Handling Unlike Coordinated Phrases in TAG by Mixing Syntactic Category and Grammatical Function

Carlos A. Prolo
Faculdade de Informática - PUCRS
Porto Alegre, RS, 90619-900, Brazil
prolo@inf.pucrs.br

Abstract

Coordination of phrases of different syntactic categories has posed a problem for generative systems based only on syntactic categories. Although some prefer to treat them as exceptional cases that should require some extra mechanism (as for elliptical constructions), or to allow for unrestricted cross-category coordination, they can be naturally derived in a grammatic functional generative approach. In this paper we explore the ideia on how mixing syntactic categories and grammatical functions in the label set of a Tree Adjoining Grammar allows us to develop grammars that elegantly handle both the cases of same- and cross-category coordination in an uniform way.

1 Introduction

Generative grammars that we commonly hear about in computational linguistics are usually based on syntactic categories. This is also the case when the formalism used is the Tree Adjoining Grammars (TAGs). Large scale handcrafted grammars for many languages have been built based on this paradigm (Joshi, 2001; XTAG Research Group, 2001; Kroch and Joshi, 1985; Abeillé and Candito, 2000; Candito, 1998; Becker, 1993; Frank, 2002; Joshi and Schabes, 1997; Abeillé and Rambow, 2000) as well as grammars extracted from corpora (Chen and Vijay-Shanker, 2000; Chiang, 2000; Hwa, 1999; Xia et al., 2001; Xia, 2001). The latter is partly due to the fact that large scale annotated corpora such as the Penn Treebank (Marcus et al., 1994; Bies et al., 1995) give primacy to syntactic categories. After all this is the most strongly sedimented generative approach at least since (Chomsky, 1957).

Computational approaches of grammar based on grammatical function such as that of Susumu Kuno (Kuno, 1987) have been given less importance. Although we can think of simply inserting functional labels in elementary trees or use them in a meta-level to generate the grammar, such as in (Candito, 1998; Kinyon, 2000; Clément and Kinyon, 2003), such tags are generally not seen as an essential part of the derivational process.

Nevertheless coordination is such an inherently functional phenomenon as we show next. Consider the sentences in (1) and (2). These are examples of regular coordination between phrases of the same category. They can easily be handled in the traditional grammar approaches of syntactic category.

(1) She flew [$_{PP}$ on May 1st and on July 4th].

(2) They sell [$_{ADJP}$ electric and electronic] products.

Now look at the cases in (3) and (4). They are different in the sense that the coordination is across categories. This poses a strong problem to the traditional grammar of syntactic categories. This has been noticed for TAGs in (Prolo, 2002). Recently this has also been tackled in the HPSG framework by (Sag, 2003) and (Abeillé, 2004). The Penn Treebank calls this constituents *UCP* for *"Unlike Coordinated Phrases"* (Bies et al., 1995). The problem is that we would need rules of the kind below (using context-free rules for short – see (Prolo, 2002) for TAGs). Basically all pairs of constituents can be coordinated but we can not assign to the resulting constituents either of the subconstituent tags.

Proceedings of the 8th International Workshop on Tree Adjoining Grammar and Related Formalisms, pages 137–140,
Sydney, July 2006. ©2006 Association for Computational Linguistics

UCP → ADVP CC PP
UCP → PP CC ADVP
UCP → ADJP CC NP
UCP → NP CC ADJP

(3) She flew [$_{??}$ yesterday and on July 4th].

(4) They sell [$_{??}$ electronic and computer] devices.

However, UCP coordination is not random. Two constituents can be coordinated only when they are fulfilling the same grammatical function (with respect to a third head). In (3) they are playing the role of adverbial adjuncts of *went*. Either one can engage in that relation individually and hence they can be coordinated while playing that role. Likewise in (4) the adjective *electronic* and the noun *computer* are both fine as left NP modifiers. Therefore they can be conjoined as such. As a final example, consider the sentences in (5). Because the direct object of the verb *know* can be realized as either an NP or a sentential complement, they can be coordinated in that role as shown in (6).

(5) I know the answer.
I know that you don't know it.

(6) I know [the answer and that you don't know it].

Clearly the recursive process of conjoining the constituents is at the grammatic functional level. We show next how we can solve this problem elegantly by mixing grammatical function and syntactic category in the set of symbols for the tree nodes of a TAG.

2 A Grammar of Grammatical Functions and Syntactic Categories

The elementary trees in our grammar are the projection of a lexical item as usual in Lexicalized TAGs. However, root nodes do not correspond to syntactic categories, but to grammatical functions. The node for the function then dominates syntactic category nodes, according to the way the function is realized syntactically. Figure 1 shows trees for an intransitive main clause and an NP subject.[1]

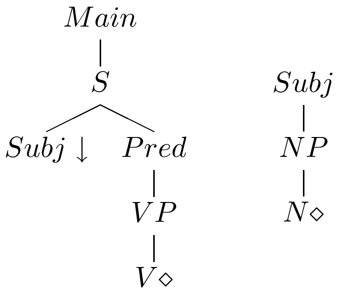

Figure 1: Elementary trees for Intransitive Main Clause and NP Subject.

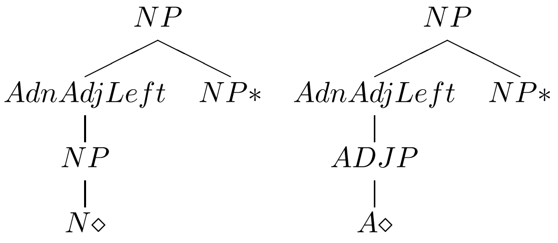

Figure 2: Elementary trees for Left Adnominal Adjuncts.

Figure 2 has trees for NP left modifiers (adnominal adjunct) realized either as an NP or an ADJP.

Finally, in Figure 3 we can see the trees for coordination of left adnominal adjuncts. Notice that they adjoin at the function node (*AdnAdjLeft*) therefore allowing for the coordination of anything that can fulfill that role, be them equal categories as in (2) or the UCP case in (4). In Figure 4 we show an additional example with a PP right NP modifier. It should be straightforward to see how to build trees for *AdnAdjRight* coordination of constituents realized by a PP or a relative clause.

In Figure 5 we finally get to subcategorization. In any approach to grammar development we have to make decisions between explicitly modeling certain restrictions in the tree structure or through features (of a *feature based* TAG). That can be seen ubiquitously in the XTAG grammar (XTAG Research Group, 2001). We can use the tree of the figure with verbs such like *eat* and *know*, having trees to realize the direct object as either an NP or a sentence. Features in the lexical items would prevent the derivation of *eat* with a sentential complement. Another approach would be to further detail the tree into one with a built in NP object

[1] Figures generally show templates where a diamond indicates where the lexical item would be substituted in, though occasionally we insert the lexical item itself.

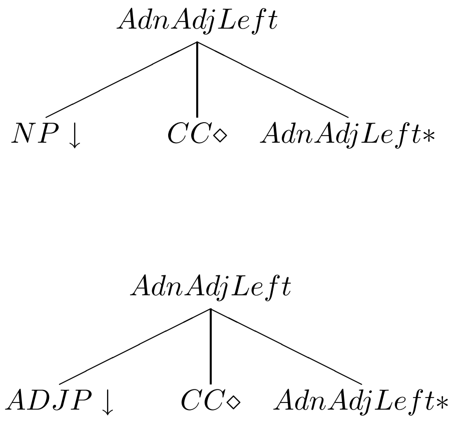

Figure 3: Elementary trees for Coordination of Left Adnominal Adjuncts.

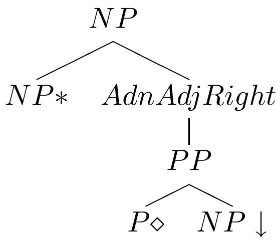

Figure 4: Elementary trees for a PP as Right Adnominal Adjunct.

and another with a sentential complement. However, realization constraints would still have to be present to allow for the coordination of only the constituents that are allowed for the specific verb. For the reader unfamiliar with grammar modeling we notice this is not a drawback of the approach. Constraints beyond those represented in the structure are constantly made as a way to avoid irrational growth of a grammar.

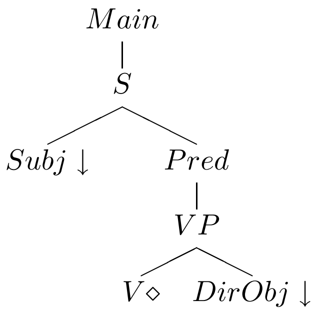

Figure 5: Elementary tree for a Verb that has a Direct Object

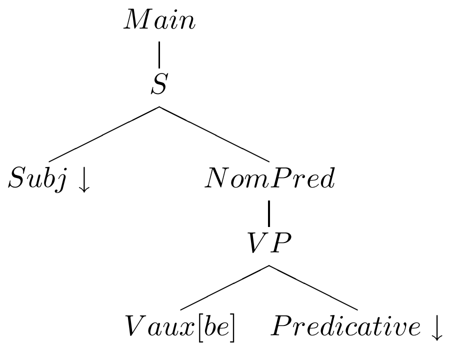

Figure 6: Elementary tree for Predicative Clauses

In Figure 6 we show still another interesting case: the predicative clauses.[2] We include it for this is a rich context for unlike coordination. One can easily see how to generate trees for coordinating NPs, PPs and ADJPs, as predicative constituents so as to allow for (7).

(7) John was [a gentlemen, always happy, and never in bad mood].

3 Conclusions

We showed in this paper how to build a Tree Adjoining Grammar of grammatical functions and syntactic categories, mixed together in a principled way of function and possible realizations. It brings the benefits of allowing handling language phenomena which are generative at each of the two sides.

In particular, we showed how it solves the problem of coordination of constituents of distinct syntactic categories.

Elementary trees are not clumsy. On the contrary they bring additional information to the structure with minimal addition of nodes. This information could otherwise be hidden in *node features*, which are generally used to represent information that would be costly to maintain explicit in the tree structure.

Finally we can see that this structure can be easily incorporated in a supervised grammar inference algorithm such as that of (Xia, 2001), provided the annotated corpus has grammatical function information. In fact this is the case in the Penn Treebank, and Xia's algorithm allows it to be used [3]. Inferring the different kinds of verbs, with respect to the functions they subcategorize for and

[2] Again this is one approach to modeling predicatives, with the auxiliary verb *be* anchoring the tree and the predicative as a substitution node. The alternative used in the XTAG grammar of having the predicative head as anchor would be possible as well.

[3] The same is true of other algorithms such as (Chen and Vijay-Shanker, 2000)'s.

their realizations is an important issue here, and is also feasible (see (Kinyon and Prolo, 2002)).

References

Anne Abeillé and Marie-Helene Candito. 2000. Ftag: A lexicalized Tree Adjoining Grammar for French. In Abeillé and Rambow (Abeillé and Rambow, 2000), pages 305–329.

Anne Abeillé and Owen Rambow, editors. 2000. Tree Adjoining Grammars: formalisms, linguistic analysis and processing. CSLI, Stanford, CA, USA.

Anne Abeillé. 2004. A lexicalist and construction-based approach to coordinations. In Stefan Mller, editor, Proceedings of the 10th International Conference on HPSG (HPSG'03), Michigan State University, Michigan, USA. Available at: http://cslipublications.stanford.edu/HPSG/4.

Tilman Becker. 1993. HyTAG: A new Type of Tree Adjoining Grammars for Hybrid Syntactic Representation of Free Word Order Lang uages. Ph.D. thesis, Universität des Saarlandes.

Ann Bies, Mark Ferguson, Karen Katz, and Robert MacIntyre. 1995. Bracketing guidelines for the Penn Treebank II style Penn Treebank Project.

Marie-Helene Candito. 1998. Building parallel LTAG for french and italian. In Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 16th International Conference on Computational Linguistics, pages 211–217, Montreal, Canada.

John Chen and K. Vijay-Shanker. 2000. Automated extraction of TAGs from the Penn Treebank. In Proceedings of the 6th International Workshop on Parsing Technologies, Trento, Italy.

David Chiang. 2000. Statistical parsing with an automatically-extracted Tree Adjoining Grammar. In Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics, Hong Kong, China.

N. Chomsky. 1957. Syntactic Structures. Mouton, The Hague.

L. Clément and A. Kinyon. 2003. Generating parallel multilingual LFG-TAG grammars using a Meta-Grammar. In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, Sapporo, Japan.

Robert Frank. 2002. Phrase Structure Composition and Syntactic Dependencies. MIT Press, Cambridge, MA, USA.

Rebecca Hwa. 1999. Supervised Grammar Induction Using Training Data with Limited Constituent Information. In Proceedings of 37th Annual Meeting of the Association for Computational Linguistics (ACL '99), pages 20–26, College Park, MD, USA.

Aravind K. Joshi and Yves Schabes. 1997. Tree-Adjoining Grammars. In A. Salomaa and G. Rozenberg, editors, Handbook of Formal Languages, volume 3, pages 69–123. Springer-Verlag, Berlin.

Aravind K. Joshi. 2001. The XTAG project at Penn. In Proceedings of the 7th International Workshop on Parsing Technologies (IWPT-2001), Beijing, China. Invited speaker.

Alexandra Kinyon and Carlos A. Prolo. 2002. Identifying verb arguments and their syntactic function in the Penn Treebank. In Proceedings of the Third International Conference on Language Resources and Evaluation (LREC), pages 1982–87, Las Palmas, Spain.

Alexandra Kinyon. 2000. Hypertags. In Proceedings of the 18th International Conference on Computational Linguistics (COLING'2000), Saarbrücken, Germany.

Anthony S. Kroch and Aravind K. Joshi. 1985. The linguistic relevance Tree Adjoining Grammar. Technical Report MS-CIS-85-16, University of Pennsylvania.

Sususmu Kuno. 1987. Functional Grammar. University of Chicago Press, Chicago, Il, USA.

Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The Penn Treebank: Annotating predicate argument structure. In Proceedings of the 1994 Human Language Technology Workshop.

Carlos A. Prolo. 2002. Coping with problems in grammars automatically extracted from treebanks. In Proceedings of the Workshop on Grammar Engineering and Evaluation, pages 36–42, Taipei, Taiwan.

Ivan Sag. 2003. Coordination and underspecification. In Jongbok Kim and Stephen Wechsler, editors, Proceedings of the 9th International Conference on HPSG (HPSG'02), Kyung-Hee University, Seoul, Korea. Available at: http://cslipublications.stanford.edu/HPSG/3/hpsg02.htm.

Fei Xia, Chung-Hye Han, Martha Palmer, and Aravind Joshi. 2001. Automatically Extracting and Comparing Lexicalized Grammars for Different Languages. In Proc. of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001), Seattle, Washington.

Fei Xia. 2001. Investigating the Relationship between Grammars and Treebanks for Natural Languages. Ph.D. thesis, Department of Computer and Information Science, University of Pennsylvania.

The XTAG Research Group. 2001. A Lexicalized Tree Adjoining Grammar for English. Technical Report IRCS 01-03, University of Pennsylvania.