Building a LCS-Based Lexicon in TAGs*

Bonnie J. Dorr
Department of Computer Science
University of Maryland
College Park, MD 20742
bonnie@cs.umd.edu

Martha S. Palmer
CIS Department, Moore School
University of Pennsylvania
Philadelphia, PA 19104
mpalmer@linc.cis.upenn.edu

1 Introduction

We are interested in building a lexicon for interlingual machine translation (MT) and in examining the formal properties of an interlingua as a language in its own right. As such it should be possible to define a lexicalized grammar for the representation of lexical entries and a set of operations over that grammar that can be used to both analyze and generate interlingua representations. The interlingua we discuss in this paper is Lexical Conceptual Structure (LCS) as formulated by Dorr (1993) based on work by Jackendoff (1983, 1990). This is described in the next section, and is followed by the presentation of a grammar for LCS as a representation language. The grammar formalism whose operations we examine with respect to their ability to compose LCS representations is Feature-Based Lexicalized Adjoining Grammar, (FB-LTAG), a version of Tree Adjoining Grammar (TAG) (Joshi et al (1975), Schabes (1990), Vijay-Shanker (1987)), and its description, along with example TAG structures, forms our final section. What we find is that the implementation of LCS as a TAG, although not completely straightforward, can be done, providing the full power of the well-defined mathematical properties of TAGs as a basis for describing the formal properties of LCS.

*We would like to acknowledge the essential contributions of Aravind Joshi, B. Srinivas, Tilman Becker, Joseph Rosenzweig, and the NLP group at Maryland, for their advice and input, and in particular Dania Egedi and Clare Voss who worked out the details of the LCS trees and their corresponding TAG representations in this paper, and who have provided a calm, considered perspective throughout. This research was supported, in part, by the Army Research Office under contract DAAL03-91-C-0034 through Battelle Corporation, by the National Science Foundation under grants NYI IRI-9357731, NSF/CNRS INT-9314583, NSF IRI-9120788, by Alfred P. Sloan Research Fellow Award BR3336, and by the Army Research Institute under contract MDA-903-92-R-0035 through Microelectronics and Design, Inc.

2 Lexical Conceptual Structure (LCS)

A central issue that arises with respect to the use of a language-independent representation in any multilingual system is that of defining a set of primitives to represent cross-linguistic phenomena. Because it is generally difficult to define such a set, many researchers have abandoned the use of an intermediate representation in multilingual applications. (See, for example, Vauquois and Boitet (1985).) However, recently, there has been a resurgence of interest in the area of lexical representation and organization (with special reference to verbs) that has initiated an ongoing effort to delimit the classes of lexical knowledge required to process natural language. (See, e.g., Grimshaw (1990), Hale and Keyser (1993), Jackendoff (1983, 1990), Levin (1993), Pustejovsky (1991), Olsen (1991), and Zubizarreta (1987).) As a result of this effort, it has become increasingly more feasible to isolate the components of meaning common to verbs participating in particular classes. These components of meaning can then be used to determine the lexical representation of verbs across languages.

The LCS approach views semantic representation as a subset of conceptual structure, i.e., the language of mental representation. It abstracts away from syntax just far enough to enable language independent encoding, while retaining enough structure to be sensitive to the requirements for multilingual processing. Jackendoff's approach includes *types* such as Event and State, which are specialized into *primitives* such as GO, STAY, BE, GO-EXT, and ORIENT. As an example of how the primitive GO is used to represent sentential semantics, consider the following sentence:

The ball rolled toward Beth.
[$_{\text{Event}}$ GO ([$_{\text{Thing}}$ BALL],
[$_{\text{Path}}$ TOWARD
([$_{\text{Position}}$ AT
([$_{\text{Thing}}$ BALL], [$_{\text{Thing}}$ BETH])])])]

This representation illustrates one dimension (i.e., the *spatial* dimension) of Jackendoff's representation. Another dimension is the *causal* dimension, which includes the primitives CAUSE and LET. These primitives take a Thing and an Event as arguments. Thus, we could embed the structure shown in the sentence above within a causative construction:

John rolled the ball toward Beth.
[Event CAUSE
  ([Thing JOHN],
   [Event GO ([Thing BALL],
              [Path TOWARD
                [Position AT
                  ([Thing BALL], [Thing BETH])])])])

Jackendoff includes a third dimension by introducing the notion of *field*. This dimension extends the semantic coverage of spatially oriented primitives to other domains such as Possessional, Temporal, Identificational, Circumstantial, and Existential.[1] For example, the primitive GO_{Poss} refers to a GO event in the Possessional field as in the following sentence:

Beth received the doll.
[Event GO_{Poss}
  ([Thing DOLL],
   [Path TO_{Poss}
     ([Position AT_{Poss}
       ([Thing DOLL], [Thing BETH])])])]

To further illustrate the notion of field, the GO primitive can be used in the Temporal and Identificational fields:

The meeting went from 2:00 to 4:00.
[Event GO_{Temp}
  ([Thing MEETING],
   [Path $FROM_{Temp}$
     ([Position AT_{Temp}
       ([Thing MEETING], [Time 2:00])])]
   [Path TO_{Temp}
     ([Position AT_{Temp}
       ([Thing MEETING], [Time 4:00])])])]

The frog turned into a prince.
[Event GO_{Ident}
  ([Thing FROG],
   [Path TO_{Ident}
     ([Position AT_{Ident}
       ([Thing FROG], [Thing PRINCE])])])]

To illustrate the use of this representation in the lexicon, consider the following example:

E: I like Mary
S: María me gusta
   (Mary (to) me pleases)

The language-independent representation for this example looks like the following:

[State BE_{Ident} ([Thing I],
  [Position AT_{Ident}
    ([Thing I], [Thing MARY])],
  [Manner LIKINGLY])]

This representation roughly means "I am in an identificational state LIKINGLY with respect to Mary." Both the Spanish and English sentences are based on this representation; the syntactic distinction (i.e., the subject-object reversal) is captured by means of parameterization in the lexicon:

Lexical Entry for like:
[State BE_{Ident} ([Thing :EXT W],
  [Position AT_{Ident}
    ([Thing W], [Thing :INT Z])],
  [Manner LIKINGLY])]

Lexical Entry for gustar:
[State BE_{Ident} ([Thing :INT W],
  [Position AT_{Ident}
    ([Thing W], [Thing :EXT Z])],
  [Manner LIKINGLY])]

The :INT/:EXT markers are examples of lexical parameterization that allow the system to account for the subject-object reversal of the *like-gustar* example.

2.1 Grammar for LCS

The current task is to explore the LCS representation in the context of an FB-LTAG model in order to test hypotheses about the interlingual representation for machine translation. Our goal is to develop a framework within which we can evaluate, formally, the expressive power of the representation language used in the lexicon, and also to determine systematically the depth of coverage with respect to different cross-linguistic phenomena.

In order to employ the TAG formalism, we must first associate a "syntax" with our "semantics." That is, we must express the wellformedness conditions on the LCS representation in terms of a "grammar," analogous to a context-free description at the level of syntactic structure. (See Figure 1.) In this grammar, curly brackets {} correspond to a choice of one, and only one, item. An example of a Path LCS would be the primitive TO with an Event and a Position as its two "arguments." Primitives correspond to the terminal nodes of the grammar (and are written in all capital letters); Types correspond to the nonterminal nodes of the grammar (and are written in lower case with an initial capital letter). Note that there are closed-class primitives (e.g., Situations and Paths) and open-class primitives (e.g., Things and Properties). There are also primitives

[1] The label Loc has been adopted to distinguish the spatial field from the non-spatial fields. Note that the spatial field is used to denote the primitives that fall in the spatial dimension. Jackendoff argues that spatial primitives are more fundamental than those of other domains (e.g., Possessional). Thus, spatial primitives have their own special status as an independent dimension.

Situation →
  LET {Thing, Event, State} {Event, State} |
  CAUSE {Thing, Event, State} {Event, State} |
  GO {Thing, Event, State} Path |
  GO-EXT {Thing, Event, State} Path |
  ORIENT Thing Path |
  STAY {Thing, Event, State} Position |
  BE {Thing, Event, State} Position
Path →
  {TO, TOWARD, FROM, AWAY-FROM, VIA}
  {Thing, Event, State}
  Position
Position →
  {AT, IN, ON, ...}
  {Thing, Event, State}
  {Thing, Event, State, Property, Time}
Thing →
  {BOOK, PERSON, ROOM, ...}
Time →
  {TODAY, SATURDAY, 2:00, 4:00, ...}
Property →
  {TIRED, HUNGRY, RED, ...}

Figure 1: LCS Wellformedness Conditions Expressed as a Context-Free Grammar

which represent a large, but finite set (e.g., Positions).

Superimposed on this grammar is a set of wellformedness conditions corresponding to the "Field" mentioned in the previous section. In the FB-LTAG framework, the Field is not specified in terms of grammar rules, but is available by means of a feature specification. The feature ensures that Locational GO primitives only take Locational Paths, for instance. The full set of wellformedness conditions is as shown in Figure 2.

3 Tree Adjoining Grammar (TAG)

FB-LTAG is a version of Tree Adjoining Grammar (TAG) (Joshi et al 1975, Schabes 1990, Vijay-Shanker 1987), that has been extended to include lexicalization and unification-based feature structures. In a TAG there are two types of elementary trees: initial trees and auxiliary trees. The frontier of an **initial tree** has as its anchor a terminal; the rest of the nodes on the frontier are non-terminals marked as substitution nodes. In addition to an anchor, and possible non-terminal substitution nodes, an **auxiliary tree** is required to have one node on the frontier marked as the foot node. The foot node must have the same category label as the tree's root node.

From a linguistic perspective, the set of elementary trees anchored by a lexical item represent the item's possible subcategorization frames. In an FB-LTAG, each lexical item is associated with a set of elementary trees, for which it is the lexical **anchor**. Each node in the tree has two sets of feature structures, the TOP and the BOTTOM. The BOTTOM feature structure contains information relating to the subtree rooted at the node, and the TOP feature structure contains information relating to the supertree at that node. Substitution nodes have only a TOP feature structure, while all other nodes have both a TOP and BOTTOM feature structure. Trees can be composed by applying two operations, substitution and adjunction, as shown in Figure 3.[2]

Field (Feature)	**Arguments**
Locational	Arg 1: {Thing, Event, State}
	Arg 2: {Thing, Event}
Temporal	Arg 1: {Event, State}
	Arg 2: {Event, State, Time}
Identificational	Arg 1: Thing
	Arg 2: {Thing, Event, Property}
Possessional	Arg 1: {Thing, Event, State}
	Arg 2: Thing
Instrumental	Arg 1: {Event, State}
	Arg 2: Thing
Perceptual	Arg 1: Thing
	Arg 2: {Thing, Event, State}
Circumstantial	Arg 1: Thing
	Arg 2: {Event, State}
Intentional	Arg 1: {Event, State}
	Arg 2: {Thing, Event, State}
Existential	Arg 1: {Thing, State}
	Arg 2: EXIST

Figure 2: Wellformedness Conditions on LCS Fields

For **substitution** to occur, there must be a non-terminal frontier node marked for substitution in an elementary tree and a corresponding elementary tree whose root has the same label as that node (Figure 3(a)). Then the substitution node is replaced by the corresponding elementary tree. Substitution only operates on the frontier of a tree. **Adjunction**, on the other hand, can operate on an internal node, actually inserting an auxiliary tree at that point (Figure 3(b)). For this to occur, the internal node in the first tree must have the same label as both the root node and the foot node of the tree being adjoined onto it. The TOP feature structure of the internal node unifies with the TOP feature structure of the root node, and the BOTTOM feature structure unifies with the BOTTOM feature structure of the foot

[2] Technically, substitution is a specialized version of adjunction, but it is useful to make a distinction between the two. These figures are used by permission from XTAG (1995). Abbreviations in the tree figure: t=top feature structure, b=bottom feature structure, tr=top feature structure of the root, br=bottom feature structure of the root, tf=top feature structure of the foot, bf=bottom feature structure of the foot, $\cup$ =unification.

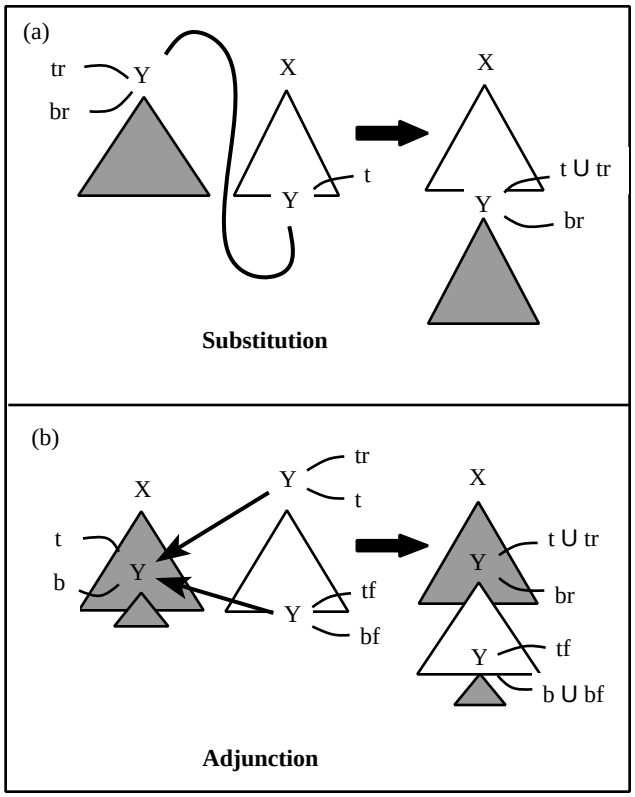

Figure 3: Substitution and Adjunction in FB-LTAG

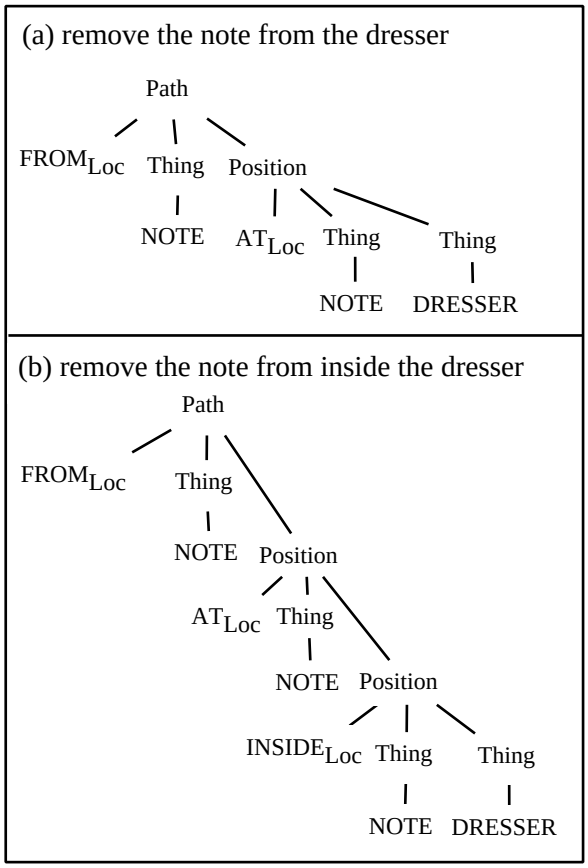

Figure 4: LCS Representations for *from* vs. *from inside*

node. For linguistic reasons, initial trees are non-recursive tree structures, whereas auxiliary trees used for adjunction are required to be recursive. For the final tree to be valid all substitution nodes must be filled, and the TOP and BOTTOM feature structures on each node must unify with each other.

4 LCS Operation Requirements

LCS structures, as described in Dorr and Voss 1994, that correspond to different syntactic elements of a sentence are composed to form the complete sentence representation. For instance, a string of prepositional phrases such as *over the hill, behind the stream, next to the woods* results in a recursive embedding of several path and position predicates. In Voss and Dorr 1994 there is a clearly defined relationship between the representation of the phrase, *from the inside of the dresser* in *remove the note from the inside of the dresser*, whose LCS is given in Figure 4(a), and the simplified version, *from the dresser* given in Figure 4(b).

At first glance, that relationship would appear to be the TAG adjunction operation. However, the necessary conditions for adjunction are not met because there is not a recursive grammar rule for Position that allows a new Position node to be adjoined underneath the AT. In the spirit of an extension proposed by Dorr and Voss (1993), we add a new grammar rule that provides this recursive definition:

Position →
{AT, IN, ON, ...}
{Thing, Event, State}
Position

Then, given the trees in Figure 5(a) and (b), adjunction can be applied to produce the tree in Figure 5(c).

The LCS *overlap* operation as defined in Dorr and Voss (1994) is more problematic. In the LCS representation of the sentence, *John lifted Mary up to the table,* there is a duplication between the UP component of LIFT, and the LCS representation of the UP TO prepositional phrase. The overlap for this example or other similar examples does not conform to adjunction as described here, since more than a single node is duplicated in both trees. It is necessary either for these duplicated nodes to be effectively merged or to find another way of representing the information. The cleanest alternative from the perspective of the TAG formalism is to represent the UP component of LIFT in the feature structure as an overridable default.

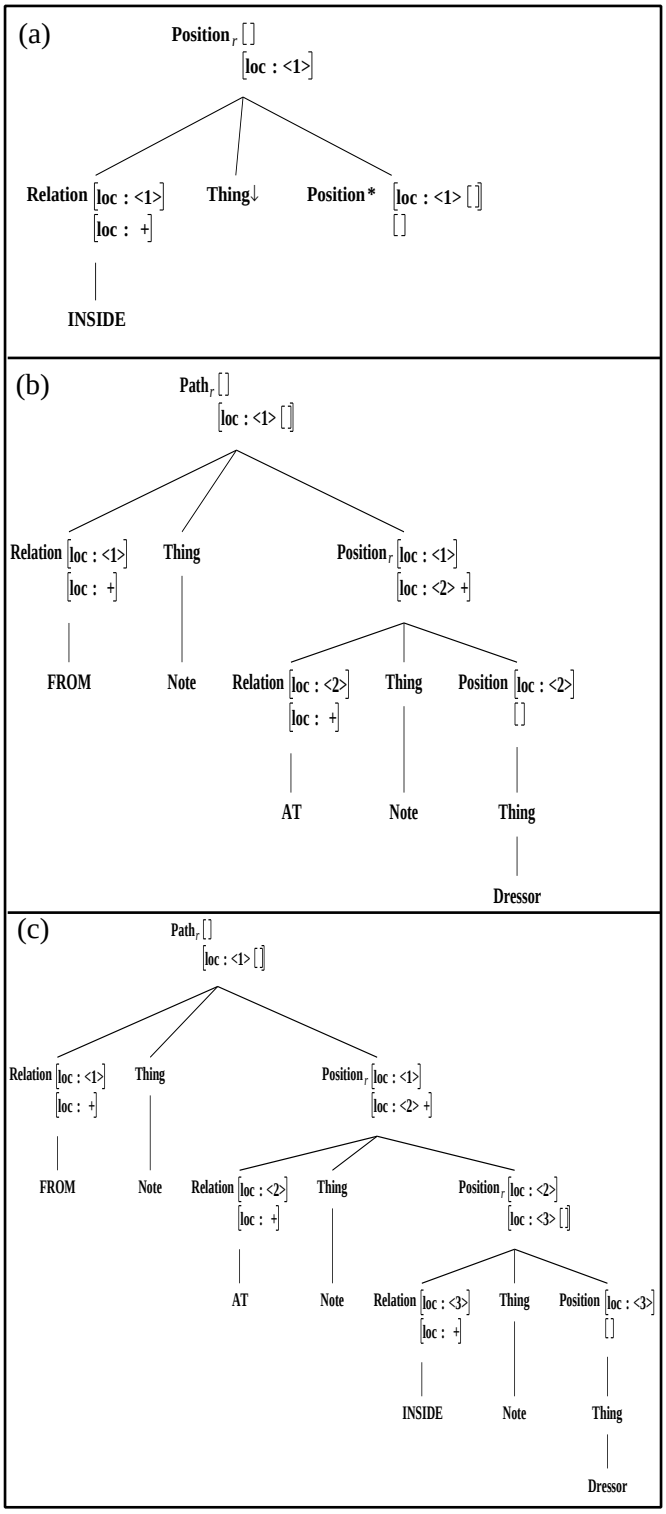

Figure 5: TAG Trees for *inside*, *from*, and *from inside*

In this case, when the LCS structure corresponding to an UP TO prepositional phrase is being adjoined, also with an UP in the feature structure, the two UPs will unify. If a different type of LCS structure needs to be adjoined, such as the structure corresponding to the DOWN prepositional phrase in *The mother lifted the child down from the carousel horse,* then the UP feature can be overridden. If there is no prepositional phrase specified, then the feature still contains the information that the direction is inherently in an UP direction. There may be particular examples where incorporating the required default information as a feature is counter-intuitive. In that case, another possibility which does not require any altering of the LCS structure would be to use partial descriptions of trees for the prepositional phrases, with multi-component adjunction so that they can be adjoined onto the initial tree as described in Shankar 1992.

5 Implications and Future Direction

We have described certain aspects of using the TAG formalism for the implementation of LCS as an interlingua. The standard operations of substitution and adjunction apply, and they can be extended to handle the overlap LCS operation. This gives us a formal structure with well-defined operations that imposes constraints on the composition of LCS, aiding in the regularization of LCS procedures. More importantly, it opens up the possibility of using the well-known mathematical properties of the TAG formalism to prove properties about LCS as an interlingua.

As discussed by Voss and Dorr (this volume), machine translation theory has not yet addressed the issues surrounding how the interlingua of a MT system should be defined or evaluated. We believe that the investigation described herein is the first step toward providing a framework in which MT developers can define and evaluate different lexical representations with respect to coverage and efficiency.

References

Dorr, Bonnie J. and Clare R. Voss, "Machine Translation of Spatial Expressions: Defining the Relation between an Interlingua and a Knowledge Representation System," in Proceedings of Twelfth Conference of the American Association for Artificial Intelligence, Washington, DC, pp. 374–379, 1993.

Dorr, B. and C. Voss (1994). "The Case for A MT Developers' Tool with a Two-Component View of the Interlingua," in Proceedings of the First Annual Association for MT in the Americas Confer-

ence on Partnerships in Translation Technology, Columbia, MD, pp. 40–47, 1994.

Grimshaw, J. (1990). *Argument Structure*, MIT Press, Cambridge, MA.

Hale, K. and J. Keyser (1993). "On Argument Structure and the Lexical Expression of Syntactic Relations," in K. Hale and J. Keyser, *The View From Building 20*, MIT Press, Cambridge, MA.

Jackendoff, R. S. (1983). *Semantics and Cognition*, MIT Press, Cambridge, MA.

Jackendoff, R. S. (1990). *Semantics Structures*, MIT Press, Cambridge, MA.

Joshi, A.K., Levy, L., Takahashi, M. (1975). "Tree Adjunct Grammars," In *Journal of Computer and System Sciences.*

Joshi, A.K., Schabes, Y. (1991), "Fixed and Flexible Phrase Structure: Coordination in Tree Adjoining Grammars," In DARPA Workshop on Spoken Language Systems.

Levin, B. (1993). *English Verb Classes and Alternations: A Preliminary Investigation*, University of Chicago Press, Chicago, IL.

Olsen, M. B. (1991). "Lexical Semantics, Machine Translation, and Talmy's Model of Motion Verbs," Linguistics Working Paper, Volume 3, Northwestern University, Evanston, IL.

Pustejovsky, J. (1991). "The Syntax of Event Structure," *Cognition*, 41.

Schabes, Y. (1990). *Mathematical and Computational Aspects of Lexicalized Grammars,* Ph.D. thesis, Computer Science Department, University of Pennsylvania, Philadelphia, PA.

Vijay-Shanker, K. (1987). *A Study of Tree Adjoining Grammars,* Ph.D. thesis, Department of Computer and Information Science, University of Pennsylvania.

Vijay-Shanker, K. (1992). *Using Descriptions of Trees in a Tree Adjoining Grammar, Computational Linguistics*, 18:4, pp. 481-517.

Vauquois B., and C. Boitet (1985). "Automated Translation at Grenoble University," *Computational Linguistics*, 11:1, pp. 28–36.

Voss, C. and B. Dorr, "Identifying the Lexical Component in a Two-Component View of the Interlingua," submitted to *Machine Translation*, 1994.

XTAG, "A Lexicalized Tree Adjoining Grammar for English," The XTAG Research Group, University of Pennsylvania Technical Report, 1995.

Zubizarreta, M. L. (1987). *Levels of Representation in the Lexicon and in the Syntax*, Foris Publications, Dordrecht, Holland/Cinnaminson, USA.