A Framework for Standardized Syntactic Annotation

Thierry Declerck

Language Technology Lab & Competence Center
Semantic Web, DFKI (GmbH)
Saarbrücken, Germany

E-mail: declerck@dfki.de

Abstract

We present in this poster actual work on the building of a standard for syntactic annotation in the framework of ISO TC37/SC4. We describe here mainly the meta-model for syntactic annotation, which is building on the actual ISO proposal for a standard for morpho-syntactic annotation (MAF) and which is embedded in running efforts for defining a generic linguistic annotation framework (LAF).

1. Introduction

There have been in the past no thorough standardisation activities in the domain of syntactic annotation, despite the numerous projects (see Abeillé, 2003) that have designed ways to implement linguistic TreeBanks, i.e. syntactically annotated corpora. For several years the Penn Treebank initiatives have served as a de facto standard, but more recent work (e.g. the Negra/Tiger initiative[1] in Germany or the ISST initiative in Italy[2]) has shown that a more coherent framework could be designed to account for both (hierarchical) constituency and dependency phenomena in syntactic annotation.
Within the European eContent LIRICS project[3], a group of international experts has started the ISO process, called SynAF (Syntactic Annotation Framework). The actual document is a revision of ISO WD 24615, which is the result of a more extended discussion, including feedback and comments from ISO experts, and which was successfully submitted for its acceptance as a Committee Draft (CD) within the ISO framework.

2. Scope of the Standard

This International Standard describes the Syntactic Annotation Framework (SynAF), a high level model for representing the syntactic annotation of textual documents.
SynAF is building on the ISO MAF proposal (CD 24611). MAF (Morpho-Syntactic Framework) is dealing with the morpho-syntactic annotation of specific segments of textual documents. The morpho-syntactic annotation framework is about *part of speech* (noun, adjective, verb, etc.), *morphological* and *grammatical* features (such as number, gender, person, mood, verbal tense).
SynAF is about the annotation of the syntactic constituency of such (groups of) morpho-syntactically annotated fragments and the syntactic dependency relations existing between those (groups of) morpho-syntactically annotated fragments. We consider that the sentence will define the boundaries of the fragments of textual documents to which SynAF will apply.
As suggested just above, syntactic annotation has at least two functions in language processing:

- To represent linguistic constituencies, like Noun Phrases (NP), describing a structured sequence of morpho-syntactically annotated items[4], where we consider also constituents built from non-contiguous elements, and
- To represent dependency relations, like head-modifier relation[5]. The dependency information can exist between morpho-syntactically annotated items within a phrase (an adjective is the modifier of the head noun within an NP) or describe a specific relation between syntactic constituents at the clausal and sentential level (i.e. an NP being the "subject" of the main verb of a clause or sentence). The dependency relation can also be stated including empty elements (like the pro-drop property in romance languages[6])

SynAF is dealing with the description of a metamodel for syntactic annotation, which means that SynAF is describing elementary linguistic (in fact syntactic) abstractions that support the construction and the interoperability of (syntactic) annotations and resources. The Thematic Domain Group 4 (TDG 4) "Syntax" associated to SynAF will propose the definition of the related data categories, which will represent a point of reference for particular tagsets used for the syntactic annotation of various languages, also in the context of various application scenarios.

[1] See: http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERCorpus/
[2] See Montemagni (2003).
[3] See lirics.loria.fr

[4] But SynAF is also designed for dealing with like empty elements or traces generated by movements at the constituency level.
[5] Including also relations between same categories, like the head-head relation between nouns in appostions or nominal coordinations.
[6] This point has been particularily stressed by the authors of the ISST framework, showing here an advantage of the two-level approach, where the dependency information do not have to map entirely to the constituencey approach. In this sense, both levels of annotation have a certain independency in relation to each other (see Montemagni, 2003).

To summarize: SynAF is concerned with a metamodel that covers both dimensions of syntactic *constituency* and *dependency,* and SynAF will propose a multi-layered annotation framework that allows the combined and interrelated annotation of language data along both lines of consideration. Also the data-categories to be proposed within TDG4 will be about the basic annotation concerning both dimensions.

3. Embedding SynAF in the LAF model[7]

We want to embed the meta-model of SynAF in the more generic Linguistic Annotation Framework (LAF) and reuse its annotation strategy. LAF provides a general framework for representing annotations that has been described elsewhere in detail (Ide and Romary, 2004, 2006). Its development has built on common practice and convergence of approach in linguistic annotation over the past 15-20 years. The core of the framework is specification of an abstract model for annotations instantiated by a *pivot format*, into and out of which annotations are mapped for the purposes of exchange.

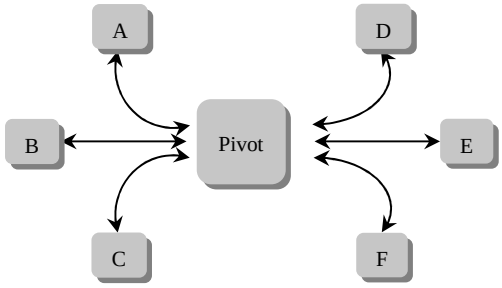

Figure 1: Use of the LAF pivot format

Figure 1 shows the overall idea for six different user annotation formats (labeled A – F), which requires two mappings for each scheme—one into and one out of the pivot format, provided by the scheme designer. The maximum number of mappings among schemes is therefore 2*n*, vs. n^2-*n* mutual mappings without the pivot. To map to the pivot, an annotation scheme must be (or be rendered via the mapping) isomorphic to the abstract model, which consists of (1) a *referential structure* for associating stand-off annotations with primary data, instantiated as a directed graph; and (2) a *feature structure representation* for annotation content. An annotation thus forms a directed graph referencing *n*-dimensional regions of primary data as well as other annotations, in which nodes are labeled with feature structures providing the annotation content. Formally, LAF consists of:

- A data model for annotations based on directed graphs defined as follows: A graph of annotations *G* is a set of vertices *V*(*G*)[8] and a set of edges *E*(*G*). Vertices and edges may be labeled with one or more features. A feature consists of a quadruple (*G', VE, K, V*) where, *G'* is a graph, *VE* is a vertex or edge in *G'*, *K* is the name of the feature and *V* is the feature value.
- A *base segmentation* of primary data that defines edges between virtual nodes located between each "character" in the primary data.[9] The resulting graph *G* is treated as an *edge graph G'* whose nodes are the edges of *G*, and which serve as the leaf ("sink") nodes. These nodes provide the base for an annotation or several layers of annotation. Multiple segmentations can be defined over the primary data, and multiple annotations may refer to the same segmentation.
- Serializations of the data model, one of which is designated as the pivot.
- Methods for manipulating the data model.

Note that LAF does not provide specifications for annotation *content categories* (i.e., the labels describing the associated linguistic phenomena), for which standardization is a much trickier matter. The LAF architecture includes a *Data Category Registry* (DCR) containing pre-defined data elements and schemas that may be used directly in annotations, together with means to specify new categories and modify existing ones (see Ide and Romary, 2004).

4. The SynAF Metamodel

While preparing SynAF, we identified some existing initiatives sharing a somehow common data model that seems to offer a good basis for the SynAF meta-model (Tiger and ISST for example, but also a longer list of corpora has been studied). Base on this study we strongly suggest the adoption of a multi-layered annotation strategy interrelating syntactic annotation for both constituency and dependency in a sound representation scheme. The studied initiatives are also offering a quite complete list of descriptors, which we started to "merge" into a first list of candidate data-categories, to be extended by data categories covering syntactic phenomena (constituency and dependency) for other languages then German and Italian

The data categories are used to decorate the meta-model for the syntactic annotation. SynAF specifications in the form of textual descriptions that describe the semantics of the modeling elements provide more complete information about the SynAF classes, relationships, and extensions. Developers shall define a data category selection (DCS) as specified for SynAF data category selection procedures.

Just below we present the actual graphical presentation of the meta-model, which is specifying the ways syntactic phenomena can be annotated.

[7] The whole section 5 is taken from (Ide, 2007).

[8] The word "vertice" is her esynonym to "node".

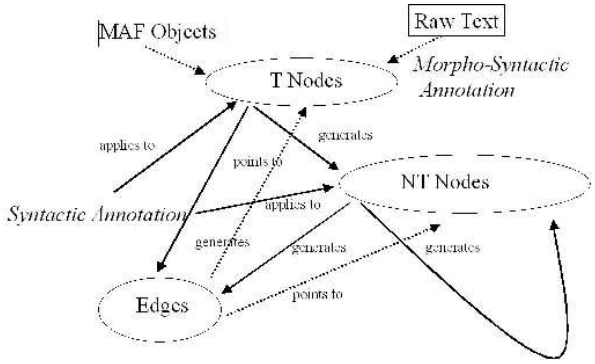

Figure 1: The SynAF metamodel

T Nodes class
The *t_nodes* class represents the terminal nodes of a syntax tree, mostly consisting of morpho-syntactically annotated words, but empty elements are allowed. The t_nodes are defined over a *span.* This can be a multiple span (for accounting for discontinuous constituents). The t_nodes are labeled with syntactic categories valid for the word level.

NT Nodes class
The n*t_nodes* class represents the non-terminal nodes of a syntax tree, mostly consisting of t_nodes and nt_nodes, but empty elements are allowed. The nt_nodes are also defined over a (possibly multiple) *span.* The nt_nodes are labeled with syntactic categories valid at the phrasal level and higher (clausal, sentential).

Edges class
The *Edges* class represents the dependency relation between nodes (both terminal and non-terminal nodes). The dependency relation is a binary one and consists of a label name and a pair of source and target nodes.

Syntactic Annotation class
The *Syntactic Annotation* class represents the application of syntactic information to MAF annotated input. It can be either a manual or an automatic application. When syntactic annotation is applied to nodes (non-terminal or terminal), then it generates either a new (non-terminal) node or a dependency edge.

5. Data Categories for SynAF

Our strategy consisted in collecting some of the most consensual syntactic annotation definitions for gaining a list of data categories for constituency (node labels) and dependency (edge labels) annotation, which will be established in the document resulting from the work in ISO TC37/SC4 TDG 4 "Syntax". In this document we present the actual list of candidates, as they have been detected in annotation initiatives like TIGER, ISST, Sparkle and EAGLES, and modified/harmonized for the purpose of this document. We do not quote the specific origin of each candidate data category. We indicate, where appropriate, language specific data categories, first for constituency labels, and then for dependency labels.

Constituency_labels	Meaning
AA	superlative phrase with am (for German)
AP	adjective phrase
AVP	Adverbial phrase
CAC	coordinated adposition
CAP	coordinated adjective phrase
CAVP	Coordinated adverbial phrase
CCP	Coordinated complementiser
CH	Chunk (non-recursive constituent)
CNP	Coordinated noun phrase
CO	coordination
CPP	Coordinated adpositional phrase
CVP	Coordinated verb phrase (non-finite)
CVZ	Coordinated infinitive with zu (for German)
NP	noun phrase
PN	proper noun
PP	adpositional phrase (prepositional and postpositional phrases
S	Sentence
VP	verb phrase (non-finite)
VZ	Infinitive with zu (for German)

SPD	prepositional phrase *di* 'of' (for Italian)
SPDA	prepositional phrase *da* 'by, from' (for Italian)
IBAR	verbal nucleus with finite tense and all adjoined elements like clitics, adverbs and negation
SV2	infinitival clause
SV3	participial clause
SV5	gerundive clause
FAC	sentential complement
FS	subordinate sentence
FINT	*+wh* interrogative sentence
F2	relative clause
CP	dislocated or fronted sentential adjuncts
COMPC	copulative/predicative complement

In the following we present the candidate data categories for dependency structures (the labels of edges in the annotation graph). Source of inspiration here were the Sparkle and the Tiger tagsets for dependency. We use also some examples taken from Sparkle

Dependency Rel	ID	Definition	Parent
Adpositional Case Marker	AC	Preposition/postposition in a PP, annotated as a sister constituent of the determiner, adjectives, noun etc	PP
Adjective Component	ADC	Component of a multi-token adjective (MTA)	MTA
Apposition	APP	"inserted" phrase, further specifying/modifying the entity described by the matrix NP.	NP PP
Adverbial phrase Component	AVC	Component of a head-less AVP	ADV
conjunct	CJ	Constituent participating in coordination	any
comparative conjunction	CM	Linguistic particles introducing a comparison in comparative constructions (for example “grosser als” in German)	
dative	DA	Dative object/`free dative' (for languages having this case in the morphology/syntax))	S VP AP AVP
head	HD	The main elements in all kind of consitutents	S VP AP AVP
postnominal modifier	MNR	Postnominal NP/PP modifier	NP PP
negation	NG	the negation particle `nicht' (also modified)	any
genitive object	OG	Genitive objects of verbs, participles and certain adjectives (for language having the geneitive case in the morphology/syntax)	
predicate	PD	Predicative AP/NP/PP, typically in a copular construction	S VP
morphological particle	PM	two cases: the infinitival `zu' (zu gehen) the adjectival (superlative) `am' (am besten)	VZ AA
relative clause	RC		NP

5. References

Abeillé, A., S. Hansen-Schirra, and H. Uszkoreit (eds.), 2003. Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora (LINC-03).

Calzolari, N., J. McNaught, and Zampolli A. (eds). 1996. EAGLES: Introduction. http://www.ilc.cnr.it/EAGLES96/edintro/edintro.html

Calzolari, N., F. Bertagna F., A. Lenci, and M. Monachini (eds). 2003. Standards and Best Practice for Multilingual Computational Lexicons. MILE (The Multilingual ISLE Lexical Entry). ISLE CLWG Deliverable, D2.2 & 3.2, Pisa.

Ide, Nancy, and Laurent Romary, 2004. A Registry of Standard Data Categories for Linguistic Annotation. Proceedings of the Fourth Language Resources and Evaluation Conference (LREC), Lisbon, 135-39.

Ide, Nancy, and Laurent Romary, 2004. International Standard for a Linguistic Annotation Framework. Journal of Natural Language Engineering, 10:3-4, 211-225.

Ide, Nancy, and Laurent Romary, 2006. Representing Linguistic Corpora and Their Annotations. Proceedings of the Fifth Language Resources and Evaluation Conference (LREC), Genoa, Italy.

Ide, Nancy. 2007. GrAF: A Graph-based Format for Linguistic Annotations. Proceedings of the LAW Workshop at ACL 2007, Prague.

Montemagni, S, F. Barsotti, M. Battista, N. Calzolari, A. Lenci O. Corazzari, A. Zampolli, F. Fanciulli, M. Massetani, R. Basili R. Raffaelli, M.T. Pazienza, D. Saracino, F. Zanzotto, F. Pianesi N. Mana, and R. Delmonte. 2003. Building the Italian Syntactic-Semantic Treebank. In Anne Abeillé (ed), Building and Using Syntactically Annotated Corpora, pages 189--210. Kluwer, Dordrecht.

Project websites:

The EAGLES Initiative: http://www.ilc.cnr.it/EAGLES96/home.html
The LIRICS Project: http://lirics.loria.fr
The SPARKLE Project: http://www.ilc.cnr.it/sparkle/sparkle.htm
The TIGER Project: http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERCorpus/