A Learnability Analysis of Argument and Modifier Structure

Leon Bergen,[1] Edward Gibson, and Timothy J. O'Donnell

MIT

Author Note

We thank Vera Demberg for extensive help in evaluating the model in an earlier version of this paper. We also thank Roni Katzir, Josh Tenenbaum, Tom Wasow, Jelle Zuidema, Mark Johnson, Jim Rogers, and audience members at ACL 2013 for many useful comments and suggestions and Jesse Snedeker, Josh Hartshorne, Erin Conwell, Dave Barner, Mike Frank, and Bob Berwick for answering questions related to this work. Finally, we thank Gerry Altmann and an anonymous reviewer for comments and questions which led to a greatly improved version of this paper.

[1] Corresponding author. Address: 43 Vassar St, Room 3037, Cambridge, MA 02139. Phone: 732-266-3200. Email: bergen@mit.edu

Abstract

We present a computational learnability analysis of the argument-modifier distinction, asking whether information present in the distribution of constituents in natural language supports the distinction and its learnability. We first develop general models of those aspects of argument structure and the argument-modifier distinction which have effects on the the distribution of constituents in sentences—abstracting away many of the implementational details of specific theoretical proposals. Combining these models with a theory of learning based on succinctness, we define two systems, the argument-only (PTSG) model and the argument-modifier (PSAG) model.

We first show that the argument-modifier (PSAG) model is able to recover the argument-modifier status of many individual constituents when evaluated against a gold standard. This provides evidence in favor of our general account of argument-modifier structure as well as providing a lower bound on the amount of information that natural language input can provide for appropriately equipped learners attempting to recover the argument-modifier status of individual constituents.

We then present a series of analyses investigating how and why the argument-modifier (PSAG) model is able to recover the argument-modifier status of some constituents. In particular, we show that the argument-modifier (PSAG) model model is able to provide a more succinct description of the input corpus than the argument-only (PTSG) model model, both in terms of lexicon size, and in terms of the complexity of individual derivations—both on the training data and for a novel heldout dataset. Intuitively, the argument-modifier (PSAG) model model is able to learn a more compact lexicon with more generalizable argument structures because it is able to "prune away" spurious modifier structure. These analyses further support our general account of argument-modifier structure and its learnability from naturalistic input.

We conclude with a discussion of the generality of our approach and the role of such computational learnability analyses to the study of grammar.

Keywords: Argument structure; language learning; grammatical representation;

probabilistic models; Bayesian nonparametrics; linguistics

A Learnability Analysis of Argument and Modifier Structure

Introduction

The flexibility and expressivity of natural language is made possible because the computational system underlying language divides labor between an inventory of stored items (e.g., morphemes, words, idioms, and other constructions), known as the *lexicon*,[2] and a set of structure-building operations which combine lexical items to create an unbounded number of new expressions. The ways in which sentences can be built are highly constrained by requirements imposed by specific lexical items. Consider the verb *put*. In its most basic meaning, this verb can only appear in sentences which contain constituents expressing: (i) who is doing the putting, (ii) what is being put, and (iii) the destination of the PUTTING event. All else being equal, a native speaker will perceive a sentence such as *John put the loaf of bread* as incomplete, expecting a phrase which expresses the destination event: *John put the loaf of bread in his kitchen cupboard*. Furthermore, *put* imposes other requirements on sentence structure, such as the requirement that object being put be expressed as a noun phrase. Without suggesting that such lexically-specified requirements represent a single, unified phenomenon, we will call phrases such as (i–iii) the *arguments* of *put* and all such information associated with a lexical item, the *argument structure* of the lexical item.

Based on such facts, linguistic theories have increasingly adopted a *lexically-driven* view of grammar. Under such an architecture, grammatical computation is characterized by a small number of structure-building functions (e.g., UNIFY, MERGE, etc.) whose operation is controlled by the argument-structure specifications of lexical items (Bresnan, 2001; Chomsky, 1995a, 1995c; Culicover & Jackendoff, 2005; Gamut, 1991; Gazdar, Klein, Pullum, & Sag, 1985; Heim

[2] In the linguistics literature, the terms *lexicon*, *lexical*, and *lexical item* are used in two senses. Typically, they refer to the inventory of stored units which are composed to create larger expressions—whether these units are morphemes, words, idioms, or other constructions. This is the sense we intend. However, under *lexicalist* approaches to grammar, it is hypothesized that word-structure and sentence-structure are handled by distinct modules of the linguistic system. In this context, *lexicon*, *lexical*, etc. are often used to refer to the entire module of grammar which computes word structure—including its productive rules—not just to stored units. Di Sciullo and Williams (1987) proposed that to avoid confusion the first sense of *lexical* (i.e., stored) be replaced with the term *listed*, and *lexical item* with *listeme*. We maintain the more standard usage, but emphasize we take no position about the number of computational modules underlying linguistic structure.

& Kratzer, 2000; Huddleston, Pullum, et al., 2002; Jackendoff, 2002; D. E. Johnson & Postal, 1980; McConnell-Ginet & Chierchia, 2000; Mel'čuk, 1988; Moortgat, 1997; Sag, 2012; Sag, Wasow, & Bender, 2003; Stabler, 1997; Steedman, 2000). Some theorists have tentatively suggested that all cross-linguistic variation might be reducible to lexical argument structure and, thus, that language learning might be reduceable to the acquisition of the lexicon (e.g., Chomsky, 1993; Steedman, 2000). In parallel, after decades of little progress, the use of lexically-driven formalisms in natural language processing has led to rapid advances in unsupervised grammar induction (Bisk & Hockenmaier, 2013; Cohn, Blunson, & Goldwater, 2010; Headden III, Johnson, & McClosky, 2009; Naseem, Chen, & Johnson, 2010; Spitkovsky, Alshawi, Chang, & Jurafsky, 2011). Taken together, these developments suggest that the adoption of lexically-driven approaches to grammar constitutes an important development in our understanding of the logical problem of language acquisition.

However, although it is clear that much of the structure of sentences can can be explained by the argument-structure requirements of lexical items, there are also constituents that do not obviously satisfy such requirements. Consider the sentence: *While preparing dinner, John thoughtlessly put the loaf of bread in his kitchen cupboard*. In this sentence, the phrases *while preparing dinner* and *thoughtlessly*, specify additional information about the time and manner of the PUTTING event, but the sentence is well-formed and interpretable without them. These phrases also differ in a number of other ways from the core arguments of the verb. For instance, while the argument-phrase specifying the doer of the PUTTING event (i.e., *John*) must appear in the subject position of the sentence (**put the loaf of bread John in his kitchen cupboard*), these other phrases can appear in a greater variety of positions (*John thoughtlessly put the loaf of bread in his kitchen cupboard, while preparing dinner*). Again, without implying that the such non-argument phrases represent a single, unified phenomenon, we refer to non-argument phrases as *modifiers*.[3]

[3] Another widely-used term is *adjunct*. However, this word tends to carry heavier theoretical implications (see below), so we prefer the more theoretically-neutral term. We also note a slight abuse of terminology. Throughout the paper the term *modifier* essentially just means "non-argument." However, there are a number of grammatical phenomena besides canonical modifiers which are also often argued to be the result of non-argument modes of composition. These includes parataxis (e.g., *The louder, the better*), parenthesis (*You are, I believe, late to the party*), apposition (*The next finalist, Susan, receives one hundred dollars*), and sometimes even coordination (e.g., *Mary ran to the store*

The existence of such (apparent) non-argument-driven structure raises two fundamental questions for grammatical theory. First, and most basically, is a formal distinction between arguments and non-arguments really needed? Historically, most grammatical theories have proposed mechanisms of non-argument composition (usually formalized as one form or another of *adjunction*). However, as we review below, there is a great deal of debate about both the nature of such mechanisms and the relevant empirical phenomena. Given this lack of consensus and the practical advantages of lexically-driven theories, one might ask whether such non-argument modes of composition are really necessary. Second, if some constituents are modifiers, this may significantly complicate the problem of lexicon learning. The status of individual phrases is frequently uncertain. For example, while in the sentence *John put the loaf of bread in his kitchen*, the phrase *in his kitchen* is typically considered an argument, it is not in the sentence *John made the loaf of bread in his kitchen* (cf. *John made the loaf of bread*). How do learners determine which phrases are lexically-specified and which are not in light of such ambiguities?

In this paper, we use computational modeling to address these two questions. First, we argue that, despite controversies in the literature, the statistics of natural language corpora provide evidence in support of non-argument modes of composition. Furthermore, this evidence is complementary to traditional linguistic evidence and suggests why the argument-/modifier-status of some phrase types has been controversial in the literature. Second, although—in principle—the existence of a grammatical argument-modifier distinction complicates the lexicon learning problem, we argue that—in practice—this problem is mitigated by the very same evidence we use to argue in favor of the psychological reality of the distinction. Information in the distribution of forms in the input can be leveraged by appropriate learning algorithms to help determine the argument- or modifierhood of individual phrases.

In order to make these arguments, we introduce two simple, idealized models of syntactic structure. We argue that lexically specified requirements on sentence structure share three logical

and John cleaned the kitchen). Although we have not performed a detailed analysis of the question, it is possible that some constituents in the corpora we will be analyzing actually represent instances of these other non-argument phenomena. Regardless, to limit the scope of our discussions we will focus on the more common case of canonical modifiers (i.e., adjuncts).

properties that have direct consequences on the distribution of forms in the input: each lexical item specifies a finite (usually small) number of arguments (**finiteness**), that tend to be obligatory (**obligatoriness**), and appear in fixed structural positions (or relations; **structural fixity**).[4] We adopt a model designed to minimally capture these properties known as the *argument-only model* and formalized using *probabilistic tree-substitution grammars* (PTSG; Bod, 1998; Scha, 1990, 1992).

We then extend this model with a non-lexical composition operator which satisfies the negation of the three properties above, allowing an unbounded number of applications (**iterability** v **finiteness**), applying non-obligatorily (**optionality** v. **obligatoriness**), and allowing constituents to be composed in a variety of structural relationships with one other (**structural flexibility** v. **structural fixity**). We call this system the *argument-modifier model* and implement it using a formalism known as *probabilistic sister-adjunction grammar* (PSAG; Chiang & Bikel, 2002; Rambow, Vijay-Shanker, & Weir, 1995).

Although these models differ in their representational assumptions, they share a common set of assumptions about learning. In particular, both models assume prior biases in favor of simplicity or *succinctness*. Following earlier work, we formalize two competing succinctness biases (Bod, Scha, & Sima'an, 2003; Cohn et al., 2010; de Marcken, 1996a, 1996b; O'Donnell, 2011, in press; Post & Gildea, 2013, *inter alia*). The first favors lexicons with smaller numbers of lexical items and the second favors simpler derivations of individual sentences. Inference in this framework corresponds to optimizing a tradeoff between these two prior biases with respect to the input data. Furthermore, this tradeoff can be understood as a special case of the standard prior-likelihood optimization invoked in Bayesian and Minimum-Description Length approaches to learning when the prior corresponds to the lexicon of stored structures and the likelihood corresponds to the process of deriving individual sentences (Berwick, 1982; Grünwald, 2007; A. S. Hsu, Chater, & Vitányi, 2013; Li & Vitányi, 2008).

[4] Note that we do not argue that these are the only properties that distinguish arguments from modifiers, merely that these properties follow logically from the lexical/non-lexical distinction and that they have predictable distributional consequences.

Our empirical arguments rely on two sets of simulations. First, we demonstrate that our formalization of argument structure and the argument-modifier distinction are empirically plausible. We do this by showing that the argument-modifier (PSAG) model correctly classifies individual phrases as arguments or modifiers when evaluated against an independently derived gold standard. Second, we demonstrate that the argument-modifier (PSAG) model provides a superior account of the linguistic input than the argument-only (PTSG) model. We do this by demonstrating that it both provides a more succinct account of the training corpus and greater generalization to new sentences.

Since the argument-modifier (PSAG) model matches linguistic intuition and provides a superior account of linguistic data, we argue that these results provide evidence against a strictly lexically-driven approach to grammatical structure. Furthermore, since the argument-modifier (PSAG) model can determine the correct argument-modifier classification from the distribution of forms in the input, we conclude that this same information can be leveraged by a learner that makes similar representational and learning assumptions. Thus, the existence of non-argument constituents needn't be major problem for the acquisition of lexical argument-structure. In the discussion, we consider the generality of this result, and its implications for linguistic theories.

Argument Structure

In this study, we are interested in *distributional* effects of non-lexical composition. With this in mind, we adopt very broad notions of *argument* and *argument structure*: any lexically-specified constraint on constituent co-occurrence. This includes verb-argument structure, but also the lexical requirements of other categories such as prepositions or nouns. For example, English prepositions like *on* or *from*, typically require an argument noun phrase, often with additional semantic requirements, such as the requirement that the object of the preposition *into* must be a space capable of containment. We also intend this notion to abstract over the various different modules or kinds of information encoded by the lexical representations of different approaches to grammar (e.g., case theory, θ-roles, ARG-ST lists, c-/f-structure, etc.).

From this distributional perspective, there are three crucially-important generalizations about lexical argument structures. First, each lexical item specifies a finite (usually small) number of arguments (**finiteness**). Second, arguments are typically obligatory (**obligatoriness**). Third, and finally, particular arguments are required to appear in fixed structural positions (with respect to the selecting lexical item; **structural fixity**). In languages, like English, which rely mostly on word order to encode structural relationships, this corresponds to fixity of order.[5]

Of course, these three properties are idealizations. There is considerable variability in the argument structure requirements of individual lexical items. For example, for some verbs an argument may be optional (e.g., *John ate*/*John ate the cake*) and for others there is more than one way to realize the same arguments (e.g., *John gave Mary the book*/*John gave the book to Mary*) Explaining such argument variability is an important problem for linguistic theory (see, for example, Levin & Rappaport Hovav, 2005, for a comprehensive review of verb-argument realization). However, work on argument structure has shown that the range of structures which are associated with particular lexical items is sharply delimited and in this paper we will model such variability as lexical ambiguity. From the perspective of the distribution of constituents in the linguistic input which we study here, this is a sufficient approximation of lexical argument structure.

There is, however, another aspect of lexical argument structure which we cannot ignore. Over the last three decades, there has been an increasing recognition of an important problem: apparent compositionality in phrasal constructions which are in fact not fully compositional. The problem arises most clearly in the case of verbal idioms such as *leave no stone unturned* or *kick the bucket*. The (idiomatic) meaning of these verb phrases cannot be predicted solely from their component words, despite the fact that they look like other verb phrases in terms of their morphology (and basic syntax), and do, in fact, have fully-compositional readings: *John kicked the bucket* could refer to John actually kicking a bucket. Like all unpredictable linguistic structure, the idiomatic interpretation of these idioms must be stored in memory. However, how

[5] Other languages employ case-marking, agreement, or other grammatical devices to express structural relations.

these interpretations are stored and what, specifically, they are associated with is a highly complex issue: *The investigators left no rock unflipped* and *The bucket was kicked by John* do not have idiomatic interpretations while *No legal stone was left unturned by the investigators* and *John will kick the bucket any day now* do.

Importantly, the idiomatic interpretation of such constructions is contingent on the co-occurrence of specific lexical items (not merely similar-meaning lexical items, as in *leave no rock unflipped*), but the constructions vary in the flexibility of the syntactic constructions in which that co-occurrence will be recognized as idiomatic (*kick the bucket* cannot be passivized, while *leave no stone unturned* can). Moreover, many idiomatic constructions themselves select for arguments. For example, the idiom *to give* `NP` *a hard time*, selects for a noun phrase direct object in much the same way as other verbs, except that it has the idiosyncratic meaning *to criticize NP*.

These facts raise a large number of highly contentious and currently controversial questions about representation and learning. However, regardless of how such observations are handled by a linguistic theory, all approaches must provide some mechanism for the storage of lexical contingencies like those in *leave no stone unturned* and these will necessarily have consequences on the distribution of forms in the linguistic input and greatly increasing the ambiguity of the learning problem. By adulthood, language users have acquired the knowledge to identify which forms and which parts of forms should be computed compositionally in the appropriate context. At the beginning of language acquisition, however, a child presented with a sentence such as *I threw the ball to the ground* has to entertain the possibility that combinations such as *throw the ball to the ground*, *throw* `NP` *to the ground*, *throw the ball to* `NP`, etc. may have some idiomatic interpretation or be otherwise stored in part or whole in the lexicon. Thus, the models that we consider in this paper will be designed to represent and learn a wide variety of argument structures, including those with compositionality that is merely apparent.

Tree-Substitution Grammars as Models of Argument Structure

A tree-substitution grammar formalizes the lexicon as an inventory of stored tree fragments, such as those shown in Figure 1 (Bod, 1998). Each tree fragment encodes the category and structural position of additional argument phrases that must be present in a complete sentence which is derived using the fragment. In a tree-substitution grammar, lexical fragments are combined via the SUBSTITUTE operation, which replaces a node at the frontier of a derivation with another tree fragment from the lexicon—subject to the condition that the category of the frontier node and the category of the substituted fragment are identical. The SUBSTITUTE operation is applied recursively until no substitutable nodes remain at the frontier, and a complete sentence has been derived.

The formalism for the tree-substitution grammar does not itself make any claims about the origin of the tree fragments in the lexicon. For this paper, we will assume that the tree fragments are generated from a context-free grammar, which will be referred to as the base system for the lexicon. For example, in order to generate the top-left tree in Figure 1, the base system would use three rules: a) $\texttt{S} \rightarrow \texttt{NP VP}$; b) $\texttt{VP} \rightarrow \texttt{V NP PP PP}$; and c) $\texttt{V} \rightarrow \texttt{put}$. After each tree fragment has been generated by the base system, it is stored in the lexicon, and can be reused during later sentence derivations.

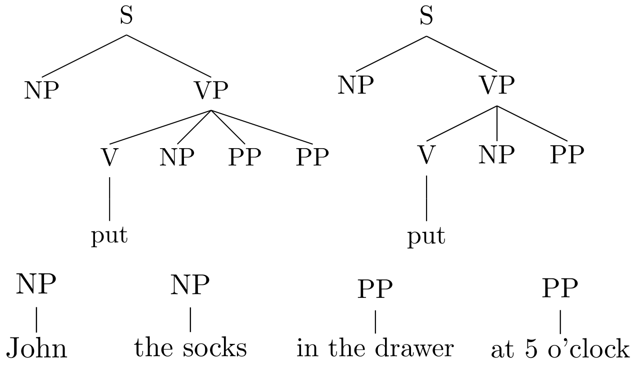


Figure 1. This figure shows the inventory of elementary trees used in our example derivations. Note that the internal constituent structure of the noun and prepositional phrases (NP and PP) has been suppressed.

Tree-substitution grammars capture the three core empirical observations about argument structure mentioned above. Each lexical fragment can only possess a fixed number of leaf argument variables (**finiteness**). All such variables must be filled in a complete derivation (**obligatoriness**). And, the position of each argument phrase is fixed relative to the lexical item which selects for it (**structural fixity**). Furthermore, since tree-substitution grammars allow combinations of elements to be stored as single tree fragments, they can capture distributional aspects of apparently-compositional structures, such as the idioms discussed in the last section.

Arguments and Modifiers

As we saw in the preceding section, all theories of grammar must necessarily include some notion of lexical argument structure (in our general sense). The question we ask in this paper is whether lexical argument structure is not only necessary, but also sufficient. Unfortunately, the answer to the question is not obvious. On one hand, nearly all theories of syntax and semantics have assumed the existence of non-argument modes of composition, suggesting that argument structure is not alone sufficient. On the other hand, there is little consensus on the correct theoretical characterization of non-argument composition or the empirical phenomena that are most relevant—theories often make conflicting or ambiguous predictions about the modifierhood of individual cases. If, in practice, almost all phrases fill some kind of argument role, perhaps modification can be reduced to lexical ambiguity, and the argument-structure-only model can be rescued at the cost of a larger lexicon.

In this section, we review earlier literature on the argument-modifier distinction. We begin by discussing the empirical phenomena which motivate the distinction, and problems which arise with these data. We then briefly outline theoretical formulations of the distinction. Finally, we conclude this section by describing the approach we adopt in this paper, introducing a formal model known known as *sister-adjunction grammars* (`SAG`; Chiang & Bikel, 2002; Rambow et al., 1995).

Notions that are similar to the argument-modifier distinction have a long pedigree in linguistics. For example, Bloomfield (1933) discusses the difference between *endocentric* phrases whose properties are predictable from their head word, and *exocentric* phrases whose properties are not (see, Somers, 1984; Vater, 1978, for further discussion of historical precedents).[6] Most modern theories of syntax make use of a slightly different terminology, distinguishing between *complements*, corresponding to arguments, and *adjuncts*, corresponding to modifiers (e.g., Adger, 2003; Bresnan, 2001). These distinctions are motivated by a variety of syntactic and semantic phenomena.[7] We first consider two semantic differences.

- **Semantic "Coreness"**: It is often observed that there is an intuitive semantic difference in the contribution of arguments and modifiers to the meaning of a sentence. Arguments constitute part of the "core" meaning of a relation, whereas modifiers are more "peripheral." For example, in verb argument structure, arguments often encode event participants, whereas modifiers typically encode circumstantial information, such as time, place, or manner (Matthews, 1981). Chomsky (1965, p. 101) points out that the sentence *he decided on the boat* has two readings, one which means "he chose the boat" and one which means "he made his decision while on the boat." Intuitively, in the former the phrase *on the boat* has an interpretation which is specifically and idiosyncratically related to the meaning of *decide* (more "cohesive"), whereas, in the latter, it does not—instead, it simply indicates place in the much the same way it would if it modified any verb phrase. Similarly, Hornstein and Lightfoot (1981) give the examples *John is a student of physics* versus *John is a student with long hair* and point out that the former attributes a single property (i.e., studying physics) to John, while the latter attributes two properties (i.e., studying something and having long hair) to John. In some cases, such differences are reflected in the availability of a semantically-equivalent, copular paraphrase for modifiers but not for arguments: *the student from Paris*/*the student that was from Paris* versus *the student of physics*/**the student that was of physics*.

[6] Somers (1984) reviews similar distinctions proposed by a number of earlier theorists including Tesniére (1959), Halliday (1970), Platt (1971), Cook (1972), and Longacre (1976).

[7] All examples in the section are drawn from the following citations: Borsley (1999); Comrie (1993); Creissels (2014); Croft (2001); Forker (2014); Haegeman (1994); Hornstein and Lightfoot (1981); Kroeger (2004); Matthews (1981); Przepiórkowski (1999a); Radford (1988); Rákosi (2006); Schütze (1995); Tallerman (2015); Vater (1978); Wichmann (2014); Zwicky (1993)

- **Latency**: If an argument is unexpressed, there must be a definite interpretation *latently* available in discourse for it (Croft, 2001; Forker, 2014; Matthews, 1981; Zwicky, 1993). In other words, in sentences such *I didn't finish* or *John ate*, the unexpressed object must be inferable in context. No similar requirement holds for modifiers.

Such observations have occasionally led to proposals that the argument-modifier distinction might be definable in terms of the truth conditions or other aspects of the meaning of sentences. For example, it has been suggested that arguments are those entities whose existence is entailed by a relation, while modifiers encode additional, non-essential information about the relation (e.g., Schikowski, Paudyal, & Bickel, 2014; Schütze & Gibson, 1999). In a sentence like *While preparing dinner, John thoughtlessly put the loaf of bread in his kitchen cupboard*, it is claimed that the existence of the agent (*John*), the theme (*the loaf of bread*), and the location (*in his kitchen cupboard*), is entailed by the `putting` event, while the manner (*thoughtlessly*) and time (*while preparing dinner*) are not.

Although it is obvious that arguments and modifiers often differ in meaning, attempts to *define* these notions in terms of meaning suffer from a number of problems. First, it isn't at all clear that notions such as entailment usefully distinguish arguments from modifiers. Presumably, the occurrence of a token instance of a `putting` event, like that described above, entails that the event took place at a particular time and in a particular manner. How does the entailment of a time or manner differ from entailment of an agent or theme? A second problem for semantic definitions of the argument-modifier distinction is the existence of syntactically obligatory constituents which do not correspond to meaningful entities in a relation. A classic example of these are semantically vacuous expletives such as *it* in *it's raining*. This pronoun fulfills the requirement in English that all sentences must have a subject, but does not correspond to a semantic participant in the `raining` event. As a result of these considerations, most theories have characterized the argument-modifier distinction in structural terms.

There are a number of syntactic phenomena which are commonly associated with the argument-modifier distinction.[8]

- **Obligatoriness**: Arguments tend to be obligatory, while modifiers are always optional.
- **Iterability**: Arguments fill semantic roles which can only be filled once (e.g., AGENT, PATIENT, THEME), whereas multiple modifiers with the same semantic role can appear in a sentence: *John inquired in the morning at ten o'clock* v. **John asked Mary Bill*, *The student of physics with long hair with glasses* v. **The student of physics of chemistry* . Relatedly, if a constituent contains multiple modifiers with the same semantic role, they can often appear in any order: *The student with long hair at the office* v. *The student at the office with glasses*.
- **Co-Occurrence with Head Words**: Arguments appear with a more restricted range of head words than modifiers (e.g., *John asked Mary/*John inquired Mary* [argument] v. *John asked in the morning/John inquired in the morning* [modifier]; *a member of parliament/*a dog of parliament* [argument] v. *a member with gray hair/dog with gray hair* [modifier]). This reflects the fact that lexical items can specify idiosyncratic syntactic and semantic selectional requirements for arguments but not for modifiers (see, also, Hartmann, Haspelmath, & Cysouw, 2014).
- **Structural Position**: Arguments tend to appear in more structurally central positions or roles. In particular, they often appear closer to the head noun of the phrase they are part of: *He laughed at the clown at ten o'clock* v. $^?$*He laughed at ten o'clock at the clown*; *a student of physics with gray hair* v. **a student with gray hair of physics*.
- **Accessibility to Syntactic Operations**: Arguments and modifiers show differential accessibility to various syntactic operations. For example, extraction of/from arguments is generally easier than extraction of/from modifiers (Huang, 1982; Ross, 1967; Truswell, 2007): *[What branch of physics]$_i$ are you a student of _$_i$?* v. **[What color hair]$_i$ are you a*

[8] The properties discussed below cover the most common structural phenomena associated with the argument-modifier distinction, but are not exhaustive (see, e.g., Schütze, 1995, for a list of tests of argumenthood for prepositional phrase dependents of nominal heads).

student with $\_\_i$*?*; *What*$_i$ *was John saying that Peter explained* $\_\_i$ v. **What*$_i$ *was John bothered because Peter explained* $\_\_i$.

- ***Pro*-Form Substitution**: Head-argument combinations can often be replaced with a pro-form, whereas head-modifier combinations often cannot: *John will wash his socks in the kitchen and Ben will do so in the bathroom* v. **John will put his socks in the kitchen and Ben will do so in the bathroom*.

- **Coordinability**: Arguments can be coordinated with arguments, and modifiers can be coordinated with modifiers, but sometimes arguments cannot be coordinated with modifiers: *The student of chemistry and of physics* and *The student with glasses and with long hair* v. **The student with glasses and of physics* and **The student of physics and with glasses*.

An important difficulty with characterizing argument-/modifierhood is that the empirical phenomena just discussed often do not align. Instead, many individual constituents pattern with arguments according to some of the criteria above, and with modifiers according to others (see, Forker, 2014; Haspelmath, 2014; Koenig, Mauner, & Bienvenue, 2003; Przepiórkowski, 1999a, 1999b; Schütze & Gibson, 1999; Tutunjian & Boland, 2008; Vater, 1978). Some difficult cases which have been discussed in the literature include instrumental prepositional phrases, passive *by*-phrases (Kay, 2005; Schütze, 1995), and locative phrases in ditransitives (Pesetsky, 1994). Moreover, it is often even more difficult to align such phenomena across languages (see, Haspelmath, 2014; Wichmann, 2014, and other papers in the same issue of *Linguistic Discovery*).

These problems have led to a situation where there is no generally agreed upon theoretical classification of phrase types and little consensus on the correct formal machinery for handling the argument-modifier distinction.[9]

There are two major classes of theoretical mechanism used to account for modifier phrases. *Lexical-modification* approaches posit rules (or other mechanisms such as lexical inheritance) which add modify the argument-structure specifications of lexical items "in place" (Bouma & van

[9] See Przepiórkowski (1999a), who gives a number of representative quotes from Fowler and Yadroff (1993), Chomsky (1995b), Rizzi (1990), and Williams (1995).

Noord, 1994; Przepiórkowski, 1999a, 1999b). The intuition underlying these approaches is that modification can be handled by mechanisms similar to those sometimes used for handling variable argument realization, such as the transitive and intransitive versions of *eat* or the double-object and prepositional-object variants of *give*.

A second approach to modeling modifier structure is the use of *structural-modification*. Structural-modification approaches treat modification as a fundamentally distinct phenomenon from argumentation. They in turn, fall into two classes. Under, *configurational* approaches to structural-modification arguments and modifiers occupy different structural positions in syntactic or semantic representations. For example, a standard treatment of the argument-modifier distinction in the X′ framework (Chomsky, 1970; Jackendoff, 1977; Kornai & Pullum, 1990) makes use of particular phrase-structure configuration. Arguments (complements) appear as sisters to the head word which selects for them, and they are dominated by a category determined by that head word (i.e., `[XP arg1 ... X ... argN]`). By contrast, modifiers are represented as *adjuncts*, that is, sisters of the phrase which they modify, with a parent node which has the same category as the modified phrase (e.g., `[XP mod1 [XP ... X ...]]`).[10] This configurational distinction mirrors formulations of the distinction in model-theoretic semantics. There, linguistic arguments are treated as *mathematical arguments* to the functions which formalize the meaning of relations. Modifiers, by contrast, are modeled as higher-order functions which take whole relations as mathematical arguments and return an object of the same semantic type as the input (see, Gamut, 1991; Heim & Kratzer, 2000; McConnell-Ginet & Chierchia, 2000).[11]

A second kind of structural-modification approach assumes that arguments and modifiers are composed by different compositional operators—we call these *operation-based* approaches. For example, some approaches to modification in the minimalist tradition have replaced the

[10] In some cases, machinery introduced to capture syntactic differences between arguments and modifiers, has been used to explain more complex constructions, providing an "escape hatch" for movement, explaining quantifier scope generalizations, and deriving linearization patterns (Chomsky, 1986; Kayne, 1994; Kracht, 1999; May, 1985).

[11] However, under neo-Davidsonian (as opposed to Montagovian) approaches to model-theoretic semantics (e.g., Davidson, 1967; Hornstein, 2008; Hunter, 2010; Parsons, 1990; Pietroski, 2005), the meaning of modifier phrases (in fact all phrases) is conjoined into the logical form of the sentence.

configurational view of classical X′-theory with one where modifiers are handled by specialized structure-building operations distinct from (internal or external) MERGE There have been many proposals (e.g., ADJOIN, PAIR-MERGE, LATE-MERGE, CONCATENATE and LABEL, INSERT and MOVE Chomsky, 1993, 1995c, 2000; Fowlie, 2013; Frey & Gärtner, 2002; Graf, 2013, 2014; Hornstein, 2008; Hunter, 2010; Lebeaux, 2000; Stepanov, 2001).[12] An early formal proposal for handling modification using distinct structure-building primitives is found in work from the *tree-adjoining grammars* tradition (Joshi, Levy, & Takahashi, 1975; Joshi & Schabes, 1997). Both mathematical models in this paper are based on proposals originally made in this literature.

Another dimension of variation between theoretical accounts of the argument-modifier distinction concerns the "direction" of the dependency between a modifier and the constituent it modifies. In lexically-driven approaches to grammar, one option for handling modification is to assume that modifier phrases "select" the phrases they modifier, rather than the other way around. This approach is standard in categorial grammar (Bouma & van Noord, 1994; Dowty, 2003; Moortgat, 1997; Steedman, 2000). Under this approach an adjective like *white* selects a noun, like *house* to form the phrase *white house*. However, there are a number of potential problems with these accounts. For example, they cannot predict ordering constraints between modifiers (Cinque, 1999, 2013; Fowlie, 2013), and reverse the traditionally-assumed head-dependent relation of the modified phrase (Dowty, 2003; Fowlie, 2013; Frey & Gärtner, 2002; Graf, 2014; Hunter, 2010). Whether modifiers are selected or selecting remains controversial.[13]

A third and final way in which theories vary in their treatment of the argument-modifier distinction is whether they make a binary or finer-grained distinction between constituent types. Some authors have proposed a three-way distinction (e.g., Briscoe & Copestake, 1999; Culicover & Jackendoff, 2005; Grimshaw, 1990; Kay, 2005; Matthews, 1981), while others propose four-, six-, and even eleven-way classifications—and sometimes even that argumenthood is a gradient property (Arka, 2014; Croft, 2001; Langacker, 1987; Mosel, 2007; Somers, 1984). Theories

[12] There have also been attempts to reduce adjuncts to specifiers (see, e.g., Adger, Pintzuk, & Plunkett, 1999; Roberts, 1997).

[13] For example, subsequent variants of HPSG published in Pollard and Sag (1987) and Pollard and Sag (1994) take opposite sides on this issue.

which posit a multi-way distinction often mix lexical- and structural-modification mechanisms to handle different phrase types (Briscoe & Copestake, 1999; Culicover & Jackendoff, 2005; Grimshaw, 1990; Kay, 2005).

In this paper, we sidestep many of these issues by restricting the focus of our investigation in three ways. First, we consider just those differences between arguments and modifiers which give rise to robust differences in the distribution of constituents in a corpus. Second, rather than giving a precise and detailed treatment of modification (and distinguishing between the multiple potential subtypes of modifiers), we attempt to capture just the notion of "non-argument"—in terms of those core distributional differences. Third, by adopting the simplified view of argument-structure discussed in the previous section, we further restrict the kinds of distributional evidence which we will study. These considerations lead to our focus on three dimensions in which arguments and modifiers differ: **obligatoriness** v. **optionality** and **finiteness** v. **iterability**, and **structural fixity** v. **structural flexibility**.[14]

Though these are not the only dimensions along which arguments and non-arguments differ, they are implicated in nearly every theoretical account in the literature. By focusing on these dimensions, we thus aim to make our results maximally independent of our choice of grammatical formalism, and generalizable to alternative grammatical theories. In the next section, we introduce *sister-adjunction grammars*, an extension of tree-substitution grammars that captures modification.

Sister-Adjunction Grammars

To model modification, we extend tree-substitution grammars by introducing a second structure-building operation, a variant of adjunction. While SUBSTITUTE must be licensed by the presence of an argument node (i.e. an empty nonterminal node at the frontier of an elementary tree), adjunction can insert constituents into otherwise fully well-formed trees. We adopt a variant of adjunction known as *sister-adjunction* which can insert a constituent as the sister to any node

[14] See Graf (2013) for a related approach.

in an existing tree (Chiang & Bikel, 2002; Rambow et al., 1995).[15] Our formalism is strongly equivalent to (unlexicalized) tree-insertion grammar and, therefore, has the same weak generative capacity as context-free grammar (Schabes & Waters, 1995).[16]

In order to derive the complete tree for a sentence, starting from a single nonterminal node of category S (i.e., the start symbol), we recursively sample arguments and modifiers according to the following procedure. For each node f with nonterminal category A on the frontier of our derivation, we perform the following two steps. First, we choose an elementary tree t with category A from our lexicon and, for each position before or after a node on the interior of t, we sister-adjoin into our derivation zero or more new nonterminal nodes, representing modifier phrases. Second, we substitute f—now with modifier category nodes—into the derivation at node n. This process then then repeats on any nonterminal nodes now on the frontier of the tree. In particular, if we have sister-adjoined a modifier node m with category X, its internal structure will be determined recursively—that is, by first choosing an elementary tree of category X from the lexicon substituting it into m.

The SISTER-ADJOIN operation captures the three core ways in which modifiers differ from our simplified formulation of lexical argument structure: (i) The decision to insert or not insert a modifier does not change the well-formedness of a generated structure (**optionality**; see, also, Graf, 2013), (ii) SISTER-ADJOIN can insert any number of modifiers at a position in a derivation (**iterability**) and, (iii) SISTER-ADJOIN can insert a modifier at any position in a constituent (**structural flexibility**).

Figure 2 illustrates two derivations of the same tree, one in a standard tree-substitution grammars without sister-adjunction, and one in our model, which we term *sister-adjunction grammars*. The tree-substitution grammar derivation, at the top of the figure, uses an elementary tree with four leaf nonterminals as the backbone for the derivation. The four phrases filling these arguments are then substituted into the elementary tree, as indicated by arrows. Note that the

[15] Sister-adjunction is also related to the alternative notion of derivation-tree for tree-adjoining grammars defined in Schabes and Shieber (1994).

[16] Our generative process is also closely related to the generative model for tree-adjoining grammars proposed in Chiang (2000).

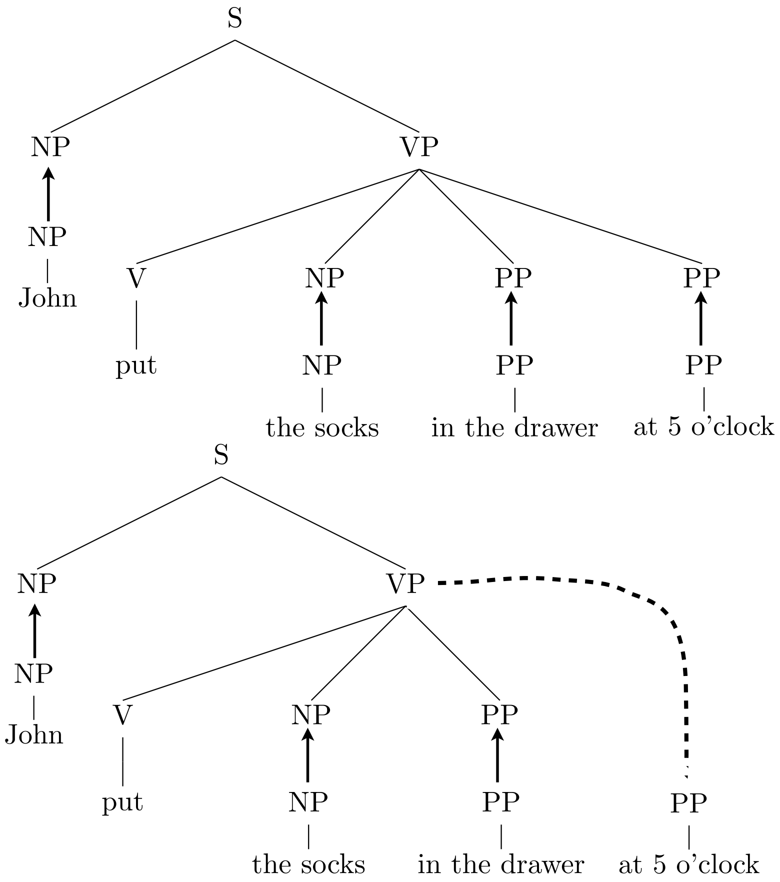


Figure 2. The first part of the figure shows how to derive the tree in TSG, while the second part shows how to use sister-adjunction to derive the same tree in our model.

prepositional phrase, *at 5 o'clock*, which is an intuitive temporal modifier, enters the derivation through an argument node in tree-substitution grammars. However, the sister-adjunction grammar in the lower part of the figure is able to insert the modifier using SISTER-ADJOIN (indicated using dotted lines) and, therefore, uses an elementary tree only three leaf nonterminals as the backbone of this derivation.

Handling Uncertainty

Neither language learners nor linguists have *a priori* knowledge of the structure of the lexicon, or about whether a token instance of a phrase is an argument or modifier. Rather, the set of lexical argument structures in a language must be learned from linguistic input, and the derivation of particular sentences must be inferred on a case-by-case basis. In order to provide broad-coverage statistical evidence for or against the sufficiency of argument-structure-only theories, we must provide a means for solving the problems of learning the lexicon and inferring the derivation of individual sentences. In this paper, we adopt a probabilistic approach to these problems, specifying prior distributions over lexica and sentence derivations for both the argument-only (PTSG) model and the argument-modifier (PSAG) model, and using probabilistic conditioning to infer language-specific lexicons and utterance-specific derivations from input data. Although we make use of probability theory to define our model, our inference framework is closely related to approaches to language learning based on description-length succinctness (e.g., Berwick, 1982, 1985; de Marcken, 1996a, 1996b; A. S. Hsu & Chater, 2010; A. S. Hsu, Chater, & Vitányi, 2011, inter alia) in the minimum description length framework (Grünwald, 2007; Li & Vitányi, 2008; Rissanen, 1978; R. J. Solomonoff, 1964a, 1964b) and can be seen as a formalization of the classical linguistic notion of the *evaluation metric* (Chomsky, 1964, 1975, 1955, 1979, 1951; Goldsmith, 2011; Rasin & Katzir, 2015). In a later section, we give formal definitions of our prior distributions, and algorithms for estimating conditional probabilities. In this section, we give an intuitive overview of the leading ideas behind the framework.

Following earlier work (de Marcken, 1996a, 1996b; Goldwater, 2006; M. Johnson, Griffiths, & Goldwater, 2007; O'Donnell, in press; Perfors, Tenenbaum, & Regier, 2011), we propose that lexicon learning is guided by two prior biases for simplicity (see, also, A. S. Hsu & Chater, 2010; A. S. Hsu et al., 2011, 2013; Phillips & Pearl, 2014). The first provides an a priori measure of the quality of proposed lexica, favoring those with fewer, more reusable lexical items. The second provides an a priori measure of the quality of the proposed derivations of individual sentences, favoring derivations which involve a small number of more probable lexical items. These two biases lead to a tradeoff. For a fixed number of sentences, if we increase the average reusability of lexical items, then we must also increase the average number of lexical items used in any derivation. Likewise, if we decrease the average number of lexical items used per derivation, we must, on average, increase the size of the lexicon. The inference problem is to find a set of lexical items and sentence derivations that best explains the distribution of forms in the input data, subject to these two prior biases.

Note that our two prior biases can be interpreted as the special case that results from applying the usual Bayesian prior/likelihood tradeoff to the problem lexical storage. The preference for more reusable lexical items results from a prior distribution which favors simpler (i.e. smaller) lexica; the preference for smaller derivations results from the likelihood, which favors derivations in which fewer random choices are made.[17] In the two sections below, we provide additional details about the implementation of our models, and intuitions about their behavior when applied to input datasets.

Simplicity Biases and Inference

As we discussed above, our models encodes two simplicity biases. The first is a bias in favor of smaller lexicons. Following Goldwater (2006), M. Johnson et al. (2007), and others, we formalize this bias using a distribution from Bayesian nonparametric statistics known as the *Pitman-Yor Process* (Pitman, 1995). A Pitman-Yor process $\mathrm{PYP}(G_0, a, b)$ can be thought of as a stochastic function (i.e., a function returning random values) with three parameters, G_0, a, b. The first parameter, G_0, is a prior distribution over possible lexical items (i.e., another stochastic function that returns sampled lexical items). In the case of the two models studied in this paper, G_0 will be some prior distribution over tree fragments that can be stored in the lexicon in principle. The other two parameters are real-values such that $0 \geq a \geq 1$ and $b > -a$.

A Pitman-Yor process works as follows. Let N be the number of lexical items sampled so far from $\mathrm{PYP}(G_0, a, b)$, let n_i be the number of times that lexical item i was previously sampled, and let K be the number of distinct lexical items that have been previously sampled (i.e., the number of lexical *types*). The first time we sample from $\mathrm{PYP}(G_0, a, b)$ a new lexical item will be chosen according to G_0, stored internally by the Pitman-Yor process, and returned to the caller. On subsequent invocations, either a previously sampled lexical item i will be returned with

[17] Note that the equivalence between our approach and minimum-description length approaches follows directly from our interpretation of the two simplicity biases as prior and likelihood (see, Grünwald, 2007; Li & Vitányi, 2008). To our knowledge the earliest version of this proposal in the context of lexicon learning was formulated by de Marcken (1996b)

probability $\frac{n_i - a}{N+b}$, or a new lexical item will be sampled from G_0, stored, and returned, with probability $\frac{aK+b}{N+b}$. Notice that these definitions favor smaller numbers of lexical items and induce a rich-get-richer dynamic on lexical item reuse. The more often a particular tree fragment is reused, the more probable it will be.

The second simplicity bias favors derivations for individual sentences that make use of a smaller number of more probable lexical items. In both the argument-only (PTSG) model and argument-modifier (PSAG) model, this bias is captured by our assumption that the probability of a derivation is the product of the probabilities of all the lexical tree fragments it contains. Because probabilities must be numbers between 0 and 1, the probability of a derivation decreases quickly (geometrically) as the number of fragments it contains increases. However, this can be mitigated somewhat if the fragments are highly likely (i.e., have probability close to 1).

Applying these two simplicity biases to the tree-substitution model, we arrive at what we call the argument-only (PTSG) model. Recent work has shown that when probabilistic tree-substitution grammars are induced from corpus data, they can achieve a good performance across variety of tasks (Bod et al., 2003; Cohn et al., 2010; O'Donnell, 2011; Post & Gildea, 2013).

To better understand the inferential behavior of the argument-only (PTSG) model, it is useful to consider a toy example. Figure 3 shows three possible solutions to the problem of inferring the correct set of stored tree fragments for a toy corpus consisting of the sentences.

Row I of Figure 3 shows the result of storing and using only the smallest, most abstract fragments of sentence structure. In this case, each particular fragment will be highly reusable, and the lexicon will be maximally compact. However, the derivations of individual utterances will necessarily make use of many lexical fragments, and, will therefore be quite complex.

Row II of the figure shows the solution at the other extreme. In this case, every utterance is stored in its entirety. This solution will result in extremely large lexicons. However, individual sentences which reoccur in the data will be generable with single reuses of individual lexical items, resulting in very low-cost derivations.

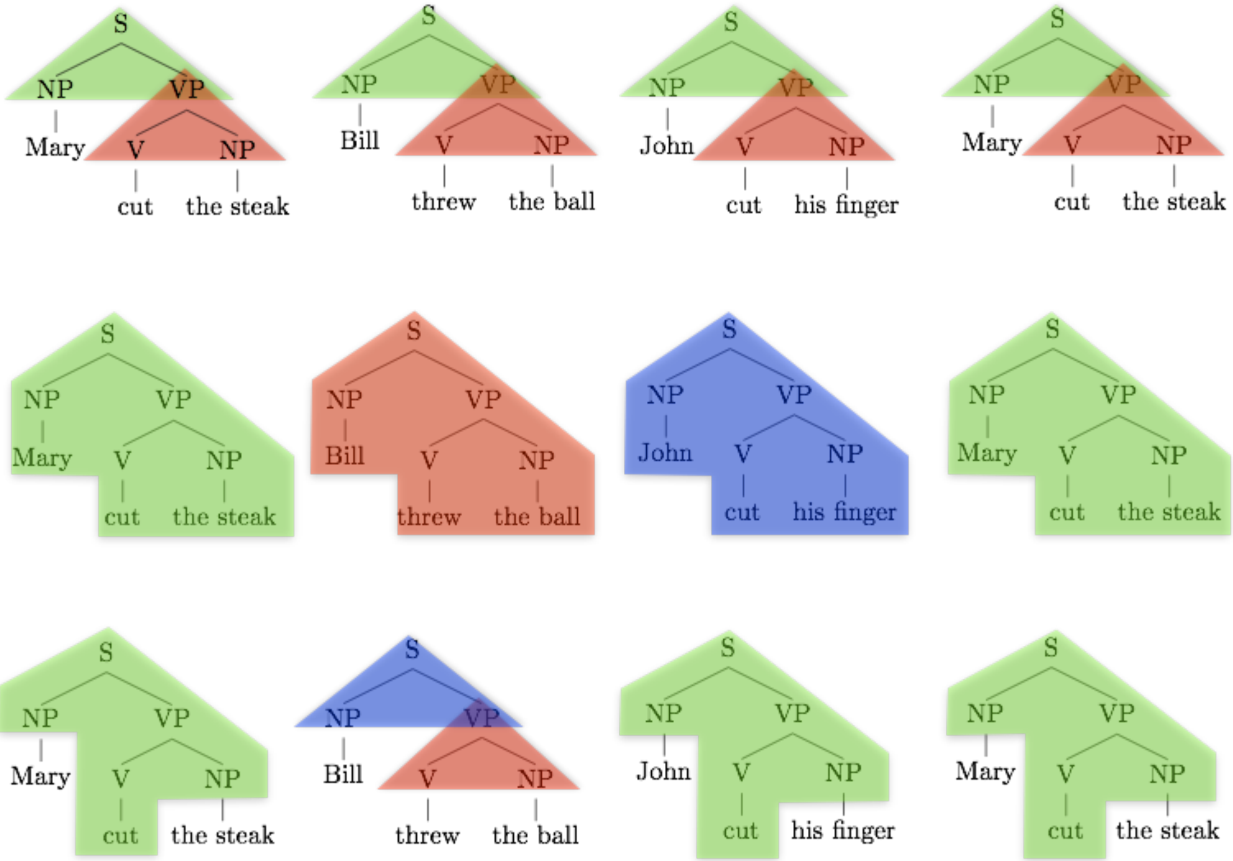


Figure 3. **Inference in the argument-only** (`PTSG`) **model**: This figure shows three possible solutions to the problem of inferring the set of stored lexical fragments and derivations of individual sentences for a toy corpus. See text for discussion.

Row III of Figure 3 shows an intermediate solution which is more optimal with respect to this dataset. By storing lexical fragments which express argument structures of intermediate complexity, this solution produces a more compact lexicon than the solution in Row 12, and simpler derivations than the solution in Row 1, providing a globally better explanation of the input forms. For any particular input data set, the number of such solutions is exponential in the number of nodes in all derivations in the input set. Our search algorithms are designed to find solutions which balance the two simplicity biases of the model.

A parallel pair of simplicity biases is used to define the distribution over modifiers. A Pitman-Yor process prior is used to encode a simplicity bias over the *types* of modifier categories sampled by the model. This prior will bias the model towards using a small set of category types when sampling modifiers. For example, if many previously sampled modifiers are adjectives, then the model will prefer to sample adjectives as modifiers in the future, rather than phrases of a different category (e.g. verbs). A second simplicity bias favors labeling only a small number of a sentence's constituents as modifiers. This bias is captured with the assumption that the probability of deriving a sequence of modifiers is the product of probabilities of the individual modifiers in this sequence. Because this product drops off exponentially in the size of the sequence, the model will (all things being equal) prefer derivations which contain a small number of modifiers.

Applying all of the simplicity biases to the sister-adjunction model, we arrive at the argument-modifier (PSAG) model. During inference, the argument-modifier (PSAG) model will attempt to find an optimal set of reusable argument-structure fragments by categorizing individual nodes in the input data as either (i) internal to a stored tree fragment, (ii) built by substitution, or (iii) built by sister-adjunction. In general, the model will categorize a node as a modifier when doing so will result in a more compact representation of the input corpus, that is, when it allows the input corpus to be explained using a smaller set of elementary trees. Intuitively, the SISTER-ADJOIN operation allows the model to "prune" out sentence structure when doing so will lead to more compact and generalizable lexical items.

Consider Figure 4. If the model posits that there are no modifiers in these sentences, then it will not identify their shared structure, and will derive them using distinct sets of elementary trees, as on the top of Figure 4. On the other hand, if it posits that the PP *before dinner* is a modifier, then it will be able to derive the core structure of these sentences using a single elementary tree, as on the bottom of Figure 4. Nodes will be identified as modifiers when, like this PP, their removal from the sentence's argument structure leads to simpler derivations of the sentences in the corpus and greater amounts of sharing in the lexicon.

Related Modeling Work

Very little earlier computational work has focused directly on distinguish arguments and modifiers. However, work on learning argument structures for verbs (i.e., *subcategorization*

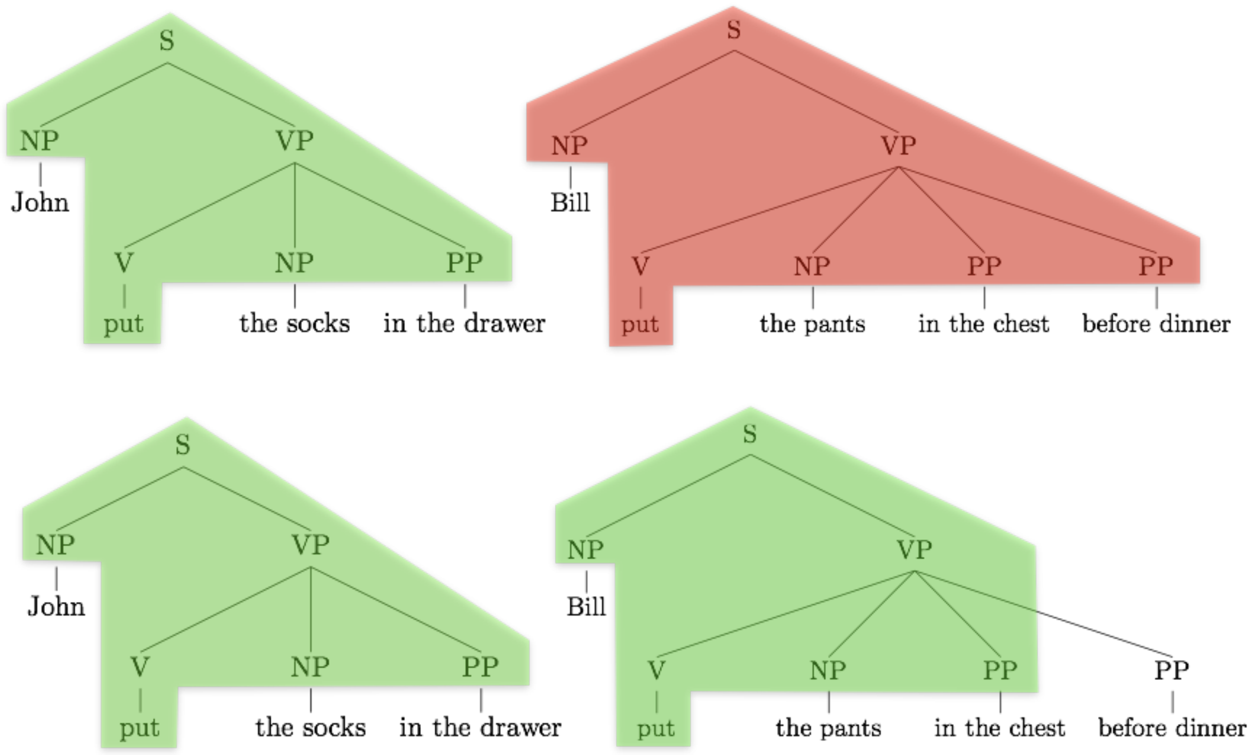


Figure 4. **The argument-only (PTSG) model v. the argument-modifier (PSAG) model**: Both the argument-only (PTSG) model and the argument-modifier (PSAG) model can derive sentences using only lexical fragments (top). However, argument-modifier (PSAG) model can "prune" constituents from derivations by attributing them to SISTER-ADJUNCTION, leading to simpler derivations and more sharing in the lexicon (bottom).

frames) has also faced the problem of determining from an input corpus which constituents should be stored as true verb arguments, and which should be treated as modifiers.[18] Webster and Marcus (1989) propose a deterministic algorithm for inducing verb-argument structures, based on the idea of taking the minimal argument structure for each verb which is consistent with the input data. Stevenson, Merlo, Kariaeva, and Whitehouse (1999), Merlo and Leybold (2001) and Merlo and Stevenson (2001) adopt a supervised approach to learning verb-argument structures based on training classifiers on linguistically inspired, hand-annotated features. Eisner (2002) defines a model of lexical metarules that relate verb subcategorization frames via sequences of transformations, and uses Bayesian methods to learn both verb-specific and verb-general

[18] In general, these earlier models have handled optional arguments via lexical ambiguity. That is, verbs which allow for an optional argument will have two lexical entries, one with, and one without the argument. Pure modification (e.g., via adjunction) is not typically modeled in these frameworks.

mappings between subcategorization frames.

A number of unsupervised approaches for classification of verb-argument structure have adopted a discriminative approach which first considers candidate subcategorization frames from a corpus and then uses statistical tests to filter these frames. Brent (1993, 1994) and Briscoe and Carroll (1997) adopt approaches that first heuristically generate candidate argument structures for individual verbs, and then classify verbs as belonging to particular argument-structure classes using a binomial test. Zeman and Sarkar (2000) and Aldezabal, Aranzabe, Gojenola, Sarasola, and Atuxta (2002), working in Czech and Basque respectively, use a number of other statistics to perform this task such as mutual information, the likelihood ratio test, and Fisher's exact test to classify candidates into arguments and modifiers.

The current work extends previous approaches to unsupervised argument structure learning in several respects. First, the model proposed here provides a uniform framework for learning verbal arguments and other types of arguments (e.g., arguments of prepositions). Second, previous models have been limited to being evaluated on only the most frequent verbs in a corpus. The classification of a verb's arguments in these models relies entirely on observations of that particular verb, and as a result the models require a large amount of training data for each verb that is considered. The argument-modifier (PSAG) model can learn other structural generalizations, for example learning that adjectives are generally modifiers, and is therefore not limited to being evaluated on only the most frequent verbs. Because of this, it will be possible to evaluate the model on a corpus that is two orders of magnitude larger than those that have been previously used (there are $\sim 4 \times 10^4$ sentences in the sections of the Penn Treebank used here, vs. $\sim 5 \times 10^2$ sentences used for evaluation in Zeman & Sarkar, 2000).

Formalization of the Model

Our model, the argument-modifier (PSAG) model, extends earlier work on induction of Bayesian TSGs (Cohn et al., 2010; O'Donnell, 2011; O'Donnell, Snedeker, Tenenbaum, & Goodman, 2011; Post & Gildea, 2009). The model uses a Bayesian non-parametric distribution—the Pitman-Yor Process (Pitman, 1995), to place a prior over the lexicon of elementary trees. This distribution allows the complexity of the lexicon to grow arbitrarily with the input, while still enforcing a bias for more compact lexicons. The model has two components: a distribution over elementary trees, similar to earlier models of Bayesian TSG induction, and a distribution over modifiers. Algorithm 1 provides pseudocode for the model, and can also be interpreted as an alternative, procedural definition of the model. Note that throughout, we will use the notation c_p to refer to the nonterminal label of a node p.

For each node p, the distribution over elementary trees rooted at that node is given by:

$$G_{c_p} | a_{c_p}, b_{c_p}, P_E \sim \text{PYP}(a_{c_p}, b_{c_p}, P_E(\cdot|c_p)) \tag{1}$$

where $P_E(\cdot|c_p)$ is a context free distribution over elementary trees with root label c_p. The hyperparameters a_{c_p}, b_{c_p} are set to $a_{c_p} = 0$, $b_{c_p} = 1$ for this paper.[19]

The context-free distribution over elementary trees $P_E(e|c)$ is defined by:

$$P_E(e|c) = \prod_{i \in I(e)} (1 - s_{c_i}) \prod_{f \in F(e)} s_{c_f} \prod_{c' \to \alpha \in e} P_{cfg}(\alpha|c'), \tag{2}$$

where $I(e)$ is the set of internal nodes in e, $F(e)$ is the set of frontier nodes, s_c is the probability that we stop expanding at a node labeled c, and $P_{cfg}(\alpha|c')$ is the probability of the context-free expansion $c' \to \alpha$. For this paper, the parameters s_c are set to 0.5. The distribution $P_{cfg}(\alpha|c')$ is defined using a distribution that is similar to the Infinite PCFG (Finkel, Grenager, & Manning, 2007; Liang, Petrov, Jordan, & Klein, 2007), which provides a Dirichlet process prior for PCFG rules[20] [21]; a similar base distribution for elementary trees is used in Cohn et al. (2010). The base

[19] Given these parameter values, the prior reduces to a Chinese Restaurant Process prior; however, the model is presented using the Pitman-Yor Process because it is strictly more general, and all of the model definitions are consistent with it.

[20] We use this nonparametric prior so that in addition to learning a distribution over elementary trees, we can also learn a distribution over context-free rules. The inferred distribution over context-free rules may substantially differ from the maximum-likelihood estimate derived from the corpus, as nodes that the model labels as modifiers are not included in the derivation of an elementary tree. This approach is also suitable to the unsupervised setting (as in Cohn et al., 2010), in which the derived trees in the corpus are not observed.

[21] Our base distribution over PCFG rules differs from the Infinite PCFG as presented in Liang et al. (2007) in a

distribution over elementary trees thus will be biased towards small elementary trees which use frequent context-free expansions.

Algorithm 1 Sister-Adjunction Grammar

```
$\beta \sim Dir(1, ..., 1)$                                [draw prior over nonterminals]
for each nonterminal sequence $c_1, ..., c_n$:
    $P_{rhs}(c_1, ..., c_n) = \frac{1}{2^n} \prod_i \beta_{c_i}$         [define base distribution for pcfg prior]
for each nonterminal $c$:
    $P_{cfg}(\cdot|c) \sim \text{DP}(a, P_{rhs}(\cdot))$          [draw distributions over context-free rules]
for each nonterminal $c$:
    for each elementary tree $e$ rooted at $c$:
        $F(e)$ = frontier of $e$, $I(e)$ = interior nodes of $e$
        $P_E(e|c) = \prod_{i \in I(e)} (1 - s_{c_i}) \prod_{f \in F(e)} s_{c_f} \prod_{c' \to \alpha \in e} P_{cfg}(\alpha|c')$
    $G_c \sim \text{PYP}(a_c, b_c, P_E(\cdot|c))$             [draw distributions over elementary trees]

$\theta \sim Dir(1, ..., 1)$                        [draw base distribution over nonterminals]
for each sequence of nonterminals $C = q_l, ..., q_1$:              [draw modifier distributions]
    if length($C$)==1
        $H_C \sim \text{DP}(\alpha, \text{Multinomial}(\theta))$
    else
        $H_C \sim \text{DP}(\alpha, H_{C'})$, where $C' = q_{l-1}, ..., q_1$

for each node $f$ on the frontier of the parse tree:
    $e \sim G_{c_f}$                        [sample an elementary tree rooted at category $c_f$]
    substitute $e$ at $f$
    for each internal node $p$ in e:
        for each argument child $d_i$ of $p$:
            j=1
            $C = c_{d_1}, s_{1,1}, ..., c_{d_i}, c_p$                              [C is the context for $d_i$]
            $s_{i,j} \sim H_C$                            [draw from the modifier distribution for $d_i$]
            while $s_{i,j} \neq STOP$                      [continue until drawing a $STOP$ symbol]
                sister-adjoin a node labeled $s_{i,j}$ between $d_i, d_{i+1}$
                j+=1
                $C = c_{d_1}, s_{1,1}, ..., c_{d_i}, s_{i,1}, ..., s_{i,j-1}, c_p$       [add sampled modifier to the context]
                $s_{i,j} \sim H_C$
```

In addition to defining a distribution over elementary trees, we also define a distribution

number of ways. First, rather than being a hierarchical Dirichlet process model, our set of nonterminal categories is fixed to be equal to the set of nonterminal categories in the treebank. Second, our rules are not fixed to be in Chomsky normal form, but rather the length of the right-hand side of each rule is sampled from a geometric distribution,and each child symbol is drawn conditioned on the parent symbol and then entire left context of the symbol, which is backed-off using the scheme of Teh (2006) andGoldwater, Griffiths, and Johnson (2006).

which governs modification via sister-adjunction. To sample a modifier, we first decide whether or not to sister-adjoin into location l in a tree. Following this step, we sample a modifier category (e.g., a PP) conditioned on the location l's *context*: its parent and left siblings. Because contexts are sparse, we use a backoff scheme based on hierarchical Dirichlet processes similar to the ngram backoff schemes defined in Teh (2006) and Goldwater et al. (2006). Let e be an elementary tree that has been substituted into the parse tree, and let p be an internal node in e. The node p will have $n \geq 1$ children derived by argument substitution: $d_1, ..., d_n$. In order to sister-adjoin between two of these children d_i, d_{i+1}, we recursively sample nonterminals $s_{i,1}, ..., s_{i,k}$ until we sample a STOP symbol:

$$P_a(s_{i,1}, ..., s_{i,k}, STOP|C_0) = (\prod_{j=1}^{k} P_a(s_{i,j}|C_j)) \cdot P_a(STOP|C_{k+1}) \tag{3}$$

where $C_j = c_{d_1}, s_{1,1}, ..., c_{d_i}, s_{i,1}, ..., s_{i,j-1}, c_p$ is the context for the j'th modifier between these children. The distribution over sister-adjoined nonterminals is defined using a hierarchical Dirichlet process to implement backoff in a prefix tree over contexts. Given the context $C = q_l, ..., q_1$ (where $l > 1$), we define the distribution H_C over sister-adjoined nonterminals $s_{i,j}$ by:

$$H_C \sim \text{DP}(\alpha, H_{C'}), \tag{4}$$

where $C' = q_{l-1}, ..., q_1$. A sample is drawn from the root of the hierarchy when the context C is of length 1 (and hence the backed-off context is empty). A Dirichlet-multinomial distribution is used as the prior in this case:

$$\theta \sim Dir(1, ..., 1)$$
$$H_C \sim \text{DP}(\alpha, \text{Multinomial}(\theta))$$

where $C = q_1$ and θ is a vector with entries for each nonterminal and an entry for the STOP symbol. The backoff scheme for sampling modifiers is illustrated in Figure 5.

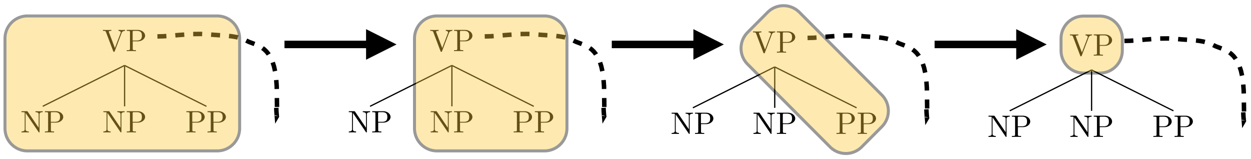


Figure 5. This illustrates the procedure for sampling a modifier at the right edge of a VP. The distribution over modifiers is conditioned on the modifier's context, which contains its VP parent and left siblings, as illustrated on the left of the figure. This distribution is estimated by successively backing off to smaller contexts.

Inference

To perform inference, we developed a local Gibbs sampler which generalizes the one proposed by Cohn et al. (2010). This sampler jointly explores the space of elementary trees and substitution/adjunction attributions for a corpus consisting of parsed sentences. At each iteration of the sampler, each node in the corpus will be annotated with one of three possible structure labels: node that is internal to an elementary tree, node that is the root of a tree which was inserted by substitution, or node that is the root of a tree which was inserted by sister–adjunction. Nodes of the first two types are considered argument nodes, while nodes of the third type are considered modifier nodes. Our Gibbs sampler randomly selects a node, and resamples its structure label from the conditional posterior given the current state of the rest of the corpus and elementary tree set.

Analyses

In this section, we will use the computational models introduced above to evaluate two questions. First, to what extent do the statistics of natural language corpora provide evidence for a distinction between arguments and non-arguments (assuming that lexical and derivational succinctness are desiderata)? Second, how much information do the statistical properties of natural language corpora contain about the distinction between arguments and non-arguments?

In order to address these questions, we will perform two sets of analyses. In the first, we will look at whether the argument-modifier (PSAG) model learns a distinction between arguments and non-arguments which agrees with how linguists have traditionally drawn the distinction, at least in relatively uncontroversial cases.[22] If the argument-modifier (PSAG) model learns to classify arguments and non-arguments in a manner similar to traditional linguistic theories, then this will provide evidence that our formalization of argument-structure and the argument-modifier distinction aligns with linguistic intuitions, and that naturalistic data provide evidence in favor of such a distinction. Moreover, to the extent that the argument-modifier (PSAG) model can recover the distinction between arguments and non-arguments, this will demonstrate that at least some aspects of the distinction are learnable in principle from the distribution of forms in natural language corpora, given suitable prior assumptions about succinctness.

In the second set of analyses, we will examine how well each model is able to describe the linguistic data from the point of view of succinctness. In particular, we will show that the argument-modifier (PSAG) model learns a more succinct representation of the input corpus than the argument-only (PTSG) model. We will also present analyses that illustrate exactly how the argument-modifier (PSAG) model's representational and inferential assumptions interact to lead to this behavior. We will also replicate these results on a set of "heldout" sentences which did not appear in the training corpus, showing that the argument-modifier (PSAG) model learns a lexicon that is able to describe novel data more succinctly than the argument-only (PTSG) model. This result has the important implication that providing mechanisms for handling modification actually increases the amount of information about argument structure that can be extracted from the corpus. This provides further evidence both of the insufficiency of models that attempt to reduce all syntactic structure to lexical argument-structure specifications, and the learnability of models which include additional mechanisms for handling modifier structure.

[22] As discussed above, there is little consensus within the field of linguistics about the precise boundary between arguments and non-arguments in difficult cases. As a result, there does not exist a gold standard corpus which specifies how to draw the distinction in every instance. We will nonetheless evaluate the argument-modifier (PSAG) model using a particular gold standard, which has been developed by a group of computational linguists. The validity of our analyses therefore relies on the assumption that a sufficient number of the gold standard judgments are non-controversial from the perspective of modern linguistic theories. There is reason to hope that this assumption is correct: linguistic theories tend to diverge mostly on the "long tail" of natural language constructions, which represent only a small portion of the sentence tokens in naturalistic corpora (see, e.g., Rimell, Clark, & Steedman, 2009).

Gold Standard Evaluation

Our first set of analyses examine the ability of the argument-modifier (`PSAG`) model to correctly classify constituents as arguments or modifiers. As we discussed above, the model was designed to capture three differences between arguments and modifiers that affect their syntactic distribution: Arguments tend to be obligatory, while modifiers tend to be optional; lexical items specify a small, finite number of arguments, while modifiers to a phrase a can be iterated; and arguments tend to appear in fixed structural positions with respect to their selecting lexical items, while modifiers tend to exhibit more flexibility in where they can appear. If the argument-modifier (`PSAG`) model is able to correctly distinguish modifier and argument phrases in the training corpus, we can conclude that naturalistic language input contains significant distributional evidence that can be brought to bear on the problem of identifying the argument-modifier status of individual constituents.

The model was trained on sections 2–21 of the Wall Street Journal portion of the Penn Treebank (Marcus, Santorini, Marcinkiewicz, & Taylor, 1999). The input consisted of approximately 40,000 parsed sentences, without any further annotations for argumentation or modification. It is important to note that under the Penn Treebank's tree annotation style, arguments and modifiers are not distinguished from each other by their hierarchical relations in the parse tree. In particular, the arguments and modifiers of a phrase are most often siblings in the tree. This contrasts with, for example, X' theory in which modifier phrases are typically the siblings of an argument's *parent*. A consequence of the Penn Treebank's flat annotation style is that the argument-modifier (`PSAG`) model could not use the hierarchical relations in the input corpus to simply read off each sentence's argument and modifier structure.

In order to evaluate the accuracy of the argument-modifier (`PSAG`) model classification of arguments and modifiers, we require a gold standard which provides annotations for arguments and modifiers in the Penn Treebank. Unfortunately, no currently available resource provides a gold-standard argument-modifier classification of all nodes in the Penn Treebank (Marcus et al., 1999). However, for a subset of the phrases in the Penn Treebank, such information is available in the PropBank corpus (Palmer, Kingsbury, & Gildea, 2005) which provides annotations of argument and modifier structure for all of the verbal predicates in the Wall Street Journal portion of the corpus. For example, the sentence *The next morning, with a police escort, busloads of executives and their wives raced to the Indianapolis Motor Speedway.* contains a single verbal predicate, *raced*. PropBank annotates all of the phrases in the sentence which are either arguments or modifiers of this verb. In this case, PropBank indicates that the phrases *busloads of executives and their wives* and *to the Indianapolis Motor Speedway* are arguments of the verb, while *the next morning* and *with a police escort* are modifiers.[23] PropBank does not annotate the arguments or modifiers of expressions which are not verbal predicates. For example, in the noun phrase *a police escort*, PropBank does not annotate the noun *police* as a modifier of the head noun *escort*.

Our first evaluation will use PropBank to assess the model's performance at classifying modifiers. During training, the argument-modifier (PSAG) model infers a label for every nonterminal node in the corpus, where the label indicates whether the node is a) internal to an elementary tree, b) an argument node at the leaf of an elementary tree, or c) a sister-adjoined node. More precisely, following each iteration of the Gibbs sampler (described above), the model has assigned a label to every nonterminal node in the corpus. After a sufficient number of iterations of the Gibbs sampler, the labels assigned to these nodes represent a sample from the posterior distribution over argument-modifier assignments to the phrases in the corpus. Our model evaluations were performed by running the Gibbs sampler for 100 iterations, and selecting the node labelings which were output on the final iteration.

For the purpose of our analyses, all sister-adjunction nodes are classified as modifiers, and all other nodes (i.e. nodes which are internal to an elementary tree or at the leaf of one) are classified as non-modifiers. We compared the model's labels to those provided by PropBank, on the subset of nodes for which PropBank provides annotations.

To show that differences in the distributions of argument and modifier phrases provide a

[23] As noted in Palmer et al. (2005), the annotation of modifiers in PropBank is non-standard in certain cases: "Although they are not considered adjuncts, NEG for verb-level negation ... and MOD for modal verbs ... are also included in this list [of verbal adjuncts] to allow every constituent surrounding the verb to be annotated." As a result, modifiers labeled NEG and MOD are excluded from our analyses.

valuable source of evidence for lexicon acquisition, we must establish that our model is able to correctly classify phrases at a rate which is better than chance. To demonstrate this, we computed the precision (i.e. number of correctly identified modifier nodes / total number of modifier nodes identified by the model) and recall (i.e. number of correctly identified modifier nodes / total number of modifier nodes in the gold-standard) of the model and compared it with two baselines. The first baseline randomly classifies each node as internal to an elementary tree, the leaf of an elementary tree, or a modifier with equal probability. Note that prior to receiving any training data, the model has no information about which phrase types are likely to be modifiers and which are likely to be arguments. The random baseline therefore represents the model's knowledge of the argument/modifier distinction prior to learning, and any improvement in the model's classification of modifiers must be attributed to information contained in the input data.

The second baseline treats every node as a modifier. We introduce this baseline in order to illustrate some basic statistics about PropBank. Table 1 shows that PropBank annotated 179,058 nodes in the corpus for their argument/modifier status; these nodes represent approximately 10% of the corpus. Among the annotated nodes, 45,507 (25%) are modifiers, meaning that 25% of the guesses of the all-modifier baseline are correct.

The precision and recall of the argument-modifier (PSAG) model and the baselines are shown in Table 1. Precision measures accuracy of modifier-predictions. The argument-modifier (PSAG) model is significantly more accurate than the random and the all-modifier baselines, demonstrating that the training data has provided information which allows the model to correctly classify many constituents.

Recall measures the coverage of gold-standard modifier nodes achieved by the models. Again, the argument-modifier (PSAG) model achieved significantly higher coverage than the random baseline, indicating that the training data contains enough information to increase the number of true modifiers that the model recognizes.

In order to better understand what the argument-modifier (PSAG) model learned about the modifiers of verbal predicates, the evaluations against PropBank were further broken down by the

Table 1
***Precision and Recall of the argument-modifier (*`PSAG`*) model:** This table shows precision and recall in identifying the modifiers of verbal predicates in the corpus. The argument-modifier (*`PSAG`*) model is compared to three baselines: an all-modifier baseline, in which every node is labeled as a modifier, a random baseline, and a version of the model that does not use context to predict modifiers.*

Model	Precision	Recall	#Guessed	#Correct Guesses	#PropBank Modifiers
All-modifier	0.25	1	179,058	45,507	45,507
Random	0.29	0.23	35,764	10,262	45,507
SAG	**0.66**	**0.52**	**36,045**	**23,698**	**45,507**

category of the modifier. Table 2 shows the results for the phrase types which occur most frequently as verbal modifiers: adverb phrases (`ADVP`s), noun phrases (`NP`s), prepositional phrases (`PP`s), and subordinate clauses (`SBAR`s). Together these categories of constituent account for more than 85% of the modifiers in the training corpus.

Table 2
This table shows labelings for modifiers of VP nodes, broken down by child category.

VP Parent				
Child Category	Model	Precision	Recall	PropBank
ADVP	Random	0.95	0.23	12,385
ADVP	SAG	0.95	0.47	12,385
NP	Random	0.04	0.23	3,345
NP	SAG	0.47	0.57	3,345
PP	Random	0.49	0.22	18,841
PP	SAG	0.56	0.54	18,841
SBAR	Random	0.40	0.22	4,552
SBAR	SAG	0.84	0.63	4,552

For the phrase categories of adverb phrases (`ADVP`s) and prepositional phrases (`PP`s), the model doubles its recall over the random baseline, and roughly maintains its baseline precision. Adverb phrases (`ADVP`s) are typically modifiers (out of 13,197 `ADVP`s annotated by PropBank, 12,384 are modifiers) when they appear within a verb phrase (`VP`) and prepositional phrases

(PPs) are frequently modifiers when they appear in this setting (out of 38,861 PPs annotated by PropBank, 18,839 are modifiers). The increase in the model's recall therefore indicates that the model learned to correctly classify many of these ADVP and PP modifiers.

In contrast to adverb and prepositional phrases, noun phrases (NPs) which appear within verb phrases are typically arguments to the verb. Out of 92,965 NPs annotated by PropBank, only 3,306 appear as modifiers. Exceptions to this generalization are cases where a noun phrase is used as an adverbial modifier, such as the noun phrase *last night* in *They played the game last night*. The precision of the model increased by a factor of 10 for NPs, indicating that it incorrectly classified many fewer non-modifier NPs. In addition, the model's precision more than doubled over the baseline, showing that among the few NPs annotated as modifiers by PropBank, the model learned to correctly classify a greater number as modifiers.

Phrases belonging to the category of subordinate clauses SBAR can serve either as arguments or modifiers. For example, in the sentence *John said that he would be late*, the subordinate clause *that he would be late* is an argument of the verb *said*. In contrast, in the sentence *The woman laughed when she heard the joke*, the clause *when she heard the joke* is a temporal modifier of the verb *laughed*. Out of 13,617 SBAR phrases annotated by PropBank, 4,551 are modifiers. The model's precision and recall on SBAR phrases was more than twice that of the random baseline, showing that the model classified fewer clausal arguments as modifiers, and correctly identified a greater number of clausal modifiers.

As we mentioned above, certain categories of constituents have highly stereotyped argument-modifier status when they appear as children of other categories. For example, adverb phrase (ADVP) children of verb phrases (VP) and adjective phrase (JJ) children of noun phrases (NP) are both typically modifiers of their parent constituents. Although PropBank only provides argument-modifier annotations for the children of verb phrases (VP nodes), it is possible to use the stereotyped behavior of these categories to examine the model's performance on the children of non-VP nodes. Tables 3–6 show the model's classification of constituents which were children of sentence-level constituents (S), prepositional phrases (PPs), noun phrases (NPs), and subordinate clauses (`SBAR`s), respectively. In each of these cases, the category of child constituents is highly indicative of their argument-modifier status. Nevertheless, we emphasize that these results are only suggestive.

For sentence-level (`S`) constituents, we analyzed three categories of child phrase: noun-phrases (`NP`s), verb-phrases (`VP`s), and (`ADVP`s). These are the three most common categories which have stereotyped argument/modifier behavior when they appear as children of (`S`) nodes. Of these three phrase types, noun and verb phrase (`NP`s and `VP`s) are not typically modifiers, whereas adverb phrases (`ADVP`s) are. For example, in *Usually, John wears a coat*, the adverb *Usually* is a modifier of the sentence while *John* and *wears a coat* are not modifiers. Table 3 shows how often the model labeled the children of sentence-level (`S`) constituents as modifiers nodes. The model accords with intuition here, most often labeling adverb phrases (`ADVP`s) but not noun or verb phrases (`NP`s or `VP`s) as modifiers.

Table 3
Labelings for modifiers of S nodes.

S Parent				
Child Category	Model	#Guessed	Corpus Total	Typically Modifier
ADVP	Random	1,393	6,063	Y
ADVP	SAG	2,331	6,063	Y
NP	Random	16,654	93,076	N
NP	SAG	1,738	93,076	N
VP	Random	16,005	89,984	N
VP	SAG	572	89,984	N

For prepositional phrases (`PP`s), we considered four categories of child constituent: adverb phrases (`ADVP`s), noun phrases (`NP`s), prepositions (`IN`s), and the infinitival *to* (`TO`s). Of these phrase types, only adverb phrases (`ADVP`s) typically modify the parent prepositional phrase. For example, in the prepositional phrase *immediately after the opening*, the adverb phrase *immediately* is a modifier while the preposition *after* and noun phrase `the opening` are not. In accord with these intuitions, Table 4 demonstrates that the model classifies most adverb phrase

(ADVP) children of prepositional phrases as modifiers, but treats prepositions (IN and TO) and noun phrases (NPs) as non-modifiers.

Table 4
Labelings for modifiers of PP nodes.

PP Parent				
Child Category	Model	#Guessed	Corpus Total	Typically Modifier
ADVP	Random	216	1,109	Y
ADVP	SAG	547	1,109	Y
IN	Random	13,972	83,848	N
IN	SAG	672	83,848	N
NP	Random	15,060	88,556	N
NP	SAG	496	88,556	N
TO	Random	1,484	8,654	N
TO	SAG	64	8,654	N

We considered four categories of subconstituents for noun phrases (NPs): determiners (e.g., *the*, *a*; DT), adjectives (JJ), other noun phrases, and prepositional phrases (PPs). Determiners (DTs) are unlikely to modify noun phrases, while adjectives (JJs) typically do modify them. For example, in the noun phrase *the big chair*, the determiner *the* is not a modifier, while the adjective *big* modifies the noun *chair*. Prepositional phrases are often modifiers (e.g., in *the resort by the sea*, the prepositional phrase *by the sea* modifies the noun *resort*), although in some cases, such as deverbal nominalizations, they are typically treated as arguments of the head noun (e.g., in the noun phrase *the destruction of the city*, the prepositional phrase *of the city* is an argument of the head noun; see, e.g., Chomsky, 1970).

Table 5 shows the modifier-classification rates of noun phrase children. The model correctly identifies determiners (DTs) as non-modifiers. However, for JJs (adjectives), the most prototypical modifiers of noun phrase, the model's performance is weaker: The number of JJs classified as modifiers is approximately the same as the random baseline. The number of PPs classified as modifiers decreased by more than half relative to the random baseline, though the implications of this are unclear: As discussed above, PPs appear frequently as the modifiers of noun phrases but also as arguments.

Table 5
Labelings for modifiers of NP nodes.

NP Parent				
Child Category	Model	#Guessed	Corpus Total	Typically Modifier
DT	Random	15,791	77,553	N
DT	SAG	1,701	77,553	N
JJ	Random	10,544	45,812	Y
JJ	SAG	9,717	45,812	Y
PP	Random	7,652	43,420	Y
PP	SAG	3,226	43,420	Y

The category `SBAR` is used to mark subordinate clauses in the Penn treebank. Here we consider the following categories of children: sentence-level constituents (`S`s) and wh-expressions (`WHADVP`s and `WHNP`s) which are used to introduce subordinate clauses (e.g., the word *when* in the sentence *The woman laughed when she heard the joke*). None of these types of constituent is typically thought of as modifying subordinate clauses. Table 6 shows, consistent with this intuition, that the model treats all three categories as non-modifiers.

Table 6
Labelings for modifiers of SBAR nodes.

SBAR Parent				
Child Category	Model	#Guessed	Corpus Total	Typically Modifier
S	Random	4,873	29396	N
S	SAG	101	29396	N
WHADVP	Random	421	2521	N
WHADVP	SAG	38	2521	N
WHNP	Random	1,383	8505	N
WHNP	SAG	79	8505	N

Discussion

We have presented two sets of results in this section. First, we have shown that the argument-modifier (PSAG) model's accuracy at classifying arguments and non-arguments substantially improves over a random baseline (which represents the model's *a priori* knowledge about the argument/modifier distinction) when evaluated on the PropBank gold standard. Second, we have shown that among phrases that are not classified by the gold standard, the argument-modifier (PSAG) model learns an argument/non-argument classification which appears linguistically reasonable on most major phrase categories.

These results have two consequences for the arguments in this paper. The argument-modifier (PSAG) model is built on the assumption of three distributional differences between lexical argument-structure-derived phrases and modifier phrases: **finiteness** v. **iterability**, **obligatoriness** v. **optionality**, and **structural fixity** v. **structural flexibility**. Since the argument-modifier (PSAG) model made use of these properties in order to classify phrases in the input corpus as arguments or non-arguments, its performance on the gold standard shows that this formalization of argument and non-argument composition corresponds to a linguistically natural distinction.

The results also show that the distributional information contained in the input corpus is sufficient for recovering of the distinction between arguments and non-arguments in many cases. The argument-modifier (PSAG) model does not have any *a priori* knowledge about which types of phrases are likely to be arguments, and it leverages only distributional information in order to infer the status of individual phrases. Thus, its performance in categorizing arguments and non-arguments must be attributable to the distributional information contained in the corpus. Although these results do not demonstrate that children use the same types of distributional information while they are learning their native language, they do, however, provide evidence that this information would be sufficient to learn the argument/non-argument distinction up to the level of accuracy achieved in these evaluations, provided that the learner makes succinctness assumptions similar to the model.

Lexicon Learning, Arguments Structure, and Succinctness

In the previous section, we showed that the argument-modifier (PSAG) model is able to correctly recover the modifier status of many constituents using only the pattern of co-occurrences between (type of) constituents in the training set. Like any implemented computational model, the argument-modifier (PSAG) model makes a number of specific assumptions. However, our gold-standard results do not reveal which aspects of the model are crucial to its performance. Since we ultimately would like to draw inferences about the representation and learning of natural languages, it is important to establish which modeling assumptions are crucial, and how they give rise to the results above. In this section, we report the results of a number of simulations and analyses designed to explore these issues.

As we discussed above, we believe there are two kinds of crucial assumptions underlying the argument-modifier (PSAG) model. First, the model makes several representational assumptions. In particular, it models a distinction between arguments and modifiers with three dimensions of difference : **obligatoriness/optionality**, **finiteness/iterability**, and **structural fixity/structural flexibility**. Second, the model makes a number of inferential assumptions. In particular, it assumes prior biases that favor small lexicons, highly reusable lexical items, and short derivations of individual forms. In this section, we show how these two kinds of assumptions interact to allow the argument-modifier (PSAG) model to use the distribution of constituents in the input to correctly identify many modifiers, learning a compact, generalizable lexicon along the way.

Intuitively, we show that because it is able to "ignore" modifiers, the argument-modifier (PSAG) model learns a more compact, generalizable lexicon, while also providing simpler derivations for individual forms. Consider a verb phrase (VP) headed by a verb like *put*. In in simplest form, *put* requires two VP-internal arguments—a noun phrase (NP) expressing the object which was put somewhere, and a prepositional phrase (PP) expressing the destination—*put his socks in the suitcase*. Across particular uses of this simple *put*-construction, the VP node will reliably have the following sequence of children: V NP PP. However, because modifiers are optional, iterable, and appear at a variety of positions within a constituent, they can greatly increase the number of different observed sequences of children in a corpus: *put his socks suddenly in the suitcase* [V NP ADVP PP], *put his socks in the suitcase suddenly without warning* [V NP PP ADVP PP], etc. However, the argument-modifier (PSAG) model is able to explain away the presence of these additional phrases using the SISTER-ADJOIN operation, and is driven to do so because this leads to a lexicon of argument structure fragments and a set of derivations of individual forms which better optimizes the tradeoff between lexicon and derivation complexity we discussed in a previous section.

In order to demonstrate this point, we must show that the inclusion of modifier structure in our model allows it to account for the data with a more compact lexicon and simpler derivations of each sentence. To do this, we compare the argument-modifier (PSAG) model to the “lesioned” argument-only (PTSG) model, which does not include modification, and show that the argument-modifier (PSAG) model is able to explain the training data using fewer, more reusable lexical items, and simpler derivations of individual forms. We also compare the performance of the argument-modifier (PSAG) model and argument-only (PTSG) model on a set of *held-out* sentences, which the models did not observe during learning. We perform this comparison in order to evaluate the generalizability of the induced grammars, showing that the argument-modifier (PSAG) model learns a grammar which generalizes better to this held-out data.

In all of the analyses in this section, we used two corpora to compare the argument structures learned by the argument-modifier (PSAG) model and argument-only (PTSG) model: the Wall Street Journal portion of the Penn Treebank, and the Brown (1973) portion of the CHILDES database MacWhinney (2000). For the WSJ, the model was trained and evaluated on the 40,000 parsed sentences from sections 2-21 (the same sentences that were used in the gold standard analyses). The CHILDES sections used here consist of approximately 30,000 child-directed utterances which were recorded between ages 1;6 to 5;1. Sentence fragments and *wh*-questions were excluded from our analyses, though the results do not differ substantially when fragments and questions are included.

The training regime was the same as in the gold standard analysis: The models received parse trees for each sentence as input. Because the CHILDES database does not provide parses, we used the corpus of parsed CHILDES sentences developed by Pearl and Sprouse (2013). In order to build this corpus, Pearl and Sprouse (2013) first used a statistical parser (the Charniak parser)[24] to find approximate parses for each sentence. Each initial parse was then corrected by a pair of trained annotators. These parse trees follow the Penn Treebank guidelines, so the WSJ and CHILDES corpora examined below were annotated in a consistent manner.

We include CHILDES in our analyses below because it is more like likely than newspaper text to be representative of the input received by a typical natural language learner learner.[25] The differences between the two corpora can be illustrated by several simple statistics. On average, the sentences in the WSJ corpus contain 25 words, while the sentences in CHILDES contain 6.5 words. The parse trees in the WSJ contain 71 nodes on average, while those in CHILDES contain 19 nodes. Finally, the average depth (i.e. the longest distance from the root node to a leaf) of the parse trees in the WSJ is 10, while the average depth in CHILDES is 5. These statistics show that the sentences in the WSJ are significantly longer and more syntactically complex than those in CHILDES.

Compression of the Training Set. In this section, we compare the ability of the argument-modifier (PSAG) model and the argument-only (PTSG) model to explain the input training data for the WSJ and CHILDES corpora. We first examine the bias for reusable lexical items. Figure 6 shows the frequencies of the $1,000$ most common stored tree fragments in the lexicons of the argument-modifier (PSAG) model and argument-only (PTSG) model, as learned on the CHILDES (left) and WSJ (right) training sets. We computed these values by first ranking the tree fragments by the frequency of their occurrence in the lexicon; this resulted in a rank for each type of tree fragment, with lower rank corresponding to greater frequency. Then, for all tree fragments below a given rank (e.g., for the tree fragments below rank 100, corresponding to the

[24] Available at ftp://ftp.cs.brown.edu/pub/nlparser/ (23 April, 2015.)

[25] Note that we did not include the CHILDES corpus in our gold-standard evaluations in the previous section because PropBank does not provide annotations for this corpus.

100 most common tree fragments), we computed the sum of the frequencies of these fragments.[26] The figure shows that the commonly used stored tree fragments learned by the argument-modifier (PSAG) model are used more often across sentences in the training corpus. The difference is more pronounced in the WSJ training set, most likely due the greater sentence complexity and greater number of modifiers in newspaper text compared to child-directed speech.

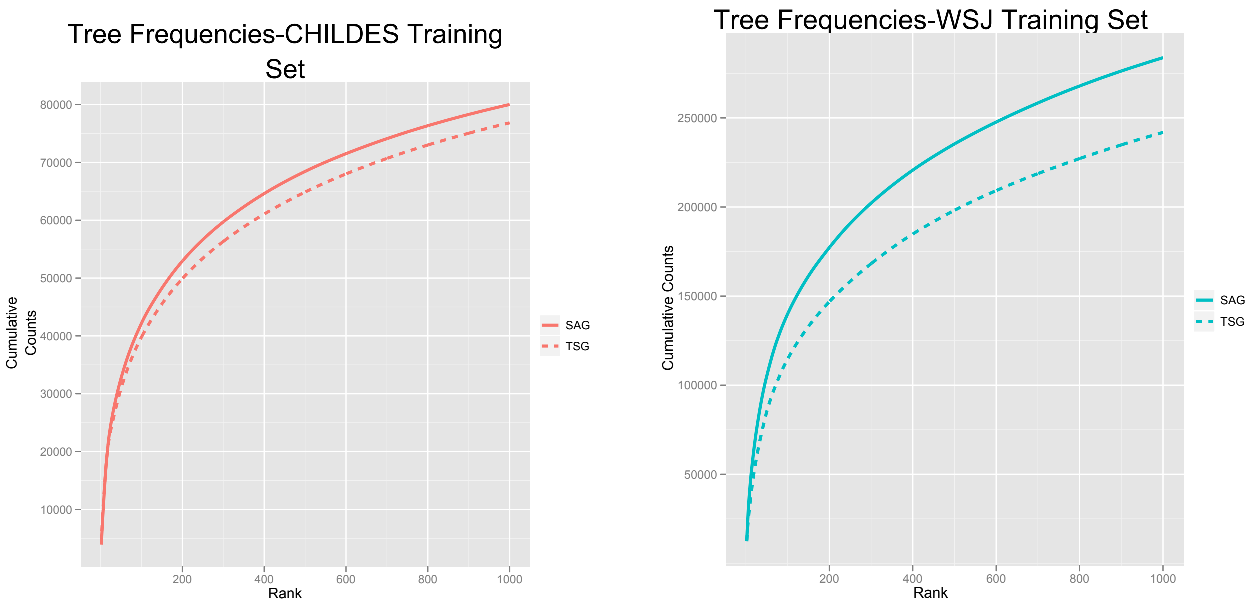


Figure 6. **Cumulative Frequencies**: The cumulative frequencies of the $1,000$ most common stored tree fragments learned by the tree-substitution grammar and sister-adjunction grammar models when trained on the CHILDES (left) and WSJ (right) training data sets. The sister-adjunction grammar model learns tree fragments which are more reusable in the training corpus.

We next examine which model was able to provide simpler derivations of individual sentences. One way to measure this, is to look at the complexity of stored tree fragments learned by each model. If a model stores tree fragments which are larger (on average), then it must account for each sentence using fewer fragments (on average). Figure 7 shows the average number of nodes (left) and average depth (right) of the $1,000$ most common elementary trees

[26] Tree fragments which were rooted at part-of-speech nodes were excluded from this and subsequent analyses. A subtree which is rooted at a part of speech necessarily consists of exactly two nodes (the part of speech and the terminal string which it is a parent of). As a result, there is only one way to parse such a subtree into tree fragments, and both models will always parse such a subtree in an identical manner.

learned by the argument-modifier (PSAG) model and the argument-only (PTSG) model on the CHILDES and WSJ corpora. These figures show that the elementary trees learned by the argument-modifier (PSAG) model are more complex than those learned by the argument-only (PTSG) model, and therefore that the derivation of individual sentences involve fewer lexical items (on average). The difference in tree fragment complexity is greater for the WSJ corpus than for CHILDES, most likely because of the greater complexity of the sentences in the WSJ: The parse trees for these sentences contain a greater number of nodes and have greater depth than those in CHILDES.

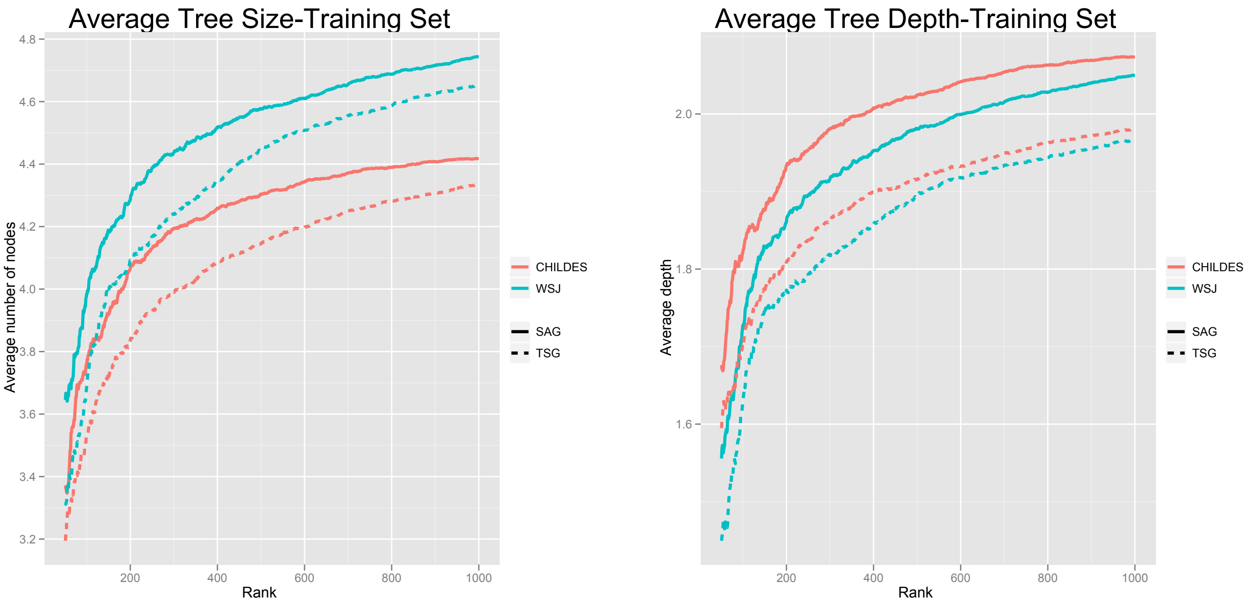


Figure 7. **Complexity of Stored Tree Fragments**: This figure shows the cumulative average number of nodes (left) and cumulative average depth (right) of the $1,000$ most common stored tree fragments learned by the sister-adjunction grammar and tree-substitution grammar models. The sister-adjunction grammar model accounts for sentences using larger trees, and, therefore, simpler derivations.

The preceding analyses indicate that the sister-adjunction model is able to learn both more reusable lexical items, and simpler derivations of each sentence than the tree-substitution model. This result is surprising. As we discussed previously, the inference performed in learning the set of lexical fragments for the argument-only (PTSG) model can be understood in terms of a tradeoff. All else being equal, smaller tree fragments are more reusable, leading to smaller lexica.

However, larger tree fragments lead to simpler derivations, since fewer are needed per derivation. Given a particular corpus, there is some set of fragments which optimizes this tradeoff.[27] If, in a particular instance, the argument-only (PTSG) model is at or near such an optimum, it can increase the reusability of some lexical item only by decreasing the simplicity of some derivation, or vice versa.

However, the present results indicate the surprising fact that the sister-adjunction model is able to learn a set of stored fragments which is more optimal by *both* measures. This is confirmed in Figure 8 which shows the proportion of nodes in the training corpus which are accounted for by the 1,000 most common stored tree fragments learned by the argument-modifier (PSAG) model and the argument-only (PTSG) model. Because it learns both more reusable *and* larger stored tree fragments, the argument-modifier (PSAG) model is able to account for the training data using a smaller number of stored items.

To understand these results better, consider the example sentences in Figure 4. The argument-modifier (PSAG) model is able to use a single elementary tree (stretching from the root S node to the verb *put*) to derive the core of both sentences. In contrast, as Figure 4 shows, the argument-only (PTSG) model will require two distinct elementary trees, one with three arguments under the VP node (for the first sentence) and one with four arguments (for the second). Thus, because the argument-only (PTSG) model can compose an optional PP such as *at 5 o'clock* separately from a sentence's core argument structure, it can re-use the same elementary tree to derive a greater number of tree configurations that appear in the corpus. This explains how the argument-modifier (PSAG) model can use its sister-adjunction operation to find more reusable elementary trees than the argument-only (PTSG) model. It is driven to do so by the its prior preference for a smaller lexicon.

The explanation of how the argument-modifier (PSAG) model is able to find more complex elementary trees (and, therefore, simpler derivations) can also be illustrated by Figure 4. The common structure shared by the two sentences is greater from the perspective of the

[27] In general, there may be more than one such optimal set of fragments.

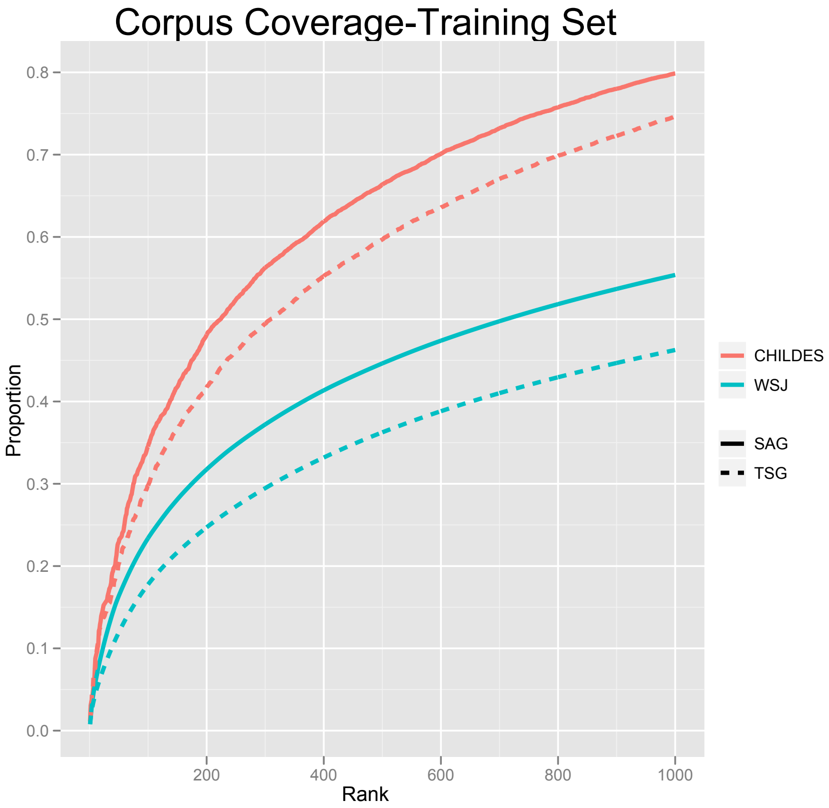


Figure 8. **Cumulative Coverage**: This figure shows the cumulative proportion of nodes in the CHILDES and WSJ corpora that are accounted for by the $1,000$ most common elementary trees. The sister-adjunction (SAG) model provides a more compressed representation of input data by simultaneously preferring more reusable lexical items and simpler derivations.

argument-modifier (PSAG) model than from that of the argument-only (PTSG) model; the argument-modifier (PSAG) model will posit a deeper elementary tree that extends from the root node to the verb *put* to explain this shared structure, while the argument-only (PTSG) model will use an elementary tree that only extends from the root to the VP node. Because the argument-modifier (PSAG) model can ignore extraneous structure such as the modifier PP, it will tend to find elementary trees which are deeper but narrower than those found by the argument-only (PTSG) model.

Figure 9 illustrates a representative example from the corpus. By using SISTER-ADJOIN to account for the ADVP node separately from the rest of the sentence's derivation, the argument-modifier (PSAG) model model was able to use a common depth-three elementary tree to

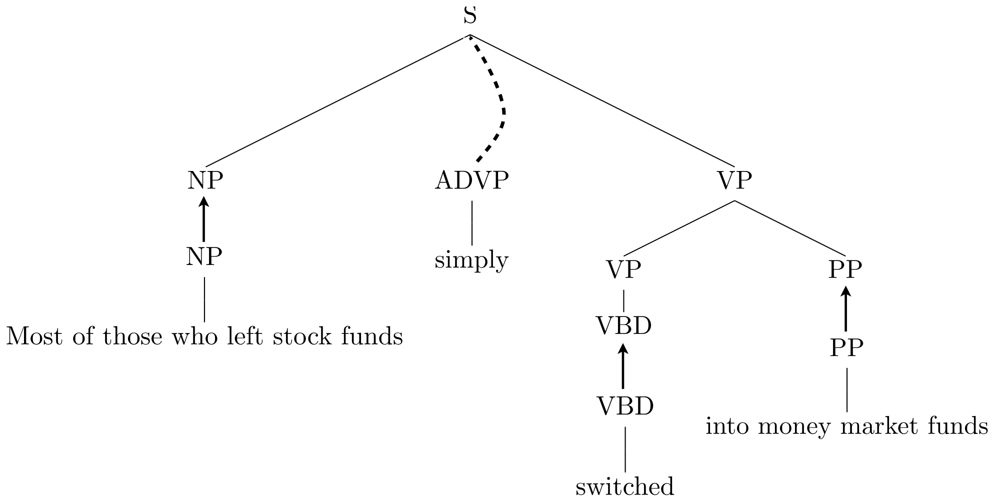


Figure 9. **Example Derivation from the WSJ Corpus:** This figure shows a portion of a derivation tree discovered by the sister-adjunction model.

derive the backbone of the sentence. By contrast, the argument-only (PTSG) model must include the ADVP node in an elementary tree; this elementary tree is much less common in the corpus.

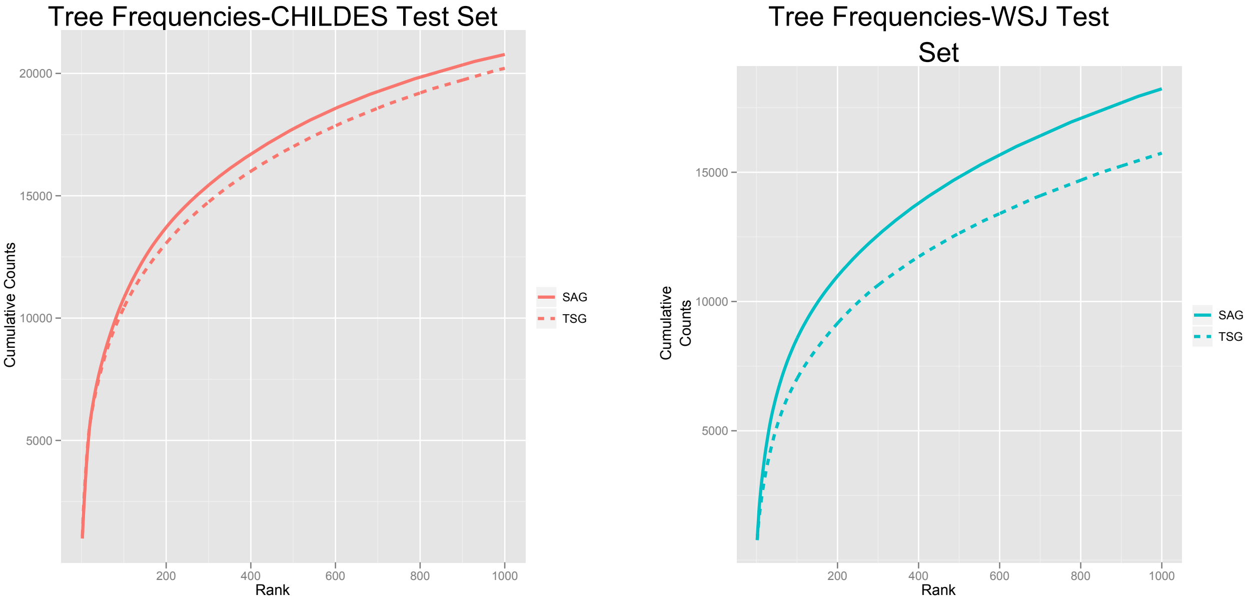
These analyses illustrate how the behavior of argument-modifier (PSAG) model leads it to find a more succinct representation of the input, and also illustrate the interaction between representational and inferential assumptions of the model. The model seeks lexical items which are both reusable and account for a large proportion of each derivation in which they are used. It can "ignore" modifiers when doing so leads to a better explanations of more the more stable aspects of syntactic structure. In short, the representational assumptions of the argument-modifier (PSAG) model allow it to hypothesize a more parsimonious theory of the input, while its inferential assumptions drive it to do so.

Compression of New Sentences. The previous analyses demonstrate that the argument-modifier (PSAG) model learns a more parsimonious representation of the input than the argument-only (PTSG) model. An important caveat, however, is that the argument-modifier (PSAG) model is a more complex grammatical formalism than the argument-only (PTSG) model. Whereas the argument-only (PTSG) model only has a single composition operation (SUBSTITUTE), the argument-modifier (PSAG) model has two composition operations (SUBSTITUTE and SISTER-ADJOIN). This means that the model has more degrees of freedom in explaining an input training set. As a result, it is possible that the argument-modifier (PSAG) model's performance does not reflect a true property of natural language in general, but rather, is an artifact of the model's greater complexity and properties particular to the input corpora. The general problem, known as *overfitting*, is that an overly complex model of a domain will often provide a better fit to observed data than the correct (simpler) model; for example, in a regression, a data set of n points can always be fit perfectly by an n-degree polynomial, even if the true relationship between the variables is much simpler (e.g., linear). A standard method to diagnose overfitting is to evaluate the model on novel data. If the model was too complex for the domain, then it will have captured spurious regularities in its input data, and will therefore generalize poorly to data outside of this input. In the present case, we ask whether evidence in favor of modification from the training corpora generalizes to a novel test set.

In order to determine whether the parsimony advantages of the argument-modifier (PSAG) model generalize to novel data, we divided the CHILDES and WSJ corpora into training and test portions. The training portion was used as input to the argument-modifier (PSAG) model and argument-only (PTSG) model, while the test portion was used for evaluating the generalizability of these grammars. For the WSJ corpus, we used the standard split: training on sections 2–21 and testing on section 23. For the CHILDES corpus, we randomly selected 80% of the sentences for training, and used the remaining 20% for test.

Our evaluation of the argument-modifier (PSAG) model follows the method in the previous sections. We compare the argument-modifier (PSAG) model to a "lesioned" argument-only (PTSG) model, and conduct similar analyses of fragment reusability and derivation complexity (fragment size). In order to perform these analyses, we applied our sampler to the test portions of the two corpora without incorporating any new tree fragments into the set of learned tree fragments. That is, after training we "froze" the set of lexical fragments (and associated counts) and did not allow any learning from the test set during inference. Thus each sentence in the test corpus was analyzed as if it were the "next" observed sentence after training. The analyses below are otherwise identical to those in the previous section.

We first examine the bias for reusable lexical items. Figure 10 shows the frequencies of the $1,000$ most common tree fragments from the lexicons of the argument-modifier (PSAG) model and the argument-only (PTSG) model, as inferred on the CHILDES (left) and WSJ (right) test sets. The figure shows that the commonly stored tree fragments learned by the argument-modifier (PSAG) model are used more often across sentences in the test corpus. The difference is again more pronounced in the WSJ test set due the greater sentence complexity and number of modifiers in newspaper text compared to child-directed speech.


Tree Frequencies-CHILDES Test Set

Tree Frequencies-WSJ Test Set

Figure 10. **Cumulative Frequencies (Generalization)**: The cumulative frequencies of the $1,000$ most commonly used fragments used in our generalization test sets from the CHILDES (left) and WSJ (right) corpora. The sister-adjunction grammar model learns tree fragments which are more reusable in these test corpora.

We next turn to the bias for simple derivations of individual sentences. As in the previous compression analyses, we measure derivation complexity by examining the size of tree fragments used to account for test sentences. Larger tree fragments imply fewer fragments per derivation. Figure 11 shows the average number of nodes (left) and average depth (right) of the $1,000$ most common elementary trees used to account for the new sentences by the argument-modifier (PSAG) model and the argument-only (PTSG) model models on the CHILDES and WSJ test corpora. These figures show that the elementary trees learned by the argument-modifier (PSAG) model are more complex than those learned by the argument-only (PTSG) model and, therefore, that the derivations of individual sentences are simpler.

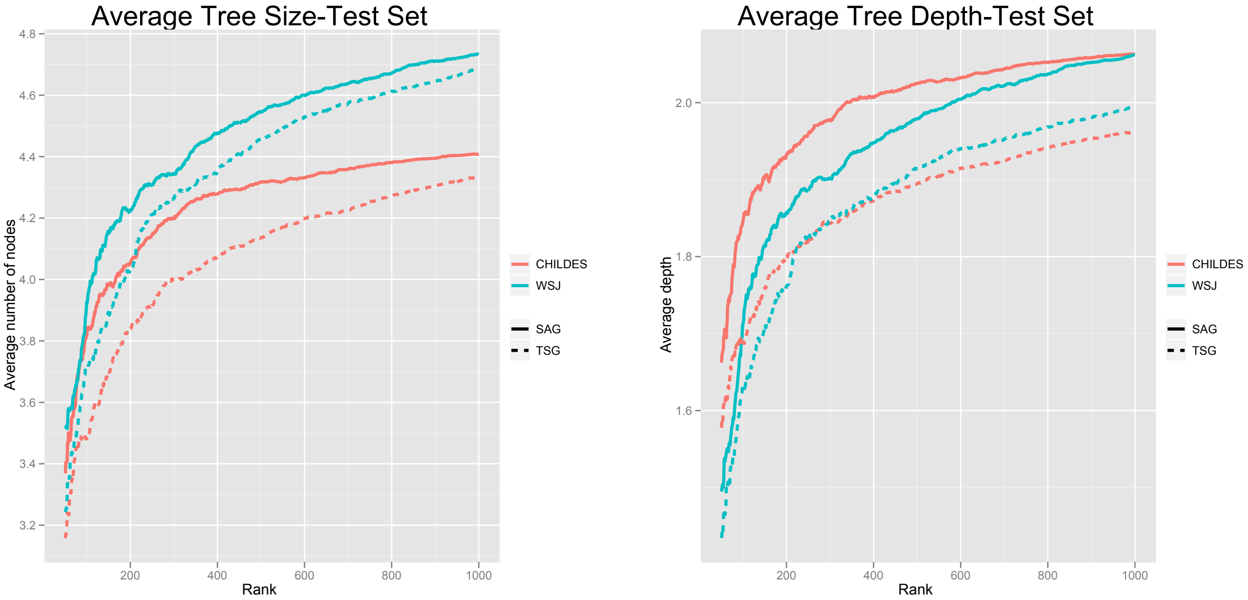


Figure 11. **Complexity of Stored Tree Fragments (Generalization)**: This figure shows the cumulative average number of nodes (left) and cumulative average depth (right) of the $1,000$ most common tree fragments used by the argument-modifier (PSAG) model and argument-only (PTSG) model to account for the novel sentences in the test data. The argument-modifier (PSAG) model accounts for sentences using larger trees, and, therefore, simpler derivations.

The preceding analyses indicate that the argument-modifier (PSAG) model is able to learn both more reusable lexical items, and simpler derivations of each sentence than the tree-substitution model. As we discussed above, this result indicates the surprising fact that the argument-modifier (PSAG) model is able to account for the data with a set of fragments which are both more reusable, and lead to simpler derivations. The current analyses show that this result obtains for new sentences as well. This is confirmed in Figure 12 which shows the proportion of nodes in the training corpus that are accounted for by the 1,000 most common stored tree fragments learned by the two models.

Discussion. The results in this section show that the argument-modifier (PSAG) model provides a more parsimonious explanation of both the input data and novel generalization data

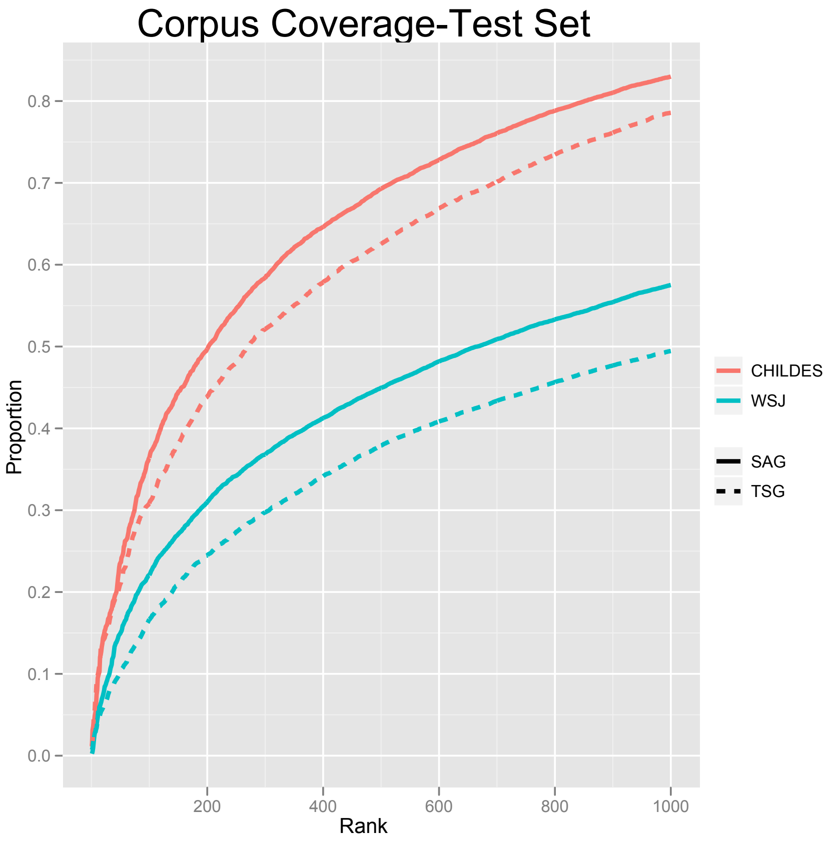


Figure 12. **Cumulative Coverage (Generalization)**: This figure shows the cumulative proportion of nodes in the CHILDES and WSJ test corpora that are accounted for by the $1,000$ most common elementary trees. The argument-modifier (PSAG) model provides a more compressed representation of input data by simultaneously preferring more reusable lexical items and simpler derivations.

than the argument-only (PTSG) model. These results demonstrate that both inferential biases of the model—the preference for simpler derivations of sentences, and for a smaller lexicon—guide how the model distinguishes arguments from non-arguments in the input corpora. The bias for simpler derivations resulted in the argument-modifier (PSAG) model discovering lexical items which were larger than those found by the argument-only (PTSG) model. The bias for a smaller lexicon resulted in the argument-modifier (PSAG) model discovering lexical items which were used more frequently than those found by the argument-only (PTSG) model. It is surprising that the model achieved both improvements simultaneously. Reducing the complexity of derivations typically requires using larger, more complex lexical items, and more complex lexical items will generally be less reusable; conversely, reducing the size of the lexicon (by using a smaller set of lexical items more frequently), typically requires using simpler lexical items, which should increase the complexity of sentence derivations.

The results show, however, that the addition of the SISTER-ADJOIN operation allowed the argument-modifier (PSAG) model to find both simpler sentence derivations and a simpler lexicon. The argument-modifier (PSAG) model achieved this result by using the SISTER-ADJOIN operation to "normalize" the trees in its lexicon. As illustrated in Figure 4, the argument-modifier (PSAG) model uses SISTER-ADJOIN to remove extraneous nodes from derivations, resulting in lexical items which occur more frequently and which extend deeper into the parse trees.

The generalization analyses in this section validate the model's simplicity biases and the use of non-argument composition. The simplicity biases in the model guide it to minimize the complexity of the lexicon and the derivations of the sentences in the training corpus. The generalization analyses show that the structure discovered through the use of these biases persists in a data set of novel sentences. This provides evidence that the model's simplicity biases are appropriate for language learning in this domain. The analyses also show that the use of non-argument composition (as represented by the SISTER-ADJOIN operation) allowed the model to learn a lexicon which generalizes better to novel sentences. This provides evidence for the linguistic reality of multiple types of composition in language: distinguishing arguments and non-arguments allowed the model to find reusable structure in the input which it could not find otherwise.

General Discussion

In this paper, we have studied argument-modifier structure, an important but historically troublesome aspect of linguistic theories. We have provided a new source of evidence in this domain, arguing (i) that (a least some cases of) the distinction can be formulated in terms of a distinction between lexical and non-lexical modes of composition, (ii) that there is statistical evidence in favor of models which draw a distinction of this form, and (iii) that argument-modifier structure is learnable from distributional evidence, up to at least the achieved level of accuracy. These results have implications for both the scientific theories of grammar and the engineering of systems of grammar learning.

Our arguments for these conclusions had several parts. First, we outlined and argued for the *representational* and *inferential* assumptions underlying the model. Our representational assumptions idealized the argument-modifier distinction in terms of lexical/extralexical modes of composition. We argued that lexicon can be idealized as a finite list of finite lists of arguments.[28] Under such a conception, there are three core properties of lexical argument structure that have consequences on the distribution of natural language constituents: **finiteness**, **obligatoriness**, and **fixity of placement**. We used *tree-substitution grammars* to capture these core properties (Bod, 1998). Since traditional notions of modifierhood often invoke the negation of one or more of these properties: **iterability** (i.e., non-**finiteness**), **optionality** (i.e., non-**obligatoriness**), and **structural flexibility** (i.e., non-**structural fixity**), we suggested modification might be defined simply as an extralexical mode of composition (at least for many canonical cases). To formalize this idea, we extended tree-substitution grammars with an additional structure-building operation, known as SISTER-ADJOIN (Chiang & Bikel, 2002; Rambow et al., 1995), which captures these three negative properties. The resulting formalism is known as *sister-adjunction grammar*. These constituted the representational assumptions behind our model.

Turning to our inferential assumptions, for any particular dataset, there are always infinitely many lexica consistent with the input, and exponentially many sets of derivations. In particular, there exist tree-substitution or sister-adjunction grammars over the same base system that are either consistent or inconsistent with any finite sample of sentences. Because the true lexicon[29] is not known in advance, we must adopt some convention for choosing (or at least ranking) hypothesized lexica and (sets of) derivations, given a dataset (i.e., we require an *evaluation metric*

[28] Of course, as we mentioned, this idealization only holds for languages which mark argumenthood primarily in terms of word-order. However, it is straightforward to generalize this to languages which make use of other mechanisms, such as case marking, by allowing each position in a list to index a particular morphological form, rather than relative position.

[29] More accurately, the "true" lexicon is never in the model class. Thus, it is the lexicon most closely approximating the ground truth given its representational limitations.

over grammars. Chomsky, 1964, 1975, 1955, 1979, 1951; Goldsmith, 2011; Rasin & Katzir, 2015). As we outlined in a previous section, we adopt a standard approach to this problem, and formalize the quality of each hypothesized grammar as a tradeoff between two prior biases which favor (i) favor smaller lexica with more reusable lexical items, and (ii) derivations of individual forms using a smaller number of more likely choices (i.e., a smaller number of more probable lexical items). Although we implement these biases using Bayesian machinery, they are instances of a much more general approach to inference (e.g., Berwick, 1982, 1985; de Marcken, 1996a, 1996b; Grünwald, 2007; A. S. Hsu & Chater, 2010; A. S. Hsu et al., 2011; Li & Vitányi, 2008; Rissanen, 1978; R. J. Solomonoff, 1964a, 1964b). Applying these ideas to tree-substitution and sister-adjunction grammars resulted in the two formal, probabilistic models we studied in this paper: the argument-modifier (PSAG) model and the argument-only (PTSG) model.

Turning to our empirical results, we first demonstrated that the argument-modifier (PSAG) model was able correctly recover the modifier status of many constituents, when evaluated against a gold-standard. Since the gold-standard data was produced and annotated by English speakers, this result provides evidence that our models matches with linguistic intuition and psychological reality. Our second set of empirical results demonstrated that the argument-modifier (PSAG) model also provided a more parsimonious explanation of the input data than the argument-only (PTSG) model, on both child-direct and adult speech and both for the training data, and on novel, generalization data. Surprisingly, the argument-modifier (PSAG) model surpasses the argument-only (PTSG) model both in terms of lexical compression and simplicity of derivations, reaching a global level of parsimony which is unachieved by the argument-only (PTSG) model. Since the argument-only (PTSG) model is a special case of the argument-modifier (PSAG) model, and both models share a parsimony bias, this result demonstrates that the input training set is more consistent with the representational assumptions of the argument-modifier (PSAG) model than the argument-only (PTSG) model. These analyses also demonstrate how and why the model behaves the way it does. The model treats constituents as a modifiers when doing so will lead to larger and more reusable lexical items.

Taken together these results indicate that the argument-modifier (PSAG) model provides a more parsimonious account of the input data that accords with well with psychological reality. Therefore, we conclude that there is statistical evidence in naturalistic linguistic input for a lexical/extralexical version of the argument-modifier distinction. Furthermore, since the inferential tools we used to reach these conclusions could also be used by natural language learners, we also conclude that argument-modifier structure is learnable, at least in part, from distributional evidence. Our results provide a lower-bound on the amount of information provided by this type of evidence; it is likely that more sophisticated models will be able to extract more information about the argument-modifier distinction from the input data.

In the remainder of this discussion, we examine the assumptions behind and implications of these results in more detail. We first address a number of small-scale assumptions implicit in this work. We then examine the generality and implications of our two main conclusions: that these results provide evidence for a certain kind of representational distinction between arguments and adjuncts, and that they also imply the learnability of such a distinction. Finally, in the last section of the discussion we turn to our assumptions about succinctness, which underlie all of the models and simulations in this paper.

Practical Issues

There are a number of assumptions we made in this work primarily for reasons of expedience. We consider these in this section.

Recall that both the argument-only (PTSG) model and argument-modifier (PSAG) model models take as a parameter a specification of the set of possible node labels (i.e., terminal and non-terminal labels) which can be used to define lexical tree fragments. In our simulations, these were simply read off the set of categories in the training treebanks. Furthermore, all simulations were conditioned on the input trees from the various training corpora. In other words, the models only had to learn to classify nodes as arguments, lexical-item-internal nodes, or modifiers.

To what degree do our results depend on the choice to condition the model on the corpus input trees? There are actually two distinct questions here. First, to what degree is the information in the treebank markup consistent with psychological reality (or, less ambitiously, consistent with linguistic theories/analyses)? Second, to what degree does providing obviously unobservable information (such as bracketing and node category labels) affect the behavior of the learning algorithm? To illustrate the difference, suppose that we somehow had access to “ground truth” parse trees, i.e. people’s true psychological representations of constituency structure. This would resolve the first question—by assumption, the representation of the training data coincides with people’s psychological representations in this case—but would not necessarily resolve the second question. It is possible that, by using the (correct) bracketing information for the corpus during training, the model would learn an argument/modifier distinction which is different from the one that people acquire—people must simultaneously infer bracketing information and argument structure during acquisition. We address each of these questions in turn.

The annotations of the Penn Treebank are widely believed to be linguistically deficient in many respects (see, e.g., Dickinson, 2005; Hogan, 2007; Vadas & Curran, 2007). Despite these concerns, the Penn Treebank annotations can still be profitably used to study the computational problems associated with language learning. In a number of other NLP tasks, state-of-the-art methods rely on models or gold standard corpora which are linguistically deficient. For example, speech recognition systems typically use n-gram models in order to model the likely input of the speaker. Though these models and corpora are deficient in many respects, they capture enough of the relevant structure of language to be useful in certain tasks. The Penn Treebank has been used to train models that perform a variety of engineering tasks, e.g. translation, relation extraction, and question answering (Gildea & Palmer, 2002; Harabagiu et al., 2000; Yamada & Knight, 2001). The ground truth solutions in each of these tasks are defined independently of the treebank annotations. For example, translation accuracy is typically evaluated by comparing the translations performed by a model to those provided by professional translators. Because the annotations from the Penn Treebank are leveraged to improve performance on these independent tasks, this provides evidence that the Treebank is capturing linguistically meaningful structure in the corpora, and gives a *prima facie* argument for the use of Penn Treebank annotations in studying the argument/modifier distinction. So long as these annotations capture a reasonable approximation of the ground truth linguistic linguistic structure in the input, then it seems likely that analyses which use these annotations will be meaningful (even if they could be improved by more accurate annotations).

Largely the same issues apply to our use of the PropBank corpus as a gold standard for our evaluations. These evaluations are only valid under the assumption that the modifier annotations found in PropBank accurately capture some aspects of the actual linguistic knowledge possessed by speakers of English. During the development of PropBank, two trained annotators independently marked each sentence in the corpus, and all annotation conflicts were resolved by a professional linguist Palmer et al. (2005). The annotators achieved an agreement rate of above 90%, indicating that the task used relatively unambiguous criteria for classifying arguments and modifiers. These criteria were orthogonal to those used by the argument-modifier (PSAG) model: annotators used the semantic role of a phrase in order to determine whether it was an argument or modifier, in contrast to the argument-modifier (PSAG) model, which used purely syntactic criteria. The fact that the argument-modifier (PSAG) model learned an argument/modifier distinction which coincided to a substantial degree with PropBank provides evidence that both sets of criteria were picking out real structure in the data.

The second question regarding our use of the Penn Treebank is whether conditioning on unobservable category and bracketing information distorts our results. More specifically: does the model learn a substantially different argument/modifier distinction than it would if it did not condition on this information? There are a number of possible reasons that this could happen. First, it may be the case that bracketing structure is not learnable from the type of distributional information that children receive during development. Of course, children do in fact learn the bracketing structures in their language, but it may be the case that additional information is necessary, e.g. strong innate biases or semantic information. If bracketing structure (or a related type of latent syntactic structure) is not learnable from distributional information, then it is not clear how the argument-modifier (PSAG) model could learn an argument/modifier distinction; the objects labeled by the argument-modifier (PSAG) model are the nodes in parse trees, and in the absence of these parse trees, there is no structure for the model to label. There have been a number of theoretical results showing that certain classes of context-free grammars, and other syntactic formalisms, are learnable from positive examples, given only limited computational resources (Chater & Vitanyi, 2007; Clark, 2013; A. S. Hsu et al., 2011; D. J. Hsu, Kakade, & Liang, 2012). Though these results are encouraging, it is not clear to what extent their modeling assumptions match the constraints on actual human learners, and therefore to what extent these results imply that inferring bracketing information should be possible for these learners. Moreover, there do not exist implemented engineering systems that can learn bracketing information (or related types of latent syntactic structure) at close to human-level accuracy. Our results therefore should be given a conditional interpretation: if bracketing information can be learned from distributional evidence, then an argument/modifier distinction can be learned (at least) to the level of accuracy achieved by the argument-modifier (PSAG) model.

A second possible issue with this modeling assumption is that a fully unsupervised model might learn a different set of phrase category labels, which would in turn lead it to make different generalizations about the types of phrases that are modifiers. As an extreme example, consider a model which does not learn a unified adjective category, but instead divides adjectives into a number of distinct categories. Such a model would not be able to express the generalization that adjectives tend to be modifiers, and as a result its inferred boundary between arguments and modifiers would likely be different than the current model's. In order to evaluate scenarios like this, we propose defining a model which can perform joint inference over syntactic categories, hierarchical structure, and argument-modifier structure. Such a model would be relatively straightforward to define as an extension of the current model. As noted above, there are substantial engineering challenges involved in implementing such a model, but algorithmic and hardware improvements may make this feasible in the near future.

Implications for Representation

Our results indicate that a lexical/extralexical variant of the argument-modifier distinction can account for a number of intuitively plausible instances of modification, as well as provide significant gains in parsimony on a realistic linguistic corpus. However, we hasten to emphasize that we are not claiming that our account explains or unifies all phenomena which have been invoked in discussions of modification in the literature. As we already discussed, there are many aspects of syntactic (and semantic) structure which we have made no attempt to capture (e.g., binding, displacement, agreement, semantic composition, etc.). Any modifier-related phenomena that require such machinery are beyond the scope of the present study. However, as far as we are aware, none of the discussions in the literature is *inconsistent* with the idea that the argument-modifier distinction is fundamentally about lexical structure in the sense we adopt here. Indeed, there is evidence that some of the properties of modifiers which are not treated in this paper are closely linked to the extra-lexical status of modification. For example, (Graf, 2013) argues that the non-extractability of modifier-internal material (Huang, 1982; Ross, 1967; Truswell, 2007) can be derived in a theory-general way from the properties of **optionality** and **independence**—a property which generalizes **iterability**. That is, he provides a mathematical characterization that explains modifier-islands using very similar principles to those that we propose.

In fact, we note that analyses of extraction are motivated by the same considerations of lexical argument structure and parsimony that motivated the present study. For example, in a sentence such as *what*$_i$ *was John hoping that Peter would explain* $\_\_i$, the identification of *what* as an extracted element follows from the fact that it is an argument of the of the embedded verb, *explain*. The addition of machinery to handle extraction (e.g., a MOVE or I-MERGE operator) to the formalism should only lead to a reduction in ambiguity in the lexicon—that is, a more succinct explanation of the input data. This, in turn, should lead to improved identification of modifiers, for two reasons. First, greater sharing between lexical items should further highlight those constituents that do not fill argument roles. And, second, phenomena such as modifier-islandhood, which result from the interaction of modification and extraction, should provide an additional source of evidence to the learner about the status of individual phrases (i.e., if a phrase cannot be extracted from, this provides evidence that it is a modifier).

This last observation raises another important point. In our framework, the argument-modifier status of a particular constituent must necessarily be inferred on a case-by-case basis. This is a logical necessity since modifierhood is defined with respect to argumenthood, and argumenthood, in turn, is defined with respect to the lexicon—whose contents cannot be known *a priori*. Of course, once a stable lexicon has been learned, a constituent which appears in some position within some syntactic configuration will typically be assigned argument or modifier status with a high degree of certainty. However, the nature of the system itself leaves open the possibility that, in some cases, a constituent may remain ambiguous. We mentioned above cases such as instrumental prepositional phrases and passive *by*-phrases (Kay, 2005; Schütze, 1995) which have proven difficult to classify. We suggest that these cases have been difficult to classify because they are, in fact, ambiguous due to their distributions in the input. If this is the case, approaches such as the one adopted here may be valuable in predicting which case are difficult and quantifying their degree of ambiguity.[30]

This brings us to a more basic question: What kinds of grammatical architectures are consistent with our results? As we reviewed earlier in the paper, there is a large variety of formal mechanisms which have been proposed to handle variability in argument realization, modification, or both. These include additional structure-building operations (like our SISTER-ADJOIN), the transformational mappings of early proposals, higher-order generalizations about dependent types such as X′-theory (i.e., *specifier*, *adjunct*, *head*), syntactic and semantic decomposition where various arguments and modifiers are selected by different (possibly null) functional elements, lexical rules which can add or delete arguments, lexical inheritance hierarchies, higher-order phrase-structure rules including mechanisms such as Kleene-star, and

[30] We also note that there is a tendency in linguistics to reject theoretical mechanisms when they fail to make exceptionless or nearly-exceptionless predictions. This is understandable given that the field has not, by and large, adopted tools which would allow it to explain degrees of ambiguity that result from ambiguity over derivations.

others. One of the fundamental motivations of this work is to try to abstract away from the details of particular proposals.

We have argued that our results will generalize to other theories which localize the difference between arguments and modifiers to a difference between lexically licensed variability and extralexical variability. Recall that we argued that variability in argument structure, such as the case of verb alternations (e.g., *John gave Mary the book*/*John gave the book to Mary*), is fairly limited, and, from the point of view of its effect on the pattern of co-occurrence of constituents, can be captured by lexical ambiguity (although this may be a poor model for other purposes). The crucial property of our treatment of modification is that it allows phrases to be inserted into a constituent with far more flexibility than argumentation, predicting a far greater variety of constituent structures. To the extent that an alternative grammatical model represents these distributional consequences of the argument/modifier distinction, we posit that it will identify argument structures similar to those found by the argument-modifier (`PSAG`) model so long as it is paired with similar assumptions about succinctness. If a model aims to identify a small set of linguistic structures which provide concise derivations of the input sentences, then its inferences regarding argument structures will be driven by similar distributional factors as the argument-modifier (`PSAG`) model, that is, it will use argument structures to explain relatively fixed, systematic co-occurrence of different phrase types in the input.

Learnability

Our second conclusion from this work was that the performance of our model also indicates that the input training data contains a considerable amount of distributional information which an appropriately constituted learner could use in both learning the lexicon, and in determining the argument/modifier status of individual constituents.

To avoid confusion, we will emphasize several claims that we are *not* making. First, we are not claiming that the learner can somehow discover the existence of a distinction between arguments and modifiers from distributional information. Our model assumes such a distinction *a* *priori*. It is rather the argument-modifier status of individual (token) constituents which it infers from distributional evidence, and it can only do so because the possibility of such differences are implied by its assumptions about grammatical computation.

Second, although we are advocating the utility of distributional evidence, we are by no means claiming that distributional evidence is the only kind of information available to the learner, or that it is sufficient on its own for identifying all instances of the argument-modifier distinction. There are numerous other semantic and syntactic correlates of the distinction; we don't doubt that these sources of evidence are often useful and sometimes necessary.

Third, although we provide evidence that such distributional evidence is available for the appropriately equipped learner, we do not claim to present evidence that real children actually use such information. However, we do not believe that children's use of such evidence is implausible, and a number of experimental findings support this contention (see, e.g., Fernandes, Marcus, Nubila, & Vouloumanos, 2006; Fisher, Gertner, Scott, & Yuan, 2010; Gertner, Fisher, & Eisengart, 2006; Yuan, Fisher, & Snedeker, 2012)

Finally, we have only provided evidence of learnability assuming a particular set of succinctness biases. These biases determine how the learner ranks different theories of the input data; theories which are themselves simpler, and which provide simpler derivations of the input data, are ranked higher. Our results do not provide evidence about the type of argument/modifier distinction that will be learned in the absence of these simplicity assumptions. We turn to these assumptions in the next section.

Succinctness Assumptions

There are an infinite number of grammars which are logically consistent with the input that the learner receives. We have assumed throughout that the learner selects from these grammars by ranking them according to the succinctness with which they explain the input. The use of succinctness as a criterion for theory selection has been widely discussed in philosophy, statistics, and cognitive science, both with respect to its normative justification as well as its appropriateness for describing human psychology. In statistics, perhaps the most general treatment comes from the theory of Solomonoff induction, which uses a distribution over the set of all possible computer programs to define simplicity preferences similar to those used in this work (de Marcken, 1996a, 1996b; Grünwald, 2007; A. S. Hsu & Chater, 2010; Li & Vitányi, 2008; Rissanen, 1978; R. Solomonoff, 1978; R. J. Solomonoff, 1964a, 1964b): theories (i.e. computable distributions over observations) are preferred when they are simple to describe, and when they provide simple descriptions of the data. It has been proven that this distribution can be used to asymptotically learn any computable theory, given a sufficient amount of data, and as a result it has been proposed as a universal normative account of learning. In cognitive science, there is a growing body of work suggesting that human inductive biases are captured by models similar to this Solomonoff induction framework (Feldman, 2000; Goodman, Tenenbaum, Feldman, & Griffiths, 2008; Piantadosi, 2011). The relation between this work and the theoretical and empirical problems of language learning has been discussed in (A. S. Hsu & Chater, 2010; A. S. Hsu et al., 2011, 2013). Such approaches are closely related to the minimum description length principle (Grünwald, 2007; Rasin & Katzir, 2015; Rissanen, 1978). The argument-modifier (PSAG) model can be understood as providing a two-part code for the input corpus, consisting of the lexicon that it learns and the derivations that it provides for the sentences in the corpus.

Conclusion

We have investigated the role of the argument-modifier distinction in grammar learning. We introduced a formalization of the distinction, the argument-modifier (PSAG) model which captures several of its core distributional features. Our first set of analyses showed that this model recovers a linguistically-plausible argument-modifier distinction from unannotated data. These analyses serve two functions. First, they help to validate the use of the argument-modifier (PSAG) model in studying the argument-modifier distinction: they suggest that the model's representational assumptions are compatible with the true underlying distinction. Second, they allow us to place a lower-bound on the amount of information that a language learner could, in principle, extract from distributional language input. A model which represents the distributional differences between arguments and modifiers more faithfully than the argument-modifier (PSAG) model may be able to leverage this type of input even more effectively.

We next compared the argument-modifier (PSAG) model to a lesioned model—the argument-only (PTSG) model—which does not represent an argument-modifier distinction. The argument-modifier (PSAG) model discovered a set of argument structures which compress the input data better than those learned by the argument-only (PTSG) model. These argument structures are both more reusable than those discovered by the argument-only (PTSG) model, and provide simpler derivations of the input sentences, suggesting that the argument-modifier (PSAG) model categorized phrases as modifiers when doing so would simplify the lexicon and its derivation of the corpus. The results also extend to a generalization comparison of the two models: the argument-modifier (PSAG) model argument structures are more reusable and simplify derivations in a held-out data set, providing evidence that an argument-modifier structure can be used to capture real structure in the data.

References

Adger, D. (2003). *Core syntax: A minimalist perspective*. Oxford, England and New York, New York: Oxford University Press.

Adger, D., Pintzuk, S., & Plunkett, B. (1999). Specifiers in generative grammar. In D. Adger, S. Pintzuk, & B. Plunkett (Eds.), *Specifiers* (pp. 1–18). Oxford, England: Oxford University Press.

Aldezabal, I., Aranzabe, M., Gojenola, K., Sarasola, K., & Atuxta, A. (2002). Learning argument/adjunct distinction for Basque. In *Proceedings of the ACL-2002 workshop on lexical acquisition* (Vol. 9, pp. 42–50).

Arka, I. W. (2014). Locative-related roles and the argument-adjunct distinction in Balinese. *Linguistic Discovery*, *12*(2).

Berwick, R. C. (1982). *Locality principles and the acquisition of syntactic knowledge* (Unpublished doctoral dissertation). Massachusetts Institute of Technology.

Berwick, R. C. (1985). *The acquisition of syntactic knowledge* [Monograph]. Cambridge, Massachusetts and London, England: The MIT Press.

Bisk, Y., & Hockenmaier, J. (2013, March). An HDP model for inducing combinatory categorial grammars. *Transaction of the Association for Computational Linguistics*, *1*, 63–74.

Bloomfield, L. (1933). *Language*. New York, New York: Henry Holt.

Bod, R. (1998). *Beyond grammar: An experience-based theory of language*. Stanford, California: Center for the Study of Language and Information,[Stanford University].

Bod, R., Scha, R., & Sima'an, K. (Eds.). (2003). *Data-oriented parsing*. Stanford, CA: CSLI.

Borsley, R. D. (1999). *Syntactic theory: A unified approach*. London, England: Edward Arnold.

Bouma, G., & van Noord, G. (1994). Contraint-based categorial grammar. In *Proceedings of the 32nd annual meeting of the association for computational linguistics.* Las Cruces, New Mexico.

Brent, M. R. (1993). From grammar to lexicon: Unsupervised learning of lexical syntax. *Computational Linguistics*, *19*(2), 243–262.

Brent, M. R. (1994). Surface cues and robust inference as a basis for the early acquisition of subcategorization frames. *Lingua*, *92*, 433–470.

Bresnan, J. (2001). *Lexical-functional syntax*. Wiley-Blackwell.

Briscoe, T., & Carroll, J. (1997). Automatic extraction of subcategorization from corpora. In *Proceedings of the 5th conference on applied natural language processing.*

Briscoe, T., & Copestake, A. (1999). Lexical rules in constraint–based grammars. *Computational Linguistics*, *25*(4), 487–526.

Brown, R. (1973). *A first language: The early stages.* Harvard U. Press.

Chater, N., & Vitanyi, P. (2007). *'ideal learning' of natural language: Positive results about learning from positive evidence.*

Chiang, D. (2000). Staistical parsing with an automatically–extracted tree adjoining grammar. In *Proceedings of the 38th annual meeting of the association for computational linguistics.* Association for Computational Linguistics.

Chiang, D., & Bikel, D. (2002). Recovering latent information in treebanks. In *Proceedings of coling 2002.*

Chomsky, N. (1964). *Current issues in linguistic theory* (C. H. van Schooneveld, Ed.). The Hague and Paris: Mouton.

Chomsky, N. (1965). *Aspects of the theory of syntax*. Cambridge, MA: The MIT Press.

Chomsky, N. (1970). Remarks on nominalization. In R. J. . P. Rosenbaum (Ed.), *Readings in english transformational grammar.* Ginn and Company.

Chomsky, N. (1975, 1955). *The logical structure of linguistic theory*. New York and London: Plenum Press.

Chomsky, N. (1979, 1951). *Morphophonemics of modern Hebrew*. New York, NY: Garland Publishing.

Chomsky, N. (1986). *Barriers*. MIT Press.

Chomsky, N. (1993). A minimalist program for linguistic theory. In K. L. Hale & S. J. Keyser (Eds.), *The view from building 20: Essays in honor of Sylvain Bromberger* (pp. 1–52).

Cambridge, Massachusetts and London, England: The MIT Press.

Chomsky, N. (1995a). Bare phrase structure. In G. Webelhuth (Ed.), *Government and binding theory and the minimalist program* (pp. 383–349). Blackwell.

Chomsky, N. (1995b). Categories and transformations. In *The minimalist program* (pp. 219–394). Cambridge, Massachusetts and London, England: The MIT Press.

Chomsky, N. (1995c). *The minimalist program*. MIT Press.

Chomsky, N. (2000). Minimalist inquiries: The framework. In R. Martin, D. Michaels, & J. Uriagereka (Eds.), *Step by step: Essays on minimalist syntax in honor of Howard Lasnik.* MIT Press.

Cinque, G. (1999). *Adverbs and functional heads: A cross-linguistic perspective*. Oxford, England and New York, New York: Oxford University Press.

Cinque, G. (2013). Cognition, universal grammar, and typological generalizations. *Lingua*, *130*.

Clark, A. (2013). Learning trees from strings: A strong learning algorithm for some context-free grammars. *The Journal of Machine Learning Research*, *14*(1), 3537–3559.

Cohn, T., Blunson, P., & Goldwater, S. (2010). Inducing tree–substitution grammars. *Journal of Machine Learning Research*, *11*, 3053–3096.

Comrie, B. (1993). Argument structure. In J. Jacobs, A. von Stechow, W. Sternefeld, & T. Venneman (Eds.), *Syntax: An international handbook* (pp. 905–914). Berlin, Germany: Walter de Gruyter.

Cook, W. A. (1972). *A case grammar matrix* (15–47 No. 6). Languages and Linguistics Working Papers (Georgetown University).

Creissels, D. (2014). Cross-linguistic variation in the treatment of beneficiaries and the argument *vs*. adjunct distinction. *Linguistic Discovery*, *12*(2).

Croft, W. (2001). *Radical construction grammar: Syntactic theory in typological perspective*. Oxford and New York: Oxford University Press.

Culicover, P., & Jackendoff, R. (2005). *Simpler syntax*. Oxford: Oxford University Press.

Davidson, D. (1967). The logical form of action sentences. *Essays on actions and events 5*, 105–148.

de Marcken, C. (1996a). Linguistic structure as composition and perturbation. In *Proceedings of the 34th annual meeting on association for computational linguistics* (pp. 335–341).

de Marcken, C. (1996b). *Unsupervised language acquisition* (Dissertation). Massachusetts Institute of Technology.

Dickinson, M. (2005). *Error detection and correction in annotated corpora* (Unpublished doctoral dissertation). The Ohio State University.

Di Sciullo, A. M., & Williams, E. (1987). *On the definition of word.* Cambridge, MA: MIT Press.

Dowty, D. R. (2003). The dual analysis of adjuncts/complements in categorial grammar. In *Modifying adjuncts* (pp. 33–66).

Eisner, J. (2002). Discovering syntactic deep structure via Bayesian statistics. *Cognitive Science*, *26*, 255—268.

Feldman, J. (2000). Minimization of boolean complexity in human concept learning. *Nature*, *407*(6804), 630–633.

Fernandes, K. J., Marcus, G. F., Nubila, J. A. D., & Vouloumanos, A. (2006). From semantics to sntax and back again: Argument structure in the third year of life. *Cognition*, *100*, B10-B20.

Finkel, J. R., Grenager, T., & Manning, C. D. (2007). The infinite tree. In *Proceedings of the 45th annual meeting of the association for computational linguistics.*

Fisher, C., Gertner, Y., Scott, R. M., & Yuan, S. (2010, March/April). Syntactic bootstrapping. *Wiley Interdisciplinary Reviews (WIREs): Cognitive Science*, *1*(2), 143–149.

Forker, D. (2014). A canonical approach to the argument/adjunct distinction. *Linguistic Discovery*, *12*(2).

Fowler, G., & Yadroff, A. (1993). The argument status of accusative measure nominals in Russian. *Journal of Slavic Linguistics*, *1*(2), 251–279.

Fowlie, M. (2013). Order and optionality: Minimalist grammars with adjunction. In *The 13th meeting of mathematics of language.*

Frey, W., & Gärtner, H.-M. (2002). On the treatment of scrambling and adjunction in minimalist grammars. In *Proceedings of formal grammar.*

Gamut, L. T. F. (1991). *Logic, language, and meaning volume II: Intensional logic and logical grammar*. University of Chicago Press.

Gazdar, G., Klein, E., Pullum, G. K., & Sag, I. A. (1985). *Generalized phrase structure grammar*. Harvard University Press.

Gertner, Y., Fisher, C., & Eisengart, J. (2006, August). Learning words and rules. *Psychological Science*, *17*(8), 684.

Gildea, D., & Palmer, M. (2002). The necessity of parsing for predicate argument recognition. In *Proceedings of the 40th annual meeting on association for computational linguistics* (pp. 239–246).

Goldsmith, J. A. (2011). The evaluation metric in generative grammar [Theoretical Discussion]. In *Proceedings of the 50th anniversay celebration of the MIT department of linguistics.*

Goldwater, S. (2006). *Nonparametric bayesian models of lexical acquisition* (Unpublished doctoral dissertation). Brown University.

Goldwater, S., Griffiths, T. L., & Johnson, M. (2006). Interpolating between types and tokens by estimating power–law generators. In *Advances in neural information processing systems 18.* Cambridge, Ma: MIT Press.

Goodman, N. D., Tenenbaum, J. B., Feldman, J., & Griffiths, T. L. (2008). A rational analysis of rule-based concept learning. *Cognitive Science*, *32*(1), 108–154.

Graf, T. (2013). The syntactic algebra of adjuncts. In *Proceedings of chicago linguistics society (CLS) 49.*

Graf, T. (2014). Models of adjunction in minimalist grammars. In *Formal grammar* (pp. 52–68). Berlin, Germany and Heidelberg, Germany: Springer.

Grimshaw, J. (1990). *Argument structure*. MIT Press.

Grünwald, P. D. (2007). *The minimum description length principle*. Cambridge, MA: The MIT Press.

Haegeman, L. (1994). *Government & binding theory*. Blackwell.

Halliday, M. A. K. (1970). Language structure and language function. In J. Lyons (Ed.), *New horizons in linguistics* (pp. 140–165). Harmondsworth: Penguin Books.

Harabagiu, S. M., Moldovan, D. I., Pasca, M., Mihalcea, R., Surdeanu, M., Bunescu, R. C., ... Morarescu, P. (2000). Falcon: Boosting knowledge for answer engines. In *Trec* (Vol. 9, pp. 479–488).

Hartmann, I., Haspelmath, M., & Cysouw, M. (2014). *Identifying semantic role clusters and alignment types via microrole coexpression tendencies.* (Manuscript)

Haspelmath, M. (2014). Arguments and adjuncts as language-particular syntactic categories and as comparative concepts. *Linguistic Discovery*, *12*(2).

Headden III, W. P., Johnson, M., & McClosky, D. (2009). Improving unsupervised dependency parsing with richer contexts and smoothing. In *Proceedings of human language technologies: The 2009 annual conference of the north american chapter of the association for computational linguistics* (pp. 101–109).

Heim, I., & Kratzer, A. (2000). *Semantics in generative grammar*. Malden, MA: Blackwell Publishing.

Hogan, D. (2007). Coordinate noun phrase disambiguation in a generative parsing model..

Hornstein, N. (2008). Adjunction, labeling, and bare phrase structure. *Biolinguistics*, *2*(1).

Hornstein, N., & Lightfoot, D. W. (1981). *Introduction to explanation in linguistics: The logical problem of language acquisition*. Addison Wesley Longman.

Hsu, A. S., & Chater, N. (2010). The logical problem of language acquisition goes probabilistic: No negative evidence as a window on language acquisition. *Cognitive Science*, *34*, 972–1016.

Hsu, A. S., Chater, N., & Vitányi, P. M. B. (2011). The probabilistic analysis of language acquisition: Theoretical, computational, and experimental analysis. *Cognition*, *120*, 380–390.

Hsu, A. S., Chater, N., & Vitányi, P. M. B. (2013). Language learning for positive evidence reconsidered: A simplicity-based approach. *Topics in Cognitive Science*, *5*, 35–55.

Hsu, D. J., Kakade, S. M., & Liang, P. S. (2012). Identifiability and unmixing of latent parse trees. In *Advances in neural information processing systems* (pp. 1511–1519).

Huang, C.-T. J. (1982). *Logical relations in Chinese and the theory of grammar* (Unpublished doctoral dissertation). Massachusetts Institute of Technology.

Huddleston, R., Pullum, G. K., et al. (2002). The cambridge grammar of english. *Language. Cambridge: Cambridge University Press*.

Hunter, T. A. (2010). *Relating movement and adjunction in syntax and semantics* (Unpublished doctoral dissertation). University of Maryland.

Jackendoff, R. (1977). *X′ syntax*. MIT Press.

Jackendoff, R. (2002). *Foundations of language*. New York: Oxford University Press.

Johnson, D. E., & Postal, P. M. (1980). *Arc pair grammar*. Princeton, New Jersey: Princeton University Press.

Johnson, M., Griffiths, T. L., & Goldwater, S. (2007). Adaptor Grammars: A framework for specifying compositional nonparametric Bayesian models. In *Advances in neural information processing systems 19*. Cambridge, MA: MIT Press.

Joshi, A. K., Levy, L. S., & Takahashi, M. (1975). Tree adjunct grammars. *Journal of computer and system sciences*, *10*(1), 136–163.

Joshi, A. K., & Schabes, Y. (1997). Tree-adjoining grammars. In *Handbook of formal languages* (Vol. 3, pp. 69–124). Berlin: Springer.

Kay, P. (2005). Argument structure constructions and the argument–adjunct distinction. In M. Fried & H. C. Boas (Eds.), *Grammatical constructions: Back to the roots* (Vol. 4, pp. 71–98). John Benjamins Publishing Company.

Kayne, R. S. (1994). *The antisymmetry of syntax*. The MIT Press.

Koenig, J.-P., Mauner, G., & Bienvenue, B. (2003). Arguments for adjuncts. *Cognition*, *89*, 67–103.

Kornai, A., & Pullum, G. K. (1990, March). The X-Bar theory of phrase structure. *Language*, *66*(1), 24–50.

Kracht, M. (1999). Adjunction structures and syntactic domains. In *Mathematics of sentence structure: Trees and their logics.*

Kroeger, P. R. (2004). *Analyzing syntax: A lexical-functional approach.* Cambridge, England: Cambridge University Press.

Langacker, R. W. (1987). *Foundations of cognitive grammar, volume 1: Theoretical prerequisites.* Stanford, California: Stanford University Press.

Lebeaux, D. S. (2000). *Language acquisition and the form of the grammar.* John Benjamins Publishing Company.

Levin, B., & Rappaport Hovav, M. (2005). *Argument realization: Research surveys in linguistics.* Cambridge University Press.

Li, M., & Vitányi, P. M. B. (2008). *An introduction to Kolmogorov complexity and its applications* (3rd ed.). New York: Springer.

Liang, P., Petrov, S., Jordan, M. I., & Klein, D. (2007). The infinite PCFG using hierarchical Dirichlet processes. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (emnlp-conll)* (pp. 688–697).

Longacre, R. E. (1976). *An anatomy of speech notions.* Lisse: de Ridder.

MacWhinney, B. (2000). The childes project. *Tools for Analyzing Talk. Part*, *1*.

Marcus, M. P., Santorini, B., Marcinkiewicz, M. A., & Taylor, A. (1999). *Treebank–3* (Tech. Rep.). Philadelphia: Linguistic Data Consortium.

Matthews, P. H. (1981). *Syntax.* Cambridge, England: Cambridge University Press.

May, R. (1985). *Logical form: Its structure and derivation.* MIT Press.

McConnell-Ginet, S., & Chierchia, G. (2000). *Meaning and grammar: An introduction to semantics.* MIT Press.

Mel'čuk, I. (1988). *Dependency syntax : Theory and practice,.* Albany, N.Y.: The SUNY Press.

Merlo, P., & Leybold, M. (2001). Automatic distinction of arguments and modifiers: the case of prepositional phrases. In *Proceedings of the 2001 workshop on computational natural language learning-volume 7* (p. 15).

Merlo, P., & Stevenson, S. (2001). Automatic verb classification based on statistical distributions of argument structure. *Computational Linguistics*, *27*(3), 373–408.

Moortgat, M. (1997). Categorial type logics. In *Handbook of logic and language* (pp. 93–177). Elsevier.

Mosel, U. (2007). A corpus based approach to valency in a language documentation project. In *Pre-ALT (association for linguistic typology) workshop on linguistic typology and language documentation.* Paris, France.

Naseem, T., Chen, H., & Johnson, M. (2010). Using universal linguistic knowledge to guide grammar induction. In *Proceedings of the 2010 conference on empirical methods in natural language processing.*

O'Donnell, T. J. (2011). *Productivity and reuse in language* (Unpublished doctoral dissertation). Harvard University.

O'Donnell, T. J. (in press). *Productivity and reuse in language: A theory of linguistic computation and storage*. Cambridge, Massachusetts and London, England: The MIT Press.

O'Donnell, T. J., Snedeker, J., Tenenbaum, J. B., & Goodman, N. D. (2011). Productivity and reuse in language. In *Proceedings of the 33rd annual conference of the cognitive science society.*

Palmer, M., Kingsbury, P., & Gildea, D. (2005). The proposition bank. *Computational Linguistics*, *31*(1), 71–106.

Parsons, T. (1990). *Events in the semantics of english* [Monograph]. The MIT Press.

Pearl, L., & Sprouse, J. (2013). Syntactic islands and learning biases: Combining experimental syntax and computational modeling to investigate the language acquisition problem. *Language Acquisition*, *20*(1), 23–68.

Perfors, A., Tenenbaum, J. B., & Regier, T. (2011). The learnability of abstract syntactic principles. *Cognition*, *118*(3), 306–338.

Pesetsky, D. (1994). *Zero syntax.* MIT Press.

Phillips, L., & Pearl, L. (2014). *The utility of cognitive plausibility in language acquisition modeling: Evidence from word segmentation.* (Manuscript)

Piantadosi, S. T. (2011). *Learning and the language of thought* (Unpublished doctoral dissertation). Massachusetts Institute of Technology.

Pietroski, P. M. (2005). *Events and semantic architecture*. Oxford: Oxford University Press.

Pitman, J. (1995). Exchangeable and partially exchangeable random partitions. *Probability Theory and Related Fields*, *102*(2), 145–158.

Platt, J. T. (1971). *Grammatical form and grammatical meaning: A tagmemic view of Fillmore's deep structure concepts*. Amsterdam, The Netherlands: North-Holland Publishing Company.

Pollard, C., & Sag, I. A. (1987). *Information-based syntax and semantics, Volume 1: Fundamentals*. Stanford, California: CSLI Publications.

Pollard, C., & Sag, I. A. (1994). *Head-driven phrase structure grammar*. University of Chicago Press.

Post, M., & Gildea, D. (2009). Bayesian learning of a tree substitution grammar. In *Proceedings of the joint conference of the 47th annual meeting of the ACL and the 4th international joint conference on natural language processing of the AFNLP.*

Post, M., & Gildea, D. (2013). Bayesian tree substitution grammars as a usage-based approach. *Language and Speech*, *56*(3), 291–308.

Przepiórkowski, A. (1999a). *Case assignment and the complement/adjunct dichotomy* (Unpublished doctoral dissertation). Neuphilologischen Fakultät der Universität Tübingen, Tübingen.

Przepiórkowski, A. (1999b). On case assignment and "adjuncts as complements.". In G. Webelhuth, J.-P. Koenig, & A. Kathol (Eds.), *Lexical and constructional aspects of linguistic meaning* (pp. 223–245). CSLI Publications.

Radford, A. (1988). *Transformational grammar: A first course*. Cambridge, England: Cambridge University Press.

Rákosi, G. (2006). *Dative experiencer predicates in Hungarian* (Unpublished doctoral dissertation). Universiteit Utrecht.

Rambow, O., Vijay-Shanker, K., & Weir, D. (1995). D–tree grammars. In *Proceedings of the 33rd annual meeting of the association for computational linguistics.*

Rasin, E., & Katzir, R. (2015). On evaluation metrics in optimality theory [Computational Model, Theoretical Discussion, Linguistic Analysis]. *Linguistic Inquiry*. (Manuscript)

Rimell, L., Clark, S., & Steedman, M. (2009). Unbounded dependency recovery for parser evaluation. In *Proceedings of the 2009 conference on empirical methods in natural language processing: Volume 2-volume 2* (pp. 813–821).

Rissanen, J. (1978). Modeling by shortest data description [Theoretical Discussion, Computational Model, Mathematical Result]. *Automaticata*, *14*(5), 465–471.

Rizzi, L. (1990). *Relativized minimality*. Cambridge, Massachusetts and London, England: The MIT Press.

Roberts, I. G. (1997). *Comparative syntax*. London, England: Arnold.

Ross, J. R. (1967). *Constraints on variables in syntax* (Unpublished doctoral dissertation). Massachusetts Institute of Technology.

Sag, I. A. (2012). Sign-based construction grammar: An informal synopsis. In I. A. Boas Hans abd Sag (Ed.), *Sign–based construction grammar* (pp. 101–107). CSLI Publications.

Sag, I. A., Wasow, T., & Bender, E. M. (2003). *Syntactic theory: A formal introduction* (2nd ed.). Stanford, CA: CSLI.

Scha, R. (1990). Taaltheorie en taaltechnologie; competence en performance. In R. de Kort & G. Leerdam (Eds.), *Computertoepassingen in de neerlandistiek* (pp. 7–22). Landelijke Vereniging van Neerlandici.

Scha, R. (1992). Virtuele grammatica's en creatieve algoritmes [Theoretical Discussion, Computational Model]. *Gramma/TTT*, *1*(1), 57–77.

Schabes, Y., & Shieber, S. M. (1994). An alternative conception of tree-adjoining derivation. *Computational Linguistics*, *20*(1), 91–124.

Schabes, Y., & Waters, R. C. (1995). Tree insertion grammar: A cubic-time parsable formalism that lexicalizes context-free grammar without changing the trees produced. *Computational Linguistics*, *21*(4), 479–513.

Schikowski, R., Paudyal, N. P., & Bickel, B. (2014). Valency classes in Chintang. In B. Comrie & A. L. Malchukov (Eds.), *Valency classes: A comparative handbook.* De Gruyter Mouton.

Schütze, C. T. (1995). *PP attachment and argumenthood* (Tech. Rep.). Cambridge, Ma: Papers on language processing and acquisition, MIT working papers in linguistics.

Schütze, C. T., & Gibson, E. (1999). Argumenthood and english prepositional phrase attachment. *Journal of Memory and Language*, *40*(3), 409–431.

Solomonoff, R. (1978). Complexity-based induction systems: comparisons and convergence theorems. *Information Theory, IEEE Transactions on*, *24*(4), 422–432.

Solomonoff, R. J. (1964a). A formal theory of inductive inference. part i [Mathematical Result, Computational Model, Theoretical Discussion]. *Information and Control*, *7*(1), 1–22.

Solomonoff, R. J. (1964b, June). A formal theory of inductive inference. part ii [Mathematical Result, Computational Model, Theoretical Discussion]. *Information and Control*, *7*(2), 224–254.

Somers, H. L. (1984). On the validity of the complement-adjunct distinction in valency grammar. *Linguistics*, *22*(4), 507–530.

Spitkovsky, V. I., Alshawi, H., Chang, A. X., & Jurafsky, D. (2011). Unsupervised dependency parsing without gold part-of-speech tags. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 1281–1290).

Stabler, E. P. (1997). Derivational minimalism. In *Logical aspects of computational linguistics.* Springer.

Steedman, M. (2000). *The syntactic process.* The MIT press.

Stepanov, A. (2001). Late adjunction and minimalist phrase structure. *Syntax*, *4*(2), 94–125.

Stevenson, S., Merlo, P., Kariaeva, N., & Whitehouse, K. (1999). Supervised learning of lexical semantic verb classes using frequency distributions. In *Proceedings of the association for computational linguistics special interest group on the lexicon SigLex99: Standardizing lexical resources.*

Tallerman, M. (2015). *Understanding syntax* (Fourth ed.). London, England and New York, New York: Routledge.

Teh, Y. W. (2006). *A Bayesian interpretation of interpolated Kneser-Ney* (Tech. Rep. No. TRA2/06). National University of Singapore, School of Computing.

Tesniére, L. (1959). *Éléments de syntaxe structurale*. Kilncksieck.

Truswell, R. (2007). Tense, events, and extractions from adjuncts. In *Proceedings from the annual meeting of the chicago linguistic society 43.*

Tutunjian, D., & Boland, J. E. (2008). Do we need a distinction between arguments and adjuncts? Evidence from psycholinguistic studies of comprehension. *Language and Linguistics Compass*, *2*(4), 631–646.

Vadas, D., & Curran, J. (2007). Adding noun phrase structure to the penn treebank. In *Proceedings of the 45th annual meeting of the association for computational linguistics* (Vol. 45, p. 240).

Vater, H. (1978). On the possibility of distinguishing between complements and adjuncts. In *Valence, semantic case and grammatical relations* (pp. 21–45). John Benjamins.

Webster, M., & Marcus, M. (1989). Automatic acquisition of the lexical semantics of verbs from sentence frames. In *Proceedings of the 27th annual meeting of the association for computational linguistics* (pp. 177–184).

Wichmann, S. (2014). Arguments and adjuncts cross-linguistically: A brief introduction. *Linguistic Discovery*, *12*(2).

Williams, E. (1995). Theta theory. In G. Webelhuth (Ed.), *Government and binding theory and the minimalist program* (pp. 97–124). Oxford, England: Blackwell.

Yamada, K., & Knight, K. (2001). A syntax-based statistical translation model. In *Proceedings of* *the 39th annual meeting on association for computational linguistics* (pp. 523–530).

Yuan, S., Fisher, C., & Snedeker, J. (2012). Counting the nouns: Simple structural cues to verb meaning. *Child Development*, *83*(4), 1382–1399.

Zeman, D., & Sarkar, A. (2000). Learning verb subcategorization from corpora: Counting frame subsets. In *Proceedings of the international conference on language resources and evaluation (LREC).*

Zwicky, A. M. (1993). Heads, bases, and functors. In G. G. Corbett, N. M. Fraser, & S. McGlashan (Eds.), *Heads in grammatical theory* (pp. 292–315). Cambridge, England: Cambridge University Press.