Dynamic TAG and lexical dependencies

Alessandro Mazzei
Dipartimento di Informatica
Università di Torino
mazzei@di.unito.it

Vincenzo Lombardo
Dipartimento di Informatica
Università di Torino
vincenzo@di.unito.it

Patrick Sturt
Department of Psychology
University of Glasgow
patrick@psy.gla.ac.uk

Incrementality is a widely held assumption that constrains the language processor to parse the input words from left to right, and to carry out a semantic interpretation of the partial structures (Marslen-Wilson, 1973). The detailed specification of the incremental syntactic process is often addressed by assuming a parsimonious version of incrementality that we can call *strong connectivity* (Stabler, 1994). Strong connectivity constrains the syntactic processor to maintain a fully connected structure throughout the whole process and is supported by some psycholinguistic evidence (Kamide et al., 2003).

In this paper we describe a constituency based dynamic grammar (cf. Milward (1994)), called *Dynamic Version of TAG* (DVTAG), that fulfills the strong connectivity hypothesis. Similarly to LTAG, a DVTAG consists of a set of elementary trees and a number of attachment operations for combining them (Joshi and Schabes, 1997). DVTAG fulfills the strong connectivity hypothesis by constraining the derivation process to be a series of steps in which an elementary tree is combined with the partial tree spanning the left fragment of the sentence. The result of a step is an updated partial structure called *left-context*. Specifically, at the processing step i, the elementary tree anchored by the i-th word in the sentence is combined with the partial structure spanning the words from positions 1 to $i-1$; the result is a partial structure spanning the words from 1 to i. Figure 1-a shows a DVTAG derivation for the sentence *John loves Mary madly*. Unlike LTAG, each node in the elementary trees is augmented with a feature indicating the lexical head that projects the node. If several unheaded nodes share the same lexical head, they are all co-indexed with a head-feature (e.g. $\_v_3$ in the elementary tree anchored by *madly* in Figure 1-a). The head-feature can contain variables in logical terms, e.g. $\_v_3$ will be unified with the lexical constant *loves* in the derivation depicted in Figure 1-a. In DVTAG the lexical anchor does not necessarily provide the head feature of the root of the elementary tree. This is trivially true for auxiliary trees, such as the tree anchored by $madly$ in Figure 1-a. However this can also occur with initial trees because initial trees can include not only the head projection of the anchor, but also other higher projections that are required to account for the full connectedness of the partial parse tree (Lombardo et al., 2004).

Lexical dependencies are useful for several NLP tasks, e.g. statistical parsing. There is a straightforward relation between lexicalized constituency grammars and lexical dependencies. In a lexicalized grammar we can associate a word with each elementary structure. Then we can view an operation on two elementary structures A and B, anchored respectively by the words w_a and w_b, as a relation between the words w_a and w_b. From this point of view, we can consider the LTAG derivation tree as a dependency tree that expresses the lexical dependencies binding the words of the derived sentence. In DVTAG we propose a different approach to computing lexical dependencies. We consider the production of a dependency tree from a DVTAG left-context as the translation of a constituency tree into a dependency tree. In particular we can transform a DVTAG left-context in a dependency tree by considering the dominance relation between the maximal projections of the terminal nodes. Using the unification mechanism on the head-feature, the dependency tree produced contains one node per lexical head, and a lexical head dominates another if the corresponding maximal projections in the derived tree stand in a dominance relation. This mechanism allows us to correctly describe the lexical dependencies of non-projective sentences. Figure 1-a depicts the evolution of the dependency tree for the sentence *John loves Mary madly.*.

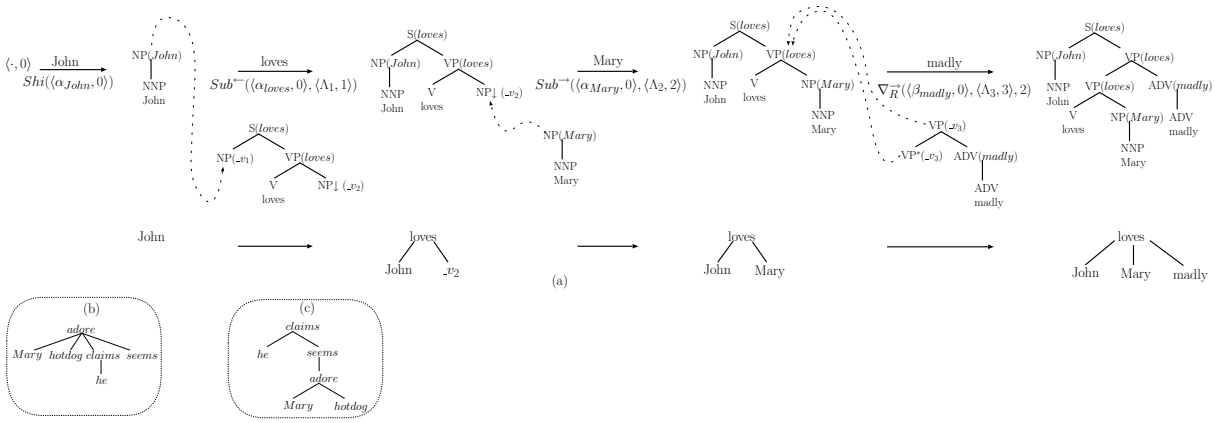

Figure 1: The DVTAG derivation and the corresponding sequence of dependency trees for the sentence *John loves Mary madly*. Since in DVTAG we always combine a *left-context* with an elementary tree, the number of attachment operations increases from two in LTAG to eight in DVTAG. In the derivation we use *initialization, inverse substitution, substitution, adjoining from the right*.

The head-feature allows us to build a dependency tree that contains the relations between the fully specified lexical items, but contains the relations involving underspecified lexical items too, e.g. the relation between $\_v_2$ and *loves* in the second step of Figure 1-a. A similar mechanism has been proposed in (Lin, 1995): Lin proposed a *percolation procedure* with the aim to extract the lexical dependencies from a constituency tree.

The DVTAG mechanism to compute lexical dependencies sheds a new light on some problematic sentences. The sentence "*Hotdog, he claims Mary seems to adore*" (the *hotdog sentence*) has some lexical dependencies that cannot be described using standard LTAG derivation trees. As pointed out by Rambow et al (Rambow et al., 2001), both the elementary auxiliary trees anchored by *claim* and *seem* are adjoined in the initial tree anchored by *adore*, while in the dependency tree describing the sentence, *seem* depends on *claim*, and *adore* depends on *seem*. In other words the elementary trees anchored by *claim* and *seem* are both children of *adore* in the LTAG derivation tree, while *seem* is a child of *claim*, and *adore* is a child of *seem* in the dependency tree. Thus the LTAG derivation tree (Figure 1-b) does not correspond to the correct dependency tree for this sentence (Figure 1-c). The key point is that LTAG uses the adjoining operation to insert both modifiers and clausal complements, and there is no way in standard LTAG to tell apart these two relations in the derivation tree. In DVTAG we can produce the correct dependency tree for the *hotdog sentence*. In fact in DVTAG we do not use the derivation structure to build the dependency tree, and the head-feature produces a simple characterization of the distinction between athematic and complement auxiliary trees. Following the Kroch's characterization (Kroch, 1989), in an athematic auxiliary tree the head-value of the foot node will be the same as that of the root node (e.g. $\_v_3$ in the elementary tree anchored by *madly* in Figure 1), while this is not true for a complementary auxiliary tree. So, in DVTAG the head-feature allows us to distinguish athematic and complement auxiliary trees straightforwardly, and so, using the conversion algorithm from constituency to dependency we are able to derive the correct dependency structure. The DVTAG mechanism to compute lexical dependencies can also correctly describe some other problematic cases, such as left-recursive dependencies (e.g. English genitive construction) and Dutch cross-serial dependencies.

References

A. Joshi and Y. Schabes. 1997. Tree-adjoining grammars. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, pages 69–123. Springer.

Y. Kamide, G. T. M. Altmann, and S. L. Haywood. 2003. The time-course of prediction in incremental sentence

processing: Evidence from anticipatory eye movements. *Journal of Memory and Language Language*, 49:133–156.

A. Kroch. 1989. Asymmetries in long distance extraction in a Tree Adjoining Grammar. In M. Baltin and A. Kroch, editors, *Alternative Conceptions of Phrase Structure*, pages 66–98. University of Chicago Press, Chicago.

D. Lin. 1995. A dependency-based method for evaluating broad-coverage parsers. In *IJCAI95*.

V. Lombardo, A. Mazzei, and P. Sturt. 2004. Competence and performance grammar in incremental parsing. In *Incremental Parsing: Bringing Engineering and Cognition Together, Workshop at ACL-2004*, pages 1–8, Barcelona, July.

W. Marslen-Wilson. 1973. Linguistic structure and speech shadowing at very short latencies. *Nature*, 244:522–523.

D. Milward. 1994. Dynamic dependency grammar. *Linguistics and Philosophy*, 17(6).

O. Rambow, D. Weir, and K. Vijay-Shanker. 2001. D-tree substitution grammars. *Computational Linguistics*, 27(1):87–121.

E. P. Stabler. 1994. The finite connectivity of linguistic structure. In C. Clifton, L. Frazier, and K. Reyner, editors, *Perspectives on Sentence Processing*, pages 303–336. Lawrence Erlbaum Associates.