SemTAG: a platform for specifying Tree Adjoining Grammars and performing TAG-based Semantic Construction

Claire Gardent
CNRS / LORIA
Campus scientifique - BP 259
54 506 Vandœuvre-Lès-Nancy CEDEX
France
Claire.Gardent@loria.fr

Yannick Parmentier
INRIA / LORIA - Nancy Université
Campus scientifique - BP 259
54 506 Vandœuvre-Lès-Nancy CEDEX
France
Yannick.Parmentier@loria.fr

Abstract

In this paper, we introduce SEMTAG, a free and open software architecture for the development of Tree Adjoining Grammars integrating a compositional semantics. SEMTAG differs from XTAG in two main ways. First, it provides an expressive grammar formalism and compiler for factorising and specifying TAGs. Second, it supports semantic construction.

1 Introduction

Over the last decade, many of the main grammatical frameworks used in computational linguistics were extended to support semantic construction (i.e., the computation of a meaning representation from syntax and word meanings). Thus, the HPSG ERG grammar for English was extended to output minimal recursive structures as semantic representations for sentences (Copestake and Flickinger, 2000); the LFG (Lexical Functional Grammar) grammars to output lambda terms (Dalrymple, 1999); and Clark and Curran's CCG (Combinatory Categorial Grammar) based statistical parser was linked to a semantic construction module allowing for the derivation of Discourse Representation Structures (Bos et al., 2004).

For Tree Adjoining Grammar (TAG) on the other hand, there exists to date no computational framework which supports semantic construction. In this demo, we present SEMTAG, a free and open software architecture that supports TAG based semantic construction.

The structure of the paper is as follows. First, we briefly introduce the syntactic and semantic formalisms that are being handled (section 2). Second, we situate our approach with respect to other possible ways of doing TAG based semantic construction (section 3). Third, we show how XMG, the linguistic formalism used to specify the grammar (section 4) differs from existing computational frameworks for specifying a TAG and in particular, how it supports the integration of semantic information. Finally, section 5 focuses on the semantic construction module and reports on the coverage of SEMFRAG, a core TAG for French including both syntactic and semantic information.

2 Linguistic formalisms

We start by briefly introducing the syntactic and semantic formalisms assumed by SEMTAG namely, Feature-Based Lexicalised Tree Adjoining Grammar and L_U.

Tree Adjoining Grammars (TAG) TAG is a tree rewriting system (Joshi and Schabes, 1997). A TAG is composed of (i) two tree sets (a set of initial trees and a set of auxiliary trees) and (ii) two rewriting operations (substitution and adjunction). Furthermore, in a Lexicalised TAG, each tree has at least one leaf which is a terminal.

Initial trees are trees where leaf-nodes are labelled either by a terminal symbol or by a non-terminal symbol marked for substitution ($\downarrow$). Auxiliary trees are trees where a leaf-node has the same label as the root node and is marked for adjunction ($\star$). This leaf-node is called a *foot* node.

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 13–16,
Prague, June 2007.

Further, substitution corresponds to the insertion of an *elementary* tree t_1 into a tree t_2 at a frontier node having the same label as the root node of t_1. Adjunction corresponds to the insertion of an *auxiliary* tree t_1 into a tree t_2 at an inner node having the same label as the root and foot nodes of t_1.

In a Feature-Based TAG, the nodes of the trees are labelled with two feature structures called top and bot. Derivation leads to unification on these nodes as follows. Given a substitution, the top feature structures of the merged nodes are unified. Given an adjunction, (i) the top feature structure of the inner node receiving the adjunction and of the root node of the inserted tree are unified, and (ii) the bot feature structures of the inner node receiving the adjunction and of the foot node of the inserted tree are unified. At the end of a derivation, the *top* and *bot* feature structures of each node in a derived tree are unified.

Semantics ($\mathbf{L}_U$). The semantic representation language we use is a unification-based extension of the PLU language (Bos, 1995). L_U is defined as follows. Let H be a set of *hole* constants, L_c the set of *label* constants, and L_v the set of *label* variables. Let I_c (resp. I_v) be the set of individual constants (resp. variables), let R be a set of n-ary relations over $I_c \cup I_v \cup H$, and let $\geq$ be a relation over $H \cup L_c$ called the *scope-over* relation. Given $l \in L_c \cup L_v$, $h \in H$, $i_1, \ldots, i_n \in I_v \cup I_c \cup H$, and $R^n \in R$, we have:

1. $l : R^n(i_1, \ldots, i_n)$ is a L_U formula.
2. $h \geq l$ is a L_U formula.
3. ϕ, ψ is L_U formula iff both ϕ and ψ are L_U formulas.
4. Nothing else is a L_U formula.

In short, L_U is a flat (i.e., non recursive) version of first-order predicate logic in which scope may be underspecified and variables can be unification variables[1].

3 TAG based semantic construction

Semantic construction can be performed either during or after derivation of a sentence syntactic structure. In the first approach, syntactic structure and semantic representations are built simultaneously. This is the approach sketched by Montague and adopted e.g., in the HPSG ERG and in synchronous TAG (Nesson and Shieber, 2006). In the second approach, semantic construction proceeds from the syntactic structure of a complete sentence, from a lexicon associating each word with a semantic representation and from a set of semantic rules specifying how syntactic combinations relate to semantic composition. This is the approach adopted for instance, in the LFG glue semantic framework, in the CCG approach and in the approaches to TAG-based semantic construction that are based on the TAG derivation tree.

SEMTAG implements a hybrid approach to semantic construction where (i) semantic construction proceeds after derivation and (ii) the semantic lexicon is extracted from a TAG which simultaneously specifies syntax and semantics. In this approach (Gardent and Kallmeyer, 2003), the TAG used integrates syntactic and semantic information as follows. Each elementary tree is associated with a formula of L_U representing its meaning. Importantly, the meaning representations of semantic functors include unification variables that are shared with specific feature values occurring in the associated elementary trees. For instance in figure 1, the variables x and y appear both in the semantic representation associated with the tree for *aime* (love) and in the tree itself.

Given such a TAG, the semantics of a tree t derived from combining the elementary trees $t_1, \ldots, t_n$ is the union of the semantics of $t_1, \ldots, t_n$ modulo the unifications that results from deriving that tree. For instance, given the sentence *Jean aime vraiment Marie* (*John really loves Mary*) whose TAG derivation is given in figure 1, the union of the semantics of the elementary trees used to derived the sentence tree is:

$$l_0 : jean(j),\ l_1 : aime(x,y),\ l_2 : vraiment(h_0),\\ l_s \leq h_0,\ l_3 : marie(m)$$

The unifications imposed by the derivations are:

$$\{x \rightarrow j, y \rightarrow m, l_s \rightarrow l_1\}$$

Hence the final semantics of the sentence *Jean aime vraiment Marie* is:

$$l_0 : jean(j),\ l_1 : aime(j,m),\ l_2 : vraiment(h_0),\\ l_1 \leq h_0,\ l_3 : marie(m)$$

[1]For mode details on L_U, see (Gardent and Kallmeyer, 2003).

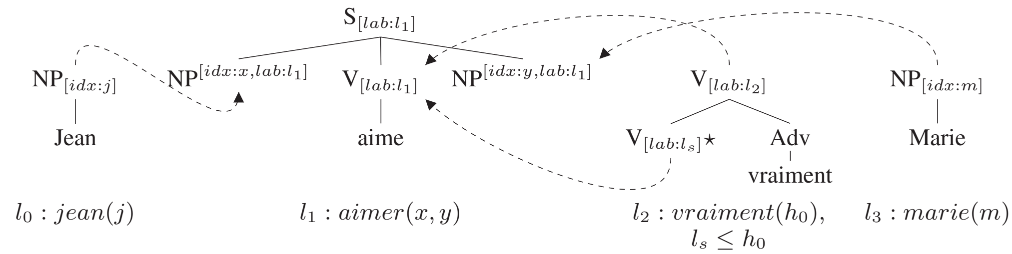

Figure 1: Derivation of "Jean aime vraiment Marie"

As shown in (Gardent and Parmentier, 2005), semantic construction can be performed either during or after derivation. However, performing semantic construction after derivation preserves modularity (changes to the semantics do not affect syntactic parsing) and allows the grammar used to remain within TAG (the grammar need contain neither an infinite set of variables nor recursive feature structures). Moreover, it means that standard TAG parsers can be used (if semantic construction was done during derivation, the parser would have to be adapted to handle the association of each elementary tree with a semantic representation). Hence in SEMTAG, semantic construction is performed after derivation. Section 5 gives more detail about this process.

4 The XMG formalism and compiler

SEMTAG makes available to the linguist a formalism (XMG) designed to facilitate the specification of tree based grammars integrating a semantic dimension. XMG differs from similar proposals (Xia et al., 1998) in three main ways (Duchier et al., 2004). First it supports the description of both syntax and semantics. Specifically, it permits associating each elementary tree with an L_U formula. Second, XMG provides an expressive formalism in which to *factorise* and *combine* the recurring tree fragments shared by several TAG elementary trees. Third, XMG provides a sophisticated treatment of variables which *inter alia*, supports variable sharing between semantic representation and syntactic tree. This sharing is implemented by means of so-called *interfaces* i.e., feature structures that are associated with a given (syntactic or semantic) fragment and whose scope is global to several fragments of the grammar specification.

To specify the syntax / semantics interface sketched in section 5, XMG is used as follows :

1. The elementary tree of a semantic functor is defined as the conjunction of its spine (the projection of its syntactic head) with the tree fragments describing each of its arguments. For instance, in figure 2, the tree for an intransitive verb is defined as the conjunction of the tree fragment for its spine (Active) with the tree fragment for (a canonical realisation of) its subject argument (Subject).

2. In the tree fragments representing the different syntactic realizations (canonical, extracted, etc.) of a given grammatical function, the node representing the argument (e.g., the subject) is labelled with an *idx* feature whose value is shared with a *GFidx* feature in the interface (where GF is the grammatical function).

3. Semantic representations are encapsulated as fragments where the semantic arguments are variables shared with the interface. For instance, the i^{th} argument of a semantic relation is associated with the *argI* interface feature.

4. Finally, the mapping between grammatical functions and thematic roles is specified when conjoining an elementary tree fragment with a semantic representation. For instance, in figure 2[2], the interface unifies the value of *arg1* (the thematic role) with that of *subjIdx* (a grammatical function) thereby specifying that the subject argument provides the value of the first semantic argument.

5 Semantic construction

As mentioned above, SEMTAG performs semantic construction after derivation. More specifically, semantic construction is supported by the following 3-step process:

[2]The interfaces are represented using gray boxes.

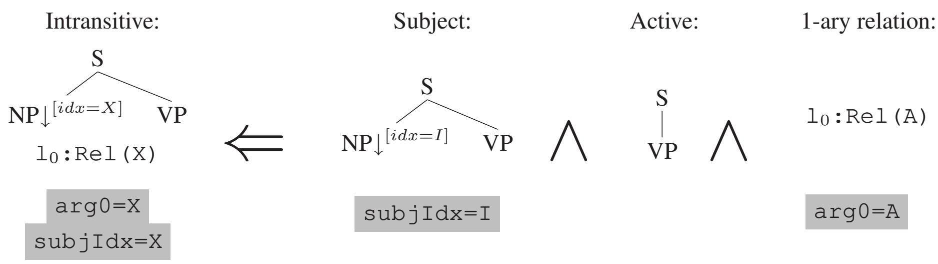

Figure 2: Syntax / semantics interface within the metagrammar.

1. First, we extract from the TAG generated by XMG (i) a purely syntactic TAG $\mathcal{G}'$, and (ii) a purely semantic TAG $\mathcal{G}''$ [3] A purely syntactic (resp. semantic) Tag is a TAG whose features are purely syntactic (resp. semantic) – in other words, $\mathcal{G}''$ is a TAG with no semantic features whilst $\mathcal{G}''$ is a TAG with only semantic features. Entries of $\mathcal{G}'$ and $\mathcal{G}''$ are indexed using the same key.

2. We generate a tabular syntactic parser for $\mathcal{G}'$ using the DyALog system of (de la Clergerie, 2005). This parser is then used to compute the derivation forest for the input sentence.

3. A semantic construction algorithm is applied to the derivation forest. In essence, this algorithm retrieves from the semantic TAG $\mathcal{G}''$ the semantic trees involved in the derivation(s) and performs on these the unifications prescribed by the derivation.

SemTag has been used to specify a core TAG for French, called SemFRag. This grammar is currently under evaluation on the *Test Suite for Natural Language Processing* in terms of syntactic coverage, semantic coverage and semantic ambiguity. For a test-suite containing 1495 sentences, 62.88 % of the sentences are syntactically parsed, 61.27 % of the sentences are semantically parsed (*i.e.*, at least one semantic representation is computed), and the average semantic ambiguity (number of semantic representation per sentence) is 2.46.

SemTag is freely available at `http://trac.loria.fr/~semtag`.

[3] As (Nesson and Shieber, 2006) indicates, this extraction in fact makes the resulting system a special case of synchronous TAG where the semantic trees are isomorphic to the syntactic trees and unification variables across the syntactic and semantic components are interpreted as synchronous links.

References

J. Bos, S. Clark, M. Steedman, J. R. Curran, and J. Hockenmaier. 2004. Wide-coverage semantic representations from a ccg parser. In *Proceedings of the 20th COLING*, Geneva, Switzerland.

J. Bos. 1995. Predicate Logic Unplugged. In *Proceedings of the tenth Amsterdam Colloquium, Amsterdam.*

A. Copestake and D. Flickinger. 2000. An open-source grammar development environment and broad-coverage english grammar using hpsg. In *Proceedings of LREC*, Athens, Greece.

Mary Dalrymple, editor. 1999. *Semantics and Syntax in Lexical Functional Grammar*. MIT Press.

E. de la Clergerie. 2005. DyALog: a tabular logic programming based environment for NLP. In *Proceedings of CSLP'05*, Barcelona.

D. Duchier, J. Le Roux, and Y. Parmentier. 2004. The Metagrammar Compiler: An NLP Application with a Multi-paradigm Architecture. In *Proceedings of MOZ'2004*, Charleroi.

C. Gardent and L. Kallmeyer. 2003. Semantic construction in FTAG. In *Proceedings of EACL'03, Budapest*.

C. Gardent and Y. Parmentier. 2005. Large scale semantic construction for tree adjoining grammars. In *Proceedings of LACL05, Bordeaux, France*.

A. Joshi and Y. Schabes. 1997. Tree-adjoining grammars. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 3, pages 69 – 124. Springer, Berlin, New York.

Rebecca Nesson and Stuart M. Shieber. 2006. Simpler TAG semantics through synchronization. In *Proceedings of the 11th Conference on Formal Grammar*, Malaga, Spain, 29–30 July.

F. Xia, M. Palmer, K. Vijay-Shanker, and J. Rosenzweig. 1998. Consistent grammar development using partial-tree descriptions for lexicalized tree adjoining grammar. *Proceedings of TAG+4*.