

# Recovering Coherent Interpretations Using Semantic Integration of Partial Parses

John Bryant

Department of Computer Science  
University of California at Berkeley  
Berkeley, CA 94720  
jbryant@icsi.berkeley.edu

## Abstract

This paper describes a chunk-based parser/semantic analyzer used by a language learning model. The language learning model requires an analyzer that robustly responds to extragrammaticality, ungrammaticality and other problems associated with transcribed language. The analyzer produces globally coherent analyses by semantically integrating the partial parses. Each resulting semantically integrated analysis is ranked by its semantic compatibility using a novel metric called semantic density.

## 1 Introduction

Consider the task faced by a child trying to learn language. The utterances confronting the child are filled with unknown words and unknown syntactic patterns. The child's grammar is in constant flux, and yet the child extracts (partial) meaning.

Chang (Chang and Maia, 2001) uses a cognitively motivated framework to model the situation children find themselves in when they learn language. The model goes utterance by utterance through a semantically labeled corpus, and whenever the semantic analysis of an utterance does not supply the salient relations in the context associated with the utterance, the model hypothesizes new grammatical mappings to account for the mismatch between the utterance's analyzed meaning and its associated context.

The language analysis that the learning framework relies on exhibits many of the standard issues motivating robust methods. Because the grammar is changing with every utterance, extragrammaticality and ungrammaticality (with respect to the current grammar) are the norm rather than the exception. In addition, the child language data functioning as a training corpus is transcribed from spoken language. As such, the utterances exhibit the usual difficulties associated with real language such as agreement errors, false starts, fragments, and constituent dislocations.

A task of this nature cannot be handled by tra-

ditional methods of language analysis because they are too brittle. Robust approaches are much more appropriate. This paper describes a system implementing such a robust approach. It combines chunk-based semantic analysis with an abductive method for recovering coherent interpretations from incomplete analyses.

Upon encountering an utterance, the system attempts syntactic/semantic analysis using a semantic chunker. If a complete analysis cannot be found, the system tries to integrate the remaining semantic chunks by merging common semantic structures. In other words, the system leverages the semantics to robustly combine and interpret recognized, but unbound semantic chunks.

Different sets of chunks can be integrated in different ways. To compare how good each possible integration is, the system uses a heuristic called semantic density. Integrations that are more dense are preferred since that suggests high semantic compatibility between the recognized chunks. The semantic density metric is related to the coherence/completeness principles used in LFG (Dalyrmples, 2001), but in addition, suggests a graded notion of semantic compatibility more akin to a notion of gradient grammaticality (Sorace and Keller, to appear).

The rest of this paper is structured as follows. The next section describes the basic architecture of the system. Section 3 provides a brief introduction to the grammar formalism the system operates on. Section 4 covers the semantic chunker. Section 5 describes the process and motivation for integrating chunks using structure merging. Section 6 details the semantic density metric and a graded notion of semantic compatibility. The last two sections describe future directions and conclude the paper.

## 2 System Architecture

As shown in Figure 1, an important aspect of the system is the tight integration between the learner and the analyzer. The analyzer extracts the semantic relations from the utterance that the learner uses

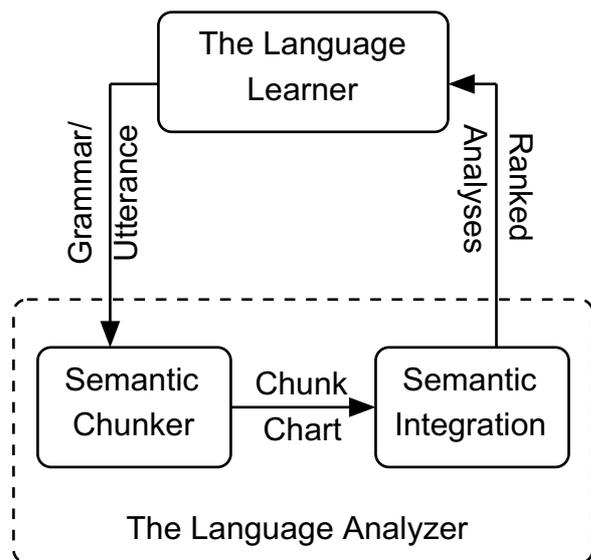


Figure 1: The learner and analyzer form a loop. The learner hypothesizes constructions to account for missing relations. The analyzer then uses the new grammar to analyze subsequent utterances.

to hypothesize new constructions. The learner provides the analyzer better grammars as time goes by, thereby making the analyzer more robust. This dynamic interaction makes it possible for the system to learn from its experience.

The language analysis component breaks down into two phases. First the language learner calls the semantic chunker with the current grammar and utterance. The first phase of analysis is semantic chunking. The chunker generates a set of semantic chunks stored as a chart.

The second phase of analysis extracts the smallest number of chunks that span the utterance from the chart, and performs semantic integration. Their common semantic structures are merged, and the resulting analyses are ranked according to the semantic density metric (see section 6). This ranked set of analyses are returned to the learner.

### 3 The Grammar

The grammar rules used for analysis are represented using a cognitively motivated grammar formalism called Embodied Construction Grammar (Bergen and Chang, 2002). The basic linguistic unit in any construction grammar (Goldberg, 1995) is the *construction*. A construction is a form-meaning pair. Each pair is a structured mapping from a lexical/syntactic pattern to its corresponding semantic and pragmatic properties.

Construction grammar rejects the assumption that

syntax and semantic are separate processing modules (Fillmore et al., 1988). Morphemes, idioms and standard syntactic constructions like subject-auxiliary-inversion are all represented by the same kind of object – the construction. Construction Grammar defines grammaticality as a combination of syntactic *and* semantic well-formedness. Using Embodied Construction Grammar as the linguistic substrate therefore requires that syntactic and semantic analysis happen simultaneously.

To describe constructions precisely, Embodied Construction Grammar (ECG) combines a grammar formalism and knowledge representation language within a unification-based framework. This makes it possible for both constructions and frame-based, schematic knowledge (Fillmore, 1982) to be expressed succinctly in the same formalism. Linking the grammar into frame-based meaning is what makes semantic integration possible (see section 5).

ECG has two basic units: the schema and the construction.

- Constructions, as discussed, are form-meaning pairs, while schemas are used to represent meaning (like frames or image schemas<sup>1</sup>).
- Schemas and constructions have roles which can be assigned an atomic value (with  $\leftarrow$ ) or coindexed (with  $\leftarrow\rightarrow$ ).
- Schemas and constructions are arranged into inheritance hierarchies with the **subcase of** keyword.
- The **self** keyword lets a schema or construction be self-referential<sup>2</sup>.

To make this more concrete, figure 2 shows the Throw lexical construction and its associated schemas. Every construction in ECG has a **form** block and a **meaning** block. These blocks are themselves special roles that are accessed using the *f* and *m* subscripts. In the case of the Throw construction, its form pole constrains its orthography feature to the string *throw*. Its meaning pole is type constrained (using the colon) to be of type (Throw-Action) schema.

The Throw-Action schema has roles for the thrower and the throwee. These roles correspond to the semantic arguments of a *throw* predicate. Roles can type constrain their fillers, and in the case of Throw-Action, the thrower must be of type *Animate* while the throwee is constrained to be of type

<sup>1</sup>Image schemas are schematic representations of cognitively-motivated spatial primitives

<sup>2</sup>ECG's **self** operator is equivalent to LFG's  $\uparrow$  operator.

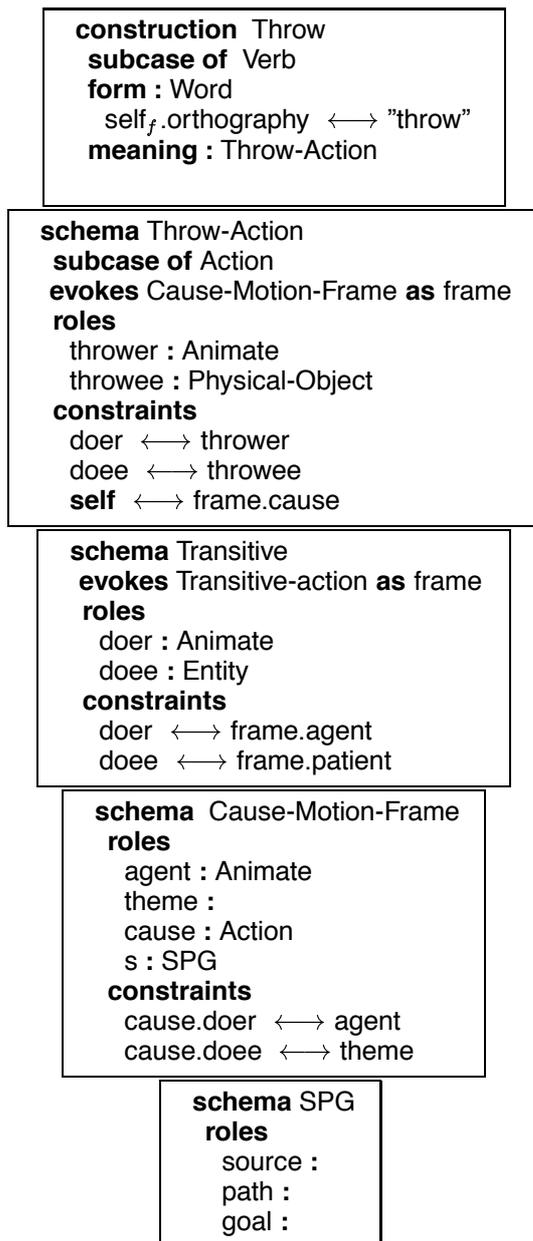


Figure 2: The *throw* lexical construction and associated schemas. This construction defines the meaning of this verb to be the Throw-Action schema. The Throw-Action schema evokes the Cause-Motion-Frame and their roles are coindexed. The SPG schema is a structured representation of a path with a source (starting point), path (way traveled) and goal (end point).

Physical-Object<sup>3</sup>.

Unique to the ECG formalism is the **evokes** oper-

<sup>3</sup>Clearly these selectional restrictions do not apply in all cases. One can certainly *throw a tantrum*, for example. Treatment of metaphorical usage is beyond the scope of both this paper and the system being described.

ator. The evokes operator makes the evoked schema locally available under the given alias. The Throw-Action schema evokes its frame, the Cause-Motion-Frame schema.

The Cause-Motion-Frame schema is the ECG representation of FrameNet's Cause-Motion frame (Baker et al., 1998; The FrameNet Project, 2004). Because *Throw* is a lexical unit associated with this frame, the corresponding Throw-Action schema evokes the Cause-Motion-Frame schema so that their roles can be coindexed. In this case, the thrower is bound to the agent while the throwee is bound to the theme.

The only commitment an evoking schema makes when it evokes some other schema is that the two schemas are related. In this way, The evokes operator provides a mechanism for *underspecifying* the relation between the evoking schema and the evoked schema. Constraints can then be used to make this relation precise.

Semantic frames are a good example of where this ability to underspecify is useful. The lexical item *throw*, for example, only profiles some of the roles in the *Cause-Motion* frame. Using the evokes operator and constraints, the Throw-Action schema can pick out which of these roles are relevant to it. Evokes thus provides an elegant way to incorporate frame-based information.

Constructions with constituents are quite similar to their lexical counterparts with two exceptions. The first is the addition of a **constructional** block to define the constituents. The second is that the form block is now used to define the ordering of the construction's constituents.

Figure 3 shows an example of a construction with constituents—the active ditransitive construction. Within the construction grammar literature, Goldberg (Goldberg, 1995) argues that the ditransitive pattern is inextricably linked with the notion of *giving*. This is represented in ECG by constraining the meaning pole of the ditransitive construction to be of type Giving-Frame.

The Active-Ditransitive<sup>4</sup> construction has four constituents, one for each grammatical function. Its first constituent is named *subject*, for example, and is constrained to be a RefExp (referring expression)<sup>5</sup>

<sup>4</sup>This representation is intentionally naive in regards to the relation between active and passive. Not only is this construction easier to describe in this form, but the language encountered by the model is sufficiently simple such that the constructions do not need to be written in full generality. Though for adult language, this is not the case. For a detailed description of how argument structure constructions can be represented compositionally see (Bryant, 2004).

<sup>5</sup>Referring expressions are constructions with a form pole

<p><b>construction</b> Active-Ditransitive  <b>subcase of</b> Active-Clause  <b>constructional</b>  subject : REFEXP[ANIMATE]  verb : VERB  obj : REFEXP  ind-obj : REFEXP[ANIMATE]  <b>form</b>  subject<sub>f</sub> meets verb<sub>f</sub>  verb<sub>f</sub> meets ind-obj<sub>f</sub>  ind-obj<sub>f</sub> meets obj<sub>f</sub>  <b>meaning</b> : Giving-Frame  self<sub>m</sub> <math>\longleftrightarrow</math> verb<sub>m</sub>.frame  self<sub>m</sub>.donor <math>\longleftrightarrow</math> subject<sub>m</sub>  self<sub>m</sub>.recipient <math>\longleftrightarrow</math> ind-obj<sub>m</sub>  self<sub>m</sub>.theme <math>\longleftrightarrow</math> obj<sub>m</sub></p>
<p><b>schema</b> Giving-Frame  <b>roles</b>  donor : Animate  theme :  means : Action  recipient : Animate  <b>constraints</b>  means.doer <math>\longleftrightarrow</math> donor  means.doe <math>\longleftrightarrow</math> theme</p>

Figure 3: The Active-Ditransitive construction and the Giving-Frame representing its meaning pole. The form block constrains the ordering on the constituents. The **meets** relation means that its left argument must be directly before its right argument. In this construction, for example, the subject constituent must be directly before the verb constituent.

with the semantic type Animate.

#### 4 Semantic Chunking

Chunkers (partial parsers) (Abney, 1996) use finite-state-machines (FSM) arranged into levels, to reliably recognize nonrecursive chunks of syntax. With this approach, each finite state recognizer corresponds to a simple syntactic rule. The levels control the interaction between recognizers, with higher-level recognizers depending on lower-level recognizers.

The semantic chunker that is integrated into the language analysis system uses the same processing scheme as Abney-style partial parsers, extending it to recognize the syntax and semantics associated with ECG constructions. This means that syntactic processing and semantic processing happen simultaneously. As a consequence, semantic information

that looks like an NP and with a meaning pole that refers to something.

is easily integrated to help minimize ambiguity.

Constructions require a very different treatment than the simple syntactic patterns recognized by FSMs. In addition to the straightforward extension of the Abney algorithm to perform unification as well as using a chart, each construction is compiled into a *construction recognizer*.

A construction recognizer searches for its constituents in the input utterance and chart. In addition to satisfying ordering constraints, candidate constituents must also satisfy the type and coindexation constraints associated with the construction. Because of this complexity, construction recognizers are more complicated than FSMs, implementing a backtracking algorithm to search for compatible constituents. For more information about the matching algorithm, see (Bryant, 2003).

#### 5 Integration Using Structure Merging

Without a complete analysis of an utterance, the system must infer exactly how a set of local, partial semantic structures best fit together into a coherent, global analysis of the utterance. The approach taken here is an abductive one in that it assumes compatible structures are the same and merges them.

Motivation for such an approach come from both linguistics and computational approaches to understanding. On the linguistic side, one needs to look no further than what Paul Kay calls the *Parsimony Principle* (Kay, 1987). The Parsimony Principle states that “Whenever it is possible to identify two roles in a text, the ideal reader does so”.

On the computational side, information extraction systems like FASTUS (Hobbs et al., 1996) use an abductive structure merging approach to build up templates describing particular kinds of events like corporate transactions. Their mechanism was intended to quickly build up consistency across utterances. This approach generalizes FASTUS’ approach to work on semantic frames within utterances as well as across utterances.

##### 5.1 Structure Merging Examples

This section illustrates how an extragrammatical utterance and an ungrammatical utterance from the CHILDES corpus (MacWhinney, 1991) can successfully be interpreted using semantic integration through structure merging.

Naomi, a child in the study, wants some flowers. Her father then responds with *I will give you one one flower* with a restart in the description of the final NP chunk. The result of this utterance’s semantic chunking analysis is shown in Figure 4.

The analysis generates two semantic chunks without any links between them. The first chunk

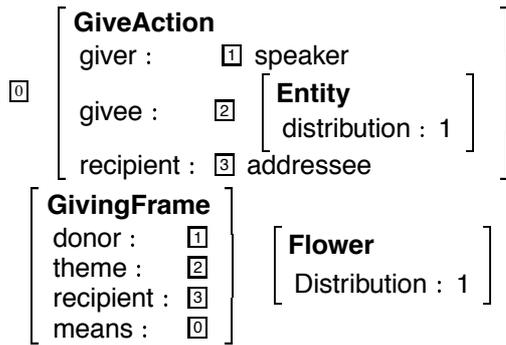


Figure 4: A simplified semantic chunk analysis of *I will give you one one flower* before semantic integration. Notice that the Flower schema is not connected to the rest of the analysis.

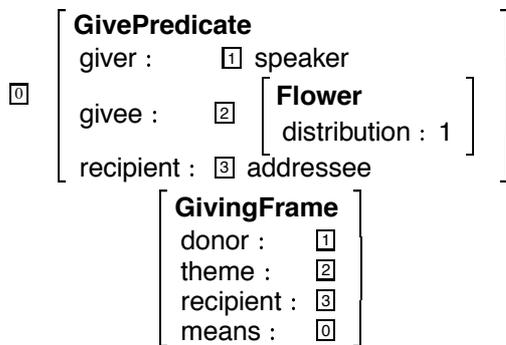


Figure 5: A simplified semantic chunk analysis of *I will give you one one flower* after semantic integration. The Entity and Flower schemas could be merged because Flower is a subtype of Entity and they had the same value for the distribution feature.

(the GiveAction and coindexed GivingFrame) corresponds to the *I will give you one* phrase in which the Giving frame's roles are filled by the speaker, the addressee and an Entity schema corresponding to the word *one*. The *one flower* chunk corresponds to the Flower schema.

Figure 5 shows the integrated analysis. Because the Entity schema and the Flower schema had compatible types<sup>6</sup> and features, the system assumed that they were the same structure. As a consequence, semantic integration generates a complete analysis.

A second more complex example illustrates some of the computational and conceptual issues associated with structure merging: Sometime later, Naomi's father is reading a book to Naomi when he utters the following ungrammatical phrase: *The lamb is looking a donkey*.

Figure 6 shows the chunk analysis of this utter-

<sup>6</sup>Flower is a subtype of Entity

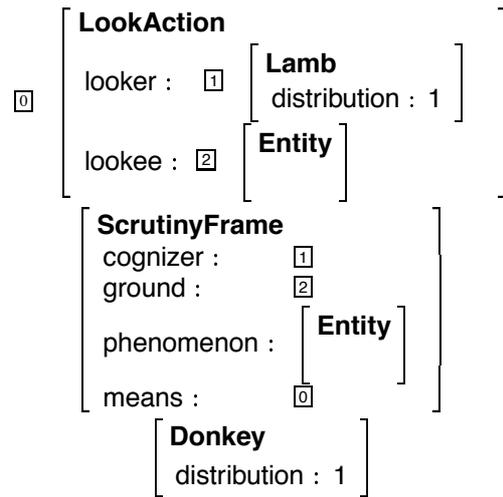


Figure 6: A simplified semantic chunk analysis of *The lamb is looking the donkey* before semantic integration. Notice that the Donkey schema is not connected to the rest of the analysis.

ance. This example generates two chunks because the subcategorization frame of the lexical item *look* is not satisfied by the chunk *a donkey*.

Running semantic integration on this set of chunks, however, results in two different possible integrations. As shown in Figures 7 and 8, the donkey can be merged with the ground or the phenomenon the Scrutiny frame<sup>7</sup> This ambiguity corresponds to whether the missing word in the utterance was intended to be *at* or *for*.

Notice that either integration would be an acceptable interpretation. In other words, these two integrations are equivalent in terms of their semantic acceptability. Certainly, however, leaving the *a donkey* structure unmerged is worse because more of the core elements of the frame would be unfilled. The semantic density heuristic (covered in the next section) formalizes this intuition.

While conceptually it is satisfying for both interpretations to be acceptable, computationally speaking, it is also worrisome. Taking a single analysis and turning it into two because of a single ambiguity signals the possibility of structure merging being NP-hard. For the short utterances associated with child language, this is not problematic. For adult

<sup>7</sup>The FrameNet project defines the Scrutiny frame as: *This frame concerns a Cognizer (a person or other intelligent being) paying close attention to something, the Ground, in order to discover and note its salient characteristics. The Cognizer may be interested in a particular characteristic or entity, the Phenomenon, that belongs to the Ground or is contained in the Ground (or to ensure that such a property of entity is not present)* (The FrameNet Project, 2004).

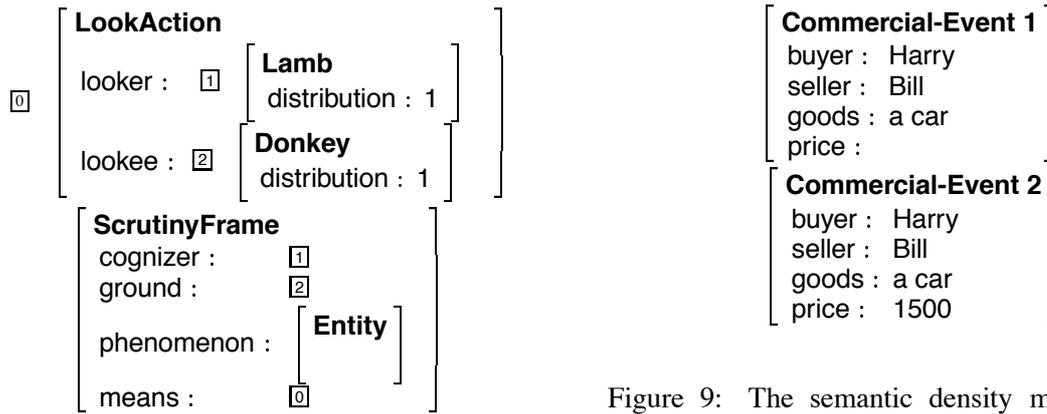


Figure 7: A simplified semantic chunk analysis of *The lamb is looking the donkey* after semantic integration. Notice that the Donkey schema is now merged with the ground role. This corresponds to the *The lamb is looking at the donkey*.

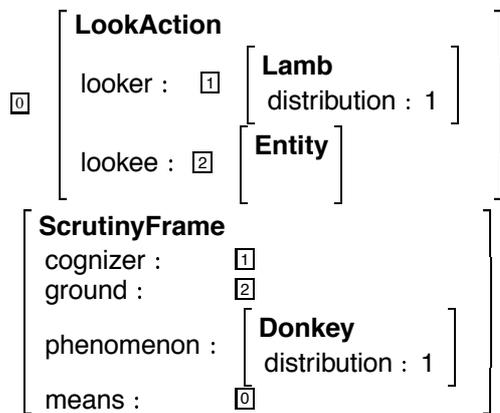


Figure 8: A simplified semantic chunk analysis of *The lamb is looking the donkey* after semantic integration. Notice that the Donkey schema is now merged with the phenomenon role. This corresponds to the *The lamb is looking for the donkey*.

language, however, this could be a serious issue. As such, an open area of research is how to design approximate merging strategies.

## 6 Semantic Density

The key insight behind our approach to making the integrated analyses is the realization that every utterance is trying to communicate a scene (more formally speaking, this scene is a semantic frame). Now assuming that a better analysis is one that more fully describes the scene, one way to compare analyses is by how completely specified the frame is. Those analyses that fill out more of the frame roles should be preferred to those that fill out fewer roles.

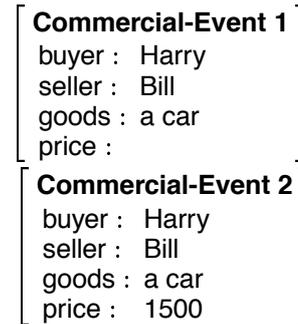


Figure 9: The semantic density metric used to compare two semantic analyses each containing a Commercial-Event frame. The Commercial-Event frame on the top has a semantic density score of .75 and the Commercial-Event on the bottom has a score of 1. Thus the second frame would be considered better because more of the frame elements are filled.

This is the motivation for the ranking heuristic that we call *semantic density*.

Semantic density compares constructional analyses based upon their semantic content. Analyses that have a higher ratio of filled slots to total slots in their semantic analysis are considered better analyses according to semantic density. Figure 9 shows a simple example of the semantic density metric in use.

Let's reconsider the example from the last section regarding *The lamb is looking a donkey*. In that case, there were two possible integrations, one where the donkey was the ground and the other where the donkey was the phenomenon. Applying the semantic density metric to those two competing analyses shows that they have equivalent semantic density. This is consistent with the intuition that either analysis is an equally acceptable interpretation of the input utterance.

The higher-level point here is that there are many ways to semantically analyze a given utterance, and some ways are better than others. While the two semantically integrated interpretations of the lamb sentence were equally good, both were better than the unintegrated utterance. Given a preference for one interpretation over the other, it makes sense to consider semantic interpretation to be a graded phenomenon much like grammaticality.

Keller (Sorace and Keller, to appear) defines the graded nature of grammaticality in terms of constraint reranking within an optimality theory context. Since structure merging and semantic density also define a gradient, they could also be stated within an optimality framework, with the most

dense analyses being considered optimal.

## 7 Related Work

Semantic analysis has a long tradition within NLP. While a broad overview describing logic, frames and semantic networks is beyond the scope of this paper<sup>8</sup>, this work builds on the frame-based tradition as well as the tradition of robust analysis to make progress towards *robust* semantic analysis.

One related traditional approach is Lexical Functional Grammar ((Dalrymple, 2001)). LFG introduced notions of completeness and coherence for its feature structures. These principles (intuitively speaking) require that certain features of an analysis (including all the semantic arguments of a predicate) be filled before an analysis can be considered well formed. Such a constraint is akin to a binary version of the semantic density metric.

Structure merging also builds on historical work. Even before FASTUS, the employment of the parsimony principle within understanding systems goes back to the completely unrelated FAUSTUS system (Norvig, 1987). Norvig's work, however, used graph based algorithms within a semantic network to perform abductive inference.

## 8 Future Work

While implementation is complete, only initial testing of the analyzer against the CHILDES data has been started. To get more complete results, a test grammar covering the parents' utterances must be completed (or learned). Once this has been finished, the number of semantic relations correctly extracted with and without structure merging can be measured.

If verified, the ideas in this paper would have to be appropriately extended to adult language. The structure merging phase, for example, would have to be redefined to use an approximate search algorithm find a good integration. Investigation of the approach described by Beale (Beale et al., 1996) seems promising.

Extremely intriguing is the extension of the semantic density metric. Currently, it is merely a first approximation of semantic preference. One obvious direction is weight different frame roles in accordance with lexical preferences. According to FrameNet data for the lexical item *look*, the *phenomenon* frame role is 50% more likely to be expressed than the *ground* role (The FrameNet Project, 2004). By including such preferences, a more complete notion of semantic preference can

be defined. Narayanan and Jurafsky (Narayanan and Jurafsky, 1998) take the first steps in this direction, integrating syntactic and semantic preferences within a probabilistic framework for a small scale preference task.

## 9 Conclusion

The analysis task found within a learning model exhibits all the canonical problems that motivate robust parsing algorithms: extra/ungrammaticality, restarts, agreement issues etc. Because of these reasons, the language learning setting seems like a good testing ground for robust analysis algorithms.

One such algorithm has been described in this paper. The algorithm is unique because of its heavy reliance on semantics. It uses knowledge at all phases of its analysis, from when it recognizes constructions using a semantic chunking approach, to when it merges the chunks' common semantic structures.

This paper also takes a small step toward defining a gradient notion of semantic analysis. Not all analyses of an utterance are created equal. One simple approach to comparing semantic interpretations is by measuring how completely they specify their schemas and frames. This is how the semantic density metric works.

More important than the particulars of these algorithms, however, is the idea that a learning model and an analysis model should be tightly coupled. Such a model makes it possible for a language understanding system to learn new language from experience. If such an idea can come to fruition, this would be the most robust language analysis algorithm of all.

## References

- Steven Abney. 1996. Partial parsing via finite-state cascades. In *Proceedings of the ESSLLI '96 Robust Parsing Workshop*.
- James Allen. 1995. *Natural Language Understanding*. The Benjamin/Cummings Publishing Company.
- Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet Project. In *Proc. COLING-ACL*, Montreal, Canada.
- Stephen Beale, Sergei Nirenburg, and Kavi Madesh. 1996. Hunter-gatherer: Three search techniques integrated for natural language semantics. In *Proceedings of AAI-96*, Portland, Oregon.
- Benjamin Bergen and Nancy Chang. 2002. Embodied construction grammar in simulation-based language understanding. Technical Report TR-02-004, ICSI. To appear in Oestman and

<sup>8</sup>See (Jurafsky and Martin, 2000) for such an overview or (Allen, 1995) for an introduction with logic.

- Reid, eds., *Construction Grammar(s): Cognitive and Cross Linguistic Dimensions*. John Benjamins.
- John Bryant. 2003. *Constructional analysis*. Master's thesis, UC Berkeley.
- John Bryant. 2004. *Towards cognitive, compositional construction grammar*. Available at [www.icsi.berkeley.edu/jbryant](http://www.icsi.berkeley.edu/jbryant).
- Nancy Chang and Tiago Maia. 2001. *Learning grammatical constructions*. In *Proceedings of the Conference of the Cognitive Science Society*.
- Mary Dalrymple. 2001. *Lexical Functional Grammar*. New York: Academic Press.
- Charles Fillmore, Paul Kay, and M. C. O'Connor. 1988. Regularity and idiomaticity in grammatical constructions: the case of *let alone*. *Language*, 64(3):501–538.
- Charles Fillmore. 1982. Frame semantics. In *Linguistics in the Morning Calm*, pages 111–138. Linguistics Society of Korea.
- Adele Goldberg. 1995. *Constructions: A Construction Grammar Approach to Argument Structure*. University of Chicago Press.
- Jerry Hobbs, Douglas Appelt, John Bear, David Israel, Megumi Kameyama, Mark Stickel, and Mabry Tyson. 1996. *Fastus: A cascaded finite-state transducer for extracting information from natural-language text*. In Roches and Schabes, editors, *Finite State Devices for Natural Language Processing*. MIT Press.
- Daniel Jurafsky and James Martin. 2000. *Speech and Language Processing*. Prentice Hall.
- Paul Kay. 1987. Three properties of the ideal reader. In Roy Freedle and Richard Duran, editors, *Cognitive and Linguistic Analyses of Test Performance*, pages 208–224. Ablex Publishing Corporation.
- Brian MacWhinney. 1991. *The CHILDES project: Tools for analyzing talk*. Erlbaum, Hillsdale, NJ.
- Srini Narayanan and Daniel Jurafsky. 1998. Bayesian models of sentence processing. In *Proceedings of the Conference of the Cognitive Science Society*.
- Peter Norvig. 1987. *FAUSTUS*. Ph.D. thesis, University of California at Berkeley.
- Antonella Sorace and Frank Keller. to appear. Gradience in linguistic data. *Lingua*.
- The FrameNet Project. 2004. <http://www.icsi.berkeley.edu/framenet>.