Lexicalized Grammar and the Description of Motion Events*

Matthew Stone*, Tonia Bleam[†], Christine Doran[‡] and Martha Palmer[†]

*Rutgers University
110 Frelinghuysen Road
Piscataway NJ 08854-8019
mdstone@cs.rutgers.edu

[†] University of Pennsylvania
3401 Walnut Suite 400A
Philadelphia PA 19104-6228
{tbleam,mpalmer}@linc.cis.upenn.edu

[‡] The MITRE Corporation
202 Burlington Road
Bedford MA 01730-1420
cdoran@mitre.org

Abstract

In natural language generation, the use of a lexicalized grammar formalism and incremental syntactic and semantic processing places strong and specific constraints on the form and meaning of grammatical entries. These principles restrict which grammatical representations are possible and suggest examples an analyst can consult to decide among possibilities. We discuss and justify a number of such constraints, and describe how they inform the design of lexical entries for motion verbs. Our entries allow a generator to match the lexical choices found in a target corpus of action descriptions by assessing how the interpretation of a verb in context contributes towards the hearer's identification of the intended action.

1. Introduction

This paper originates in a project of tailoring a natural language generation system called SPUD, for sentence planning using description (Stone & Doran, 1997), to generate instructions for action in a concrete domain. The desired behavior for the system is specified by a corpus of edited, naturally-occurring action instructions whose form and content the system must mirror. The input to the system consists of three components: a representation of the context in which instruction is to be issued; a set of communicative goals describing the content that the instruction should make available to the audience; and a database of facts describing the GENERALIZED INDIVIDUALS such as paths, places and eventualities involved in the action (Bach, 1989; Hobbs, 1985). The task is further complicated because the content and organization of this input database must suit a variety of other tasks, such as animation (Badler *et al.*, 1998).

Such a generation task demands a detailed model of how the available input determines appropriate linguistic elements to arrange in output. The problem of LEXICAL CHOICE illustrates this. English offers a wide range of verbs to describe events in which an agent moves some object along a path; any motion instruction obliges the generator to choose just one. Uses of verbs differ syntactically in the kinds of optional elements that accompany them; they differ semantically both in the constraints they place on the motion event itself and in the links they establish between the event and the speaker and hearer's mutual knowledge of the environment. As we shall see, often many verbs, in many syntactic frames, can truly and appropriately describe

* The bulk of this work was performed while the authors were located at and supported through IRCS, Penn (NSF-STC SBR 8920230). Thanks to Aravind Joshi, Alistair Knott and Bonnie Webber.

each event. Nevertheless, we find a constrained and consistent pattern of lexical choice across naturally-occurring instructions. In order to mirror lexical choice in SPUD, we must provide a computational account of lexical items through which SPUD can exhibit the same consistency. SPUD is based on the widely-espoused view that sentence generation is goal-directed activity (Appelt, 1985; Dale, 1992; Moore, 1994; Moore & Paris, 1993); SPUD's repertoire of communicative action is determined by a declarative lexicalized grammar. To plan a sentence, SPUD searches among the derivations admitted by the grammar for a true sentence whose interpretation achieves the system's communicative goals in the current context. Clearly, then, to mirror a specified corpus of instructions, the grammar provided to SPUD must characterize the words and constructions used in the corpus accurately and comprehensively. It must describe forms syntactically, so that they are combined appropriately, but it must also describe them semantically and pragmatically, in order to support a useful assessment of interpretation.

In this paper we articulate a methodology for constructing lexicalized grammatical resources for generation systems such as SPUD, and show how this methodology allows us to ensure that SPUD deploys its lexical and syntactic options as observed in a corpus of desired output. Our methodology involves guidelines for the construction of syntactic structures, semantic representations and the interface between them, but the basic principle behind all of these guidelines is this: THE REPRESENTATION OF A GRAMMATICAL ENTRY MUST MAKE IT AS EASY AS POSSIBLE FOR THE GENERATOR TO EXPLOIT ITS CONTRIBUTION IN CARRYING OUT FURTHER PLANNING. This principle responds to two concerns. First, our research has revealed many characteristic uses of language in which a single entry helps achieve multiple communicative goals (Stone & Webber, 1998). This is an important way in which a generator needs to be able exploit the contribution of an entry it has already used, in line with our principle. Second, SPUD is currently constrained to greedy or incremental search for reasons of efficiency. At each step, SPUD picks the entry whose interpretation goes furthest towards achieving its communicative goals. As the generator uses its grammar to build on these greedy choices, our principle facilitates the generator in arriving at a satisfactory overall utterance.

2. Syntax

We collected occurrences of the verbs *slide, rotate, push, pull, lift, connect, disconnect, remove, position* and *place* in the maintenance manual for the fuel system of the American F16 aircraft; in this manual, each operation is described consistently and precisely. Syntactic analysis of instructions in the corpus and the application of standard tests allowed us to cluster the uses of these verbs into five syntactic classes; these classes are consistent with each verb's membership in a distinct Levin class (Levin, 1993). Differences among these classes include whether the verb lexicalizes a path of motion (*rotate*), an endpoint (*position*), or a change of state (*disconnect*); and whether a spatial complement is optional (as with the verbs just given) or obligatory (*place*). The data in (1) illustrate these alternatives.

(1) a Rotate valve one-fourth turn clockwise. [Path]
b Rotate halon tube to provide access. [No path]
c Position one fire extinguisher near aircraft servicing connection point. [Endpoint]
d Position drain tube. [No endpoint]
e Disconnect generator set cable from ground power receptacle. [Change of state]
f Disconnect coupling. [No source argument]
g Place grommet on test set vacuum adapter. [Endpoint, required]

We crafted syntactic entries for these verbs as trees in Lexicalized Tree-Adjoining Grammar, LTAG (Joshi *et al.*, 1975; Schabes, 1990). Our entries respect three requirements that reflect the analysis of the corpus and the generator's need to build on the syntax of entries it selects.

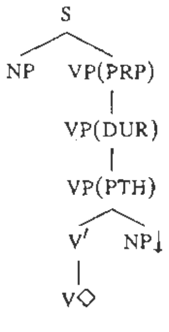
1. The grammar must associate each verb with its observed range of complements and modifiers, in the observed orders.

2. All optional elements, regardless of interpretation, must be represented in the syntax as modifiers, using the LTAG operation of adjunction. This allows the generator to select an optional element when it is needed to achieve communicative goals not otherwise satisfied. Recall that, in LTAG, a substitution site indicates a constituent that must be supplied syntactically to obtain a grammatical sentence; we call a constituent so provided a SYNTACTIC ARGUMENT. The alternative way of elaborating a sentence is to rewrite a node so as to include additional material (generally optional) specified by an auxiliary tree; we call material so provided a SYNTACTIC ADJUNCT. If optional elements are represented as syntactic adjuncts, it is straightforward to select one whenever its potential benefit is recognized. With other representations—for example, using alternative syntactic entries some of which include a syntactic argument position (substitution site) for the "optional" constituent—the representation can result in artificial dependencies or even dead-end paths in the search space in generation. To use this representation successfully, the generator would have to anticipate how the sentence would be fleshed out later in order to select the right entry early on.

3. The appropriate order of complements and modifiers for a verb must be represented using hierarchies of nodes in the verb's elementary tree. In a fixed word-order language like English, the nodes we add reflect different semantic classes which tend to be realized in a particular order; in a free word-order language, we might instead introduce ordering nodes based on information-structure status. Introducing such nodes decouples the generator's search space of derivations from the overt output word-order. It allows the generator to select complements and modifiers in any search order, while still realizing the complements and modifiers with their correct surface order. Again, alternative designs—representing word-order in the derivation itself or in features that clash when elements appear in the wrong order—introduce dependencies into the search space for generation that make it more difficult for the generator to build on its earlier choices successfully.

The latter requirements induce certain differences between our trees and other LTAG grammars for English, such as the XTAG grammar (Doran *et al.*, 1994), even in cases when the XTAG trees do describe our corpus. For example, we associate *slide* with the tree in (2); the structure reflects the optionality of the *path* constituent and makes explicit the observed characteristic order of constituents specifying path (PTH), duration (DUR) and purpose (PRP).

(2)



3. Syntax/semantics interface

SPUD adopts an ontologically promiscuous semantics (Hobbs, 1985): each entry used in the derivation of an utterance contributes a constraint to its overall semantics. The role of the syntax/semantics interface is to determine when the constraints contributed by different grammatical entries describe the same generalized individuals. For example, take the phrase *slide the sleeve quickly*. The corresponding constraints describe an event e in which agent x *slides* object y along path p; describe an individual z that is a *sleeve*; and describe an event e' that is *quick*. The syntax/semantics interface provides the guarantee that $e = e'$ and $y = z$—i.e., that the sliding is what is quick and that the sleeve is what is slid. See (Hobbs, 1985; Hobbs *et al.*, 1993) for more details on ontologically promiscuous semantics.

Note that this strategy contrasts with other approaches to LTAG semantics, such as (Candito & Kahane, 1998), which describe meanings primarily in terms of function-argument relations. (It is also possible to combine both function-argument and constraint semantics, as in (Joshi & Vijay-Shanker, 1999; Kallmeyer & Joshi, 1999).) Like Hobbs, we use semantic representations as a springboard to explore the relationships between sentence meaning, background knowledge and inference—relationships which are easiest to state in terms of constraints. In addition, the use of constraints harmonizes with our perspective that a basic generation task is to construct extended descriptions of individuals (Stone & Webber, 1998; Webber *et al.*, 1999).

In general, to express the semantic links between multiple entries in a derivation, we associate each node in a syntactic tree with the individuals that the node describes. We refer to the collection of individuals that label the nodes in an entry as the SEMANTIC ARGUMENTS of the entry. When one tree combines with another by substitution or adjunction, a node in one tree is identified with a node in the other tree; at the same time the corresponding entities must be unified. Thus for example by labeling the foot VP node for *quickly* with e' and the corresponding VP node for *slide* with e, we can derive the identity $e = e'$ for *slide quickly*.

Our notion of semantic arguments is clearly distinguished from the notion of syntactic argument that we used in section 2 to characterize the syntactic structure of entries. Each syntactic argument position corresponds to one semantic argument (or more), since the syntactic argument position is a node in the tree which is associated with some individuals: semantic arguments. However, semantic arguments need not be associated with syntactic argument positions. For example, in a verb entry, we do not have a substitution site that realizes the eventuality that the verb describes. But we treat this eventuality as a semantic argument to implement a Davidsonian account of event modifiers, cf. (Davidson, 1980). Meanwhile, optional constituents that specify paths or places may be best modeled syntactically as modifiers, using the syntactic operation of adjunction. Optional constituents nevertheless can be taken to specify semantic arguments by associating their adjunction sites with references to the entities they specify (e.g., the paths or places). Because we count these implicit and unexpressed entities as semantic arguments, our notion is broader than that of (Candito & Kahane, 1998) and is more similar to Palmer's essential arguments (Palmer, 1990). It is a substantive question for grammar design WHICH entities SHOULD be acknowledged as semantic arguments for a given entry.

We make use of three tests to determine whether a particular syntactic modifier of a verb phrase describes the overall eventuality argument of the verb—this makes it an adjunct for the purposes of semantics as well—or whether it specifies some other semantic argument of the verb. The tests are: a DO SO test and an EXTRACTION test (explained here), and a PRESUPPOSITION test (explained in the following section). Together, these tests provide strong and specific guidance for designing the syntax/semantics interface in a generation grammar. (Of course, these tests are not perfect and may on occasion reveal difficult or ambiguous cases.)

1. The DO SO test succeeds when a modifier of a verb can be varied across ellipsis with *do so* naturally. The infinitivals in (3a), which provide different reasons for Kim and Sandy, pass the test; the locative PPs in (3b) fail the test, as they cannot be taken to describe Kim and Chris's separate destinations:

(3) a Kim left early to avoid the crowd. Sandy did so to find one.
b #Kim ran quickly to the counter. Chris did so to the kiosk.

A successful DO SO test suggests the modifier describes the event or action directly. A failed one suggests the modifier contributes a description of an entity that is independently related to the event or action—in other words, that the modifier specifies a semantic argument (e.g., the destination in (3b)). A theoretical explanation for the test can be given in terms of a semantic view of ellipsis such as (Hardt, 1999), where *do so* recovers an action discourse referent that has been introduced by an earlier predicate on events. When a modifier makes a predication on an event, there are two actions available for *do so*: the modified action and (as (3a) illustrates) the unmodified one. When a modifier instead makes a predication on a participant in the event, the only action referent for *do so* is that contributed by the main verb. In such cases, the DO SO test fails because we do not have a suitable action referent or a way of determining what role the new participant plays.

2. The EXTRACTION test applies to classes of syntactic modifiers of VP headed by a closed-class item. The test succeeds if it is grammatical to extract from inside the syntactic modifier (in a *wh*-question, for example), as in (4a); it fails otherwise, as in (4b).

(4) a What did you remove the rabbit from? (A: the hat)
b #What did you remove the rabbit at? (A: the magic show)

Passing the extraction test suggests that an optional constituent specifies a semantic argument. In LTAG, extraction describes a relation among trees in a tree family that have essentially the same meaning and differ only in syntax. On one formalization (Xia *et al.*, 1998), these relationships between trees are realized as descriptions of structure to add to elementary trees, or transformations. An "extraction transformation" that introduces the entity *l* in the syntax/semantics interface and relates *l* to the available entity *e* in the semantics cannot be represented this way. However, if some semantic argument *l* is referenced in the original tree, the extraction analogue to this tree can easily realize *l* differently. If we describe the source location as the semantic argument *l* in (4a) for example, the new realization involves an initial *wh*-NP substitution site describing the source *l*, and the corresponding stranded structure of the PP *from t*. (Note that failure of the extraction test would be inconclusive in cases where syntax independently ruled extraction out.)

4. Semantics

Semantic analysis of the instructions in the F16 corpus revealed that differences among verbs often involve links that the verbs impose between the action and what is known in the context about the environment in which the action is to be performed. The following illustration is representative. In the aircraft vent system, pipes are sealed together using a sleeve, which fits snugly over the ends of adjacent pipes; and a coupling, which snaps shut around the sleeve and holds it in place. At the start of maintenance, one *removes* the coupling and *slides* the sleeve away from the junction between the pipes. Afterwards, one *(re-)positions* the sleeve at the junction and *(re-)installs* the coupling around it. In the F16 corpus, these actions are always described using these verbs.

This use of verbs reflects the general design and function of the equipment as well as the motions themselves. For example, the motion involved in sliding the sleeve away is just the reverse of the motion involved in positioning the sleeve back. Since the verb *slide* indicates smooth motion ALONG A SURFACE (but not direction), *slide* seems to describe both actions equally well. The verb *position*, meanwhile, is used to describe a motion that leaves its object in some definite location, where the object can perform some intended function. In the case of the sleeve, it would only be IN POSITION when straddling the pipes whose junction it seals.

We capture such distinctions in SPUD using a two-part lexical semantics.

1. The ASSERTION contributes new relationships among generalized individuals to the discourse. For example, the assertion of a motion verb might specify what manner of motion or what trajectory of motion is involved in an event.
2. The ANAPHORIC PRESUPPOSITION is interpreted by a process of resolution linking it to salient facts and individuals from background knowledge and the conversational record. (Space precludes a description of the resolution process, but see (van der Sandt, 1992) for a theoretical account and (Stone, 2000) for the implementation.) Motion verbs generally carry such presuppositions: for instance, they presuppose a current location for the object (they assert this to be the beginning of the path traveled). But such presuppositions also distinguish lexical items. For example, *slide* presupposes a surface the object starts out in contact with; the object is asserted to remain in contact with this surface during the sliding. Meanwhile, *position* presupposes some "position" where the object carries out its intended function; the object is asserted to wind up at this position. Presuppositions can also evoke salient referents from the discourse history; for instance *reposition* presupposes a suitable prior motion event.

This formalism for presupposition is the basis for our third test for semantic arguments, the PRESUPPOSITION TEST. Any individual that is referenced in the presupposition of the verb must be treated as a semantic argument, even if a syntactic constituent that specifies that individual is optional. As suggested by (Saeboe, 1996), to apply the presupposition test in designing a lexical entry, we can compare the interpretation of a sentence with a modifier, such as *from the power adaptor* in (5a), to a corresponding sentence without the modifier, as in (5b):

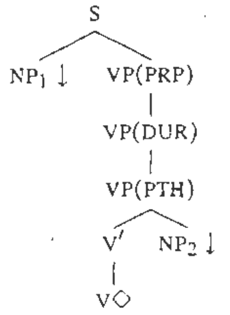
(5) a (Find the power cable.) Disconnect it from the power adaptor.
b (The power cable is attached to the power adaptor.) Disconnect it.

If the entity specified by the modifier can be identified implicitly as discourse-bound—so that the sentence without the modifier can have the same interpretation as the sentence with the modifier, as in (5)—then the modifier must express a presupposed semantic argument. (Again, this is a partial diagnostic since semantic arguments need not always be presupposed.)

Let us pause to motivate our methodology of specifying lexical presuppositions as well as lexical assertions—and our tests for designing the syntax/semantics interface—in terms of our overriding goal: to allow the generator to build on its choices as easily as possible. The requirement to assert only what is true and to presuppose what is shared restricts which verbs are applicable in any context. At the same time, however, assertion and presupposition provide constraints on interpretation that can reduce ambiguity or trigger further inferences. They can thereby help the hearer identify the speaker-intended action. For example, the verb's presupposition may combine with other constraints contributed by the verb's complements to identify the participants in a described action (Stone & Webber, 1998). Of course, the generator can build on the presupposition of the verb this way only if it represents the interpretation of the presupposition and keeps track of the semantic arguments of the verb in order to model further elements as providing additional constraints on these arguments.

(6) fleshes out our earlier sample entry, for *slide*. The tree gives the syntax for one element in the tree family associated with *slide*; the feature structures associated with nodes show the syntax/semantics interface for this tree; the associated formulas describe the semantics of the entry in terms of presuppositions and assertions about the individuals referenced in the tree.

(6) a Syntax and syntax/semantics interface:



$$\text{S, VP(PRP), VP(DUR)} \rightarrow \begin{bmatrix} \text{event} : \textsc{e} \\ \text{tense} : \langle \textsc{s},\textsc{r} \rangle \end{bmatrix}$$

$$\text{VP(PTH)}, \text{V}' \rightarrow \begin{bmatrix} \text{event} : \textsc{e} \\ \text{path} : \textsc{path} \\ \text{changed} : \textsc{obj} \end{bmatrix}$$

$$\text{NP}_1 \rightarrow \begin{bmatrix} \text{ind} : \textsc{agent} \end{bmatrix}$$

$$\text{NP}_2 \rightarrow \begin{bmatrix} \text{ind} : \textsc{obj} \end{bmatrix}$$

b Presupposition: *located-at-start*(R,OBJ,PATH), *along-surface*(PATH)

c Assertion: *caused-motion*(R,E, AGENT,OBJ,PATH)

5. Conclusion: grammar and lexical choice

The manual's consistent alternation between *slide* and *position* casts into relief the problem of lexical choice with which this paper opened. We close by suggesting how the methodology we have outlined here—formulating a grammar that matches a corpus and allows the generator to build on and exploit the entries it selects—leads to the construction of generation resources that can account for such alternation.

First, observe that the syntax and the syntax/semantics interface put *slide* and *position* on an equal footing. We can settle on a syntactic tree for each verb that best fits the context as in (Stone & Doran, 1997); we have designed these trees so that either choice can be fleshed out by further constituents into a satisfactory utterance.

To choose one verb in construction over the other, we must look at the INTERPRETATION of the two entries. A key part of this interpretation is the way the hearer resolves the presupposition. For example, the hearer resolves *position the sleeve* by finding in the common ground SOME sleeve and SOME position where it belongs. Part of SPUD's task is to ensure that the hearer will arrive at the SAME resolution that the generator intends; for *position the sleeve*, that of course means identifying the INTENDED sleeve and the INTENDED position for it. Depending on the context, it may be necessary to elaborate the description of an action, by adding additional words and additional presuppositions with them, to make the hearer's resolution of the presupposition unique. (Such an elaboration might yield *position the wing-vent sleeve*.)

This characterization of the speaker's communicative goals and the hearer's interpretation directly informs our lexical choice. Different presuppositions determine different possible resolutions, depending on the properties of salient objects in the common ground. The fewer resolutions that there are after selecting a verb, the more the verb assists the hearer in identifying the needed action. This gives a reason to prefer one verb over another. In our example, general background indicates that each sleeve only has a single place where it belongs, at the joint; meanwhile, there may be many "way points" along the pipe to slide the sleeve to. This makes the anaphoric interpretation of *position* less ambiguous than that of *slide*; to obtain an equally constrained interpretation with *slide*, an additional identifying modifier like *into its position* would be needed. This favors *position* over *slide*.

References

APPELT D. (1985). *Planning English Sentences*. Cambridge.

BACH E. (1989). *Informal Lectures on Formal Semantics*. SUNY.

BADLER N., BINDIGANAVALE R., BOURNE J., PALMER M., SHI J. & SCHULER W. (1998). A parameterized action representation for virtual human agents. In *Embodied Conversational Characters*.

CANDITO M. & KAHANE S. (1998). Can the TAG derivation tree represent a semantic graph? an answer in the light of Meaning-Text Theory. In *TAG+4*.

DALE R. (1992). *Generating Referring Expressions*. MIT.

DAVIDSON D. (1980). The logical form of action sentences. In *Essays on actions and events*, p. 105–148. Clarendon.

DORAN C., EGEDI D., HOCKEY B. A., SRINIVAS B. & ZAIDEL M. (1994). XTAG System—a wide coverage grammar for English. In *COLING*.

HARDT D. (1999). Dynamic Interpretation of Verb Phrase Ellipsis. *Linguistics and Philosophy*, **22**, 187–221.

HOBBS J., STICKEL M., APPELT D. & MARTIN P. (1993). Interpretation as abduction. *Artificial Intelligence*, **63**, 69–142.

HOBBS J. R. (1985). Ontological promiscuity. In *ACL*, p. 61–69.

JOSHI A. & VIJAY-SHANKER K. (1999). Compositional semantics with lexicalized tree-adjoining grammar (LTAG). In *International Workshop on Computational Semantics*, p. 131–145.

JOSHI A. K., LEVY L. & TAKAHASHI M. (1975). Tree adjunct grammars. *J. of the Computer and System Sciences*, **10**, 136–163.

KALLMEYER L. & JOSHI A. (1999). Factoring predicate argument and scope semantics: underspecified semantics with LTAG. In *12th Amsterdam Colloquium*.

LEVIN B. (1993). *English Verb Classes and Alternations*. University of Chicago.

MOORE J. (1994). *Participating in Explanatory Dialogues*. MIT.

MOORE J. D. & PARIS C. L. (1993). Planning text for advisory dialogues: capturing intentional and rhetorical information. *Computational Linguistics*, **19**, 651–695.

PALMER M. (1990). *Semantic Processing for Finite Domains*. Cambridge.

SAEBOE K. J. (1996). Anaphoric presuppositions and zero anaphora. *Linguistics and Philosophy*, **19**, 187–209.

SCHABES Y. (1990). *Mathematical and Computational Aspects of Lexicalized Grammars*. PhD thesis, University of Pennsylvania.

STONE M. (2000). Towards a computational account of knowledge, action and inference in instructions. *J. of Language and Computation*.

STONE M. & DORAN C. (1997). Sentence planning as description using tree-adjoining grammar. In *ACL*, p. 198–205.

STONE M. & WEBBER B. (1998). Textual economy through close coupling of syntax and semantics. In *INLG*, p. 178–187.

VAN DER SANDT R. (1992). Presupposition projection as anaphora resolution. *J. of Semantics*, **9**, 333–377.

WEBBER B., KNOTT A., STONE M. & JOSHI A. (1999). Discourse relations: A structural and presuppositional account using lexicalised TAG. In *ACL*, p. 41–48.

XIA F., PALMER M., VIJAY-SHANKER K. & ROSENZWEIG J. (1998). Consistent grammar development using partial tree-descriptions for lexicalized tree-adjoining grammars. In *TAG+4*.