

A Computational Framework for Non-Lexicalist Semantics

Jimmy Lin

MIT Computer Science and Artificial Intelligence Laboratory
Cambridge, MA 02139
jimmylin@csail.mit.edu

Abstract

Under a lexicalist approach to semantics, a verb completely encodes its syntactic and semantic structures, along with the relevant syntax-to-semantics mapping; polysemy is typically attributed to the existence of different lexical entries. A lexicon organized in this fashion contains much redundant information and is unable to capture cross-categorial morphological derivations. The solution is to spread the “semantic load” of lexical entries to other morphemes not typically taken to bear semantic content. This approach follows current trends in linguistic theory, and more perspicuously accounts for alternations in argument structure. I demonstrate how such a framework can be computationally realized with a feature-based, agenda-driven chart parser for the Minimalist Program.

1 Introduction

The understanding of natural language text includes not only analysis of syntactic structure, but also of semantic content. Due to advances in statistical syntactic parsing techniques (Collins, 1997; Charniak, 2001), attention has recently shifted towards the harder question of analyzing the meaning of natural language sentences.

A common lexical semantic representation in the computational linguistics literature is a frame-based model where syntactic arguments are associated with various semantic roles (essentially frame slots). Verbs are viewed as simple predicates over their arguments. This approach has its roots in Fillmore’s Case Grammar (1968), and serves as the foundation for two current large-scale semantic annotation projects: FrameNet (Baker et al., 1998) and PropBank (Kingsbury et al., 2002).

Underlying the semantic roles approach is a lexicalist assumption, that is, each verb’s lexical entry completely encodes (more formally, projects) its syntactic and semantic structures. Alternations in argument structure are usually attributed to multiple lexical entries (i.e., verb senses). Under the lexicalist approach, the semantics of the verb *break* might look something like this:

- (1) **break(agent, theme)**
agent: subject theme: object
- break(agent, theme, instrument)**
agent: subject theme: object
instrument: oblique(*with*)
- break(theme)**
theme: subject
- ...

The lexicon explicitly specifies the different subcategorization frames of a verb, e.g., the causative frame, the causative instrumental frame, the inchoative frame, etc. The major drawback of this approach, however, is the tremendous amount of redundancy in the lexicon—for example, the class of prototypical transitive verbs where the agent appears as the subject and the theme as the direct object must all duplicate this pattern.

The typical solution to the redundancy problem is to group verbs according to their argument realization patterns (Levin, 1993), possibly arranged in an inheritance hierarchy. The argument structure and syntax-to-semantics mapping would then only need to be specified once for each verb class. In addition, lexical rules could be formulated to derive certain alternations from more basic forms.

Nevertheless, the lexicalist approach does not capture productive morphological processes that pervade natural language, for example, *flat.V* → *flatten.ADJ* or *hammer.N* → *hammer.V*; most frameworks for computational semantics fail to capture the deeper derivational relation-

ship between morphologically-related terms. For languages with rich derivational morphology, this problem is often critical: the standard architectural view of morphological analysis as a preprocessor presents difficulties in handling semantically meaningful affixes.

In this paper, I present a computational implementation of Distributed Morphology (Halle and Marantz, 1993), a non-lexicalist linguistic theory that erases the distinction between syntactic derivation and morphological derivation. This framework leads to finer-grained semantics capable of better capturing linguistic generalizations.

2 Event Structure

It has previously been argued that representations based on a fixed collection of semantic roles cannot adequately capture natural language semantics. The actual inventory of semantic roles, along with precise definitions and diagnostics, remains an unsolved problem; see (Levin and Rappaport Hovav, 1996). Fixed roles are too coarse-grained to account for certain semantic distinctions—the only recourse, to expand the inventory of roles, comes with the price of increased complexity, e.g., in the syntax-to-semantics mapping.

There is a general consensus among theoretical linguists that the proper representation of verbal argument structure is event structure—representations grounded in a theory of events that decompose semantic roles in terms of primitive predicates representing concepts such as causality and inchoativity (Dowty, 1979; Jackendoff, 1983; Pustejovsky, 1991b; Rappaport Hovav and Levin, 1998). Consider the following example:

- (2) He sweeps the floor clean.

$$[[\text{DO}(\text{he}, \text{sweeps}(\text{the floor}))] \text{CAUSE} \\ [\text{BECOME} [\text{clean}(\text{the floor})]]]]$$

Dowty breaks the event described by (2) into two subevents, the activity of sweeping the floor and its result, the state of the floor being clean. A more recent approach, advocated by Rappaport Hovav and Levin (1998), describes a basic set of event templates corresponding to Vendler’s event classes (Vendler, 1957):

- (3) a. $[x \text{ACT}_{\langle \text{MANNER} \rangle}]$ (activity)
 b. $[x \langle \text{STATE} \rangle]$ (state)
 c. $[\text{BECOME} [x \langle \text{STATE} \rangle]]$ (achievement)
 d. $[x \text{CAUSE} [\text{BECOME} [x \langle \text{STATE} \rangle]]]$ (accomplishment)
 e. $[[x \text{ACT}_{\langle \text{MANNER} \rangle}] \text{CAUSE} [\text{BECOME} [x \langle \text{STATE} \rangle]]]$ (accomplishment)

A process called *Template Augmentation* allows basic event templates to be freely “augmented” to any other event template. This process, for example, explains the resultative form of surface contact verbs like *sweep*:

- (4) a. Phil swept the floor.

$$[\text{Phil ACT}_{\langle \text{SWEEP} \rangle} \text{floor}]$$

 b. Phil swept the floor clean.

$$[[\text{Phil ACT}_{\langle \text{SWEEP} \rangle} \text{floor}] \text{CAUSE} \\ [\text{BECOME} [\text{floor} \langle \text{CLEAN} \rangle]]]]$$

Following this long tradition of research, I propose a syntactically-based event representation specifically designed to handle alternations in argument structure. Furthermore, I will show how this theoretical analysis can be implemented in a feature-driven computational framework. The product is an agenda-driven, chart-based parser for the Minimalist Program.

3 A Decompositional Framework

A primary advantage of decompositional (non-lexicalist) theories of lexical semantics is the ability to transparently relate morphologically related words—explaining, for example, categorial divergences in terms of differences in event structure. Consider the adjective *flat* and the deadjectival verb *flatten*:

- (5) a. The tire is flat.
 b. The tire flattened.

Clearly, (5a) is a stative sentence denoting a static situation, while (5b) denotes an inchoative event, i.e., a transition from “tire is not flat” to “tire is flat”. One might assign the above two sentence the following logical form:

- (6) a. $\text{BE}(\text{tire}, [_{\text{state}} \text{flat}])$
 b. $\text{ARG}_{\delta}(\text{tire}, e) \wedge \text{BECOME}(\text{BE}([_{\text{state}} \text{flat}]), e)$

In Davidsonian terms, dynamic events introduce event arguments, whereas static situations do not. In (6b), the semantic argument that undergoes the change of state (ARG_{δ}) is introduced externally via the event argument.

Considering that the only difference between *flat*._{ADJ} and *flatten*._V is the suffix *-en*, it must be the source of inchoativity and contribute the change of state reading that distinguishes the verb from the adjective. Here, we have evidence that derivational affixes affect the semantic representation of lexical items, that is, fragments of event structure are directly associated with derivational morphemes. We have the following situation:

- (7)
$$\llbracket \text{flat} \rrbracket = [_{\text{state}} \text{flat}]$$

$$\llbracket \text{is flat} \rrbracket = \lambda x \text{BE}(x, [_{\text{state}} \text{flat}])$$

$$\llbracket \text{-en} \rrbracket = \lambda s \lambda x \text{ARG}_{\delta}(x, e) \wedge \text{BECOME}(\text{BE}(s), e)$$

$$\llbracket \text{flat-en} \rrbracket = \lambda x. \text{ARG}_{\delta}(x, e) \wedge \\ \text{BECOME}(\text{BE}([_{\text{state}} \text{flat}]), e)$$

In this case, the complete event structure of a word can be compositionally derived from its component morphemes. This framework, where the “semantic load” is

spread more evenly throughout the lexicon to lexical categories not typically thought to bear semantic content, is essentially the model advocated by Pustejovsky (1991a), among many others. Note that such an approach is no longer lexicalist: each lexical item does not fully encode its associated syntactic and semantic structures. Rather, meanings are composed from component morphemes.

In addition to *-en*, other productive derivational suffixes in English such as *-er*, *-ize*, *-ion*, just to name a few, can be analyzed in a similar way. In fact, we may view morphological rules for composing morphemes into larger phonological units the same way we view syntactic rules for combining constituents into higher-level projections, i.e., why distinguish $VP \rightarrow V + NP$ from $V \rightarrow \text{Adj} + \text{-en}$? With this arbitrary distinction erased, we are left with a unified morpho-syntactic framework for integrating levels of grammar previously thought to be separate—this is indeed one of the major goals of Distributed Morphology. This theoretical framework translates into a computational model better suited for analyzing the semantics of natural language, particularly those rich in morphology.

A conclusion that follows naturally from this analysis is that fragments of event structure are directly encoded in the syntactic structure. We could, in fact, further postulate that all event structure is encoded syntactically, i.e., that lexical semantic representation is isomorphic to syntactic structure. Sometimes, these functional elements are overtly realized, e.g., *-en*. Often, however, these functional elements responsible for licensing event interpretations are not phonologically realized.

These observations and this line of reasoning has not escaped the attention of theoretical linguists: Hale and Keyser (1993) propose that argument structure is, in fact, encoded syntactically. They describe a cascading verb phrase analysis with multiple phonetically empty verbal projections corresponding to concepts such as inchoativity and agentivity. This present framework builds on the work of Hale and Keyser, but in addition to advancing a more refined theory of verbal argument structure, I also describe a computational implementation.

4 Event Types

Although the study of event types can be traced back to Aristotle, it wasn't until the twentieth century when philosophers and linguists developed classifications of events that capture logical entailments and the co-occurrence restrictions between verbs and other syntactic elements such as tenses and adverbials. Vendler's (1957) four-way classification of events into states, activities, accomplishments, and achievements serves as a good starting point for a computational ontology of event types. Examples of the four event types are given below:

(8)	<i>States</i>	<i>Activities</i>
	know	run
	believe	walk
	<i>Accomplishments</i>	<i>Achievements</i>
	paint a picture	recognize
	make a chair	find

Under Vendler's classification, activities and states both depict situations that are inherently temporally unbounded (atelic); states denote static situations, whereas activities denote on-going dynamic situations. Accomplishments and achievements both express a change of state, and hence are temporally bounded (telic); achievements are punctual, whereas accomplishments extend over a period of time. Tenny (1987) observes that accomplishments differ from achievements only in terms of event duration, which is often a question of granularity.

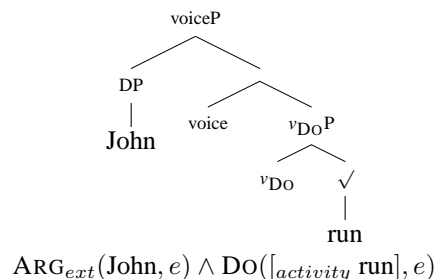
From typological studies, it appears that states, change of states, and activities form the most basic ontology of event types. They correspond to the primitives BE, BECOME, and DO proposed by a variety of linguists; let us adopt these conceptual primitives as the basic vocabulary of our lexical semantic representation.

Following the non-lexicalist tradition, these primitives are argued to occupy functional projections in the syntactic structure, as so-called light verbs. Here, I adopt the model proposed by Marantz (1997) and decompose lexical verbs into verbalizing heads and verbal roots. Verbalizing heads introduce relevant eventive interpretations in the syntax, and correspond to (assumed) universal primitives of the human cognitive system. On the other hand, verbal roots represent abstract (categoryless) concepts and basically correspond to open-class items drawn from encyclopedic knowledge. I assume an inventory of three verbalizing heads, each corresponding to an aforementioned primitive:

- (9) v_{DO} [+dynamic, -inchoative] = DO
 v_{δ} [+dynamic, +inchoative] = BECOME
 v_{BE} [-dynamic] = BE

The light verb v_{DO} licenses an atelic non-inchoative event, and is compatible with verbal roots expressing activity. It projects a functional head, *voice* (Kratzer, 1994), whose specifier is the external argument.

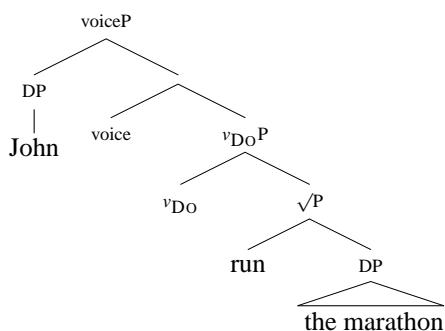
- (10) John ran.



The entire voiceP is further embedded under a tense projection (not shown here), and the verbal complex undergoes head movement and left adjoins to any overt tense markings. Similarly, the external argument raises to [Spec, TP]. This is in accordance with modern linguistic theory, more specifically, the subject-internal hypothesis.

The verbal root can itself idiosyncratically license a DP to give rise to a transitive sentence (subjected, naturally, to selectional restrictions). These constructions correspond to what Levin calls “non-core transitive sentences” (1999):

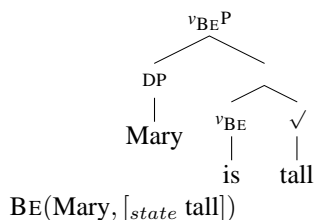
(11) John ran the marathon.



$\text{ARG}_{ext}(\text{John}, e) \wedge \text{DO}([\text{activity } \text{run}(\text{marathon})], e)$

Similarly, v_{BE} licenses static situations, and is compatible with verbal roots expressing state:

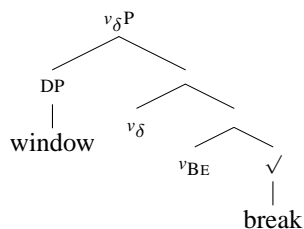
(12) Mary is tall.



$\text{BE}(\text{Mary}, [\text{state } \text{tall}])$

The light verb v_{δ} licenses telic inchoative events (i.e., change of states), which correspond to the BECOME primitive:

(13) The window broke:

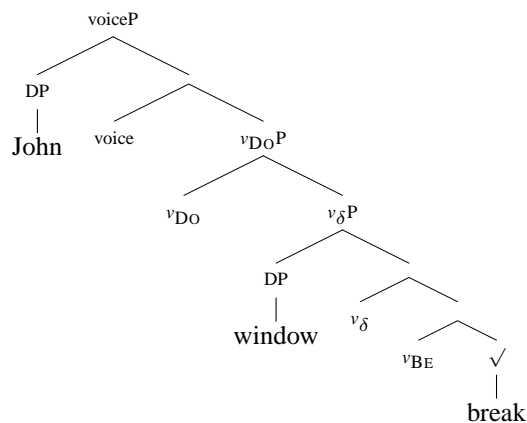


$\text{ARG}_{\delta}(\text{window}, e) \wedge \text{BECOME}(\text{BE}([\text{state } \text{break}], e))$

The structure denotes an event where an entity undergoes a change of state to the end state specified by the root. $v_{\delta}P$ can be optionally embedded as the complement

of a v_{DO} , accounting for the causative/inchoative alternation. Cyclic head movement (incorporation) of the verbal roots into the verbalizing heads up to the highest verbal projection accounts for the surface form of the sentence.

(14) John broke the window.

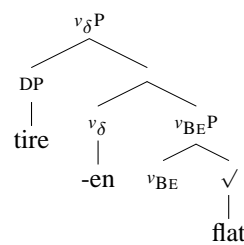


$\text{CAUSE}(e_1, e_2) \wedge \text{ARG}_{ext}(\text{John}, e_1) \wedge$
 $\text{DO}([\text{activity } \text{undef}], e_1) \wedge \text{ARG}_{\delta}(\text{window}, e_2) \wedge$
 $\text{BECOME}(\text{BE}([\text{state } \text{break}], e_2))$

Note that in the causative form, v_{DO} is unmodified by a verbal root—the manner of activity is left unspecified, i.e., “John did something that caused the window to undergo the change of state break.”

Given this framework, deadjectival verbs such as *flatten* can be directly derived in the syntax:

(15) The tire flattened.



$\text{ARG}_{\delta}(\text{tire}, e) \wedge \text{BECOME}(\text{BE}([\text{state } \text{flat}], e))$

In (Lin, 2004), I present evidence from Mandarin Chinese that this analysis is on the right track. The rest of this paper, however, will be concerned with the computational implementation of my theoretical framework.

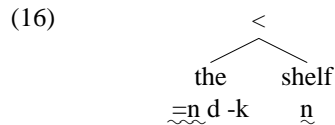
5 Minimalist Derivations

My theory of verbal argument structure can be implemented in a unified morpho-syntactic parsing model that interleaves syntactic and semantic parsing. The system is in the form of an agenda-driven chart-based parser whose foundation is similar to previous formalizations of Chomsky’s Minimalist Program (Stabler, 1997; Harkema, 2000; Niyogi, 2001).

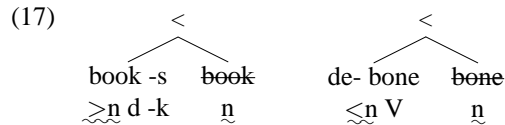
Lexical entries in the system are minimally specified, each consisting of a phonetic form, a list of relevant features, and semantics in the form of a λ expression.

The basic structure building operation, MERGE, takes two items and creates a larger item. In the process, compatible features are canceled and one of the items projects. Simultaneously, the λ expression associated with the licenser is applied to the λ expression associated with the licensee (in theoretical linguistic terms, Spell-Out).

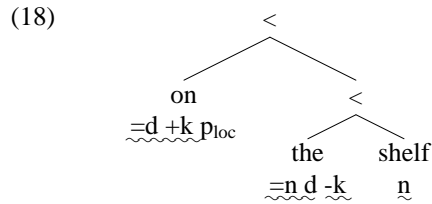
The most basic feature is the $=x$ licenser feature, which cancels out a corresponding x licensee feature and projects. A simple example is a determiner selecting a noun to form a determiner phrase (akin to the context free rule $DP \rightarrow \text{det noun}$). This is shown below (underline indicates canceled features, and the node label $<$ indicates that the left item projects):



The features $>x$ and $<x$ trigger head movement (incorporation), i.e., the phonetic content of the licensee is affixed to the left or right of the licenser's phonetic content, respectively. These licenser features also cancel corresponding x licensee features:



Finally, feature checking is implemented by $+x/-x$ features. The $+x$ denotes a need to discharge features, and the $-x$ denotes a need for features. A simple example of this is the case assignment involved in building a prepositional phrase, i.e., prepositions must assign case, and DPs much receive case.



Niyogi (2001) has developed an agenda-driven chart parser for the feature-driven formalism described above; please refer to his paper for a description of the parsing algorithm. I have adapted it for my needs and developed grammar fragments that reflect my non-lexicalist semantic framework. As an example, a simplified derivation of the sentence “The tire flattened.” is shown in Figure 1.

The currently implemented system is still at the “toy parser” stage. Although the effectiveness and coverage

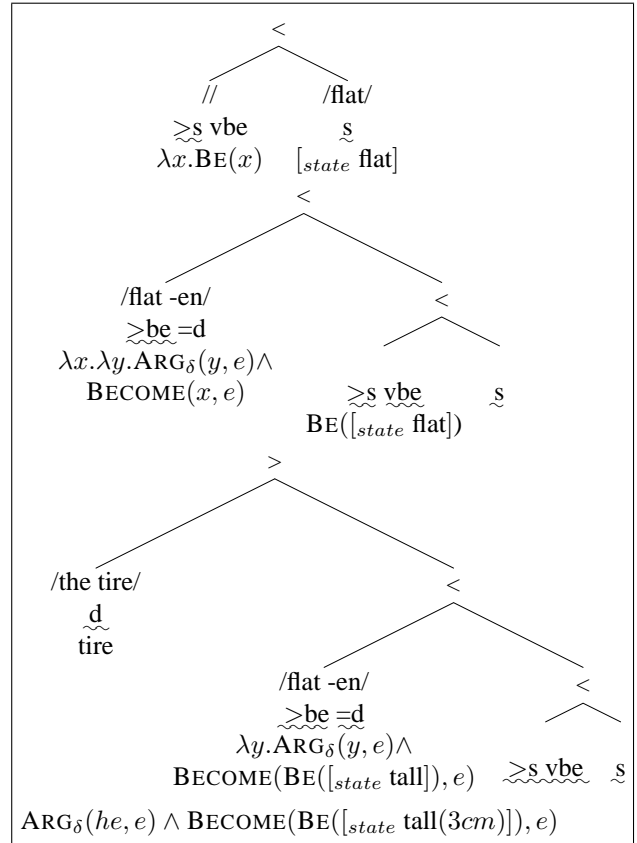


Figure 1: Simplified derivation for the sentence “The tire flattened.”

of my parser remains to be seen, similar approaches have been successful at capturing complex linguistic phenomena. With a minimal set of features and a small number of lexical entries, Niyogi (2001) has successfully modeled many of the argument alternations described by Levin (1993) using a Hale and Keyser (1993) style analysis. I believe that with a suitable lexicon (either hand crafted or automatically induced), my framework can be elaborated into a system whose performance is comparable to that of current statistical parsers, but with the added advantage of simultaneously providing a richer lexical semantic representation of the input sentence than flat predicate argument structures based on semantic roles.

6 Conclusion

A combination of factors in the natural development of computational linguistics as a field has conspired to narrow the diversity of techniques being explored by researchers. While empirical and quantitative research is the mark of a mature field, such an approach is not without its adverse side-effects. Both syntactic and semantic parsing technology faces a classic chicken-and-egg prob-

lem. In order for any new framework to become widely adopted, it must prove to be competitive with state-of-the-art systems in terms of performance. However, robust parsing cannot be achieved without either laboriously crafting grammars or a massive dedicated annotation effort (and experience has shown the latter method to be superior). Therein, however, lies the catch: neither effort is likely to be undertaken unless a new framework proves to be quantitatively superior than previously established methodologies. Lacking quantitative measures currently, the merits of my proposed framework can only be gauged on theoretical grounds and its future potential to better capture a variety of linguistic phenomena.

References

- Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING/ACL 1998)*.
- Eugene Charniak. 2001. Immediate head parsing for language models. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL-2001)*.
- Michael Collins. 1997. Three generative lexicalized models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL-1997)*.
- David Dowty. 1979. *Word Meaning and Montague Grammar*. D. Reidel Publishing Company, Dordrecht, The Netherlands.
- Charles J. Fillmore. 1968. The case for case. In E. Bach and R. Harms, editors, *Universals in Linguistic Theory*, pages 1–88. Holt, Rinehart, and Winston, New York.
- Kenneth Hale and Samuel Jay Keyser. 1993. On argument structure and the lexical expression of syntactic relations. In Kenneth Hale and Samuel Jay Keyser, editors, *The View from Building 20: Essays in Linguistics in Honor of Sylvain Bromberger*. MIT Press, Cambridge, Massachusetts.
- Morris Halle and Alec Marantz. 1993. Distributed morphology and the pieces of inflection. In Kenneth Hale and S. Jay Keyser, editors, *In The View from Building 20*, pages 111–176. MIT Press, Cambridge, Massachusetts.
- Henk Harkema. 2000. A recognizer for minimalist grammars. In *Proceedings of the Sixth International Workshop on Parsing Technologies (IWPT 2000)*.
- Ray Jackendoff. 1983. *Semantics and Cognition*. MIT Press, Cambridge, Massachusetts.
- Paul Kingsbury, Martha Palmer, and Mitch Marcus. 2002. Adding semantic annotation to the Penn TreeBank. In *Proceeding of 2002 Human Language Technology Conference (HLT 2002)*.
- Angelika Kratzer. 1994. The event argument and the semantics of voice. Unpublished manuscript, University of Massachusetts, Amherst.
- Beth Levin and Malka Rappaport Hovav. 1996. From lexical semantics to argument realization. Unpublished manuscript, Northwestern University and Bar Ilan University.
- Beth Levin. 1993. *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press, Chicago, Illinois.
- Beth Levin. 1999. Objecthood: An event structure perspective. In *Proceedings of the 35th Annual Meeting of the Chicago Linguistics Society*.
- Jimmy Lin. 2004. *Event Structure and the Encoding of Arguments: The Syntax of the English and Mandarin Verb Phrase*. Ph.D. thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.
- Alec Marantz. 1997. No escape from syntax: Don't try morphological analysis in the privacy of your own lexicon. In *Proceedings of the 21st Annual Penn Linguistics Colloquium*.
- Sourabh Niyogi. 2001. A minimalist implementation of verb subcategorization. In *Proceedings of the Seventh International Workshop on Parsing Technologies (IWPT-2001)*.
- James Pustejovsky. 1991a. The generative lexicon. *Computational Linguistics*, 17(4):409–441.
- James Pustejovsky. 1991b. The syntax of event structure. *Cognition*, 41:47–81.
- Malka Rappaport Hovav and Beth Levin. 1998. Building verb meanings. In Miriam Butt and Wilhelm Geuder, editors, *The Projection of Arguments: Lexical and Compositional Factors*. CSLI Publications, Stanford, California.
- Edward Stabler. 1997. Derivational minimalism. In Christian Retoré, editor, *Logical Aspects of Computational Linguistics*. Springer.
- Carol Tenny. 1987. *Grammaticalizing Aspect and Affectiveness*. Ph.D. thesis, Massachusetts Institute of Technology.
- Zeno Vendler. 1957. Verbs and times. *Philosophical Review*, 56:143–160.