject to change slightly as we review our annotation work.

1.3 Annotation Workflow

Four annotators are currently working on the project.[7] The annotation workflow for each portion of text (ca 2000 words) is the following: 1/ two different annotators perform individual annotation of the same portion of text; 2/ the same people perform a crossed-correction for most obvious errors by the annotators; 3/ two different proofreaders perform a second-step comparison and deal with complex cases.

2 NotaBene RDF Annotation Tool

Stein (2008, 165-168) has given a comprehensive specification of what the features of the annotation tool should be. Most importantly, we adopt the principle that the software should provide a convenient interface to manually annotate the syntactic relations between words and also to perform comparisons. *NotaBene RDF Annotation Tool* free software (still in alpha version) focuses on those features.[8] An SRCMF-specific plugin has been designed for manual annotation and annotation comparisons.

2.1 General Presentation

As explained in (Mazziotta, forthcoming), NotaBene is an attempt to use Semantic-Web techniques to provide textual data with linguistic annotations. This data has to be valid XML that identifies every taggable token with a unique identifier (e.g.: an `@xml:id` attribute) that is interpreted as a URI. It uses RDF formalisms (Klyne and Carroll, 2004)[9] to store annotations and OWL ontologies to describe terminologies (Bechhofer et al., 2004). NotaBene focuses on multiple conceptualisation and allows concurrent visualisations of the same text/annotation[10]. The use of RDF rather than the more commonly used XML makes it easier to cross several overlapping analysis without having to elaborate complex jointing procedures (Loiseau, 2007).

[7] Currently, the four annotators work part-time on the annotation task, hence, one could say there is the equivalent of two full-time annotators.

[8] It is freely available at `https://sourceforge.net/projects/notabene/`. Note that the documentation is still very sparse; please contact the author if you intend to use the program.

[9] See also the current NotaBene conceptual specification `http://notabene.svn.sourceforge.net/viewvc/notabene/trunk/doc/specification.pdf`, that explains how the RDF model has been restricted.

[10] Furthermore, it can show concurrent terminologies applied to the same text, but we will not discuss it here.

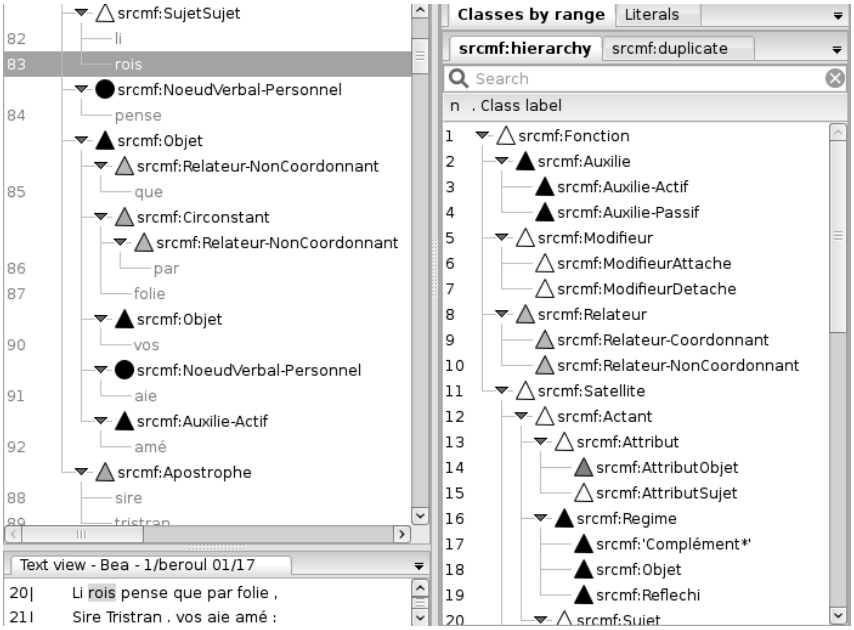

Figure 1: NotaBene SRCMF Working environment

Each visualisation is associated with one or more OWL ontologies. The current terminology is visible on the right panel of the application (see fig. 1, showing some SRCMF-specific classes).[11]

Visualisations are dynamically linked with the RDF data structure, which is updated on-the-fly.

2.2 SRCMF Plugin for Syntactic Annotation

For the sake of ergonomics, it turned out to be easier to represent syntactic structures using a constituent-like visualisation. By identifying the governor of each structure, we can use such a visualisation to represent a dependency graph, as there is evidence (Robinson, 1970) of formal equivalence on the two descriptions –we will discuss this later on (see section 2.4). Hence, the main plugin for syntactic annotation is a tree-like widget in which words are displayed vertically from top to bottom in the order of the text. Here is an example of a fully annotated sentence to introduce the interface:

> Li rois pense que par folie, Sire Tristran, vos aie amé ["The king thinks that it was madness that made me love you, Lord Tristan"] –Béroul, in (Defourques and Muret, 1947, v. 20)

As it can be seen on the left panel in fig. 1, the text is wrapped in a hierarchy of folders that mainly represent labelled subtrees[12]. Within each clause, a disc is used to visually identify the main governor, whereas triangles mark its dependents.

At the beginning of the annotation task, the plugin shows a simple list of words, which are selected and wrapped into folders that represent the linguistic analysis of the text. This can be done either by using customisable keyboard shortcuts or by pointing and clicking with the mouse.

A simultaneous view of the running text, preserving references and punctuation, is synchronised with the tree widget (see at the bottom-left corner of fig. 1).

2.3 Comparison Procedures

NotaBene's ability to display concurrent annotations of the same text is used to compare the results of the syntactic analysis by two annotators. It identifies structures that differ by not having the same contents or label. As it can be seen in fig. 2, the same structure has not been understood in the same way by the first (who places the *Apostrophe* at the main clause level) and by the second annotator (who places it at the subordinate clause level). At the application level, NotaBene simply sees that the *Objet* folder on the right pane con-

[11] Although the figure shows a tree, the class hierarchy is a graph. See n. 12 for some translations of the labels.

[12] The tag labels translate roughly (the *srcmf* prefix is the namespace of the project): *Phrase* "Clause", *SujetSujet* "Subject", *Objet* "Object", *Circonstant* "Adjunct", *NœudVerbal...* "Finite Verb", *Auxilie...* "Non-finite auxiliated form", *Relateur...* "Conjunction/preposition", *Apostrophe* "Vocative".

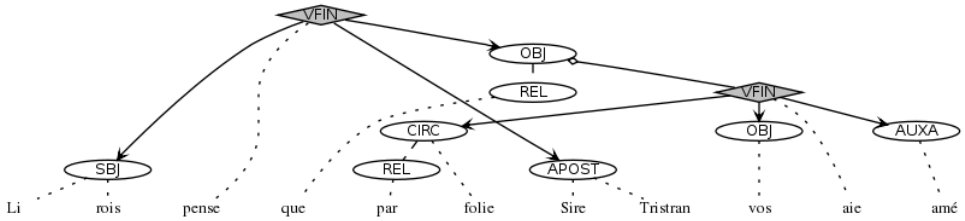

Figure 3: DOT Graph Export

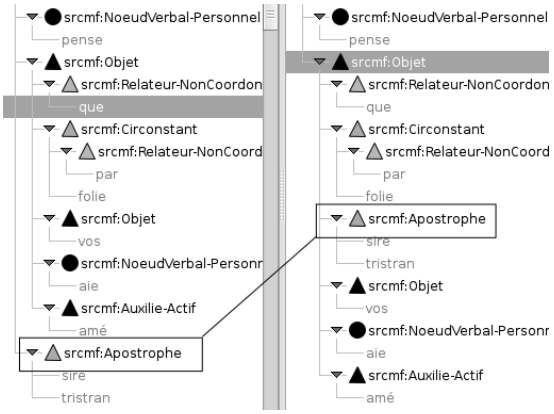

Figure 2: Comparison (boxes manually added)

tains an additional *Apostrophe* and focuses on the *Objet* structure on the right, and the first word of the structure on the left. The person who performs the comparison can immediately choose the right interpretation, and correct the erroneous analysis.

2.4 Export Capabilities

The RDF data model underlying the tree widget mimicks the tree structure and needs to be converted to create a genuine dependency graph. As the tree structure identifies SRCMF-specific governors (formally equivalent to heads in Head-Driven Phrase Structure Grammar), the transformation is relatively easy[13]. The resulting dependency RDF graph can be validated against the ontology and additional class restrictions defining the annotation model, but this feature still needs to be implemented in NotaBene.

It is possible to create as many filters as necessary to transform the RDF graph into other data structures, using NotaBene as an interface. At first, we have decided to focus on two objectives: 1/ corpus exploration; 2/ analysis rendering for the purpose of human reading.

[13] Although the description of coordination relations – which is difficult in a dependency-based framework (Kahane, 2001, 6-7)– requires a more complex algorithm.

The best syntactic corpus exploration tool we know about is TigerSearch (Brants et al., 2002).[14] The TigerSearch documentation defines the TigerXML format to represent dependency or constituency structures. TigerSearch corpora can be queried using a specific formalism and displays the analysis in a tree-like from.

TigerSearch tree display is not sufficient to represent our syntactic model – mainly because complex relations involving coordinations are surimpressed on the tree drawing, creating too many nodes to be conveniently readable. To enhance the readablility of the syntactic relations, we export our RDF graph into graphviz DOT files,[15] to render an elegant representation of the syntactic structures –fig. 3 (node labels are self-explanatory).

3 Conclusion and "TODO's"

The use of NotaBene satisfies the annotators of the SRCMF project, providing a convenient means to add manual annotations, compare parallel analyses and export data structures to other formalisms and tools.

In order to increase the quality of the project output, further implementations will at first deal with: 1/ data validation, using OWL reasoners[16]; 2/ *a posteriori* comparisons between POS annotation and syntactic annotation

Acknowledgements

The project is funded from 2009 to 2012 by the *Agence Nationale de la Recherche*, France and the *Deutsche Forschungsgemeinschaft*, Germany. We would like to thank Brigite Antoine, Beatrice Barbara Bichoff, Tom Rainsford, Achim Stein and Jean-Christophe Vanhalle for proofreading.

[14] See `http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERSearch/`.

[15] `http://www.graphviz.org/`.

[16] Using Integrity Constraint Validation, currently being added to Pellet semantic reasoner software, see `http://clarkparsia.com/`.

References

David Bamman, Marco Passarotti, Roberto Busa, and Gregory Crane. 2008. The annotation guidelines of the latin dependency treebank and index thomisticus treebank: the treatment of some specific syntactic constructions in latin. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odjik, Stelios Piperidis, and Daniel Tapias, editors, *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, may. European Language Resources Association (ELRA).

Sean Bechhofer, Frank Van Harmelen, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein, editors. 2004. *OWL Web Ontology Language Reference. Reference. W3C Recommendation 10 February 2004*.

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER Treebank. In *Proceedings of The First Workshop on Treebanks and Linguistic Theories (TLT2002) 20th and 21st September 2002, Sozopol, Bulgaria*.

L. M. Defourques and E. Muret, editors. 1947. *Béroul. Le roman de Tristan. Poème du XII^{e} siècle*. Champion, Paris, 4 edition.

Céline Guillot, Alexei Lavrentiev, and Christiane Marchello-Nizia. 2007. La Base de Français Médiéval (BFM): états et perspectives. In Kunstmann and Stein (Kunstmann and Stein, 2007b), pages 143–152.

Claude Hagège. 1999. *La structure des langues*. Number 2006 in Que sais-je? Presses Universitaires de France, Paris, 5 edition.

Sylvain Kahane. 2001. Grammaires de dépendance formelles et théorie sens-texte. In *Actes TALN 2001, Tours, 2-5 juillet 2001*.

Graham Klyne and Jeremy J. Carroll, editors. 2004. *Resource Description Framework (RDF): Concepts and Abstract Syntax W3C Recommendation 10 February 2004*.

Pierre Kunstmann and Achim Stein. 2007a. Le Nouveau Corpus d'Amsterdam. (Kunstmann and Stein, 2007b), pages 9–27.

Pierre Kunstmann and Achim Stein, editors. 2007b. *Le Nouveau Corpus d'Amsterdam. Actes de l'atelier de Lauterbad, 23-26 février 2006*. Steiner, Stuttgart.

Gilbert Lazard. 1984. La distinction entre nom et verbe en syntaxe et en morphologie. *Modèles linguistiques*, 6(1):29–39.

Sylvain Loiseau. 2007. Corpusreader: un dispositif de codage pour articuler une pluralité d'interprétations. *Corpus*, 6:153–186.

France Martineau. 2008. Un corpus pour l'analyse de la variation et du changement linguistique. *Corpus*, 7:135–155.

Nicolas Mazziotta. forthcoming. Logiciel *NotaBene* pour l'annotation linguistique. annotations et conceptualisations multiples. *Recherches Qualitatives*.

Alain Polguère and Igor Mel'čuk, editors. 2009. *Dependency in linguistic description*. John Benjamins, Amsterdam and Philadelphia.

Jane Robinson. 1970. Dependency structures and transformational rules. *Language*, 46:259–285.

Achim Stein. 2008. Syntactic annotation of Old French text corpora. *Corpus*, 7:157–171.