This article presents the Syntactic Reference Corpus of Medieval French (SRCMF). The corpus is composed of texts taken from the two major Old French corpora, the Base de Français Médiéval and the Nouveau Corpus d'Amsterdam. This contribution describes some of the core principles of the annotation model, which is based on dependency grammar, as well as the annotation procedure and representation formats.

1. Introducing the SRCMF

The project SRCMF\(^1\) builds a syntactic dependency annotation on top of the two principal Old French (henceforth “OF”) corpora: the Base de Français Médiéval ("BFM"\(^2\), Guillot et al. 2007) and the Nouveau Corpus d'Amsterdam ("NCA"\(^3\), Stein et al. 2006, Stein/Kunstmann 2007). The annotation principles rely on the concept of dependency (close to the models of Tesnière 1965 and Mel'čuk 2009) and sentences are described as a hierarchy of connected words rather than a tree of immediate constituents. One reason for choosing such a model is that dependency is more appropriate to give an account of a language with a relatively free word order such as OF, compared to the more rigid SVO order of Modern French: in OF the verb is often, but not always, in the second position after any kind of initial constituent, it is less constrained with respect to the order of the dependents of the verb) and with respect to adjacency conditions (e.g. of heads and modifiers or of auxiliaries and main verbs). A second reason is the desire to introduce as few theoretical assumptions as possible: for example, the SRCMF grammar does not postulate the existence of movement and therefore has no empty nodes created by traces, neither does it assume a

\(^1\) Funded by the Agence nationale de la recherche (ANR) and the Deutsche Forschungsgemeinschaft (DFG), 1.3.2009-29.2.2012. For more information see the SRCMF wiki on <https://listes.cru.fr/wiki/srcmf>


position for the empty subjects. This distinguishes the SRCMF project from the first major syntactic resource for medieval French: the corpus *Modéliser le changement: les voies du français* (MCVF)\(^4\) contains, for Old and Middle French (until 1500), about 72,000 annotated sentences with PENN-style constituent structure annotation (Martineau 2008, 2009). A third reason is that the goal of the SRCMF project is to provide not only a reference corpus for syntactic research but also for the training of dependency parsers.

2. The SRCMF grammar model

2.1. General principles

A word is represented by a node that depends (as a *dependent*) on its *governor* (we also use the term "head"). The inflected verb is the topmost governor. Each dependency relation is labelled with its function. Following the specifications of the *NotaBene* annotation tool (Mazziotta 2010a, 2010b), SRCMF uses a class hierarchy for syntactic structures and functions. The structures and functions and their abbreviations ("tags") are listed in table 1, where structures are distinguished by "[S]". Each dependency relation is expressed by the triple "governor-function-dependent".

<table>
<thead>
<tr>
<th>Tag</th>
<th>Function</th>
<th>Tag</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apst</td>
<td>apostrophe</td>
<td>NgPrt</td>
<td>negative particle</td>
</tr>
<tr>
<td>AtObj</td>
<td>attribute of object</td>
<td>NMax [S]</td>
<td>non-maximum structure</td>
</tr>
<tr>
<td>AtSj</td>
<td>attribute of subject</td>
<td>NSnt [S]</td>
<td>non-sentence</td>
</tr>
<tr>
<td>Aux</td>
<td>auxiliation</td>
<td>Obj</td>
<td>object</td>
</tr>
<tr>
<td>AuxA</td>
<td>active auxiliation</td>
<td>Regim</td>
<td>oblique</td>
</tr>
<tr>
<td>AuxP</td>
<td>passive auxiliation</td>
<td>Rfc</td>
<td>reflexive clitic</td>
</tr>
<tr>
<td>Circ</td>
<td>adjunct</td>
<td>Rfx</td>
<td>reflexive pronoun</td>
</tr>
<tr>
<td>Insr</td>
<td>comment clause</td>
<td>RelC</td>
<td>coordinating relator</td>
</tr>
<tr>
<td>Cmpl</td>
<td>complement</td>
<td>RelINC</td>
<td>non-coordinating relator</td>
</tr>
<tr>
<td>GpCoo [S]</td>
<td>coordinated group</td>
<td>SjImp</td>
<td>impersonal subject</td>
</tr>
<tr>
<td>Coo [S]</td>
<td>coordination</td>
<td>SjPer</td>
<td>personal subject</td>
</tr>
<tr>
<td>Intj</td>
<td>interjection</td>
<td>Snt [S]</td>
<td>sentence</td>
</tr>
<tr>
<td>ModA</td>
<td>attached modifier</td>
<td>VFin [S]</td>
<td>finite verb</td>
</tr>
<tr>
<td>ModD</td>
<td>detached modifier</td>
<td>VInf [S]</td>
<td>infinitival verb</td>
</tr>
<tr>
<td>Ng</td>
<td>negation</td>
<td>VPar [S]</td>
<td>participle verb</td>
</tr>
</tbody>
</table>

Table 1: tagset of SRCMF syntactic categories

\(^4\) The MCVF corpus is freely available on <http://www.voies.uottawa.ca> and on CD-Rom.
The SRCMF model does not use null elements (empty nodes or traces). This is avoided by encoding the linear surface order of words without assuming movement of any kind. Discontinuous structures, which occur very frequently in free word order languages like OF, are connected by the dependency relations alone, thus accepting crossing branches in the representation. However, the model uses duplicated forms in some special cases. In the relative clause (1), the relative pronoun *qui* is a non-coordinating relator (RelNC) whose duplicate is a subject (SjPer). This allows the user to retrieve the complete argument structure of verbs regardless of the clause type.\(^5\)

(1) *Souffrance si est semblable a esmeraude qui toz jorz est vert.*
Sufferance such is like an emerald which all day is green.
*(Queste del Saint Graal, 124)*

In (2), the contracted form *nes* (*ne+les*) is a negation (Ng), its duplicate is an object (Obj):

(2) *sovent dit qu’ or veut morir s’ il nes ocit.*
often says that now wants die if he not+them kills
*(Tristan de Béroul v.1985-6)*

Duplicated forms are linked by a special type of relation, different from the dependency relation.

### 2.2. Governing nodes and functional elements

The selection of the governing node is crucial for a dependency annotation. Whereas some dependency models prefer functional nodes as heads (thus coming closer to generative approaches), the SRCMF model prefers the main lexical node: each structure is headed by the lexical head (verb, noun, adjective, adverb). According to the principles of dependency grammar, each main clause must contain a finite verb (VFin) as the top node of the structure. This means that coordinated main clauses as in (3) are analysed as two separate clauses, governed by *monte* and *part* (see also Mazziotta, in

---

\(^5\) Again, this approach is different from the Turin University Treebank, where a trace-filler system accounts for discontinuous structures and where slash categories are used for nodes which combine more than one function (e.g. subject and verb in causative constructions, see Bosco 2004:152ss).
The fact that lexical heads are generally preferred over functional heads as top nodes of a structure is an important feature which also distinguishes the SRCMF model from some other dependency annotations, like TUT. In our example sentence (4) the main clause is governed by the inflected verb (i.e. the first inflected element of the verb complex, here a). This verb immediately dominates the verb of the subordinate clause (entra).

The functional category (e.g. the conjunction que) depends on the verb. Similarly, prepositional phrases are headed by the noun, the preposition (entre) depends on the noun (cuises).

(4) Elle a juré […] qu' entre ses cuises nus n' entra
   She has sworn that between her thighs no one not entered
   (Tristan de Béroul, v. 4235)

The dependency of functional elements is shown in (5), where the governing nodes are printed in bold and the functional categories are underlined.

(5) (VFin a (SjPer elle) (AuxA juré) (Obj entra (RelNC qu') (SjPer nus) (Ng n')
    (Circ cuises (RelNC entre) (Det ses)))))

The structure in (5) also shows that in complex verb forms the finite verb (auxiliary or modal) dominates the non-finite verb (participle or infinitive): thus, juré depends on a at the same level as the subject elle.

One reason for preferring lexical governors is that functional categories are often absent in medieval French (genitives without preposition, nouns without determiner, relative clauses without relative pronoun etc.).

3. Annotation

3.1. The annotation procedure

Due to the limited size of the OF corpora (about 3 million words in each
corpus, BFM and NCA, with a considerable number of shared texts), the SRCMF project adopted a manual annotation procedure during the three-year funding period in order to provide resources which are as reliable as possible.

NotaBene is a tool for manual syntactic annotation (Mazziotta, 2010b). It makes it possible to create and modify the syntactic annotation by means of a graphic interface. It allows the user to manipulate tree structures, to add free comments to any node of the structure as well as to search and list them. Script-based semi-automatic correction is also provided, and text-specific or user-specific annotations can be created by simple modification of labels. NotaBene can compare two versions of the same text and highlight the differences in the annotations. RDF graphs are used ("resource description format"; see Bechhofer et al. 2004) for the internal representation of the annotation, and dependency relations (i.e. governor-function-dependent triples) are expressed by RDF triples which form a directed graph. The RDF data is encoded in a W3C-defined XML format which can easily be converted. Although NotaBene can be freely adapted to other annotation tasks, a number of its functions are closely linked with the workflow of the SRCMF project (figure 1).

![Annotation workflow of the SRCMF project](image)

6 NotaBene is open-source and freely available on http://sourceforge.net/projects/NotaBene/
The manual annotation procedure has been designed for attaining a high level of accuracy by means of redundancy. At the first level ("manual annotation", in fig. 1), two annotators produce two separate analyses of a text. At the next level ("correction 1"), they compare their analyses in order to eliminate annotation errors. In the next step ("correction 2"), two correctors compare and review both versions using the comparison function of the NotaBene tool, decide about cases of syntactic ambiguity, and produce the final version. This step is also executed using NotaBene, and the final result is therefore encoded in RDF graphs and will be published in that format, which contains the complete information of the syntactic analysis.

3.2. Distribution formats and queries

The last two steps shown in figure 1 are not part of the annotation procedure proper, but they exemplify the formats which can be derived from the RDF graphs. Currently, NotaBene can convert RDF into dot (GraphViz) format to visualize graph images as well as into the two application-oriented formats TigerXML and CoNLL.

TigerXML has been specified for the TigerSearch query software (IMS, Stuttgart; Lezius 2002) and has been chosen because TigerSearch provides a user-friendly environment for syntactic queries, either as a stand-alone application\(^7\) or as a plugin for the TXM platform\(^8\). Since TigerXML was conceived for the representation of constituent graphs (where words have to be terminal nodes), some modifications were necessary. TigerXML is being developed further in the tiger2 project, one of whose goals consists in representing both constituency and dependency analyses simultaneously in the same graph.\(^9\)

The other export format is the standard tabular format used in dependency parsing, as defined by the Conference on Computational Natural Language

\(^7\) For Windows, Mac and various versions of Unix, see http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERSearch/oldindex.shtml

\(^8\) TXM was developed in the project Textométrie at the École Normale Supérieure of Lyon, see Heiden et al. (2010).

\(^9\) TigerXML is currently being elaborated in the tiger2 project. One of its goals consists in representing both constituency and dependency analyses simultaneously in the same graph. For more information see http://korpling.german.hu-berlin.de/tiger2/
Learning (in the CoNLL 2009 shared task). One of the goals of the manual annotation is to provide a reliable gold-standard for the training of dependency parsers. Promising tests were made with the mate-tools (Bohnet 2010; Björkelund et al. 2010): unlike other graph-based dependency parsers, the mate parser implements a "maximum spanning tree" which not only considers the nodes depending directly on a given node, but also the grandchildren and sibling nodes.

Due to this technique, mate is well suited for the SRCMF grammar model: as explained in section 2.2, our grammar is verb-centered, i.e. the verb is the top node of main clauses as well as of subordinate clauses, and functional categories are dependent on the lexical ones. For the automatic analysis however, functional categories provide important information. Consider the example given in (4): for a dependency parser without "maximum spanning tree", subordination would be a mere verb-verb dependency (a juré–entra). The mate parser, however, looks further ahead to the functional category (qu’) and – judging by these very first tests – performs quite well even for complex structures like coordinations or subordinate clauses of this kind. In the unabbreviated version of the sentence (6), the coordinated predicates (a juré 'has sworn' and et mis en vo 'and put in oath') as well as the subordinate clause (qu'entre...) were analyzed correctly, although the parser had been trained on only 3,000 manually annotated sentences. The parser output shown in figure 2 shows that only nus was erroneously analyzed as an attributive adjective (ModA) instead of an indefinite subject pronoun.

(6) Elle a juré et mis en vo qu'entre ses cuises nus n' entra.

Fig. 2: Automatic dependency annotation with the mate parser

5. Conclusions and perspectives

The SRCMF project is work in progress, and the manual annotation of the BFM and NCA corpora will be pursued even after the end of the funding period. The results will be published from 2012 on. The first tests with
dependency parsers like *mate* have encouraged us to conclude that the combination of manually annotated training corpora and automatic parsing could be an interesting perspective for the continuation of the project.

References


Sprachverarbeitung (IMS).


