Integrating compositional semantics into a verb lexicon

Hoa Trang Dang, Karin Kipper and Martha Palmer
Department of Computer and Information Sciences
University of Pennsylvania
200 South 33rd Street
Philadelphia, PA 19104, USA
{htd,kipper,mpalmer}@linc.cis.upenn.edu

Abstract

We present a class-based approach to building a verb lexicon that makes explicit the close association between syntax and semantics for Levin classes. We have used Lexicalized Tree Adjoining Grammars to capture the syntax associated with each verb class and have augmented the trees to include selectional restrictions. In addition, semantic predicates are associated with each tree, which allow for a compositional interpretation.

1 Introduction

The difficulty of achieving adequate hand-crafted semantic representations has limited the field of natural language processing to applications that can be contained within well-defined sub-domains. Despite many different lexicon development approaches (Mel'cuk, 1988; Copestake and Sanfilippo, 1993; Lowe et al., 1997), the field has yet to develop a clear consensus on guidelines for a computational lexicon. One of the most controversial areas in building such a lexicon is polysemy: how senses can be computationally distinguished and characterized. We address this problem by employing compositional semantics and the adjunction of syntactic phrases to support regular verb sense extensions. This differs from the Lexical Conceptual Structure (LCS) approach exemplified by Voss (1996), which requires a separate LCS representation for each possible sense extension. In this paper we describe the construction of VerbNet, a verb lexicon with explicitly stated syntactic and semantic information for individual lexical items, using Levin verb classes (Levin, 1993) to systematically construct lexical entries. We use Lexicalized Tree Adjoining Grammar (LTAG) (Joshi, 1987; Schabes, 1990) to capture the syntax for each verb class, and associate semantic predicates with each tree.

Although similar ideas have been explored for verb sense extension (Pustejovsky, 1995; Goldberg, 1995), our approach of applying LTAG to the problem of composing and extending verb senses is novel. LTAGs have an extended domain of locality that captures the arguments of a verb in a local manner. The association of semantic predicates to a tree yields a complete semantics for the verb. Moreover, the operation of adjunction in LTAGs provides a mechanism for extending verb senses.

2 Levin classes

Levin verb classes are based on the ability of a verb to occur in diathesis alternations, which are pairs of syntactic frames that are in some sense meaning preserving. The fundamental assumption is that the syntactic frames are a direct reflection of the underlying semantics. However, Levin classes exhibit inconsistencies that have hampered researchers' ability to reference them directly in applications. Many verbs are listed in multiple classes, some of which have conflicting sets of syntactic frames. Dang et al. (1998) showed that multiple listings could in some cases be interpreted as regular sense extensions, and defined intersective Levin classes, which are a more fine-grained, syntactically and semantically coherent refinement of basic Levin classes. We represent these verb classes and their regular sense extensions in the LTAG formalism.

3 Lexicalized Tree Adjoining Grammars

3.1 Overview of formalism

Lexicalized Tree Adjoining Grammars consist of a finite set of initial and auxiliary elementary trees, and two operations to combine them. The minimal, non-recursive linguistic structures of a language, such as a verb and its complements, are captured by initial trees. Recursive structures of a language, such as prepositional modifiers which result in syntactically embedded VPs, are represented by auxiliary trees.

Elementary trees are combined by the operations of substitution and adjunction. Substitution is a simple operation that replaces a leaf of a tree with a new tree. Adjunction is a splicing operation that replaces an internal node of an elementary tree with an auxiliary tree. Every tree is associated with a lexical item of the language, called the *anchor* of the tree. The tree represents the domain over which the lexical item can directly specify syntactic constraints, such as subject-verb number agreement, or semantic constraints, such as selectional restrictions, all of which are implemented as features.

LTAGs are more powerful than context free grammars (CFG), allowing localization of so-called unbounded dependencies that cannot be handled by CFGs. There are critical benefits to lexical semantics that are provided by the extended domain of locality of the lexicalized trees. Each lexical entry corresponds to a tree. If the lexical item is a verb, the corresponding tree is a skeleton for an entire sentence with the verb already present, anchoring the tree as a terminal symbol. The other parts of the sentence will be substituted or adjoined in at appropriate places in the skeleton tree in the course of the derivation. The composition of trees during parsing is recorded in a *derivation tree*. The derivation tree nodes correspond to lexically anchored elementary trees, and the arcs are labeled with information about how these trees were combined to produce the parse. Since each lexically anchored initial tree corresponds to a semantic unit, the derivation tree closely resembles a semantic-dependency representation.

3.2 Semantics for TAGs

There is a range of previous work in incorporating semantics into TAG trees. Stone and Doran (1997) describe a system used for generation that simultaneously constructs the semantics and syntax of a sentence using LTAGs. Joshi and Vijay-Shanker (1999), and Kallmeyer and Joshi (1999), describe the semantics of a derivation tree as a set of attachments of trees. The semantics of these attachments is given as a conjunction of formulae in a flat semantic representation. They provide a specific methodology for composing semantic representations much like Candito and Kahane (1998), where the directionality of dominance in the derivation tree should be interpreted according to the operations used to build it. Kallmeyer and Joshi also use a flat semantic representation to handle scope phenomena involving quantifiers.

4 Description of the verb lexicon

VerbNet can be viewed in both a static and a dynamic way. The static aspect refers to the verb and class entries and how they are organized, providing the characteristic descriptions of a verb sense or a verb class (Kipper et al., 2000). The dynamic aspect of the lexicon constrains the entries to allow a compositional interpretation in LTAG derivation trees, representing extended verb meanings by incorporating adjuncts.

Verb classes allow us to capture generalizations about verb behavior. Each verb class lists the thematic roles that the predicate-argument structure of its members allows, and provides descriptions of the syntactic frames corresponding to licensed constructions, with selectional restrictions defined for each argument in each frame.[1] Each frame also includes semantic predicates describing the participants at various stages of the event described by the frame.

Verb classes are hierarchically organized, ensuring that each class is coherent – that is, all its members have common semantic elements and share a common set of thematic roles and basic syntactic frames. This requires some manual restructuring of the original Levin classes, which is facilitated by using intersective Levin classes.

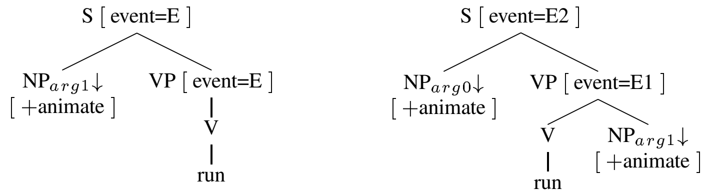
5 Compositional Semantics

We use TAG elementary trees for the description of allowable frames and associate semantic predicates with each tree, as was done by Stone and Doran. The semantic predicates are primitive enough so that many may be reused in different trees. By using TAGs we get the additional benefit of an existing parser that yields derivations and derived trees from which we can construct the compositional semantics of a given sentence.

We decompose each event E into a tripartite structure in a manner similar to Moens and Steedman (1988), introducing a time function for each predicate to specify whether the predicate is true in the preparatory ($during(E)$), culmination ($end(E)$), or consequent ($result(E)$) stage of an event.

Initial trees capture the semantics of the basic senses of verbs in each class. For example, many

[1] These restrictions are more like preferences that generate a preferred reading of a sentence. They may be relaxed depending on the domain of a particular application.



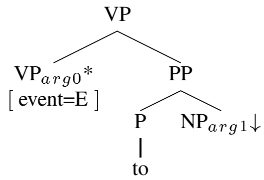
$motion(during(E), X_{arg1})$ $\quad$ $cause(during(E2), X_{arg0}, E1) \wedge motion(during(E1), X_{arg1})$

Figure 1: Induced action alternation for the *Run* verbs

verbs in the *Run* class can occur in the induced action alternation, in which the subject of the intransitive sentence has the same thematic role as the direct object in the transitive sentence. Figure 1 shows the initial trees for the transitive and intransitive variants for the *Run* class, along with their semantic predicates. The entity in motion is given by $arg1$, associated with the syntactic subject of the intransitive tree and the direct object of the transitive tree. The event denoted by the transitive variant is a composition of two subevents: $E1$ refers to the event of $arg1$ running, and $E2$ refers to the event of an entity ($arg0$) causing event $E1$.

Predicates are associated with not only the verb trees, but also the auxiliary trees. We use a flat semantic representation like that of Kallmeyer and Joshi, and the semantics of a sentence is the conjunction of the semantic predicates of the trees used to derive the sentence. Figure 2 shows an auxiliary tree for a path prepositional phrase headed by "to", along with its associated semantic predicate.

When the PP tree for "to the park" is adjoined into the intransitive tree for "John ran", the semantic interpretation is the conjunction of the two predicates *motion(during(E),john)* $\wedge$ *goal(end(E),john,park)*; adjunction into the transitive tree for "Bill ran the horse" yields *cause(during(E2),bill,E1)* $\wedge$ *motion(during(E1),horse)* $\wedge$ *goal(end(E1),horse,park)*. In both cases, the argument $X_{arg0.arg1}$ (john or horse) for the auxiliary tree is nonlocal and comes from the adjunction site.[2] The arguments are recovered from the derivation tree, following Candito and Kahane. When an initial tree is substituted into another tree, the dependency mirrors the derivation structure, so the variables associated with the substituting tree can be referenced as arguments in the host tree's predicates. When an auxiliary tree is adjoined, the dependency for the adjunction is reversed, so that variables associated with the host tree can be referenced as arguments in the adjoining tree's predicates.



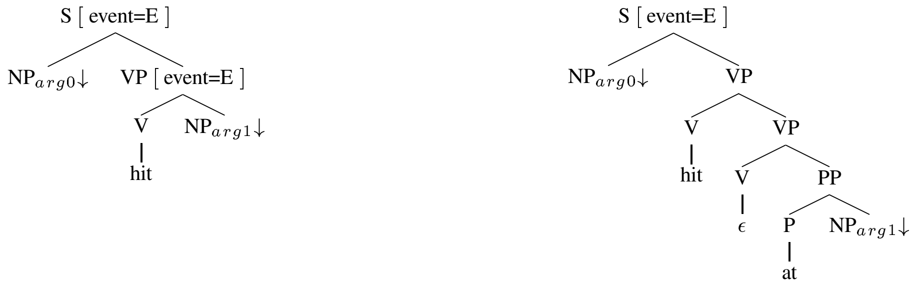
$goal(end(E), X_{arg0.arg1}, X_{arg1})$

Figure 2: Auxiliary path PP tree

The tripartite event structure allows us to express the semantics of classes of verbs like change of state verbs whose description requires reference to a complex event structure. In the case of a verb such as "break", it is important to make a distinction between the state of the object before the end of the action and the new state that results afterwards. This event structure also handles the conative construction, in which there is an intention of a goal during the event, that is not achieved at the end of the event. The example of the conative construction shown in Figure 3 expresses the intention of hitting something. Because the intention is not satisfied the semantics do not include the predicates *manner(end(E),forceful,X_{arg0})* $\wedge$ *contact(end(E),X_{arg0},X_{arg1})*, that express the completion of the contact with impact event.

The ability of verbs to take on extended senses in sentences based on their adjuncts is captured in a

[2] $X_{arg0.arg1}$ is the variable associated with the entity in motion ($arg1$) in the tree to which the PP adjoins ($arg0$).

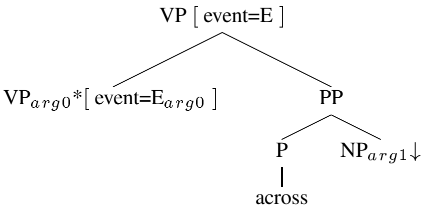


$$manner(during(E), directedmotion, X_{arg0}) \wedge$$
$$contact(end(E), X_{arg0}, X_{arg1}) \wedge$$
$$manner(end(E), forceful, X_{arg0})$$

$$manner(during(E), directedmotion, X_{arg0})$$

Figure 3: Syntax and semantics of transitive and conative construction for *Hit* verbs

natural way by the TAG operation of adjunction and our conjunction of semantic predicates. The original *Hit* verb class does not include movement of the direct object as part of the meaning of hit; only sudden contact has to be established. By adjoining a path PP such as "across NP", we get an extended meaning, and a change in Levin class membership to the *Throw* class. Figure 4 shows the class-specific auxiliary tree anchored by the preposition "across" together with its semantic predicates, introducing a motion event that immediately follows (*meets*) the contact event.



$$meets(E_{arg0}, E) \wedge$$
$$motion(during(E), X_{arg0.arg1}) \wedge$$
$$via(during(E), X_{arg0.arg1}, X_{arg1})$$

Figure 4: Auxiliary tree for "across"

6 Conclusion

We have presented a class-based approach to building a verb lexicon that makes explicit and implements the close association between syntax and semantics, as postulated by Levin. The power of the lexicon comes from its dynamic aspect that is based on the LTAG formalism, for which we already have a large English grammar. Palmer et al. (1998) defined compositional semantics for classes of verbs implemented in LTAG, representing general semantic components (e.g., motion, manner) as features on the nodes of the trees. Our use of separate logical forms gives a more detailed semantics for the sentence, so that for an event involving motion, it is possible to know not only that the event has a *motion* semantic component, but also which entity is actually in motion. This level of detail is necessary for applications such as animation of natural language instructions (Bindiganavale et al., 2000). Another important contribution of this work is that by dividing each event into a tripartite structure, we permit a more precise definition of the associated semantics. Finally, the operation of adjunction in TAGs provides a principled approach to representing the type of regular polysemy that has been a major obstacle in building verb lexicons.

Researching whether a TAG grammar for VerbNet can be automatically constructed by using development tools such as Xia et al. (1999) or Candito (1996) is part of our next step. We also expect to be able to factor out some class-specific auxiliary trees to be used across several verb classes.

7 Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments. This research was partially supported by NSF grants IIS-9800658 and IIS-9900297 and CAPES grant 0914-95.

References

Rama Bindiganavale, William Schuler, Jan M. Allbeck, Norman I. Badler, Aravind K. Joshi, and Martha Palmer. 2000. Dynamically Altering Agent Behaviors Using Natural Language Instructions. *Fourth International Conference on Autonomous Agents*, June.

Marie-Hélène Candito and Sylvain Kahane. 1998. Can the TAG derivation tree represent a semantic graph? An answer in the light of Meaning-Text Theory. In *Proceedings of the Fourth TAG+ Workshop*, pages 21–24, Philadelphia, PA, August.

Marie-Hélène Candito. 1996. A Principle-Based Hierarchical Representation of LTAGs. In *Proceedings of COLING-96*, Copenhagen, Denmark.

Ann Copestake and Antonio Sanfilippo. 1993. Multilingual lexical representation. In *Proceedings of the AAAI Spring Symposium: Building Lexicons for Machine Translation*, Stanford, California.

Hoa Trang Dang, Karin Kipper, Martha Palmer, and Joseph Rosenzweig. 1998. Investigating regular sense extensions based on intersective Levin classes . In *Proceedings of COLING-ACL98*, Montreal, Canada, August.

Adele E. Goldberg. 1995. *Constructions. A Construction Grammar Approach to Argument Structure*. University of Chicago Press, Chicago, Ill.

Aravind K. Joshi and K. Vijay-Shanker. 1999. Compositional semantics with Lexicalized Tree-adjoining Grammar: How much underspecification is necessary? . In *Proceedings of the Third International Workshop on Computational Semantics (IWCS-3)*, pages 131–145, Tilburg, The Netherlands, January.

Aravind K. Joshi. 1987. An introduction to tree adjoining grammars. In A. Manaster-Ramer, editor, *Mathematics of Language*. John Benjamins, Amsterdam.

Laura Kallmeyer and Aravind Joshi. 1999. Underspecified semantics with LTAG . In *Proceedings of Amsterdam Colloquium on Semantics*.

Karin Kipper, Hoa Trang Dang, and Martha Palmer. 2000. Class-based construction of a verb lexicon. In *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI-2000)*, Austin, TX, July-August.

Beth Levin. 1993. *English Verb Classes and Alternation, A Preliminary Investigation*. The University of Chicago Press.

J.B. Lowe, C.F. Baker, and C.J. Fillmore. 1997. A frame-semantic approach to semantic annotation. In *Proceedings 1997 Siglex Workshop/ANLP97*, Washington, D.C.

I. A. Mel'cuk. 1988. Semantic description of lexical units in an explanatory combinatorial dictionary: Basic principles and heuristic criteria. *International Journal of Lexicography*, I:3:165–188.

M. Moens and M. Steedman. 1988. Temporal ontology and temporal reference. *Computational Linguistics*, 14:15–38.

Martha Palmer, Joseph Rosenzweig, and William Schuler. 1998. Capturing Motion Verb Generalizations in Synchronous TAG. In Patrick Saint-Dizier, editor, *Predicative Forms in Natural Language and in Lexical Knowledge Bases*. Kluwer Press.

James Pustejovsky. 1995. *The Generative Lexicon*. MIT Press, Cambridge, Massachusetts, USA.

Yves Schabes. 1990. *Mathematical and Computational Aspects of Lexicalized Grammars*. Ph.D. thesis, Computer Science Department, University of Pennsylvania.

Matthew Stone and Christine Doran. 1997. Sentence Planning as Description using Tree Adjoining Grammar. In *Proceedings of ACL-EACL '97*, Madrid, Spain.

Clare Voss. 1996. *Interlingua-based Machine Translation of Spatial Expressions*. Ph.D. thesis, University of Maryland, Department of Computer Science.

Fei Xia, Martha Palmer, and K. Vijay-Shanker. 1999. Toward semi-automating grammar development. In *Proceedings of the 5th Natural Language Processing Pacific Rim Symposium(NLPRS-99)*, Beijing, China.