Ambiguity Resolution in a Cognitive Model of Language Comprehension

Peter Lindes (plindes@umich.edu)
John E. Laird (laird@umich.edu)
University of Michigan, 2260 Hayward Street
Ann Arbor, MI 48109 USA

Abstract

The Lucia comprehension system attempts to model human comprehension by using the Soar cognitive architecture, Embodied Construction Grammar (ECG), and an incremental, word-by-word approach to grounded processing. Traditional approaches use techniques such as parallel paths and global optimization to resolve ambiguities. Here we describe how Lucia deals with lexical, grammatical, structural, and semantic ambiguities by using knowledge from the surrounding linguistic and environmental context. It uses a local repair mechanism to maintain a single path, and shows a garden path effect when local repair breaks down. Data on adding new linguistic knowledge shows that the ECG grammar grows faster than the knowledge for handling context, and that low-level grammar items grow faster than more general ones.

Keywords: Natural language understanding; cognitive models; Soar; construction grammar; Embodied Construction Grammar; local repair; ambiguity resolution; garden path effect.

Introduction

In previous work, we described the development of a cognitive model of language comprehension (Lindes and Laird, 2016; 2017), implemented in Soar (Laird, 2012), that incorporates the Embodied Construction Grammar (ECG) cognitive linguistic theory of grammar (Feldman, Dodge, and Bryant, 2009; Bergen and Chang, 2013). A key part of our model is that it attempts to model human comprehension processes. This is done by using parsing that is incremental and word by word, eagerly applying all available knowledge sources at each step, while maintaining a single syntactic and semantic interpretation. Our work is inspired by previous cognitive model-based theories, such as NL-Soar (Newell, 1990; Lehman et al. 1991; Lewis, 1993), and is consistent with the recent "Now–or-Never bottleneck" proposal of Christiansen and Chater (2016).

Traditional natural language processing approaches focus on syntactic analysis of isolated sentences (Hale, 2014). Techniques for resolving ambiguities include multiple parallel paths, using statistics from corpora, global optimization, and producing a ranked list of possible parses. These methods lack contextual knowledge to resolve ambiguities to produce accurate, grounded meanings in context. Their success is at the cost of relaxing constraints imposed by an incremental model of human processing.

Although our system, called Lucia, has been successful in supporting language understanding for an embodied robotic agent (Lindes and Laird, 2016), a significant question is whether incremental, word-by-word approaches can handle the many types of ambiguity that can arise in language understanding. Parsers developed for ECG (Bryant, 2008) and Fluid Construction Grammar (FCG; Steels and Hild, 2012) do not attempt to model incremental parsing, but instead treat parsing as optimization over a complete sentence, with no commitment to word-by-word processing. Thus, these other approaches do not treat the issues of dealing with ambiguity that arise in incremental parsing.

In this paper, we explore the problem of ambiguity in incremental language processing. We build on previous work by Lewis (1993), where local repair is used to recover from some types of syntactic ambiguity, but we extend this to other forms of lexical, grammatical, structural, and semantic ambiguity, taking advantage of the contextual knowledge that is available during processing. Comparison to detailed human performance data is outside the scope of our current research. In the following, we discuss the basic operation of the system, and explore how it deals with different ambiguous situations.

Basic Comprehension

Lucia is built within a Soar agent called Rosie (Mininger and Laird, 2016) that learns new tasks involving robotic object manipulation and navigation. It uses a grammar for a domain-specific subset of English written in the formal language of ECG (Bryant, 2008). A program translates the ECG grammar into Soar production rules that we call G rules. Another set of Soar rules that connect to the embodied context of the agent, are written by hand, and are called C rules. Together these rules process language input to produce meaningful messages that Rosie uses to perform actions and learn new tasks.

Grammars in the ECG language are made up of two kinds of "items:" constructions and schemas. Each schema defines the structure of a certain kind of meaning element and defines its "roles" or "slots." A construction is a pairing of a form with a meaning. There are three types of ECG constructions. Lexical constructions (L cxns) recognize input words. Phrasal constructions (P cxns) combine one or more constituents already recognized into a higher-level structure. General constructions (G cxns) do not recognize specific forms, but augment instances of other constructions that are marked as their subcases. Any construction can evoke a schema to represent its meaning and provide constraints to specify how to populate the slots of the schema.

Semantic parsing is carried out incrementally, with processing done greedily for each word, as in the incremental approach called "Chunk-and-Pass," which Christiansen and Chater (2016) claim models human comprehension. The basic operation is a word cycle in which a new word is received, a lexical access operator retrieves one or more senses of that word (L cxns), and then further processing is performed. The further processing includes operators that recognize and apply phrase level constructions (P cxns) and operators that ground the meanings built from the grammar to the perceptions and actions of the agent using C rules.

The current state of the parse is represented by a stack in working memory that contains a sequence of construction instances that have been recognized but not yet incorporated as constituents of a higher level construction. During lexical access, one or more L cxn instances are added to the current state. Then a P cxn that matches the current state, if any, creates a new instance of itself on the stack, removing its constituents from the stack and adding them as its children, to form a new "chunk." This can happen several times in a single word cycle. When a construction instance is created, its corresponding meaning structure is also built. These meaning structures trigger grounding operators that look for something to ground this meaning, either in the agent's perceptual model or its general background knowledge.

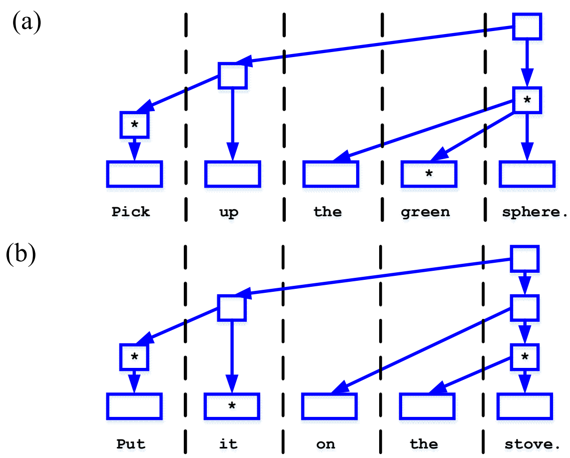

Figure 1: Examples of word-by-word comprehension.

Figure 1 shows some example parses. The word processing cycles are separated by vertical dotted lines. Each rectangle is a construction instance, with L cxns shown larger. An asterisk means a grounding operator was used. Meaning structures are not shown. Within each cycle, operators are executed from the bottom up. When the whole sentence has been processed and the result is a single construction instance, that construction is interpreted to produce a message to tell the robot what to do. If the processing does not produce a single result, the parse fails.

The Lucia comprehender has been applied to a corpus of several hundred sentences previously used with the Rosie system. The grammar and context rules have been developed sufficiently to correctly comprehend 130 of those sentences. A variety of sentential forms are comprehended, including the examples in (1).

(1) a. `The sphere is green.`
b. `Store the large green sphere on the red triangle.`
c. `Pick a green block that is larger than the green box.`
d. `Drive to the wall.`
e. `Go until there is a doorway.`
f. `If the green box is large then go forward.`
g. `What is inside the pantry?`
h. `Where is the red triangle?`
i. `Is the large triangle to the right of the green sphere?`
j. `Drive down the hall until you reach the end.`
k. `Fetch a soda.`

A variety of declarative, interrogative, and imperative sentences are handled, including ones with relative clauses and conditional clauses. In many of the 130 sentences, various kinds of lexical, syntactic, and semantic ambiguities must be handled. Below we examine some of these cases.

Handling Ambiguities

Here we analyze how Lucia handles instances of lexical, grammatical, structural, and semantic ambiguities, as well as garden path sentences. For each type of ambiguity, we give some specific examples and show how Lucia resolves them using different types of contextual knowledge within its incremental, word-by-word approach to comprehension.

Lexical Ambiguities

Lucia has several strategies for dealing with words that have different meanings depending on the context.

Resolution by Syntactic Context Many function words have meanings that vary depending on the syntactic context. For example, *up* can be a particle together with a verb as in *pick up*, or it can be a preposition. Various forms of *to be*, such as *is*, have many possible uses. When possible, Lucia uses the strategy of having a single construction for the word defined in the grammar and instantiated during lexical access, and then resolving the correct meaning from the syntactic context by what phrasal construction uses that word. This follows the principle in construction grammar theory that both words and larger constructions contribute to meaning (Goldberg, 1995). Consider some of the many uses of *is* in (2):

(2) a. `The sphere` *`is`* `green.`
b. `The red triangle` *`is`* `on the stove.`
c. `Go until there` *`is`* `a doorway.`
d. *`Is`* `the large orange block a sphere?`

Is can declare an object property (2a) or a relation (2b). With *there, is* can declare the existence of something (2c). *Is* can also introduce a question (2d). None of this information is derived during lexical access, but is added as phrasal constructions are recognized.

Multiple Senses, Immediate Resolution Content words often have multiple senses, with context needed to select from them. In these cases, the grammar defines two or more alternative lexical constructions. A phrasal construction that recognizes one of them chooses that one and deletes the others, as in (3):

(3) a. `The` *`sphere`* `is` *`red`*`.`
 b. `Where is the` *`red`* `triangle?`
 c. `Is this a` *`sphere`*`?`

These three sentences show different senses for both *sphere* and *red*. *Sphere* produces two senses, a noun and a class name. The noun sense is recognized by one P cxn in (3a), while *a sphere* in (3c) is recognized by a different P cxn that uses the class sense, discarding the noun. In both (3a) and (3b) *red* is recognized as a property, but in (3a) it is declared to apply to *the sphere,* while in (3b) it is used as an adjective to modify *triangle*.

That can be deictic (4a) to refer to something being pointed to, or can be used to introduce a relative clause (4b). Both senses are generated in lexical access. A P cxn that matches the context then selects one of the senses and deletes the other.

(4) a. `Put` *`that`* `in the pantry.`
 b. `Pick up the green block` *`that`* `is on the stove.`

Multiple Senses, Delayed Resolution The word *square* can be a property to be applied, a noun, or an adjective:

(5) a. `This is a` *`square`*`.`
 b. `Put the` *`square`* `in the` *`square`* `box.`

All three senses are generated by lexical access each time. For a property application as in (5a), that sense is chosen by a P cxn and the others discarded. In the first case in (5b), the noun is chosen similarly.

The second case in (5b) is more complicated: in processing this instance of *square,* the noun will be chosen as before. When *box* is being processed, the system recognizes that the chosen sense is wrong, and an operator called *snip* is selected, which deletes the P cxn for *the square.* Next, the previously discarded adjective sense of *square* replaces the noun sense. Now the whole phrase *the square box* can be recognized. Many nouns can be used as adjectives like this.

The case of *square* as an adjective illustrates the delayed resolution strategy. In immediate resolution, other senses are not completely forgotten; they are linked to the chosen sense and can be brought back and selected in a later context. This is one kind of repair process that makes incremental parsing possible. These strategies make it possible for the comprehender to maintain only a single path in its parse state, yet still have enough information available to make a local repair when necessary.

Resolution by Semantic Context Some lexical ambiguities must be resolved by semantic rather than syntactic context. The meaning of *bank*, for example, depends on whether the semantic context is related to rivers or finances. Lucia has access to semantic information, both in the part of the sentence that has already been processed and in the more general discourse context. At the moment, none of the sentences we have worked with have needed this kind of resolution, but this can be easily added when needed.

Grammatical Ambiguities

Lucia uses one of two strategies when multiple phrasal constructions match a given parse state. The first is simple: when two different constructions match at the same time, if one matches more constituents than the other, then the more specific one (the one with the greater span) is chosen. When processing *sphere* in Figure 1a, either the noun by itself could be recognized or the phrase *the green sphere*. The longer, more specific match is preferred to the shorter, more general one.

There are cases where two constructions with the same span match the same parse state. In order to choose a more specific option over a more general one in these cases, there are preference rules to select the more specific one.

(6) a. *`The`* `sphere is green.`
 b. `This is` *`a`* `sphere.`

In (6) we have two phrases with *sphere*. Either could be recognized by a noun phrase construction, but in (6b) the phrase should be interpreted as a property that can be applied to the subject of the sentence rather than a noun phrase to ground to an object. Two preference rules, one for a definite and one for an indefinite determiner, make the distinction.

Structural Ambiguities

Often the immediate context suggests one way of integrating a word into the ongoing parse, but later on that decision turns out to be wrong, as in *the square box* where the word *square* should be an adjective and not a noun. Of particular importance are the attachment of prepositional phrases and relative or subordinate clauses. Lucia implements a strategy of local repair, similar to that used by Lewis (1993), to resolve these ambiguities, as the following examples show.

(7) a. `Pick up the green block on the stove.`
 b. `Put the green sphere in the pantry.`
 c. `Pick up the green block that is on the stove.`

d. `Put the green block that is on the stove in the pantry.`

e. `Move the green rectangle to the left of the large green rectangle to the pantry.`

Sentence (7a) appears to be complete after processing *block*. However, there are more words. After processing *stove,* there is a prepositional phrase that could either modify *the green block* or provide a target location for the verb. In this case, it should modify the noun phrase, since *pick up* does not expect a target location. However, that noun phrase has already been consumed by the clause construction and is no longer available on the stack as a constituent, so the system is at an impasse. What can be done?

The answer is a variant of the *snip* operator described earlier, which was introduced by Lewis (1993). This version deletes the clause construction to expose the noun phrase for *the green block* on the stack. Then that noun phrase is combined with the prepositional phrase to form a new referring expression that is grounded to that particular green block, which happens to be on the stove. Figure 2 shows two steps of this process.

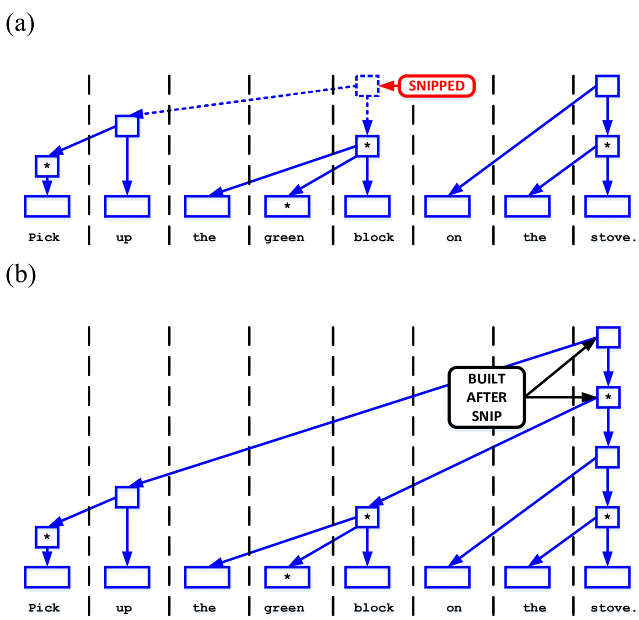

Figure 2: A local repair using *snip*

Figure 2a shows the state of the parse when we reach the impasse. At this point, a *snip* is performed to delete the clause construction shown with dotted lines, allowing the creation and grounding of the expression for *the green block on the stove*, as in Figure 2b. Finally, a new clause construction is created with this new referring expression.

Another aspect of grounded comprehension is shown by (7a). *The green block* is first grounded to a set of four green blocks that all exist in the current environment. If the sentence ended here, the comprehender would have two choices: either pick one of the four at random or report that it sees four possible meanings and ask for clarification. However, when the full expression *the green block on the stove* has been processed, grounding yields a single green block, which is currently on the stove. This shows an example of resolving ambiguous semantics through grounding.

Semantic Ambiguities

The current Lucia system resolves several problems using semantic information built into its grammar. One example is the different prepositional phrase attachments chosen for sentences (7a) and (7b). The two verbs *pick up* and *put* are not simply processed as instances of some general verb part of speech. Instead, distinct meaningful constructions for the two verbs are treated differently in the grammar, causing one to require a prepositional phrase and the other not. This is an example of how grammatical constructions, not just lexical items, carry meaning, as Goldberg (1995) insists.

Prepositions give another interesting example of this effect. Consider the two sentences in (8).

(8) a. `Go` *`to`* `the kitchen.`

b. `Go` *`down`* `the hall.`

Most generative grammar approaches produce the same exact grammatical structure for both of these sentences. Such an approach fails in an incremental semantic parse that must produce actionable meanings. The final messages that are to be sent to the robot for these two sentences are different. For (8a), the message specifies a specific waypoint as the goal of the *go* action, whereas for (8b) no specific goal is given, just an object representing *the hall* to guide the motion.

When sentence (8b) was first encountered while building Lucia's grammar, we realized that not all prepositions are the same. Consider a number of other possible prepositions that could have appeared in one of these sentences: *across*, *along*, *around*, *behind*, *in*, *into*, *out of*, *past*, *through*, *to the left of*, and so on. Some of these would work perfectly well in one of the sentences while making the other infelicitous[1]. Whether some of these make sense in certain sentences may depend on the noun that follows or the main verb of the sentence. Each of these prepositions seem to describe a trajectory in space, which may or may not have a terminating point. An interesting mental exercise is to try to imagine a diagram of the trajectory expected for each of the prepositions listed in each of the given sentences or in a similar one.

To deal with this problem, some refactoring was done in the part of the grammar dealing with prepositions. In (8a), *to* is treated as an ordinary preposition. For *down* in (8b) we created a new construction that can only be a constituent of a corresponding special subcase of a prepositional phrase. These constructions provide an alternative way of parsing

[1] Linguists use the term *infelicitous* to describe a sentence which is syntactically correct but does not make sense semantically.

depending on the particular preposition involved, which then allows building a different meaning structure.

This is another example of constructions carrying meaning, and shows key characteristics of a constructionist approach to grammar. In this approach we seek to define many specific constructions to build meaning into the grammar, rather than a minimal number of meaningless phrase labels to cover the language. This fits with psychological theories of children's language acquisition that emphasize children learning very specific constructions first and then gradually generalizing them (Tomasello, 2003).

Garden Path Sentences

"Garden path sentences" are grammatically correct, but are difficult for humans to parse correctly, at least at first. It appears that humans make a wrong decision early on in the parse, and later on, no local repair mechanism is sufficient to correct the problem. The Lucia theory produces this effect as we see with (9).

(9) `The horse raced past the barn fell.`

Lewis (1993) provides a theory of garden paths. He describes three possible causes: there is a lack of structural cues to trigger repair, the syntactic relation that needs to be altered is no longer available, or the system has not learned an alternative solution through previous deliberation.

The Lucia analysis of this sentence is consistent with this theory, as shown in Figure 3. First, *the horse raced* looks like a whole sentence using the past tense of *race* and discarding its past participle sense. Later a correct parse is found for *The horse raced past the barn*. Now when *fell* arrives, there is no way to integrate it into the sentence, because of the wrong choice that was made to use *raced* as a simple past tense verb rather than a past participle. This creates a garden path effect.

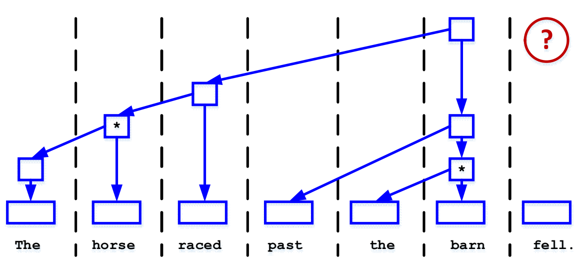

Figure 3: A garden path sentence.

Why does local repair not work here? Because when the system gets to the impasse, the change that needs to be made is at *raced,* which is two layers back on the stack and two layers deep in the hierarchy. This is not local enough for local repair to work, consistent with Lewis's second reason.

If the grammar only has the past participle sense of *raced*, Lucia produces a correct analysis. A deliberative repair process might produce the correct parse. Neither humans nor Lucia can do this as part of automatic parsing.

Taken together, the examples above show that an incremental comprehension system can resolve many lexical, grammatical, structural, and semantic ambiguities, and at the same time produce garden path effects.

Adding to Linguistic Knowledge

Currently, Lucia has no mechanism for learning new vocabulary, new phrasal constructions, or new concepts. The principle that meaningful language relies on many very specific constructions organized in a network with some generalities (Goldberg, 2006), rather than a few general rules, suggests that adding linguistic knowledge by hand will not scale up to something approaching general human language. Thus, even if our comprehension mechanisms are sufficient, the system will be limited in its application if it is unable to acquire new language. A means of acquisition is an essential goal for future work.

However, by analyzing Lucia's development, we can make some predictions about learning. In Lucia, the linguistic knowledge has grown incrementally. To process each new sentence, we coded new constructions and schemas in ECG and added new context rules when necessary. We expect that the G rules, which encode items in the grammar, would grow faster than the C rules which perform contextual processing.

Figure 4 shows how the number of Soar production rules of each type grew as the number of sentences comprehended grew from 42 to 130. Many more grammar rules than context rules were added, and the number of grammar rules grew more rapidly than the number of context rules.

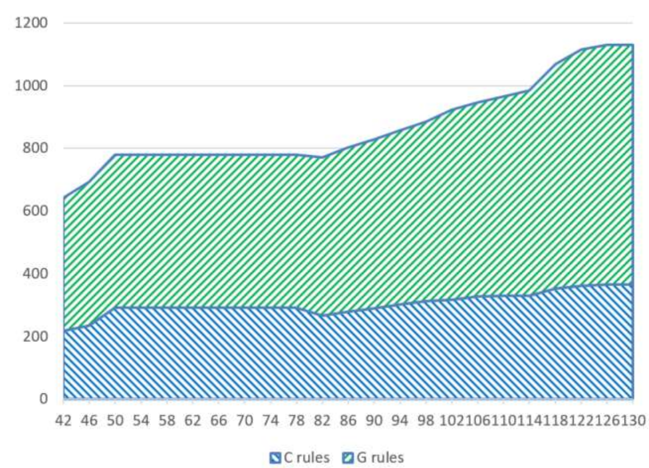

Figure 4: Growth of C & G rules as language coverage increases.

Figure 5 gives a different perspective on this growth data. Here we show the growth in ECG items, both constructions and schemas. Constructions are further broken down into lexical constructions (L cxns), phrasal constructions (P cxns), and general constructions (G cxns). We see that lexical constructions and schemas are growing faster than the more general construction types, confirming that the more specific items grow faster.

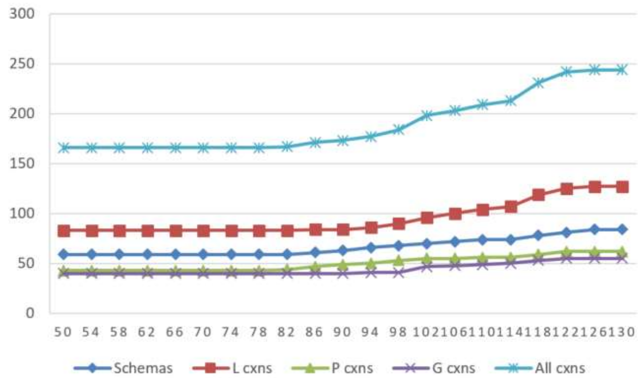

Figure 5: Growth of ECG items as language grows.

Conclusions

The results from Lucia are consistent with the claim that a comprehension system using a human-like, integrated, incremental parsing approach, within a cognitive architecture, and with construction grammar, can incrementally resolve a variety of linguistic ambiguities. They also are consistent with one type of breakdown that arises in garden path sentences in a way similar to humans.

How scalable is this approach? There are many linguistic forms that it does not handle: past and future tenses, auxiliary verbs, conjunctions, metaphor, and on and on. Nevertheless, as Figures 4 and 5 show, as new forms have been addressed, most of the new knowledge required has been expressible in the ECG grammar and has not required changes to the underlying context rules

The techniques we have described for handling ambiguity, however, depend mostly on the context operators. They provide grounded semantics, select among grammatical alternatives, and perform local repairs. This is consistent with the theory that human-like comprehension relies heavily on context to resolve ambiguities.

The current approach requires coding context rules by hand. In the future, we will attempt to enhance the ECG language to encode contextual constraints and/or use context-dependent retrievals with spreading activation in long-term declarative memory (Jones et al., 2016).

Also in future work we intend to explore comparing detailed processing data from Lucia to the large amount of available human performance data, and to datasets other than the Rosie sentence corpus we have considered here.

Acknowledgment

The work described here was supported by the AFOSR under Grant Number FA9559-15-1-0157. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressly or implied, of the AFOSR or the U.S. Government.

References

Bergen, B. and N. Chang (2013). Embodied Construction Grammar. In Thomas Hoffman and Graeme Trousdale, eds, *The Oxford Handbook of Construction Grammar*. Oxford University Press, New York.

Bryant, J. E. (2008). *Best-Fit Constructional Analysis*. PhD dissertation in Computer Science, University of California at Berkeley.

Christiansen, M. H. and N. Chater (2016). The Now-or-Never bottleneck: A fundamental constraint of language. *Behavioral and Brain Sciences (2016)*, 1-72.

Feldman, J., E. Dodge, and J. Bryant (2009). Embodied Construction Grammar. In *The Oxford Handbook of Linguistic Analysis*, Bernd Heine and Heiko Narrog eds. Oxford Handbooks Online.

Goldberg, A. E. (1995). *Constructions: A Construction Grammar Approach to Argument Structure*. The University of Chicago Press.

Goldberg, A. E. (2006). *Constructions at work: The nature of generalization in language.* Oxford University Press.

Hale, J. T. (2014). *Automaton Theories of Human Sentence Processing*. Stanford: CLSI Publications.

Jones, S. J., A. R. Wandzel, and J. E. Laird (2016) Efficient Computation of Spreading Activation Using Lazy Evaluation. *Proceedings of the 14th International Conference on Cognitive Modeling (ICCM)*.

Laird, J. E. (2012). *The Soar Cognitive Architecture*. Cambridge, MA: The MIT Press.

Lehman, J. F., R. L. Lewis, and A. Newell (1991). Integrating Knowledge Sources in Language Comprehension. *The Soar Papers, Vol. II*. Cambridge, MA: MIT Press.

Lewis, R. L. (1993). *An Architecturally-based Theory of Human Sentence Comprehension*. PhD dissertation in Computer Science, Carnegie Mellon University.

Lindes, P. and J. E. Laird (2016). Toward Integrating Cognitive Linguistics and Cognitive Language Processing. *Proceedings of the 14th International Conference on Cognitive Modeling (ICCM 2016*.

Lindes, P. and J. E. Laird (2017). Cognitive Modeling Approaches to Language Comprehension Using Construction Grammar. Accepted for publication with the *AAAI Spring Symposium 2017* track on *Computational Construction Grammar and Natural Language Understanding.*

Mininger, A. and J. E. Laird (2016). Interactively Learning Strategies for Handling References to Unseen or Unknown Objects. *Advances in Cognitive Systems 4*.

Newell, A. (1990). *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press.

Steels, L. and M. Hild, eds. (2012). *Language Grounding in Robots.* Springer.

Tomasello, M. (2003). *Constructing a Language: A Usage-Based Theory of Language Acquisition*. Harvard University Press, Cambridge, MA.