Consistent Grammar Development Using Partial-Tree Descriptions for Lexicalized Tree-Adjoining Grammars

Fei Xia, Martha Palmer, K. Vijay-Shanker, Joseph Rosenzweig
Institute for Research in Cognitive Science
University of Pennsylvania
400A, 3401 Walnut Street
Philadelphia, PA 19104,USA
fxia/mpalmer/vshanker/josephr@linc.cis.upenn.edu

1 Introduction

An important characteristic of an FB-LTAG is that it is lexicalized, i.e., each lexical item is anchored to a tree structure that encodes subcategorization information. Trees with the same canonical subcategorizations are grouped into tree families. The reuse of tree substructures, such as wh-movement, in many different trees creates redundancy, which poses a problem for grammar development and maintenance (Vijay-Shanker and Schabes, 1992). To consistently implement a change in some general aspect of the design of the grammar, all the relevant trees currently must be inspected and edited. Vijay Shanker and Schabes suggested the use of hierarchical organization and of tree descriptions to specify substructures that would be present in several elementary trees of a grammar. Since then, in addition to ourselves, Becker, (Becker, 1994), Evans et al. (Evans et al., 1995), and Candito(Candito, 1996) have developed systems for organizing trees of a TAG which could be used for developing and maintaining grammars.

Our system is based on the ideas expressed in Vijay-Shanker and Schabes, (Vijay-Shanker and Schabes, 1992), to use partial-tree descriptions in specifying a grammar by separately defining pieces of tree structures to encode independent syntactic principles. Various individual specifications are then combined to form the elementary trees of the grammar. Our paper begins with a description of our grammar development system and the process by which it generates the Penn English grammar as well as a Chinese TAG. We describe the significant properties of both grammars, pointing out the major differences between them, and the methods by which our system is informed about these language-specific properties. We then compare our approach to other grammar development approaches for LTAG such as the specification of TAGs in DATR (Evans et al., 1995) and Candito's implementation (Candito, 1996).

2 System Overview

In our approach, three types of components – subcategorization frames, blocks and lexical redistribution rules – are used to describe lexical and syntactic information. Actual trees are generated automatically from these abstract descriptions. In maintaining the grammar only the abstract descriptions need ever be manipulated; the tree descriptions and the actual trees which they subsume are computed deterministically from these high-level descriptions.

2.1 Subcategorization frames

Subcategorization frames specify the category of the main anchor, the number of arguments, each argument's category and position with respect to the anchor, and other information such as feature equations or node expansions. Each tree family has one canonical subcategorization frame.

2.2 Blocks

Blocks are used to represent the tree substructures that are reused in different trees, i.e. blocks subsume classes of trees. Each block includes a set of nodes, dominance relation, parent relation, precedence relation between nodes, and feature equations. This follows the definition of the tree descriptions specified in a logical language patterned after Rogers and Vijay-Shanker(Rogers and Vijay-Shanker, 1994).

Blocks are divided into two types according to their functions: subcategorization blocks and transformation blocks. The former describes structural configurations incorporating the various information in a subcategorization frame. For example, some of the subcategorization blocks used in the development of the English grammar are shown in Figure 1.[1]

When the subcategorization frame for a verb is given by the grammar developer, the system will automatically create a new block (of code) by essentially selecting the appropriate primitive subcategorization blocks corresponding to the argument information specified in that verb frame.

The transformation blocks are used for various transformations such as wh-movement. These transformation blocks do not encode rules for modifying trees, but rather describe the properties of a particular syntactic construction. Figure 2 depicts our representation of phrasal extraction. This can be specialized to give the blocks for wh-movement, topicalization, relative clause formation, etc. For example, the wh-movement block is defined by further specifying that the ExtractionRoot is labeled S, the NewSite has a +wh feature, and so on.

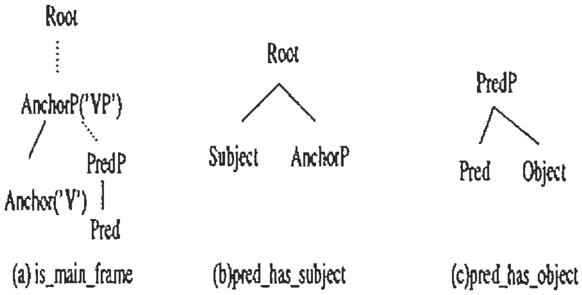

Figure 1: Some subcategorization blocks

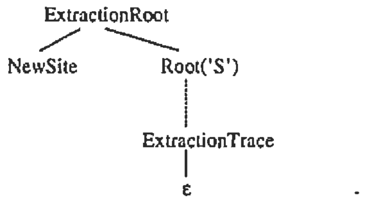

Figure 2: Transformation block for extraction

2.3 Lexical Redistribution Rules (LRRs)

The third type of machinery available for a grammar developer is the Lexical Redistribution Rule (LRR). An LRR is a pair (r_l, r_r) of subcategorization frames, which produces a new frame when applied to a subcategorization frame s, by first *matching*[2] the left frame r_l of r to s, then combining information in r_r and s. LRRs are introduced to incorporate the connection between subcategorization frames. For example, most transitive verbs have a frame for active(a subject and an object) and another frame for passive, where the object in the former frame becomes the subject in the latter. An LRR, denoted as passive LRR, is built to produce the passive subcategorization frame from the active one. Similarly, applying dative-shift LRR to the frame with one NP subject and two NP objects will produce a frame with an NP subject and an PP object.

Besides the distinct content, LRRs and blocks also differ in several aspects:

[1] In order to focus on the use of tree descriptions and to make the figures less cumbersome, we show only the structural aspects and do not show the feature value specification. The parent, (immediate dominance), relationship is illustrated by a plain line and the dominance relationship by a dotted line. The arc between nodes shows the precedence order of the nodes are unspecified. The nodes' categories are enclosed in parentheses.

[2] Matching occurs successfully when frame s is compatible with r_l in the type of anchors, the number of arguments, their positions, categories and features. In other words, incompatible features etc. will block certain LRRs from being applied.

- They have different functionalities: Blocks represent the substructures that are reused in different trees. They are used to reduce the redundancy among trees; LRRs are introduced to incorporate the connections between the closely related subcategorization frames.
- Blocks are strictly additive and can be added in any order. LRRs, on the other hand, produce different results depending on the order they are applied in, and are allowed to be non-additive, i.e., to remove information from the subcategorization frame they are being applied to, as in the procedure of passive from active.

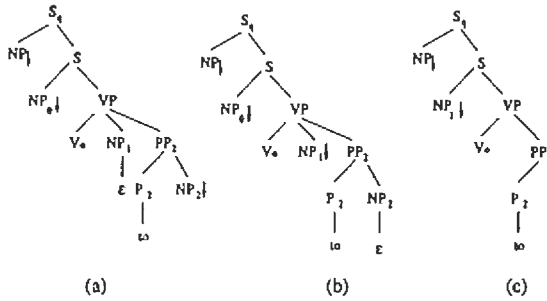

Figure 3: Elementary trees generated from combining blocks

2.4 Tree generation

To generate elementary trees, we begin with a canonical subcategorization frame. The system will first generate related subcategorization frames by applying LRRs, then select subcategorization blocks corresponding to the information in the subcategorization frames, next the combinations of these blocks are further combined with the blocks corresponding to various transformations, finally, a set of trees are generated from those combined blocks, and they are the tree family for this subcategorization frame. Figure 3 shows some of the trees produced in this way. For instance, the last tree is obtained by incorporating information from the ditransitive verb subcategorization frame, applying the dative-shift and passive LRRs, and then combining them with the wh-non-subject extraction block.

3 Generating grammars

We have used our tool to specify a grammar for English in order to produce the trees used in the current English XTAG grammar. We have also used our tool to generate a large grammar for Chinese. In designing these grammars, we have tried to specify the grammars to reflect the similarities and the differences between the languages. The major features of our specification of these two grammars are summarized in Table 1.

	English	Chinese
examples of LRRs	passive dative-shift ergative	bei-construction object fronting ba-construction
examples of transformation blocks	wh-question relativization declarative	topicalization relativization argument-drop
# LRRs	6	12
# subcat blocks	34	24
# trans blocks	8	15
# subcat frames	43	23
# trees generated	638	280

Table 1: Major features of English and Chinese grammars

By focusing on the specification of individual grammatical information, we have been able to generate nearly all of the trees (91.3% - 638 out of the 699) from the tree families used in the current English grammar developed at Penn[3]. Our approach, has also exposed certain gaps in the Penn grammar. We are encouraged with the utility of our tool and the ease with which this large-scale grammar was developed.

We are currently working on expanding the contents of subcategorization frame to include trees for other categories of words. For example, a frame which has no specifier and one NP complement and whose predicate is a preposition will correspond to PP → P NP tree. We'll also introduce a modifier field and semantic fea-

[3] We have not yet attempted to extend our coverage to include punctuation, it-clefts, and a few idiosyncratic analyses that are included in the sixty trees we are not generating.

tures, so that the head features will propagate from modifiee to modified node, while non-head features from the predicate as the head of the modifier will be passed to the modified node.

4 Comparison to Other Work

Evans, Gazdar and Weir (Evans et al., 1995) also discuss a method for organizing the trees in a TAG hierarchically, using an existing lexical representational system, DATR (Evans and Gazdar, 1989). Since DATR can not capture directly dominance relation in the trees, these must be simulated by using feature equations.

There are substantial similarities and significant differences in our approach and Candito's approach, which she applied primarily to French and Italian. Both systems have built upon the basic ideas expressed in (Vijay-Shanker and Schabes, 1992) for organizing trees hierarchically and the use of tree descriptions that encode substructures found in several trees. The main difference is how Candito uses her dimensions in generating the trees. Her system imposes explicit conditions on how the classes appearing in the hierarchy can be combined, based on which dimension they are in. For example, one condition states that only a terminal node (leaf node of a hierarchy) of the second dimension can be used in constructing a tree. Therefore two redistributions (such as passive and causative) can be used in a single tree only when a new passive-causative terminal node is first created manually. In contrast, our approach automatically considers all possible applications of LRRs, and discards those that are inconsistent.

5 Conclusion

We have described a tool for grammar development in which tree descriptions are used to provide an abstract specification of the linguistic phenomena relevant to a particular language. In grammar development and maintenance, only the abstract specifications need to be edited, and any changes or corrections will automatically be proliferated throughout the grammar. In addition to lightening the more tedious aspects of grammar maintenance, this approach also allows a unique perspective on the general characteristics of a language. Defining hierarchical blocks for the grammar both necessitates and facilitates an examination of the linguistic assumptions that have been made with regard to feature specification and tree-family definition. This can be very useful for gaining an overview of the theory that is being implemented and exposing gaps that remain unmotivated and need to be investigated. The type of gaps that can be exposed could include a missing subcategorization frame that might arise from the automatic combination of blocks and which would correspond to an entire tree family, a missing tree which would represent a particular type of transformation for a subcategorization frame, or inconsistent feature equations. By focusing on syntactic properties at a higher level, our approach allows new opportunities for the investigation of how languages relate to themselves and to each other.

References

Tilman Becker. 1994. Patterns in metarules. In *Proceedings of the 3rd TAG+ Conference*, Paris, France.

Marie-Helene Candito. 1996. A principle-based hierarchical representation of ltags. In *Proceedings of COLING-96*, Copenhagen, Denmark.

Roger Evans and Gerald Gazdar. 1989. Inference in datr. In *EACL-89*.

Roger Evans, Gerald Gazdar, and David Weir. 1995. Encoding Lexicalized Tree Adjoining Grammars with a Nonmonotonic Inheritance Hierarchy. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics(ACL '95)*, Cambridge, MA.

James Rogers and K. Vijay-Shanker. 1994. Obtaining Trees from their Descriptions: An Application to Tree Adjoining Grammars. *Computational Intelligence*, 10(4).

K. Vijay-Shanker and Yves Schabes. 1992. Structure sharing in lexicalized tree adjoining grammar. In *Proceedings of the* 15th *International Conference on Computational Linguistics (COLING '92)*, Nantes, France.