Incremental Nonmonotonic Sentence Interpretation through Semantic Self-Organization

Marshall R. Mayberry, III
Department of Psycholinguistics
Saarland University
Saarbrücken, Germany 66041
martym@coli.uni-sb.de

Risto Miikkulainen
Department of Computer Sciences
The University of Texas
Austin, TX 78712
risto@cs.utexas.edu

Technical Report AI08-12
December 2008

Abstract

Subsymbolic systems have been successfully used to model several aspects of human language processing. Yet, it has proven difficult to scale them up to realistic language. They have limited memory capacity, long training times, and difficulty representing the wealth of linguistic structure. In this paper, a new connectionist model, InSomNet, is presented that scales up by utilizing semantic self-organization. InSomNet was trained on semantic dependency graph representations from the Redwoods Treebank of sentences from the VerbMobil project. The results show that InSomNet learns to represent these semantic dependencies accurately and generalizes to novel structures. Further evaluation of InSomNet on the original spoken language transcripts shows that it can also process noisy input robustly, and its performance degrades gracefully when noise is added to the network weights, underscoring how InSomNet tolerates damage. It interprets sentences nonmonotonically, i.e., it generates expectations and revises them, primes future inputs based on semantics, and coactivates multiple interpretations in the output. In other words, while scaling up it still retains the cognitively valid behavior typical of subsymbolic systems.

1 Introduction

The empirical approach to natural language processing over the past twenty years has emerged through a variety of statistical learning techniques. These techniques have capitalized on large corpora from which knowledge can be automatically extracted rather than stipulated *a priori*. Statistical techniques built on symbolic representations have proven successful on tasks such as part-of-speech (POS) tagging and syntactic parsing. On the other hand, in domains for which a symbolic description would be virtually impossible, such as robust sentence processing and modeling various types of human aphasias, the subsymbolic approach has proven equally powerful.

These *subsymbolic* (alternatively, connectionist, or neural network) systems are inspired by the biological brain as systems that process information through massively interconnected simple processing units operating in parallel. In this framework concepts are not represented by discrete symbols, but rather are distributed across all processing units. These *distributed representations* automatically give rise to a number of plausibly cognitive behaviors as revealed by psycholinguistic studies of human sentence processing (HSP). They are robust to noise, damage, and incomplete information; they may represent multiple concepts in parallel that are graded with respect to the statistical properties of their training data; and they generalize automatically to new input.

Despite their success in modeling cognitive effects in natural language processing tasks, the main challenge for subsymbolic models has been scaling them up to the complexity of real-world language. The principal problem is representation. So far, these systems have been limited to relatively simple syntactic or case-role analyses of sentences from toy grammars. In order to be useful as large-scale cognitive models, subsymbolic systems need to be able to incrementally render a sentence into a detailed semantic interpretation that indicates how constituents in the sentence, such as words and phrases, are related.

The purpose of this article is to show that incremental, robust semantic interpretation of sentences from a sizable real-world corpus of dialogues using neural networks results in cognitively plausible behavior. A subsymbolic system can be trained to read a sentence with complex grammatical structure into a holistic representation of the semantic features and dependencies of the sentence. This research breaks new ground in three important respects. First, the model described in this article, the Incremental Nonmonotonic Self-Organization of Meaning Network (InSomNet), is the only subsymbolic system to convert sentences into explicit and detailed semantic representations derived from a hand-annotated treebank of real-world sentences. Second, whereas almost all previous work has focused on the representation and learning of syntactic tree structures (such as those in the Penn Treebank), InSomNet produces semantic dependency graphs. Third, the challenge of developing a subsymbolic scheme for handling such graph structures led to the method of self-organizing the semantic frames in a novel way to represent structure subsymbolically, resulting in a number of interesting cognitive behaviors that will be analyzed in this work.

In Section 2 below, the cognitive phenomena InSomNet was designed to model and the semantic formalism that contributed to the design of the model's architecture are reviewed. Section 3 then presents a detailed description of the InSomNet architecture. Section 4 demonstrates that InSomNet scales up to a sizable corpus of semantically annotated sentences. Section 5 demonstrates InSomNet's robustness to noisy input as well as damage to the network itself. In Section 6, the cognitive plausibility of the model is evaluated with respect to its psycholinguistic validity. These results are discussed in relation to each other in Section 7, and Section 8 concludes the article.

Throughout the article, the following font conventions are adopted:

- italics refer to lexical items (*boy*) and predicates (*hit*);
- bold refers to predicate labels (**h1**), and subcategorization (**IX**) and thematic roles (**Agent**);
- sans serif refers to network components (Hidden Layer and Map Node); and
- small caps refer to network architectures (InSomNet and Srn).

2 Background

The symbolic and statistical NLP communities utilize syntactic grammars to parse a sentence into a unique, linguistically preferred, syntactic tree as typically determined by grammaticality judgments with respect to acceptability (Chomsky, 1965). Because such grammars tend to license a multitude of possible parse trees for a given input sentence, it is necessary to rank the possible parses produced by a grammar so that the preferred parse tree reliably comes out on top. Such parse selection is usually called *parse disambiguation*, but some of the the ambiguity that makes ranking multiple parse trees licensed by a grammar often stems from the lack of sufficient constraints in the grammar itself to eliminate ungrammatical sentences rather than inherent properties of language. Yet, as noted in a footnote on page 1 in Collins (1999), parsing can also be understood as the task of finding all syntactically well-formed parse trees for a sentence, setting aside the burden of disambiguation and ranking. This latter notion of parsing has generally been adopted in the psycholinguistics community where the nature of ambiguity itself and how humans so effortlessly accommodate it is of theoretical interest. To make this distinction clear, *semantic parsing* (sometimes simply called *comprehension* in the computational model literature, Rohde & Plaut, 2003) is taken as the task of converting a sentence into a linguistically well-founded semantic interpretation based on the thematic roles of the words in the sentence, albeit one that may implicitly feature syntactic constraints. The term comprehension is used in this sense in this article, and *parsing* is then restricted to the identification and activation of the linguistic structures comprising an interpretation. Subsymbolic systems, as primarily models of human performance rather than grammatical competence, usually superimpose the representations of all interpretations for a given sentence in a single vector of real numbers, with each interpretation graded according to likelihood.

This section begins with a description of the particular cognitive behaviors that InSomNet has been designed to model. The foundational components of subsymbolic systems are then presented, followed by a chronology of previous subsymbolic approaches to cognitively plausible sentence processing. Finally, the section concludes with a detailed description of the linguistic foundations of the semantic formalism used to annotate complex linguistic representations of sentences drawn from the established VerbMobil corpus of business travel-arrangement dialogues (Wahlster, 2000).

2.1 Cognitive Issues in Natural Language Understanding

A key characteristic of natural language is *ambiguity*. Ambiguity is believed to arise as a natural consequence of situated language acquisition: A child must learn the combinatorically daunting task of grounding concepts denoted by sounds in a language to objects and events in its immediate environment (Altmann, 1998), a process that cannot be guaranteed to be one-to-one. Ambiguity pervades language at all levels of traditional linguistic description, from phonetic to morphological to lexical to syntactic to semantic to pragmatic to discourse, and the boundaries between these levels are not always distinct. This categorical fuzziness is but one manifestation of *gradience* in natural language (Crocker & Keller, 2006) and, more particularly, encapsulated in the *grain problem* (Mitchell, Cuetos, Corley, & Brysbaert, 1995): Which level of description or some weighted combination thereof is most informative to natural language understanding and when? How are these levels delineated?

The ubiquitous prepositional phrase-attachment ambiguity, that is used in this paper to motivate both the design of the InSomNet model and how it represents a sentence, is comically captured in Groucho Marx's famous quip:

> I shot an elephant in my pajamas. How he got in my pajamas, I'll never know.

Prepositional phrase attachment ambiguities cannot be reliably resolved on the basis of syntax alone. In the most general case, world knowledge and context are the only guides to the intended meaning. Indeed, the reason the quip is funny is that the second sentence forces the reinterpretation of the first sentence: People assume "in my pajamas" modifies the verb to indicate that the subject is wearing them, based on their own personal experience. The revised modifier interpretation in the immediate context provokes the humorous mental image of an elephant in pajamas.

In the worst case, ambiguities give rise to the much-studied pathological *garden-path* sentences that sometimes force reanalysis (Small, Cottrell, & Tanenhaus, 1988; Marslen-Wilson, 1989). In the general case, however, people resolve local ambiguities in real-time without even being overtly aware that there are other possible interpretations. This *facility* is another characteristic of natural language: Scarcely a sentence is formed spontaneously that is without some type of error ranging from minor ungrammaticalities to dysfluencies (Crocker & Keller, 2006). Yet, people are able to communicate with one another with remarkable ease, speed, and accuracy. This *Performance Paradox* for *garden-variety* sentences underscores the characteristic *robustness* of human language understanding (Crocker, 2005).

One likely insight into the facility with which people understand everyday language comes from the now well-established characteristic of *incremental processing*: Sentence comprehension proceeds apace of reading or speech production with minimal buffering, as it must if memory resources are not to be exhausted by the combinatoric explosion of interpretations that would result if comprehension required the indefinite retention of sentence components until enough information had been accumulated to permit disambiguation (Garrett, 1990; Barton, Berwick, & Ristad, 1987; Church & Patil, 1982). Although Jurafsky (1996) suggested that dynamic programming for parsing control and rule integration may alleviate some of this inefficiency at the syntactic level, he nevertheless conceded that it may not generalize to interpretation as a whole.

Incremental interpretation, on the other hand, is more tractable because unviable interpretations can be pruned away, limiting combinatorial complexity. In statistical models, where only a given number of the most promising candidates are maintained at a time in a beam of candidate parses, the utility of this approach is obvious, but suffers from the drawback that pruned away candidates are unrecoverable without explicit reanalysis. In subsymbolic models, on the other hand, it is unusual for a trace of even an unlikely interpretation to vanish completely, but these models are susceptible to the aforementioned limited memory capacity that affects all interpretations (Bengio, Simard, & Frasconi, 1994), although accurate recovery of unlikely interpretations can be quite problematic. However, much headway has been made in the past two decades to improve the memory capacity of subsymbolic models, resulting from refinements of the standard feedforward network (Ffn) backpropagation algorithm (Rumelhart, Hinton, & Williams, 1986), such as Backpropagation-through-time (Bptt; Williams & Zipser, 1989; Lawrence, Giles, & Fong, 2000) and Long Short-Term Memory (Lstm; Hochreiter & Schmidhuber, 1997).

Human language comprehension is also characterized by the real-time *integration* of diverse sources of information. In addition to the integration of words in the input into a developing interpretation of an isolated sentence on the basis of lexical, syntactic, and semantic information sources, studies also show that non-linguistic sources of information bear directly on comprehension in situated language settings. The ability of people to integrate diverse sources of information is part and parcel of their ability to *adapt* to the (dynamic) availability of those sources, while at the same time weighting their relevance to an unfolding sentence.

The case for incremental interpretation is strongly supported by psycholinguistic studies. The seminal work of Marslen-Wilson (1975), for example, not only supported the view that sentence interpretation is incremental, but also was one of the first studies to highlight its integrative and adaptive characteristics:

> This report presents evidence that sentence perception is most plausibly modeled as a fully interactive parallel process: That each word, as it is heard in the context of normal discourse is immediately entered into the processing system at all levels of description, and is simultaneously analyzed at all these levels in the light of whatever information is available at each level at that point in the processing of the sentence.

These characteristics of incremental, integrative, and adaptive language processing are buttressed by the observation that people actively *anticipate* likely sentence continuations (Federmeier, 2007). Research by Altmann and Kamide (1999) showed that linguistic knowledge such as selectional restrictions of verbs could facilitate prediction of upcoming role fillers. Further evidence for these characteristics of language processing has accrued from numerous studies in the *visual world paradigm*, wherein people's gazes in a visual scene are monitored while they listen to an utterance that may or may not be relevant to the scene. What people look at in a scene often reveal how they are understanding an utterance.

There is further evidence that multiple interpretations of a sentence are initially coactivated and the nonviable interpretations later suppressed (Dahan & Gaskell, 2007; Hudson & Tanenhaus, 1984). Moreover, not all ambiguities are always resolved, and indeed some ambiguities, such as modification by a prepositional phrase, may remain underspecified (i.e., the prepositional phrase may not be said conclusively to modify any particular constituent, and may even modify several at the same time; Hindle & Rooth, 1993).

That is, natural language representations are not simply compositional, but consist of multiple superimposed and weighted interpretations in parallel that often represent more than the sum of their parts. Nor do these representations accrue monotonically as a sentence is processed, with each word processed incorporated into a developing interpretation with minimal structural modification. Rather, evidence suggests that human sentence processing is also *nonmonotonic*. Nonmonotonicity can be seen in many guises, ranging from outright reanalysis of a sentence online to the modulation of coactivated interpretations due to structural and semantic priming (Onifer & Swinney, 1981; MacDonald, Just, & Carpenter, 1992; MacDonald, 1993), and thus arises as a direct result of both incremental and anticipatory sentence processing. Connectionist models exhibit similar behavior. For instance, consider the sentence *The princess threw the ball in the stadium for the pitcher/diplomat.* (Mayberry & Miikkulainen, 1994). The first noun, *princess*, that is processed biases the interpretation of *ball* as a formal dance, but the next noun, *stadium*, re-biases it toward an article of sports equipment. If the final noun in the sentence is *diplomat*, it overrides the interpretation of the sports sense back toward the formal-dance sense of ball; if it is *pitcher*, it reinforces the sports sense. This behavior, termed *semantic flipping*, is also evident in people as they navigate several possible sentence continuations. Words processed later in a sentence can change the developing interpretation radically, but the network can still recover from an early interpretation that is at odds with later words. This way, the network can more effectively resolve lexical ambiguities, attachments, and anaphoric references during the course of incremental interpretation (cf., Onifer & Swinney, 1981; MacDonald et al., 1992; MacDonald, 1993). A cognitively plausible model should exhibit similar nonmonotonicity. The components of such a model are described next.

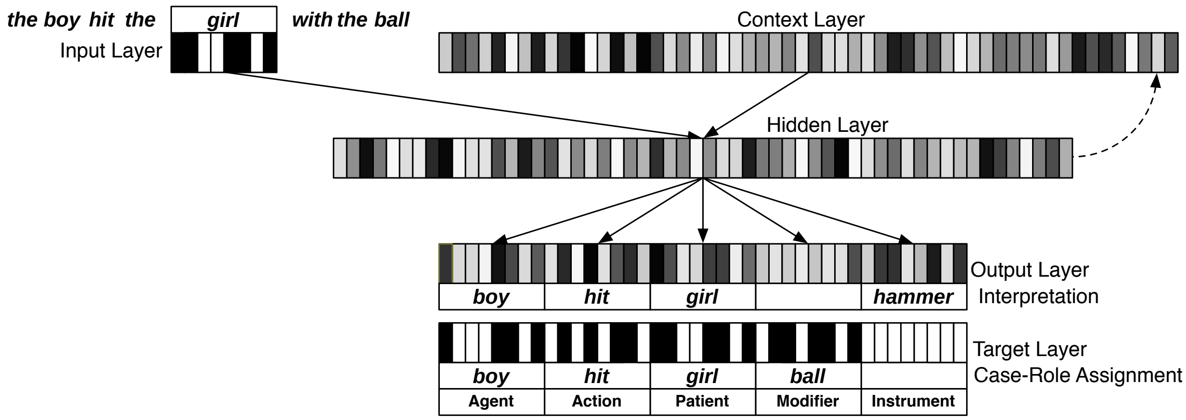


Figure 1: **Case-role analysis with an SRN.** The input sentence *the boy hit the girl with the ball* is presented to the case-role analysis network. The activation pattern of the current input word, *girl* and the Context Layer that represents the sentence processed thus far is propagated through a set of weights to the Hidden Layer. The Hidden Layer activation is further propagated to the five Output Layer assemblies chosen to stand for selected semantic roles. The network anticipates the word *hammer* as an instrument for *hit*, based on its training experience. However, the weakly activated modifier *ball* will become strongly activated when that word is later read in as input. In this manner, SRNs can be used to convert sequential input into a linguistic representation that evolves over the course of processing.

2.2 Connectionist Foundations

Three types of neural networks have dominated the connectionist sentence processing field: the *Simple Recurrent Network* (SRN; Elman, 1990) for processing temporal sequences such as sentences, the *Recursive Auto-Associative Memory* (RAAM; Pollack, 1990) for linguistic structures represented as trees, and the *Self-Organizing Map* (SOM; Kohonen, 1990, 1995) for clustering these representations.

2.2.1 The Simple Recurrent Network (SRN)

The SRN is essentially an FFN modified to process sequences of patterns. Figure 1 shows an SRN in the middle of processing the sentence *the boy hit the girl with the ball* with the word representation for *girl* loaded into the Input Layer. The Context Layer holds a copy of the previous Hidden Layer representing the sentence prefix *the boy hit the ...* that has already been processed. The Input and Context Layers are propagated together to the Hidden Layer, which forms a representation to serve as memory for the next word in the sentence, *with*. The Hidden Layer representation is then propagated to the Output Layer, for which the sentence-final interpretation (boy_{agt} hit_{act} $girl_{obj}$ $ball_{mod\ -\ ins}$) serves as the target. As shown in Figure 1, this interpretation specifies that *ball* is a modifier of *girl*, not an instrument of *hit* (which is thus left blank). The Hidden Layer is copied to the Context Layer, the representation for the next word, *with*, is loaded into the Input Layer, and the process repeated in this manner for each subsequent word in the sentence.

As each word in the sentence is read in, the memory represented by the Hidden Layer degrades. The reason is that the Hidden Layer is a fixed-length vector, and is forced to represent previous items less accurately in order to accommodate new ones. This *memory problem* has limited the use of the SRN to learning simple sentences and targets such as localist or feature-encoded word representations.

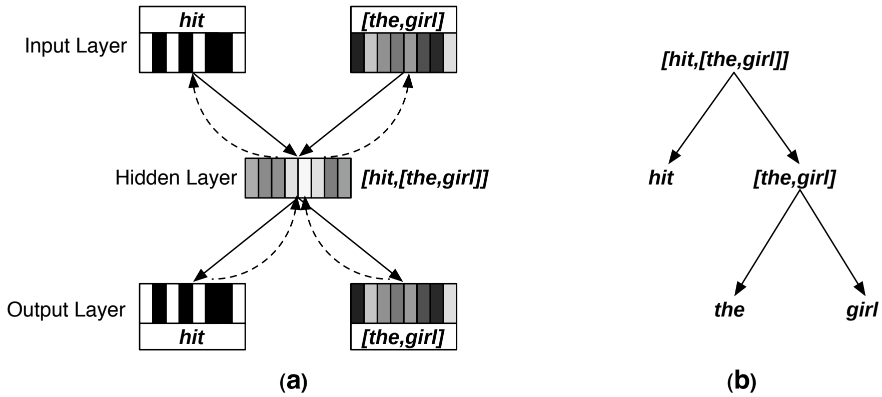


Figure 2: **Encoding Compositional Structure with** Raam**.** In order to encode a tree structure, such as that in (b), into a fixed-length representation, the tree's constituents need to be compressed. In the binary tree Raam shown here in (a), two input representations, *hit* and *[the,girl]* are combined into a single representation, *[hit,[the,girl]]*. This compressed representation develops in the Hidden Layer through autoassociating the Input and Output Layers. The subtree *[the,girl]* is itself a compressed representation of its constituents, *the* and *girl* developed in the Hidden Layer through autoassociation. In this manner, progressively deeper structures can be encoded recursively from simpler constituents. All the weights (indicated by solid arrows) are trained through backpropagation. However, accuracy of repeated compressed encodings is limited due to memory degradation.

InSomNet is based on the line of research that uses Srns to map a sentence into a set of case roles in a manner similar to that shown in Figure 1. Based on the theory of thematic case roles (Fillmore, 1968), case-role analysis assumes that the syntactic structure of the sentence is specified beforehand, and the goal is to assign the proper roles to the words in the sentence. For example, given a simple sentence with subject, verb, object, and a *with*-clause, the network's task is to assign those words to the thematic roles **Agent**, **Act**, **Patient**, and **Modifier** or **Instrument** depending on the selectional preferences of the words in the training corpus (as determined by frequency of correlation; Miikkulainen & Dyer, 1991; McClelland & Kawamoto, 1986). As before, the sentence is read in one word at a time, and the network is trained to map the sentence into the words that fill the case roles for that sentence. By giving these same words as static targets for each word of the sentence, the network will develop expectations and defaults. In Figure 1, the network has only read the word *girl*, but is already expecting *hammer* as an **Instrument** because sentences in the training corpus with *hit* occur most frequently with *hammer* as an **Instrument**. When the word *ball* is later read in, both the **Modifier** and **Instrument** slots assume representations for *ball*, reflecting their equal frequency in the corpus. If, instead of *ball*, the word were *doll*, which only occurs in the modifier sense, then the **Instrument** slot would become deactivated. This strengthening and weakening of multiple coactivated senses shows how the network nonmonotonicity resolves ambiguities. This behavior is desirable for a cognitively motivated system, and has served as a constraint on the implementation of InSomNet.

2.2.2 Recursive Auto-Associative Memory (Raam)

In order to represent structures in a neural network, it is necessary to be able to combine two or more substructures into one. The Raam network shown in Figure 2 provides such a mechanism. Raam is

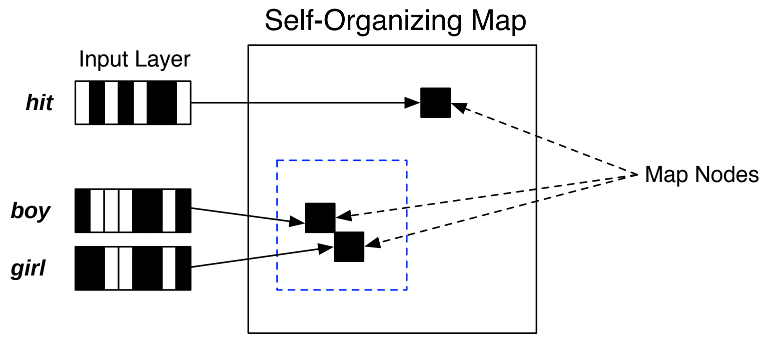


Figure 3: **Clustering Representations with** SOM**.** The figure shows three input patterns *hit*, *boy*, and *girl* mapped to a Self-Organizing Map (SOM). There is a single set of weights between the Input Layer and each Map Node of the SOM. Trained on a set of patterns, the SOM effectively clusters its input space while preserving its topology. Thus, similar input patterns, such as *boy* and *girl* in the figure, will be mapped to neighboring nodes, while dissimilar patterns, such as *hit*, are mapped to a Map Node in a different region of the map. The self-organization algorithm develops a local response by activating the Map Node whose weights are closest to an input pattern. These weights and those of neighboring Map Nodes are then adapted towards the input's activation pattern. Over time, nearby Map Nodes become similar and thus respond to similar input patterns. Such a map can be used to represent multiple linguistic structures distinctly.

also a two-layer FFN backpropagation network that is trained to reproduce, or *autoassociate*, its input at its output by developing a compressed representation of its input in the Hidden Layer. Because the compressed representation has the same length as its constituents, it can be used for building up yet deeper structures in a recursive fashion.

For example, in Figure 2, the tree *[hit,[the,girl]]* is encoded from *hit* and *[the,girl]*. The subtree *[the,girl]* has already been encoded earlier from *the* and *girl*. The leaf representations for *hit*, *the*, and *girl* must be specified beforehand. Usually they are given a simple localist or feature encoding.

Like the SRN, RAAM suffers from a memory problem. As deeper structures are encoded, information from earlier constituents such as the leaves themselves is gradually degraded to the point where it cannot be accurately recovered. The reason is that te Hidden Layer has a fixed length and, consequently, the more items it is forced to represent, the less accurately it is able to represent them. Therefore, RAAM has proven difficult to scale up to the complexity of natural language.

2.2.3 Self-Organizing Map (SOM)

The SOM is a neural network that develops a two-dimensional clustering of its input space of patterns. It preserves the topology (i.e., relative distances between patterns) of that space by mapping similar input patterns to nearby Map Nodes on the SOM. In Figure 3 each input pattern is shown mapped onto a particular Map Node called the maximally-responding unit, or *winner*, as typically measured by the Euclidean distance between the node's weights and the input pattern.

The training algorithm proceeds by randomly initializing the weights between each Map Node in the map and the Input Layer and setting the *neighborhood* (the dashed square in Figure 3) to half the diameter of the map. The input patterns are presented one at a time in random order and the map is searched for the Map Node whose weights most closely match the input pattern.

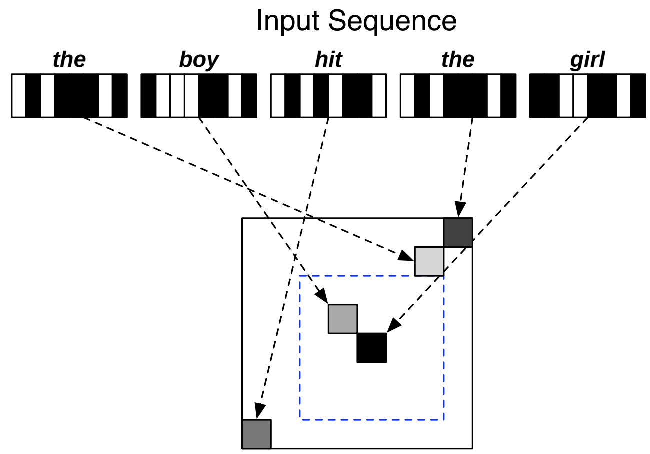


Figure 4: **The Sequential Activation Retention and Decay Network.** SardNet is a Som for sequences, and is also trained using a variant of the Som algorithm. The primary modification is that winning units remain activated, but are removed from further competition for the remainder of the sequence and decayed by a predetermined factor. In this manner, the map develops a unique representation for the input sequence. Accordingly, SardNet is used to alleviate the memory problem of the Srn.

The weights of the winner and all the nodes in its neighborhood are updated according to the Som adaptation rule to approximate the current input better (Kohonen, 1990, 1995) and coerce neighboring nodes to represent similar patterns. The neighborhood and *learning rule*, which is the amount of weight update on each trial, are reduced according to a predefined schedule that often is linearly or exponentially decaying. When the process of self-organization is completed (i.e., inputs are consistently mapped to corresponding Map Nodes, the weights between each Map Node and the Input Layer come to represent prototypes, called *codebook vectors*, that cluster the input patterns into regions on the map with similar features, weighted according to their respective frequencies in the input space.

The Sequential Activation Retention and Decay Network (SardNet; James & Miikkulainen, 1995; Mayberry & Miikkulainen, 1999b), shown in Figure 4, is a variant of the Som designed to cluster *sequences* of input patterns in addition to the patterns themselves. As each pattern is read in, the corresponding winning Map Node in SardNet is activated at a value of 1.0, the rest of the map decayed by a factor such as 0.9, and the winning node removed from further competition. If the same input pattern occurs more than once, or two input patterns map to the same node, the next closest available node is activated. The SardNet architecture can be used to greatly ameliorate the memory problem of the Srn, as will be shown in Section 3.3.1.

2.3 Previous Connectionist Models of Sentence Processing

A large number of connectionist models of sentence processing have been proposed over the years that make use of the Srn and Raam. They all face the same challenges of long training time, weak memory for sequences, and the associated difficulty in building up sizable structures. As a result, a few symbolic-subsymbolic hybrid systems have been developed that use symbolic components to get around these problems such as ParSec (Jain, 1991) and Nnep (Henderson, 1994). However, they suffer many of the drawbacks associated with symbolic systems, including brittleness, catastrophic failure, and the need to hand-code knowledge into the system. Both the purely subsymbolic and hybrid approaches will be briefly reviewed in this section.

2.3.1 Jain's ParSec System

ParSec (Jain, 1991) is an incremental connectionist parser of spoken language that learns to assign words into a three-level structure for phrases, clauses, and the full sentence through a series of feed-forward "mapping" modules. The parser was trained on the *Conference Registration Task*, which consisted of 204 sentences taken from a 400-word lexicon. It proved to be robust to speech effects such as restarts and repairs, as well as ungrammaticality and fillers, when tested on the *Air Travel Information Service* (ATIS) task. However, there is little structural information in the output representations, other than that implied by the phrases and clauses themselves. How these constituents relate to each other is left unspecified. Regardless, ParSec clearly demonstrated some of the most attractive properties of neural networks such as robustness to noisy and ungrammatical input.

2.3.2 Reilly's Raam-based Parser

Reilly demonstrated one of the earliest and henceforth most common approaches to subsymbolic parsing of syntactic structure (Reilly, 1992). A Raam network was first trained to encode a set of 16 syntactic parse trees into distributed representations. An Srn was then trained to read the representation of the sentence one Part-of-Speech (POS) tag at a time, outputing the representation of the complete parse tree at every step. For example, as each item in the input sequence `d n p d n v d a n` was read in successively, the Srn attempted to produce the parse result `[[[d,n],[p,d,n]]],[v,[d,[a,n]]]]`. The Srn was then tested on four novel sentences, and the final parse results were decoded to determine how well the system learned the constituent structures.

Reilly's architecture was able to learn 11 of the 16 training sentences. In the four test sentences, it was able to produce some substructures correctly, but failed to generate the correct parse for the complete sentence. Although Reilly did not directly state that the Srn is unable to retain enough information in memory to produce the correct parse at the output, this problem can be inferred from the results given in the paper. The Srn failed to learn those trees with the deepest structure in both the training and test sets. Reilly did, however, fault the limited capacity of both the Raam and Srn networks for the poor generalization to novel sentences.

2.3.3 Sharkey and Sharkey's Modular Parser

Sharkey and Sharkey took Reilly's approach a step further by incorporating a feedforward network to map the output from an Srn to Raam parse tree representations (Sharkey & Sharkey, 1992). Three different connectionist architectures, an Srn, a feedforward network, and a Raam, were all trained separately and then combined into a four-layer architecture that could parse structurally ambiguous sentences in terms of prepositional phrase attachment, embedded clauses and relative clauses.

The Raam decoder network was trained to develop the structures that served as output of the integrated network. The Srn encoder network was trained on the standard prediction task of the next work in the input. The feedforward network was trained to map the Hidden Layer representation at the end of the process to the corresponding RAAM representation so that the structure could be decoded into its proper constituents. The network was able to disambiguate all of the 1280 sentences on which it was trained and 75.6% of the 320 novel sentences in the test set. Cluster analysis of the hidden unit activations revealed that the SRN was capturing significant structural information.

Sharkey and Sharkey's model was one of the first purely modular approaches to connectionist sentence processing, but generalization was still very limited. Sharkey and Sharkey did not provide an explicit analysis of the actual errors that the network made, concentrating their discussion rather on the structures that the network was able to capture, but the poor generalization results implicate the limited memory capacity of the SRN parser and the RAAM decoder.

2.3.4 Berg's XERIC Parser

In the XERIC parser (Berg, 1992), RAAM and SRN were integrated into a five-layer architecture. The encoder part of a RAAM network was inserted between the input and Context Layers and the Hidden Layer of the SRN, and the decoder part was inserted between the Hidden Layer and the Output Layer. The input was a feature encoding of the current word, which was propagated through the encoder to the Hidden Layer, and further propagated through the decoder to the Output Layer. The output corresponded to a template structure based on a simplified form of X-Bar theory (Sells, 1985), composed of a specifier, a phrasal head, and up to two complements. Features included the syntactic category of the word, its number and person, and several parameters that depended on the category (such as tense for verbs). Additionally, a nine-unit ID tag distinguished words with identical syntactic features.

The network was trained on a corpus of 1000 fairly simple, unambiguous sentences using a variant of BPTT modified for RAAM. Training was not perfect, with accuracy ranging between 91% and 99% depending on the depth of the X-Bar structures used to represent the sentences. On the testing corpora (also consisting of 1000 sentences), the accuracy ranged between 89% and 99%. Berg's analysis showed that 53% of the errors the network made were due to the ID tags in the output units. However, the ID units constituted only a tenth of the total number of output units. Thus, a disproportionate number of the errors resulted from distinguishing sentence constituents. This type of error is a hallmark of the memory problem that has prevented the scaling of recurrent neural networks to more than a small inventory of interesting linguistic phenomena.

Nonetheless, XERIC was one of the first integrated architectures that could handle and represent complex recursive phrase-structure constructs. An integrated architecture is desirable because it makes it possible to encode and build structure simultaneously without losing valuable error information at module interfaces.

2.3.5 Ho and Chan's Confluent Preorder Parser (CPP)

The Confluent Preorder Parser (CPP; Ho & Chan, 1997) also learned to process sentences one word at a time, but the output representations were generated by a preorder traversal of the sentences' parse trees by a variant of RAAM designed to process sequences called SRAAM (Pollack, 1990). Because the sentence consisted of only terminals, the sequence generated by the preorder traversal yielded more elements than there were in the sentence itself. To deal with them, Ho and Chan (1997) used a dual-ported SRAAM (Chrisman, 1991) to develop representations for the preorder traversal sequences. In the dual-ported SRAAM, a process called confluent inference forced the Hidden Layer to develop representations that subserved two different tasks: the development of a representation of the surface sentence and, at the same time, the preorder traversal of that sentence's parse tree.

The network was trained on 82 sentences, and tested on 30 sentences from a syntactic grammar adapted from Pollack (1990) and was able to generalize to 93.75% of the test set. This is a reasonable amount of generalization, but the corpus of sentences was relatively small and uncomplicated, and featured no semantic constraints that might have made the task easier.

The authos noted that a drawback of this approach was that the network did not develop representations that corresponded to the internal structure of a sentence. For example, internal syntactic phrase structures within a POS-tag sentence, such as `p d a n`, would still need to be trained separately for the parser to develop a representation (e.g. `[p,[d,[a,n]]]`) for those structures due to the preorder traversal strategy. The parser only learned to parse complete sentences, which was undesirable given the wealth of psycholinguistic research showing such phrases are processed as meaningful constituents in themselves and then integrated into the meaning of the sentence that is being developed during parsing.

2.3.6 Miikkulainen's SPEC

The Subsymbolic Parser for Embedded Clauses (SPEC; Miikkulainen, 1996) is a system based on the SRN (the Parser) specifically designed to handle relative clauses. In order to accomplish this task, two other components, the Stack and the Segmenter, were included in the system. The Parser read in words one at a time and formed a case-role representation at its output. The Stack was a RAAM network used to save and restore the context of the Parser as the relative clause segments were encountered during parsing. The Segmenter, a simple FFN, used the Parser's Hidden Layer and the next input word to learn clause boundaries. It used the boundary information to control the execution of the Parser via the right context from the Stack, depending on the level of the relative clause being parsed. Each module was trained separately with basic clause constructs so that, when the system was integrated, it could generalize to novel sentence structures. Trained on only 100 randomly selected sentences from a simple context-free grammar with semantic restrictions (Elman, 1991), the network successfully parsed the entire corpus of 98,100 sentences.

When the representation on the Stack was artificially lesioned, SPEC exhibited plausible cognitive performance. Shallow center embeddings were easier to process, as were sentences with strong semantic constraints in the role bindings. When the Parser made errors, it usually switched the roles of two words in the sentence, as do people in similar situations. A symbolic representation of the Stack would make modeling such behavior very difficult.

SPEC generalized remarkably well, but that generalization ability was essentially due to its specialized design for relative clauses. Moreover, the output for the Parser was simply case-role representations which were only as deep as the level of embedding of relative clauses. However, an investigation into the cognitive performance of the network revealed the expected memory limitations of the Stack and the Parser resulting from repeated compressions.

2.3.7 Lane and Henderson's Ssn

The Simple Synchrony Network (Ssn; Lane & Henderson, 2001) is a Srn-based system that used Temporal Synchrony Variable Binding (Tsvb; Shastri & Ajjanagadde, 1993) to represent structures and generalize across sentence constituents. The introduction of Tsvb was motivated by the *binding problem*, which arises when multiple entities are each represented with multiple features. Tsvb indicates which features are bound to which entities through the use of synchrony of activation pulses. A Srn component of Ssn handled the non-pulsing units in the system that were designed to represent information about the sentence as a whole, while the pulsing units contained information about constituents. This approach was developed to deal with the $O(n^2)$ potential parent-child dependencies between constituents. The use of incremental processing allowed the Ssn to focus on how to incorporate the current input word into the syntactic structure that had been processed up to that point.

Like Ho and Chan (1997)'s parser, the Ssn was not demonstrated on a corpus of real sentences, but only on a simplified corpus consisting of part-of-speech (POS) tags. As the Ssn incrementally processed a sentence, it generated POS tag for the parent, as well as the intermediate parent-child dependencies holding between that parent and previous sentence constituents. In this way, all parent-child dependencies between parent and children accumulated in the output, so that the full syntactic structure of the sentence was generated piecemeal in the output at the end of the sentence. To recover the complete parse, these structures had to be reassembled off-line.

Applied to a subset of the SUZANNE corpus, the most successful version of the architecture achieved between 70% and 80% average precision/recall, which is comparable to the performance of contemporary parsers using Probabilistic Context-Free Grammars (Pcfgs).

2.3.8 Rohde's Cscp

The Connectionist Sentence Comprehension and Production model (Cscp; Rohde, 2002) is a system based on the Srn that was designed to both comprehend a sentence incrementally and to reproduce it from its compressed encoding in a 500-unit hidden Message Layer. The system was composed of two modules, the Message Encoder/Decoder System and the Comprehension, Prediction, and Production System, which were trained separately. The Encoder/Decoder System first encoded a set of **Proposition**s of the form **(action, role, role-filler)**. For example, the **Proposition** (*chased, agent, dog*) specifies that, in the sentence *the dog chased the car*, the *dog* is the agent of *chased*. A set of such triples has been shown to be equivalent to a semantic network (Hinton, 1981). Message decoding was accomplished through a Query Network (St. John & McClelland, 1988; John & McClelland, 1990), that combined the Message Layer with a Proposition Query Layer, in which one of the three components of the **Proposition** was missing. Consequently, all three components had to be probed; moreover, this process was repeated for each new proposition presented to the network, resulting in $3n(n+1)/2$ queries and quadratic training time in the number of propositions. Error was calculated at the Proposition Response Layer and backpropagated to the Proposition Query Layer.

The other module of the Cscp comprised the Comprehension and Production Systems. Three-syllable encodings of words (some words were forced into three syllables through abbreviation) were presented to the Comprehension System at the Word Input Layer. Word input proceeded through a Hidden Layer to the Comprehension Gestalt Layer and then to the Message Layer, both of which were recurrent. The Production System proceeds in the reverse order from the Message Layer to the recurrent Production Gestalt Layer through a Hidden Layer and finally out to a Prediction Layer. The Prediction Layer, as the name implies, predicted the next word in the sentence.

The CSCP is basically a model of human sentence processing that models empirical experiments on a wide range of psycholinguistic phenomena including nested relative clauses and main verb/reduced relative, sentential complement, subordinate clause, and prepositional phrase attachment ambiguities. It was also designed to model production errors and structural priming effects.

The model was trained on about four million sentences generated according to the *Penglish* grammar, a hand-crafted PCFG with limits on the depth of recursion. The probabilities for the rules in Penglish were statistically determined from the Penn Treebank.

The CSCP model represents the state-of-the-art in connectionist modeling of psycholinguistic data. It is not a parsing system, but can only fill in queries presented to it. Moreover, it is trained on sentences and propositions generated from a well-designed and consistent grammar of English that has been tailored to the phenomena under study. As such, it lacks a number of common word types that show up in standard corpora such as modals, predicate adjectives, comparatives or superlatives, gerunds, infinitives, and proper nouns. Questions and commands have been left out, as have subjunctive mood, perfect progressive, and compound future tenses. Nevertheless, it was able to parse significantly more realistic language than previous subsymbolic systems.

2.3.9 Discussion of Connectionist Modeling

Two of the approaches taken up here were based on predicting the next word in the sentence (Reilly and Sharkey and Sharkey). Prediction is easier for the SRN to handle because it does not have to retain as much information in memory as it would if it were being trained to output structured representations. In Berg's XERIC parser, the network did learn structured linguistic representations, but the phrases were fairly simple overall. It was trained as a combination of a SRN and RAAM network and, so, suffered the same memory problems that arise from compression, as previously discussed above in Section 2.2.2.

Another difference between the models described reveals two common strategies in connectionist modeling. All of the systems are *modular*, except Berg's XERIC model, which is an *integrated* system (although CSCP could also be considered mainly an integrated system, given the complexity of the components, except that the Comprehension and Production Systems were trained separately). The modular approach allows the modules of the system to be trained independently of one another, which can greatly simplify training. Yet, an integrated system makes no assumptions about how the system should be broken up into modules and, furthermore, often results in better training because there are no module interfaces through which accuracy can be lost. Moreover, from a cognitive standpoint, an integrated system more closely resembles the observed functioning of the human language faculty, although this point remains controversial (see Marslen-Wilson, 1975; Fodor, 1983; McRae, Spivey-Knowlton, & Tanenhaus, 1998). The integrated system develops its own internalized clustering of the training space, and so, in a sense, generates its own functional modules with soft boundaries. This approach is used with INSOMNET in order to keep open the possibility of extending the model as a true language acquisition system.

The large-scale parsers are much more powerful than the early systems. The SSN is a *syntactic* parser that performs well on the task on which it was evaluated, but there is a strict limit on the number of phases that can keep the pulsing units distinct. The CSCP is an impressive demonstration of a connectionist system that is able to make the subtle distinctions that are of central interest in psycholinguistics. However, it did not model extragrammatical phenomena that pervade language use (although speech errors in production were indeed modeled). Because the CSCP is a purely connectionist system, there is little doubt that this is a research avenue that can be explored further. A more serious limitation of the approach is that it can only respond to questions asked of it instead of forming explicit sentence interpretations.

Indeed, the main issue in connectionist parsing is how the structures should be built up from the input sentence. Previous approaches have either attempted to map the input sentence to its parse tree directly (Reilly, Sharkey and Sharkey, and Berg), or map the sentence incrementally to a preorder traversal of the sentence's parse tree (Ho and Chan), or respond to relevant queries only (Rohde). None of these approaches reveals how the same constituent (such as a phrase or clause) can appear in multiple places in the sentence. For example, the representation for the phrase *the boy* in the sentence *the girl who saw the boy liked the dog* is embedded in the representation for *the girl who saw the boy.* A more realistic linguistic approach would identify and treat this phrase separately and then integrate it into the sentence representation during parsing. Miikkulainen comes closest to this approach, but the parser was designed to specifically handle relative clauses, relying on a controller to symbolically "push" and "pop" the appropriate context representations from the neural network Stack in order to generate case-role representations at the output.

The INSOMNET model performs no such explicit stack manipulations, but rather relies on the self-organization of semantic features in order to determine the proper storage of intermediate semantic parse results. The linguistic basis for the semantic encoding used in the model is described next.

2.4 Linguistic Foundations

Until very recently, almost all work on applying machine learning techniques to natural language processing focused on POS-tagging and syntax. Large-scale syntactic corpora are available in these areas, making meaningful analysis possible. The Penn Treebank (Marcus, Santorini, & Marcinkiewicz, 1993) has become the *de facto* standard for such analyses. Yet, this focus on tractable problems has led to a "scaling up by dumbing down" trend in NLP research, and not much progress has been made toward the ultimate goal of producing semantic interpretations (Mooney, 1999; Spark-Jones & Willett, 1997). The reason is that it has been difficult to annotate large-scale corpora with rich semantic interpretations.

Recently, such a corpus of deep semantic annotations called the Redwoods Treebank has become available, making it possible to train parsers with semantic representations. These representations are based on a general semantic framework called Minimal Recursion Semantics (MRS; Copestake, Flickinger, Sag, & Pollard, 2005). A brief overview of the Redwoods Treebank and MRS is given below, with emphasis on those aspects used in INSOMNET.

2.4.1 Redwoods Treebank

The Redwoods Treebank (Oepen, Flickinger, Toutanova, & Manning, 2002) is a corpus developed under the Linguistic Grammars Online (LinGO) project at the Center for the Study of Language and Information (CSLI) at Stanford University. The Redwoods Treebank has been designed to

- Sentence: *every man loves some woman*
- Predicate Logic Representation
  1. $every(x, man(x),\ some(y, woman(y),\ love(x, y)))$
  2. $some(y, woman(y),\ every(x, man(x),\ love(x, y)))$
- Minimal Recursion Semantics Representation

  h1: *every*(x, **h4**, **h6**) (Predicate Logic interpretation 1: **h6** $\equiv$ **h2**)
  h2: *some*(y, **h5**, **h7**) (Predicate Logic interpretation 2: **h7** $\equiv$ **h1**)
  h3: *love*(x, y)
  h4: *man*(x)
  h5: *woman*(y)

Figure 5: **Minimal Recursion Semantics vs. Predicate Logic.** This figure demonstrates how MRS provides a more tractable representation scheme for handling scopal ambiguity than Predicate Logic on the sentence *every man loves some woman.* By employing flat semantics through the labeling of predicates, MRS makes it possible to leave the scope of the quantifiers *every* and *some* underspecified, whereas in Predicate Logic, the scoping decision must be made immediately. The scope handles (**h6** or **h7**) do not need to be instantiated until context makes it clear which interpretation is preferred, avoiding costly structural revision.

provide more detailed syntactic and semantic analyses than standard corpora used in computational linguistics. The sentences in the Redwoods Treebank are taken from the VerbMobil project (Wahlster, 2000), consisting of transcribed face-to-face dialogues in a travel arrangement domain. The sentences have been annotated with Head-driven Phrase Structure Grammar (HPSG; Pollard & Sag, 1994) analyses licensed by the English Resource Grammar (ERG; Copestake & Flickinger, 2000; Flickinger, 2000), a large-scale, broad-coverage grammar of English. The June 10, 2001, release of the Redwoods Treebank was used in this study, and the target parses were verified by a single annotator.

2.4.2 Minimal Recursion Semantics

While HPSG representations can be unwieldy to use with neural networks, their semantic components can be mapped to a simpler formalism called Minimal Recursion Semantics (MRS). In contrast to the standard Predicate Logic representations of linguistic dependencies, MRS uses flat semantics. In MRS, predicate labels called *handles* (i.e., *pointers* or *addresses*) serve as fillers for argument slots in the set of MRS frames. In this manner, MRS avoids the nesting of predicates necessary in Predicate Logic, and thereby also avoids the problem of combinatorial explosion that arises when ambiguity requires the maintenance of numerous structures with nested constituents during sentence processing.

Figure 5 compares representations of the sentence *every man loves some woman* in Predicate Logic and in MRS. This sentence is a classic example of scopal ambiguity, and it is also interesting from a cognitive standpoint. The default interpretation, based on real-world knowledge, strongly favors the Predicate Logic interpretation 1 in Figure 5. This formula states that there is a different woman for each man that he loves. The reason for this interpretation is that the predicate *some* falls within the scope of *every.* People generally overlook the other possible interpretation, given in the second formula, in which *some* has wide scope, governing *every.* Here, the meaning is that there is one and the same woman that every man loves (a notion that usually only crops up in poetry in which that woman may be metaphorically understood as Aphrodite). In Predicate Logic, both interpretations have to be maintained in parallel until the ambiguity can be resolved. Given

Abbreviation	*Semantic Role*
EVT	*event* feature structure
FRI	*full referential index* feature structure
SA	*state-of-affairs*
A0	*arg0* (predicative) argument
A1	*arg1* (external) argument
A3	*arg3* (patient) argument
BV	*bound variable*
RE	*restriction*
IX	*instance*
EV	*event*

Table 1: **Semantic Annotations.** The Erg links argument structure to a set of semantic roles. This table describes the semantic annotations, together with abbreviations used in the examples in this section. A more comprehensive list is provided in the Appendix.

the proliferation of ambiguity in natural language, the likely combinatorial explosion of possible interpretations is a distinct disadvantage.

To avoid such a combinatorial explosion, Mrs uses handles to label predicates, such as the **h4** handle for $man(x)$ in Figure 5. Predicate arguments then use the handles to point to their fillers. An example is the quantifier $every(x$, **h4**, **h6**), which is a ternary predicate with implicit arguments *bound variable*, *restriction*, and *scope*. The restriction (**RE**) is filled by the predicate labeled **h4** (i.e., *every* is restricted to the set of men), and the bound variable (**BV**) x that ranges over that set. The handle **h6** in the *scope* (**SC**) slot is an example of underspecification: There is no predicate labeled **h6** among the predicates in the set. In order to specify the scope, the handle **h6** must be identified with another handle. Thus, in Figure 5, the default interpretation of the sentence is recovered when **h6** is instantiated with the handle **h2** that labels the predicate $some(y$, **h5**, **h7**). Similarly, the poetic interpretation is invoked with the instantiation of **h7** in the scope slot of the *some* quantifier with the handle **h1** of *every.* Leaving the argument handles uninstantiated keeps the ambiguity unresolved until the sentence or its context provides the proper instantiation. Predicate Logic, on the other hand, would require that a scoping decision be made immediately. In an incremental cognitive system, such an early commitment can make it difficult to revise this decision if later context favors a different interpretation because predicates may have to be completely renested.

Figure 6 is a graphical illustration of the Mrs representation for the sentence *the boy hit the girl with the ball*, that features the very common prepositional phrase attachment ambiguity, using abbreviations for semantic annotations from the Erg shown in Table 1. This sentence and its representation will be used throughout the article to motivate the design and operation of InSomNet. Each node in the graph has a handle and an identifier (given by the corresponding word in the sentence if there is one, and otherwise by the semantic relation) that names a given predicate. The arguments of the predicate are represented by the arcs emanating from the predicate's node; the labels on the arcs are abbreviations for the argument role names. Taken together, the arcs give the predicate's subcategorization, and the nodes to which they point are the fillers. Because the Mrs representation is a DAG, some nodes are leaves. These leaves are called *feature nodes* that characterize the predicates that reference them. (The particular features are not shown in the figure due to space considerations, but are provided in Figure 8 in Section 3.1 that describes the semantic graph in terms of frames.) The

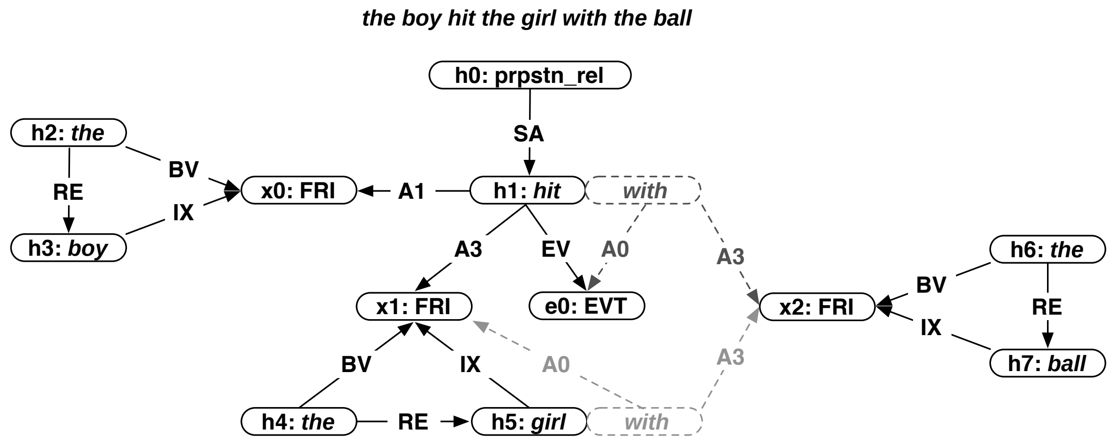


Figure 6: **An** Mrs **Representation of Prepositional Phrase Attachment.** This graph represents the sentence *the boy hit the girl with the ball*. Nodes in the graph are labeled with a handle and the word in the sentence (or semantic relation if there is no corresponding word, such as for phrasal or clausal graph nodes). The arcs emanating from a node indicate subcategorization and are labeled with the arguments that the word or semantic relation takes. The handles that fill those roles are the labels of the nodes to which the arcs point. The top node in the graph is labeled by the handle **h0** and semantic relation **prpstn_rel**, indicating the sentence is a declarative proposition. The **SA** (*state-of-affairs*) arc of the proposition points to the main predication of the sentence in the node labeled by the handle **h1** and word *hit*. The **EV** arc of *hit* points to the handle **e0** and semantic type **EVT** (an *event*). Each noun phrase in the sentence is represented by a set of three nodes for the determiner, the noun, and a corresponding index. For example, the noun phrase *the boy* comprises the determiner, *the* with the handle **h2**, and its roles, **BV** and **RE**, are filled by **x0** and **h3**, respectively. The noun *boy* has a single role, **IX**, filled by the handle of the index **x0**. The index has semantic type **FRI** (*full referential index*). The agent of *hit*, indicated by the arc labeled **A1**, is also filled by the **x0** handle. Similarly, the patient, **A3**, is filled by the handle **x1**, which is the index for *girl*. Lastly, *ball* is the object of the preposition *with*, and so the handle of its index **x2** fills the preposition's **A3** role. The two interpretations of the sentence are illustrated by two *with* nodes. Modifier attachment is represented by a dashed node that shares the handle of the node modified. In the verb-attachment case, the *with* node has the same handle **h1** as *hit*, and its **A0** role is the index handle **e0**. In the noun-attachment case, the *with* node has the same handle **h5** as *girl*, and its **A0** role is the index handle **x1**. In this manner, Mrs formalism is flexible: Either interpretation may be selected according to context, especially in an incremental sentence processing system where the context unfolds over time.

majority of them are distinguished by handles that begin with **x** (for nominal predicates) and **e** (for verbal predicates); two other types (**d** for degree modifiers, and **v** for conjunctions) are less common in the current version of the corpus. Features such as as gender, person, and number for nominal predicates are provided in a corresponding *full referential index feature structure* with semantic type **FRI** that fills the noun's **IX** role slot. For example, **x1** in Figure 6 characterizes the noun *girl* with the features **fem** (feminine), and **3sg** (third person singular). Similarly, verbal predicates are characterized by a corresponding *event feature structure* with semantic type **EVT** that fills the **EV** argument slot and specifies features such as aspect, mood, and tense. Thus, **e0** characterizes the verb *hit* with the features **-asp** (non-aspectual), **ind** (indicative), and **past**. Non-leaf predicates labelled by handles that begin with **h**. The Appendix provides a more complete description of the components of Mrs representations as used in the simulations in this study.

the boy hit the girl with the ball

- **h0: prpstn_rel(h1)**
- **h1:** *hit*(**x0, x1, e0**) (Instrumental interpretation: **h8** ≡ **h1** and **h9** ≡ **e0**)
- **h2:** *the*(**x0, h3,** _)
- **h3:** *boy*(**x0**)
- **h4:** *the*(**x1, h5,** _)
- **h5:** *girl*(**x1**) (Modifier interpretation: **h8** ≡ **h5** and **h9** ≡ **x1**)
- **h6:** *the*(**x2, h7,** _)
- **h7:** *ball*(**x2**)
- **h8:** *with*(**h9, x2**)

Figure 7: **Structural Ambiguity in** Mrs**.** As in the scopal ambiguity described in Figure 5, prepositional phrase attachment ambiguity can be represented in Mrs through underspecification. In this case, both the preposition's handle **h8** and its first argument **h9** are instantiated to resolve the ambiguity. Identifying (or *sharing*) **h8** with the verb *hit*'s handle **h1** and **h9** with the its event index **e0** gives the instrumental interpretation, whereas instantiating **h8** with the handle **h5** of *girl*(**x1**) and **h9** with its index **x1** gives the modifier interpretation. The ambiguity remains unresolved if **h8** and **h9** are left uninstantiated. In this manner, prepositional phrase ambiguity can be resolved without costly revision.

In order to understand the Mrs example completely, the semantic annotation used in Figure 6 needs to be briefly explained. Figure 7 also shows a representation of these predicates similar to the Mrs structure described earlier in Figure 5.

The sentence is a declarative proposition, as indicated by the semantic relation **prpstn_rel** in the top node labeled by the handle **h0**. Sentence types subcategorize for a single role, *state-of-affairs*, which is indicated in Figure 6 by the arc labeled **SA**. The filler for this role is the handle **h1**, which is the label of the node for the main verbal predication of the sentence, the word *hit*. Verbs have an *event* role, and transitive verbs, such as *hit*, have an *arg1* role and an *arg3* role, which correspond to the thematic roles of **Agent** and **Patient**, respectively. These three roles are represented in the figure by the arcs labeled **EV**, **A1**, and **A3**.

The semantics of the noun phrases in the sentence are represented by three sets of nodes. Each set represents the determiner, the noun, and an index that indicates the features of the noun, such as gender, number, and person (also not shown). The determiner is a quantifier which subcategorizes for the *bound variable* (**BV**) and *restriction* (**RE**) arguments (the *scope* role is empty in this release of the Redwoods Treebank). The **BV** role is filled with the noun's index, and the **RE** role, with the noun's handle. The noun has a single *instance* (**IX**) role, filled with its index. For example, in the noun phrase, *the boy*, the node for the noun *boy* is labeled with the handle **h2**, which fills the **RE** role for the governing determiner *the*. The index of *boy* is the handle **x0**, which labels the node with semantic type **FRI**, indicating that *boy* is a *full referential index*. The index handle **x0** binds the determiner and noun through their **BV** and **IX** roles, respectively, and fills the **A1** role of *hit* to indicate that *boy* is the agent of the verb.

Similarly, the instance handle **x1** fills the verb's **A3** role to denote the patient. The handle **x2** is the index for the noun *ball* and fills the **A3** role of the preposition *with*. The preposition can either modify the verb for the instrumental sense of the sentence, or the noun for the modifier sense. In Mrs, modification is represented by *conjunction* of predicates; for example *big red ball* would be denoted by $\bigwedge$[*big*(x), *red*(x), *ball*(x)]. The n-ary connective $\bigwedge$ is replaced by a handle, which is distributed across the operands so that each predicate has the same handle (an operation called *handle-sharing*). In the case of verb-attachment, the verb *hit* and the preposition *with* both share the handle **h1**, and the preposition's **A0** role is filled with the verb's event structure handle **e0**. For noun-attachment, *with* has the same handle **h5** as *girl*, and its **A0** role points to the index **x1** of *girl*.

The flat semantics representation of MRS is ideal for a connectionist system such as INSOMNET because it is no longer necessary to represent arbitrarily nested structures in a fixed vector. Previous systems (such as Berg, 1992; Mayberry & Miikkulainen, 2000, 1999a, 2005) that encoded trees with RAAM suffered from memory problems. Using MRS, in contrast, the meaning of a sentence is encoded as a directed acyclic graph (DAG) distributed over a grid of cells. The cells of the grid are trained to hold the node contents of the MRS representation (such as the one depicted in Figure 6), together with their subcategorization arcs implemented as context-sensitive pointers to other cells in the grid that hold the argument fillers. Furthermore, underspecification – represented in MRS by an unfilled argument slot – is naturally represented in connectionist assemblies as superimposed activation patterns. They can be construed as the maintenance of likely fillers without a clear commitment to any one of them until later processing provides disambiguating information, at which point, the filler takes on the appropriate interpretation. Such a representational scheme is well suited for neural networks, as will be described next.

3 The InSomNet Architecture

In this section, the design decisions that went into the development of INSOMNET will be motivated by showing how a directed acyclic graph such as that shown in Figure 6 used to represent the MRS interpretation of a sentence can be accommodated naturally in a connectionist system, as described next in Section 3.1. The components of INSOMNET's architecture will be described in Section 3.2, followed by how these components are activated and trained during sentence processing in Sections 3.3 and 3.4, respectively.

3.1 Semantic Representation

The first step in defining the INSOMNET architecture was to develop a way of representing the MRS annotations in a neural network. To this end, the MRS graph from Figure 6 was rendered as a set of frames, called a *frameset*, to serve as targets for INSOMNET, as shown in Figure 8. Because the MRS graph has already been described in detail in Section 2.4.2, this section will focus on the the frames that INSOMNET generates as output upon processing a sentence. Each frame has the format

| **Handle Word Semantic-Relation Subcategorization <Arguments>** |.

For example, the node labeled **h1:** *hit* in the MRS graph in Figure 8 is represented by the frame

| **h1** ***hit*** **_arg13_rel A0A1A3DMEV _ x0 x1 _ e0 _** |

The first element, **h1**, is the **Handle** (node label) of the frame. Other frames can include this element in their argument slots to represent a dependency to this frame. That is, the **Handle** serves as a *pointer* to other nodes in the semantic graph. For example, **h1** fills the *state-of-affairs* (**SA**) slot in the topmost node, **h0: prpstn_rel**, as indicated by the arc labeled **SA**. The second element, *hit*, gives the **Word** for this frame (where applicable; many frames, such as the **h0: prpstn_rel** frame that

the boy hit the girl with the ball

```
| h0 __   prpstn_rel        SA          h1   __   __  __   __  __ |
| h1 hit  arg13_rel         A0A1A3DMEV  __   x0   x1  __   x0  __ |
| h1 with miscprep_ind_rel  A0A3DMEV    e0   x2   __  __   __  __ |
| e0 __   EVT               DVASMOTN    bool -asp ind past __  __ |
| h2 the  def_explicit_rel  BVDMRESC    x0   __   h3  __   __  __ |
| h3 boy  diadic_nom_rel    A3IX        __   x0   __  __   __  __ |
| x0 __   FRI               DVGNPNPT    __   masc 3sg prn  __  __ |
| h4 the  def_explicit_rel  BVDMRESC    x1   __   h5  __   __  __ |
| h5 girl diadic_nom_rel    A3IX        __   x1   __  __   __  __ |
| h5 with miscprep_ind_rel  A0A3DMEV    x1   x2   __  __   __  __ |
| x1 __   FRI               DVGNPNPT    __   fem  3sg prn  __  __ |
| h6 the  def_explicit_rel  BVDMRESC    x2   __   h7  __   __  __ |
| h7 ball reg_nom_rel       IX          x2   __   __  __   __  __ |
| x2 __   FRI               DVGNPNPT    __   neu  3sg prn  __  __ |
```

Figure 8: MRS **Frameset Format for** INSOMNET**.** This set of frames corresponds to the MRS dependency graph in Figure 6. The frames determine the output representation of INSOMNET and encode the semantics of the sentence being processed. The first four fields, **Handle**, **Word**, **Semantic-Relation**, and **Subcategorization**, are a part of every frame and describe the nodes in the graph. How the following six fields are interpreted depends on the value of the **Subcategorization**. This field encodes the names of the arguments taken by the **Semantic-Relation** in the frame and has at most six fields. These arguments correspond to the labeled arcs in the graph. For example, the node labeled **h1:** *hit* is a transitive verb denoted by the semantic relation **arg13_rel**, and has a **Subcategorization** abbreviated asf **A0A1A3DMEV**. Three of the roles, **A1** (*arg1*), **A3** (*arg3*), and **EV** (*event*), have fillers (**x0**, **x1**, and **e0**, respectively), which are represented by arcs from the **h1:** *hit* node. The other two arguments, **A0** (*arg0*) and **DM** (*dimension*), are left unfilled, and are represented by "_" (**Null**). Because the sentence *the boy hit the girl with the ball* is ambiguous, there are two frames containing the **Word** *with*: one that shares its handle (**h1**) with the *hit* frame, and the other that shares its handle (**h5**) with the *girl* frame. These frames correspond to the *with* nodes in the graph. Note that handle sharing is indicated by literally labelling the *with* nodes with the handles of their attachment nodes. During training, only one of these frames is presented, so that INSOMNET is only exposed to unambiguous interpretations. It must learn to represent both interpretations. The frameset format thus has a regular structure that makes it possible to encode it as a neural network.

denote phrasal and clausal constituents, have no corresponding word in the input sentence). The third element is the **Semantic-Relation** for the frame. The **Semantic-Relation** serves as the node label in the graph unless the frame has a corresponding **Word**. The third element, **A0A1A3DMEV**, represents the **Subcategorization** and is shorthand for the sequence **A0 A1 A3 DM EV**, the argument roles that the semantic relation takes given in a canonical order (the roles **A0** and **DM** are not shown in Figure 6 because their slots are empty in Figure 8 as indicated by **"_"** or **Null**). In this case, the subcategorization type indicates that *hit* is a transitive verb with three arguments: the agent is the frame to which the **A1** arc points; the patient, the frame to which the **A3** arc points; and the event, the frame to which the **EV** arc points. The arc labels themselves are abbreviations for MRS argument role names (e.g., **A1** is *arg1*, **EV** is *event*, and **BV** is *bound variable*) shown in Figure 8. The rest of the frame <_ **x0 x1** _ **e0**> lists the **Arguments** (fillers) in one-to-one correspondence to these subcategorization roles. For example, the argument **x0** fills the **A1** agent

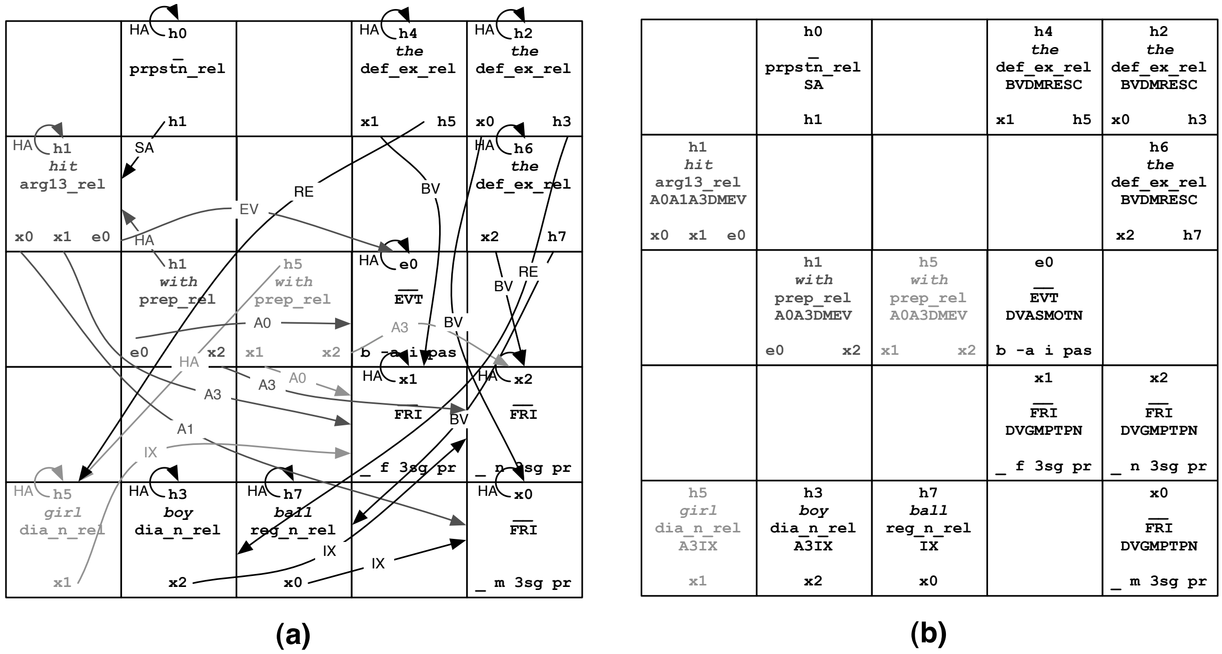


Figure 9: **Representing Semantic Graphs in Neural Networks.** The grid (a) represents the basic symbolic information in the nodes and values from Figure 8, with labeled arcs for roles and their fillers (b). Subcategorization information is used to indicate roles and fillers in the actual representation instead of explicit arcs. Each cell in the grid denotes an assembly over which a pattern of activation will develop that encodes the symbolic information. This approach allows encoding of the Mrs graph in a neural network representation.

role, **x1**, the **A3** patient role, and **e0**, the **EV** event role in the subcategorization **A0A1A3DMEV**. The frames labelled with **x0** and **e0** in Figure 8 are called *feature frames* and correspond to the feature nodes described in Section 2.4.2. Feature frames, as leaves of the DAG, never have pointers as fillers, but rather list features of their corresponding predicates.

It is important to point out two properties of Mrs **Handle**s that have influenced the design of InSomNet. First, a given **Handle** may refer to several frames. As explained in Section 2.4.2, handle-sharing is used to denote predicate conjunction to represent modification or linguistic relations that are optional (in both whether and where they occur). It can also represent linguistic relations that occur more than once and therefore may require more than one frame to represent, such as adjuncts (as in the example above), modifiers (such as adjectives and relative clauses), or verb-particle constructions (e.g., *work* something *out*).

Second, in the symbolic Mrs specification, **Handle**s are arbitrary designators (e.g., the label **h1** has no meaning in itself). However, in a neural network, **Handle**s have to be encoded as patterns of activation that can be learned. In InSomNet, the **Handle**s are designed to be dynamically associated with patterns that represent core semantic features, such as predicate argument structure and role fillers. In this way, InSomNet can learn to generalize these semantic features to process novel sentences. How these **Handle** patterns are developed is an important aspect of the model that will be described in detail in Section 3.4.3.

Figure 9 shows how a frame-based representation such as that illustrated in Figure 8 can be represented in a Som. Each cell in the Som holds a distributed representation that encodes the

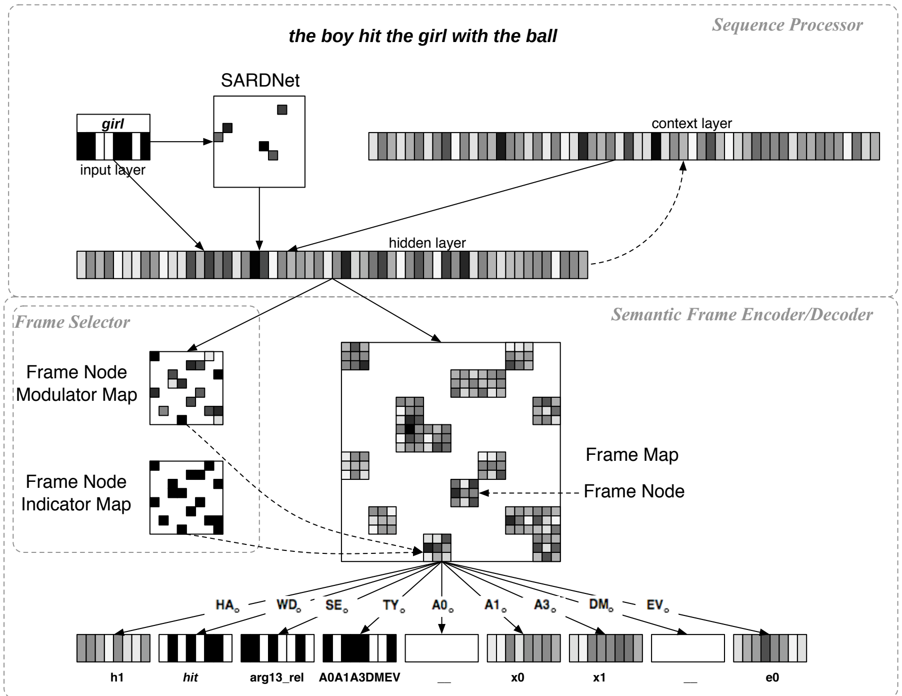


Figure 10: **The InSomNet Architecture.** The InSomNet model consists of three operational modules based on how they function together. The Sequence Processor reads the input sentence in one word at a time and activates both the Frame Selector and the Frame Encoder. The Frame Encoder encodes the Mrs dependency graph representing the semantic interpretation of the sentence. The Frame Selector select frames in the Frame Map in a graded manner corresponding to an *a posteriori* probability that those frames belong to the current semantic interpretation. The solid arrows indicate weights and the dashed arrows represent a conceptual correspondence between the modules.

components of individual Mrs frames. The assignment of which cell holds which frame representation is determined through semantic self-organization, which will be taken up in greater detail in Section 3.4.3. The codebook vector of each cell is the handle that labels the contents of the cell. The pattern in a cell can be decoded to access the elements of the encoded frame. In the grid of Figure 9 (a), the arcs point from decoded fillers to the cell they denote. What roles these fillers fill are captured in the encoded subcategorization information for each frame, as shown Figure 9 (b). This representation completely describes the graph in Figure 8 in a Som-based neural network. Because these frames are encoded through distributed representations, a *decoder* network is required to pull out their individual components. How this is done in InSomNet is described next.

3.2 Architecture Overview

The InSomNet sentence parsing architecture shown in Figure 10 consists of three operational modules, each with two components:

Sequence Processor

- An Srn to process the input sequence one word at a time.
- A SardNet Map that retains an exponentially decaying activation of the input sequence.

Frame Encoder/Decoder

- A Frame Map that encodes the frames of the MRS dependency graph in Frame Nodes.
- A Frame Node Decoder that generates the components of the frame representations.

Frame Selector

- A Frame Node Modulator Map that modulates the activation levels of the Frame Nodes.
- A self-organized Frame Node Indicator Map indicating which Frame Nodes are part of the sentence interpretation. It also serves as the target for the Frame Node Modulator Map.

Altogether, these components represent intergrated modules that handle different aspects of parsing a sentence into a semantic representation. The SRN component, i.e., the Sequence Processor, reads the input sentence incrementally. The Frame Encoder/Decoder distributes the semantic representation over the grid of frame representations, and the Frame Selector determines which nodes are part of the sentence interpretation. The integrated system is thus able process complex sentences robustly.

3.3 INSOMNET Activation

Having provided an overview of the INSOMNET architecture, the activation of the modules in the course of processing an input sentence is presented in this section. The training of the model is described in Section 3.4.

3.3.1 Sequence Processor

A SRN forms the basis for the INSOMNET architecture. The SRN reads a sequence of word representations one at a time as its input. The node in the SARDNET Map that is available and closest to the input word is activated, and the rest of the map is decayed by a factor of 0.9. Both the pattern in the Input Layer, the SARDNET Map, and the Context Layer are propagated through the network's Hidden Layer to develop a semantic interpretation of the sentence at its output. At each time step, a copy of the Hidden Layer is saved and used as input during the next step, together with the next word. In this way, each new word is interpreted in the context of the entire sequence read so far, and the final sentence interpretation is gradually formed at the output.

3.3.2 Frame Encoder/Decoder

The Frame Map is the primary innovation of INSOMNET. It allows the semantic interpretation of an input sentence to be developed incrementally and in a graded but distinct manner. In the current implementation, the Frame Map consists of a 12×12 array of Frame Nodes. Each node itself is a 144-dimensional vector. (However, these model parameters are somewhat arbitrary: Larger and smaller maps will also work, affecting only the resolution of the semantic entries.) Thus, the Frame Map as a whole functions as a second Hidden Layer of the SRN. As an input sentence is processed, a number of Frame Map nodes will become strongly activated; that is, a particular pattern of activation develops over the 144 units of each of the 144 Frame Nodes. These patterns of activation are compressed representations of the MRS frames that constitute the semantic interpretation of the current sentence. Which MRS frame a given Frame Node represents is not stipulated *a priori* as part of the network design; rather, INSOMNET must learn to associate nodes with frames that have similar semantic structure. How it does so using the process of semantic self-organization will be taken up in Section 3.4.3, along with the encoding and decoding of MRS frame components. As a result of this self-organization of MRS frame representations, similar frames cluster together on the map. For example, patterns encoding determiners occupy one section of the map, the various types of verbs another, nouns yet another, and so on. Morever, although each node becomes tuned to MRS frames with particular semantic features, no given Frame Node is dedicated to a given frame. Rather, the nodes are flexible enough to represent different frames, depending on what is needed to represent the input sequence. For example in Figure 10, the node at the bottom center of the Frame Map decodes into the | **h1** ***hit*** **_arg13_rel A0A1A3DMEV _ x0 x1 _ e0** | frame for this particular sentence. In another sentence, it could represent a different verb with a slightly different subcategorization type (say, **A0A1A4DMEV** for one sense of the verb *think*) and its associated arguments. This feature of the architecture allows the Frame Map to represent semantic dependency graphs dynamically, making it flexible enough to generalize to novel sentences.

3.3.3 Frame Selector

As each word is read in by the Sequence Processor, INSOMNET indicates which frames are included in the current semantic interpretation by activating units on the Frame Node Modulator Map. The units of this map correspond one-to-one with the Frame Nodes in the Frame Map. Their activation represents how strongly their corresponding nodes in the Frame Map should be activated. This correspondence allows frames to be selected in a graded manner. This process is useful in cognitive modeling of psycholinguistic effects such as expectations and defaults, semantic priming, activating multiple senses to represent ambiguity, and revising interpretations nonmonotonically in the course of sentence processing. The Frame Node Indicator Map is used as the target in training the Frame Node Modulator Map, as will be explained in the next section.

3.4 Training INSOMNET

Now that processing in INSOMNET has been described module by module, it is appropriate to consider how the network is trained. INSOMNET can be trained in a single phase, but using two distinct phases results in slightly better performance. In the first phase, only the components using self-organization, i.e., the SARDNET Map and the Frame Node Indicator Map, are trained. Once these maps have settled, the second phase begins, during which the backpropagation-based components are trained. An overview of the training process is first presented, followed by the training of each module in detail.

The model parameters, i.e., the SRN learning rate and the Frame Node Indicator Map neighborhood and learning rate, were all updated on epoch t_n starting at $t_0 = 0$ according to the schedule

$$t_{n+1} = \lceil 1.5 * t_n + 1 \rceil. \tag{1}$$

For example, if the last parameter update t_4 occurred on epoch 10, the next parameter update t_5 would occur on epoch 16. The initial learning rate for the Frame Node Indicator Map was 0.4 and decayed by a factor of 0.9. The neighborhood of the Frame Node Indicator Map was initialized to half the diameter of the Frame Map (i.e., 6) and decremented with each update. Once the Frame Node Indicator Map had stopped self-organizing (when its learning rate fell below 0.001), the Sequence

Processor was initialized with a learning rate of 0.01, and was subsequently decayed by 0.9 according to the same schedule.

3.4.1 Overview

The Sequence Processor receives error signals from both the Frame Selector and Frame Encoder/Decoder modules (Figure 10). Target frames are presented to the Frame Encoder/Decoder module, and the output from each Frame Node that holds the compressed frame representation is compared with the target to generate an error signal. All the error signals are backpropagated through the Frame Encoder to the Sequence Processor, and the appropriate weight changes made.

The error signal from the Frame Selector is generated by comparing the Frame Node Modulator Map with the Frame Node Indicator Map. The Frame Node Indicator Map itself is self-organized on the basis of compressed representations. This process is described in detail below in Section 3.4.4.

3.4.2 Sequence Processor

The Sequence Processor is trained with BPTT to improve the network's ability to process longer sequences over standard backpropagation. With BPTT, the SRN is effectively trained as if it were a multi-layer feedforward network, with the constraint that the weights between each layer are shared.

The SARDNET Map (see Figure 4) enhances this ability by allowing the SRN to latch information in order to propagate gradients back to the beginning of the sentence. Because SARDNET develops a localist encoding of the input sequence, in principle its representation may be brittle and hard to generalize. Yet, because the map is coupled with the distributed representation in the Hidden Layer of the SRN, the Sequence Processor proves to be quite robust. The reason is that the sequence that develops on the map is itself a distributed pattern, such that similar sentences entail similar SARDNET representations. Therefore, the SRN not only retains long-distance dependency information in order to develop the correct output, but also does it robustly and generalizes to new sequences.

3.4.3 Frame Encoder/Decoder

Previous models of incremental sentence interpretation applied an SRN to the task of case-role analysis. A major obstacle to scaling up this approach was that the number and types of roles had to be specified beforehand as part of the network's architecture (e.g., the five output role assemblies shown in Figure 1), thus posing a strict limit on the complexity of the sentences the model could handle.

To solve this problem of *a priori* stipulating the role a given output represents, the Frame Map was developed so that each frame of the MRS graph that represents the interpretation of the sentence being processed is assigned to a unique Frame Node, as shown in Figure 9 (b). INSOMNET learns this frame/node assignment on the basis of self-organizing the compressed representations of the MRS frames in the training corpus. The Frame Node Indicator Map, discussed in more detail in Section 3.4.4 below, is a self-organizing map trained on compressed frame representations. The nodes in the Frame Node Indicator Map are in one-to-one correspondence with the Frame Nodes in the Frame Map. The compressed frame representations are developed by autoassociating their components using RAAM. The compressed frame representation is then cleaned up using *pointer quantization*, discussed next.

Pointer Quantization. In any given corpus, there are as many **Handles** as there are distinct frames, taking into consideration handle-sharing as explained in Section 3.1. In any given sentence

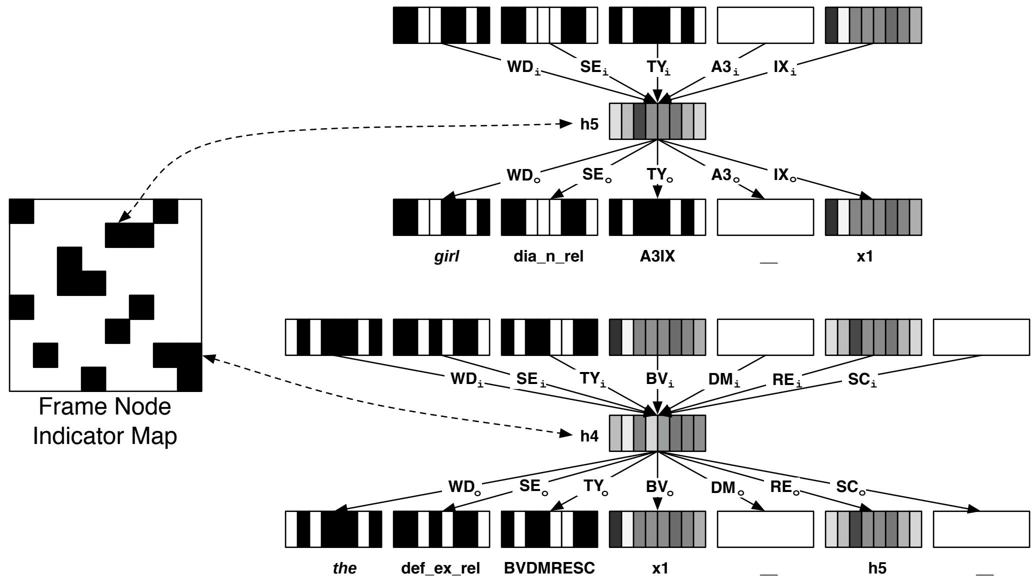


Figure 11: **Handles as the Basis of Semantic Self-organization.** In this figure, compressed representations for two frames with **Handle**s **h4** and **h5** are developed using RAAM to autoassociate their respective components through shared weights, (e.g., the **Word** input, $\mathbf{WD}_\mathrm{i}$, and output, $\mathbf{WD}_\mathrm{o}$, weights). The *restriction* (**RE**) slot of the **h4** frame (*the*) references the **h5** frame (*girl*); thus, the compressed representation for the **h5** frame is developed before that of **h4**. The compressed representation for the **h5** frame is *not* used in the encoding of the **h4**. Rather, the codebook vector of the unit in the Frame Node Indicator Map closest to the compressed representation for the **h5** frame is used instead. This substitution, called *pointer quantization*, facilitates learning by restricting the representations of **Handle**s to the small set of 144 codebook vectors developed in the Frame Node Indicator Map, which has been semantically self-organized on the basis of the compressed representations of the MRS frames in the training corpus.

in the corpus, there are sometimes identical frames, such as when two nouns share the same features. In these cases the compressed representations of such frames developed through autoassociating their components via RAAM will also be identical. To ensure that identical (or very similar) frames are assigned to unique Frame Nodes on the FRAME MAP, the prototype codebook vectors that have developed during the self-organization of the Frame Node Indicator Map on the basis of the compressed frame representations are substituted for these compressed frame representations, as shown in Figure 11. This substitution, called pointer quantization, has two primary advantages. First, it constrains the number of distinct pointer representations to the number of units in the Frame Node Indicator Map, thus facilitating training. Second, in those cases where two identical frames would map onto the same unit in a regular SOM, the base architecture for the Frame Node Indicator Map is SARDNET, in which winning nodes are removed from further competition. Thus, identical (or very similar) frame representations are assigned to distinct nodes with distinct codebook vectors. In this way, the quantization of the compressed **Handle** representations function as *content-addressable* pointers to the Frame Node that in turn encodes the complete frame.

For example in Figure 11, the frame constituents of | ***girl*** **diadic_nom_rel A3IX _ x1** | (without the **Handle** field) are autoassociated through a RAAM network through input and output weights for each role of the frame (e..g., the **Word** input, $\mathbf{WD}_\mathrm{i}$, and output, $\mathbf{WD}_\mathrm{o}$, weights), resulting in a compressed representation of the frame **h5** in the Hidden Layer. The closest codebook vector to this compressed representation in the Frame Node Indicator Map is assigned the (arbitrary) handle **h5**, the label for this frame. The codebook vector is then used instead of the compressed representation as fillers for other frames that reference **h5**. Similarly, the frame constituents (which now include the **h5 Handle** representation) of | ***the*** **_def_rel BVDMRESC x1 _ h5 _** | are autoassociated through the RAAM network, with the resulting codebook vector in the map to the Hidden Layer assigned the **Handle h4**. Note, of course, that the first frame must be compressed before the second so that the second can use its **Handle h5** in its *restriction* (**RE**) slot. A compressed representation for each frame in the MRS dependency graph of a sentence can be developed because the graph is acyclic. The leaves of the graph (the feature frames) serve as stable starting points, and compressed representations of the frame corresponding to each frame are developed recursively. While the handles (e.g., **h4** and **h5**) are arbitrarily generated during training, the patterns they designate have semantic content and, therefore, are not arbitrary.

Once all handles for the frames in the MRS dependency graph have been developed, they are presented to the Frame Node Indicator Map in reverse order to preserve the original sentence order in the semantic graph as it is decoded. This map is also trained as a SARDNET map (see Figure 4), but with a decay rate of 1.0, since the nodes are supposed to indicate simply whether a frame is present or not. At this point, which frame is supposed to be in which Frame Map node is known, so during training, the Frame Map serves as a grid of second Hidden Layers and the frame components are decoded in the Output Layer. The appropriate frame components are presented as targets for the patterns in the Output Layer, and the resulting error signals are backpropagated through the Frame Node Decoder weights to the corresponding Frame Node and on to the first Hidden Layer of the SRN. It is important to note that the Frame Node Decoder weights are the same output weights used to develop the frame handles shown in Figure 11; e.g., the **Word** $\mathbf{WD}_o$ weight matrix. This weight-sharing further ensures that the representations developed in the Frame Nodes are similar to the frame representations developed in the RAAM network, and, consequently, that the Frame Node representations exhibit the same semantic self-organization as the codebook vectors in the Frame Node Indicator Map on which they are based.

Self-organizing maps are conventionally trained with static targets (or those drawn from an input space with a fixed probability distribution). However, using compressed representations developed with RAAM requires training the Frame Node Indicator Map with a set of moving targets. These targets are nevertheless slowly converging to stable patterns. The moving targets complicate training INSOMNET because as long as the Frame Node Indicator Map is training, the patterns can oscillate among nodes. Yet, it is precisely this oscillation among nodes that gives the Frame Map its topological organization and flexibility. Competing nodes develop similar patterns, while allows the map to represent frames robustly and the system to generalize to new frames.

3.4.4 Frame Selector

Recall that the Frame Node Indicator Map indicates which nodes in the Frame Map encode the semantic frames of the sentence being processed. If a unit is active on the Frame Node Indicator Map, then the pattern in the corresponding Frame Node is a target frame and should be decoded into an MRS frame. Learning these target frames is the task of the Frame Node Modulator Map.

The Frame Selector takes the Hidden Layer activation in the Sequence Processor and propagates it to the Frame Node Modulator Map (Figure 12). The output is compared with the self-organized

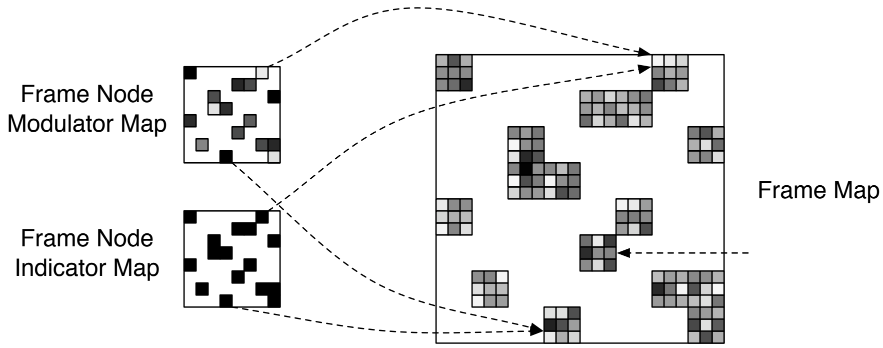


Figure 12: Frame Map **Self-organization.** The Frame Nodes in the Frame Map are in one-to-one correspondence with the units in the Frame Node Indicator Map and the Frame Node Modulator Map. The activation of a unit in the Frame Node Indicator Map indicates that the pattern in the corresponding Frame Node is a target frame and should be decoded into an MRS frame. The Frame Node Indicator Map is self-organized based on the compressed frame encodings that become the representations for the frame handles (Figure 11). The Frame Node Modulator Map uses the Frame Node Indicator Map as its target to learn which frames should be active for any given sentence. In this manner, it is possible to learn both to identify which frame should be active and induce the self-organization of the Frame Map at the same time.

Frame Node Indicator Map, and the resulting error signal is backpropagated to the Hidden Layer. In this way, the Frame Node Modulator Map output comes to represent the network's confidence that the activated frames are part of the semantic interpretation of the input.

Training the Frame Selector network is not started until the Frame Node Indicator Map is finished self-organizing (e.g., the neighborhood is at 0, or the learning rate is below the 0.001 threshold). This process allows the compressed representations developed through RAAM to stabilize.

3.5 Putting It All Together

INSOMNET is an incremental sentence parsing and comprehension model that uses the novel technique of self-organizing nodes in the Frame Node Indicator Map on the basis of compressed representations of semantic MRS frames. These nodes identify which Frame Nodes in the Frame Map can be decoded into the frames of the MRS dependency graph that represents the sentence. At the same time, the Frame Node Modulator Map indicates the network's confidence that the frames are part of the current interpretation.

This approach allows both scaling the model up to sentences of realistic length and complexity, and modeling psycholinguistic effects such robustness, expectations, ambiguity, and semantic priming. In the next section, the basis on which INSOMNET is evaluated on these tasks will be described.

4 Model Performance

Cognitive behavior in subsymbolic systems such as INSOMNET results from the soft constraints that arise in computation over massively interconnected simple processing units. Yet, the primary task of semantic parsing undertaken in this article is generally considered a symbolic one. Nevertheless, the model incorporates graded parsing through the Frame Node Modulator Map, which represents INSOMNET's confidence that any given frame is part of the sentence's interpretation as it is processed *we can go until say one or two*

h0	__	prpstn_rel	SA	h1	__	__	__	__	__
h1	*can*	arg4_event_rel	A0A4DMEV	__	h2	__	e0	__	__
e0	__	EVT	DVASMOTN	bool	-asp	ind	pres	__	__
h4	__	def_rel	BVDMRESC	x1	__	h2	__	__	__
h5	__	def_rel	BVDMRESC	x2	__	h2	__	__	__
h2	*go*	arg1_rel	A0A1DMEV	__	x0	__	e1	__	__
h3	__	def_rel	BVDMRESC	x0	__	h6	__	__	__
h6	*we*	pron_rel	IX	x0	__	__	__	__	__
x0	__	FRI	DVGNPNPT	__	gen	1pl	s1p	__	__
h2	*until*	independent_rel	A0A3DMEV	e1	e2	__	__	__	__
e1	__	EVT	DVASMOTN	bool	-asp	mod	tns	__	__
h2	*say*	degree_rel	DGDM	d0	__	__	__	__	__
h2	*one*	number_hour_rel	APDMHRIXMI	__	d0	*one*	x1	__	__
h2	*or*	conj_rel	CALHLIRHRI	e2	__	x1	__	x2	__
e2	__	CRI	DV	bool	__	__	__	__	__
x1	__	FRI	DVGNPNPT	bool	neu	3sg	prn	__	__
h2	*two*	number_hour_rel	APDMHRIXMI	__	d0	*two*	x2	__	__
x2	__	FRI	DVGNPNPT	bool	neu	3sg	prn	__	__
d0	__	DI	DV	bool	__	__	__	__	__

Figure 13: **Example frameset.** This figure shows an example of the semantic dependency graph representation for the sentence *we can go until say one or two*. The graph contains considerable linguistic information, including modifiers (such as the frame for the word *say*), feature information like gender, person and number for nominals (nouns and pronouns), mood, tense, and aspect for verbal elements (verbs and adjectives), as well as basic feature information for other categories (such as adverbs). This representation of real-world language is used to train and test InSomNet's parsing and comprehension performance.

incrementally. In this section, InSomNet's ability to produce the targeted Frame Nodes (*parsing*) and their contents (*comprehension*) was evaluated on the medium-sized Redwoods Treebank of richly semantically annotated sentences. Because these Mrs representations are based on pointers while Frame Node activation is graded, the evaluation must take into account how the system is to be penalized if a pointer designates a weakly activated frame. Precision and recall measures are therefore used to identify an optimal Frame Node activation threshold so that parsing and comprehension are maximized. While such weak node activation complicates performance analysis, it is actually an important innovation that allows the system to parse language in a cognitively valid manner.

4.1 Mrs Dataset

The evaluation dataset is a corpus of 4817 sentences from the Redwoods Treebank for which at least one analysis was selected as correct by the treebank annotator. Of these, 168 sentences had more than one preferred analysis, and one had as many as seven analyses. These ambiguous sentences were kept in the dataset in order to determine whether InSomNet could learn to maintain several interpretations in parallel. Duplicate sentences with the same interpretation, such as the ubiquitous *let us see*, were removed. The dataset contains the full MRS representation, resulting in 18,225 unique frames overall.

Figure 13 illustrates the semantic dependency graph used in the current study for the sentence *we can go until say one or two*. The graph contains considerable linguistic information, including verb subcategorization, modals (e.g., *can*), modifiers (e.g., *say*), and conjunctions (e.g., *or*). As mentioned in Section 3.1, the MRS representation also contains feature frames that encode feature information like gender, person and number for nominals (nouns and pronouns); mood, tense, and aspect for verbal elements (verbs and adjectives); as well as basic feature information for other categories (such as degree modifiers and conjunctions).

Morphemes were represented as separate tokens in the input sequence, and irregular forms were replaced by their stems. For example, in the sentence *you said you were getting in tuesday night*, the input word *said* is rendered as *say -d* and *getting*, as *get -ing*, with the extra morphemes *-d* and *-ing* processed in separate steps. There were six such morphemes: *-s*, *-ing*, *-er*, *-est*, *-ly*, and *-d* (used for both simple past and perfect participles). Common inflected words, such as *am*, *are*, *were*, etc., were left unchanged. Such preprocessing is not strictly necessary, but allows focusing the study on semantic processing without confounding it with morphology.

All non-handle frame constituents, i.e., **Subcategorization**, **Semantic-Relation**, and **Word**, were given random representations, so that the burden of learning the task fell solely on INSOMNET. The pointer components, **Handle** and **Argument**, develop their own representations as described in Section 3.4.3.

The original version of the MRS dataset had 1,177 semantic relations defined in the ERG. Yet, in most cases, the semantic relation could be distinguished by the word embedded in its definition (e.g. *or* in **_or_rel**), rendering the semantic relation redundant and therefore uninformative. Table 2 shows the number of lexical distinctions in the dataset along with the number of words having that many senses. For example, 845 words (80.2% of the total) were associated with one semantic relation in the corpus, as in the **_or_rel** example above, whereas two words had eight senses: *get* and *like*. Additionally, there were 68 distinct semantic relations for which there was no corresponding word in the sentence, such as **prpstn_rel** and **unspec_loc_rel**.

This correlation between the word as it appears in the input sentence and its semantic relation means that the network must learn to associate the correct semantic relations with the words in the input. This mapping is arbitrary because word and semantic representations were randomly generated. INSOMNET was able to learn this association very well. Nevertheless, in many cases a single input word still had several senses that were distinguished by the semantic relations in the ERG. In such cases, the ERG was used to separate out the word and abstract the semantic relation up one level in its type hierarchy. As an example, the word *say* in Figure 13 occurs with four distinct semantic relations in the MRS dataset, and can be identified with four abstract relations in the HPSG type hierarchy:

_say_approx_rel, as in *we can go until say one or two*, becomes *say* and **degree_rel**;

_say_rel, as in *a moment ago you say -d five o'clock* becomes *say* and **arg123_rel**;

_say_h_rel, as in *but did you say you were free this friday* becomes *say* and **arg134_rel**;

_say_loc_rel as in *at two o'clock on the twenty eighth we say -d* becomes *say* and **arg12_rel**.

This process yielded 104 abstract relations, 71 of which were already in the original dataset. The remaining 33 new relations made distinctions between verb types, some adjectives (such as those

sense count	1	2	3	4	5	6	7	8
word count	845	124	51	19	5	1	1	2

Table 2: **Word sense statistics.** This table shows the degree of semantic ambiguity of words in the MRS dataset. Slightly more than 80% of the words were unambiguous, whereas only four had more than five senses. Thus, INSOMNET had to learn to disambiguate roughly 20% of the words based on sentence context.

that can take extraposed "it"), adverbs, prepositions, and, particularly, nouns (many of which were divided among temporal distinctions). This pairing of words and abstract relations allowed the original semantic relation to remain unique, and at the same time allowed further generalization.

4.2 Evaluation Criteria

If the semantics of a sentence could have been represented as a tree encoded and decoded with RAAM, evaluation of the network's performance would have been straightforward: Decode the sentence-final interpretation by traversing the tree representation and count the number of constituents that are closest to their targets. However, because the semantic representation used in INSOMNET is a directed acyclic graph represented using pointers, two criteria are required to evaluate the accuracy of the semantic representation. The first, termed *parsing*, counts the number of frames in the semantic representation of a sentence that are activated in the Frame Map, as indicated by their level of activation in the Frame Node Modulator Map. This measure requires the identification of a *threshold* above which the activation of a node would be deemed a part of the targeted semantic representation. The second, termed *comprehension*, measures how well the compressed representation in a given Frame Node decodes into its targeted constituents, including pointers. This measure must also take into consideration the accuracy of the pointers, i.e., whether (a) they point to the correct node in the Frame Map, and (b) that node is activated above threshold.

The loosely modular network architecture of INSOMNET makes such a graded evaluation possible and informative. The Frame Encoder/Decoder component is responsible for comprehension and the Frame Node Modulator Map component, for frame selection; these components operate together to yield a semantic interpretation of a sentence. This and Section 5 focus on the accuracy of the semantic representation at the end of sentence processing. Section 6 demonstrates that INSOMNET is cognitively plausible by evaluating the network's performance during incremental sentence processing.

The evaluation metric, called the *Combined Comprehension and Parsing Metric* (CCPM), was designed to measure performance in terms of the semantic parsing and comprehension criteria taken together, as will be described next.

4.3 INSOMNET Performance Measure

Parsing performance is evaluated based on how many of the units in the Frame Node Modulator Map that correspond to target frames are activated. Because these units are graded, indicating the network's confidence that any given frame is part of the target frame set, a *Frame Node Activation Threshold* is needed to determine whether a unit and its corresponding frame is to be counted as part of a sentence's interpretation or not.

This threshold is determined using precision and recall, which are commonly employed in computational linguistics to gauge how well a system is able to distinguish relevant items from irrelevant

Node	Frame	# Fields	# Correct	Activation	Status
0	**h0: prpstn_rel**	1	1	1.00	*tp*
6	**h7: *ball***	5	5	0.60	*fn*
12	**h5: *girl***	6	6	0.85	*tp*
13	**h3: *boy***	6	6	0.90	*tp*
18	**x0: FRI**	8	8	1.00	*tp*
23	**e0: EVT**	8	8	1.00	*tp*
26	**x2: FRI**	8	8	0.85	*tp*
27	**x1: FRI**	8	8	0.95	*tp*
32				0.90	*fp*
37	**h5: *with***	8	8	0.85	*tp*
44	**h1: *with***	8	8	0.20	*fn*
49				0.85	*fp*
54	**h2: *the***	8	8	0.95	*tp*
55	**h4: *the***	8	8	0.90	*tp*
59	**h1: *hit***	9	7	1.00	*tp*
63	**h6: *the***	8	8	0.65	*fn*

Table 3: **Example Data.** This table provides example data corresponding to Figure 14 for the sentence *the boy hit the girl with the ball* as evaluated at the end of sentence.

items, where items in our system are the set of Frame Nodes, $\mathcal{N}$, comprising a sentence's interpretation. Thus, *true positives* (*tp*) are the target frame nodes InSomNet selects (activates above threshold), *false positives* (*fp*) are the non-target frame nodes it selects, and *false negatives* (*fn*) are the target frame nodes it fails to select. (*True negatives* are not often taken into account because they typically constitute the largest proportion of all items and, consequently, do not contribute much information to evaluation.) In this framework, *precision* (P) is the proportion of frame nodes selected by InSomNet that are also in the target set $\mathcal{N}$:

$$P = \frac{tp}{tp + fp},$$

and *recall* (R) is the proportion of target frame nodes InSomNet actually selected:

$$R = \frac{tp}{tp + fn}.$$

These scores are typically combined into a score for overall performance called the *F-measure* (F), defined as

$$F = \frac{2PR}{P + R},$$

assuming that precision and recall are weighted equally. The optimal threshold T, which serves as the parsing measure, is the value that maximizes the F-measure, i.e.,

$$T = \max_{\mathcal{N}} F.$$

InSomNet's *comprehension* performance is evaluated based on how accurately the target Frame Node patterns are decoded into their components regardless of their activation, i.e., as the proportion

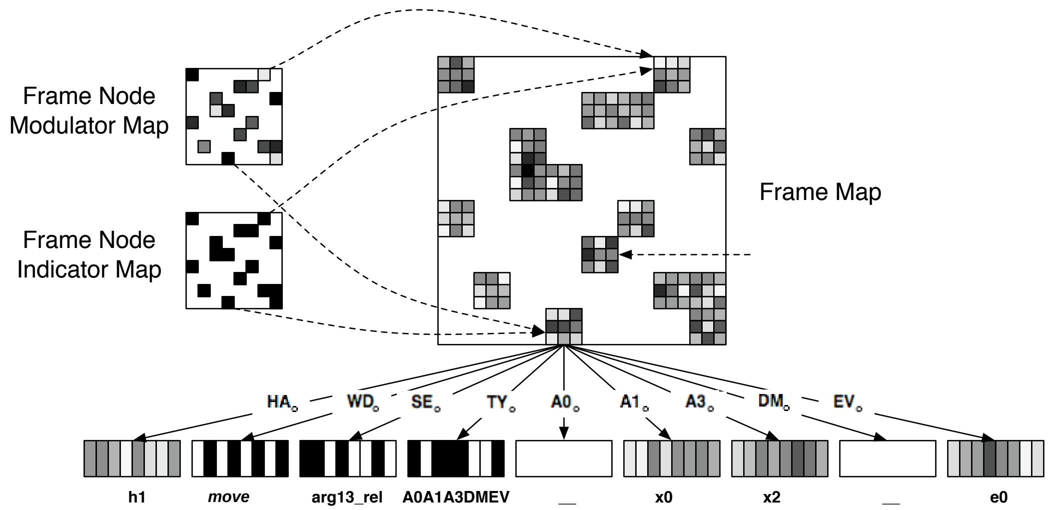


Figure 14: **Evaluation of Comprehension and Parsing.** Evaluation of InSomNet's comprehension and parsing performance is based on how well the Frame Nodes are activated and decoded into their proper components. For example, in this figure, there are 16 nodes, one of which is shown decoded. In this Frame Node (bottom middle), all but the **Word** *move* and one **Argument (A3)** role filler **x2** are decoded correctly (cf. Figure 10), so comprehension for this frame would be $\frac{7}{9} = 0.7778$. If all remaining 90 fields among the true positives decode correctly, the comprehension measure is $A = \frac{97}{99} = 0.9798$. The parsing measure is evaluated with respect to how many units denoting targets in the Frame Node Modulator Map are above threshold $T = 0.85$. Two nodes (32 and 49) are false positives; three nodes (6, 44, and 63) are false negatives. Thus, precision $P = \frac{11}{13} = 0.8462$, recall $R = \frac{11}{14} = 0.7857$, and the F-Measure F for this sentence is computed as 0.8148. The CCPM C is thus $0.9798 \cdot 0.8148 = 0.7983$.

of all decoded fields (t) that are correct (c): $A = \frac{c}{t}$. Note that the comprehension measure does not penalize false positives or negatives because these are already accounted for in the parsing measure.

The Combined Comprehension and Parsing Metric (CCPM) C is then simply $C = A\,T$.

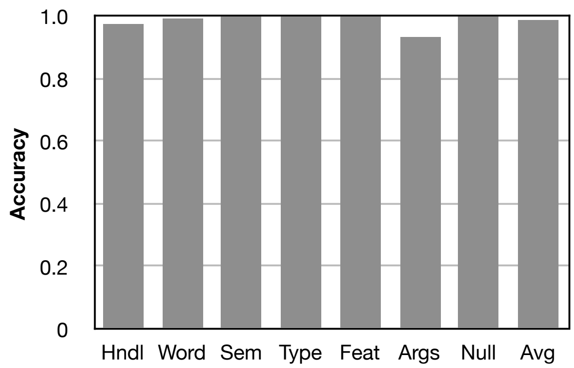
To provide an illustration of how the CCPM is computed on InSomNet's interpretation of the sentence *the boy hit the girl with the ball*, Table 3 lists the relevant data of all nodes but the true negatives. A threshold $T = 0.85$ is assumed to have been computed over the example corpus, and the parsing and comprehension measures are applied to the sentence-final Frame Map shown in Figure 14.

First, InSomNet's parsing measure on this example is evaluated. There are a total of 16 Frame Nodes activated. Eleven are true positives (nodes 0, 12, 13, 18, 23, 26, 27, 37, 54, 55, 59); three are false negatives (nodes 6, 44, and 63); and two are false positives (nodes 32 and 49). Thus, precision $P = \frac{11}{13} = 0.8462$, recall $R = \frac{11}{14} = 0.7857$, and the F-measure $F = \frac{2 \cdot 0.8462 \cdot 0.7857}{0.8462+0.7857} = 0.8148$.

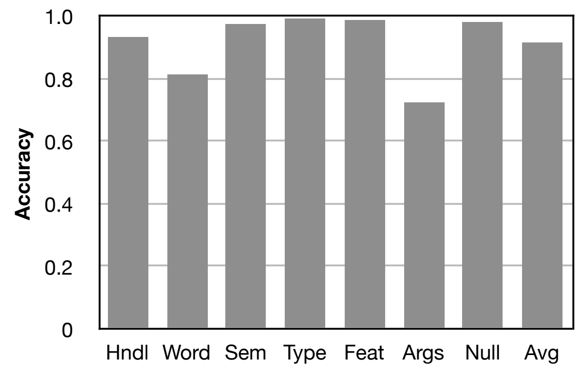
Next, InSomNet's comprehension measure is evaluated. Among the target nodes listed in Table 3, only node 59 has decoding errors: the **Word** field is *move* instead of *hit*, and the **A3** (patient) role is **x2** instead of **x1** (cf. Figure 10). Thus, the comprehension measure is $A = \frac{97}{99} = 0.9798$ on this sentence. The CCPM C for this sentence is then calculated as $0.9798 \cdot 0.8148 = 0.7983$.

4.4 Performance Results

The performance of InSomNet was evaluated using the CCPM on the Mrs dataset using tenfold cross-validation. Figure 15 provides plots on both the Training (a) and Test Sets (b) to facilitate



(a) Training



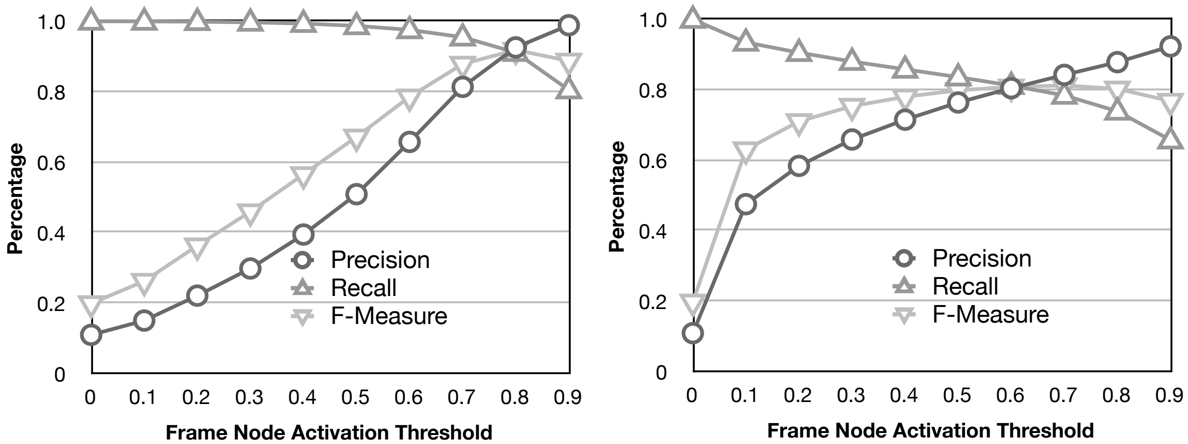
(b) Test

Figure 15: **Comprehension Performance.** The x-axis shows the six MRS components, **Handle** (Hndl), **Word**, **Semantic-Relation** (Sem), **Subcategorization** (Type), **Argument** (Args), and **Feature** (Feat), **NULL** fields, together with the overall **Average** (Avg) that were individually analyzed to identify strengths and weaknesses in INSOMNET's representational capability. The y-axis gives the accuracy for each component. The graphs show that INSOMNET performs well on the Training Set (a) and generalizes well to the Test Set (b) for all other arguments except the **Argument** and **Word** fields, which are most affected by data sparsity.

comparison of their respective comprehension performances: the x-axis lists the six MRS components **Handle**, **Word**, **Semantic-Relation**, **Subcategorization**, **Argument**, and **Feature**, and a **Null** component to indicate empty fillers, along with the overall **Average**. Accuracy for each component is plotted on the y-axis. All but the **Word** and **Argument** components generalize well. The **Word**s account for 47.3% of targets in the MRS dataset. Yet many of these words rarely occur in a given Training Set during cross-validation, so INSOMNET is unable to learn them well. Likewise, the **Argument** components are quantized pointer representations to 18,225 distinct frames in the complete MRS dataset, some of which point to wrong frames. This pointer problem is especially difficult with long sentences, where pointers and feature frames can become disassociated. As an example, the frames having the semantic relations **def_rel** and **arg4_event_rel** in the frameset for the sentence *we can go until say one or two* shown in Figure 13 point to feature frames with the handles **x0**, **x1**, **x2**, and **e0**. INSOMNET accurately reproduces the features, but sometimes loses track of which frames point to these feature frames when they are compressed into handles and quantized. Pointer quantization of these feature frames enforces their distinction by assigning them to unique nodes on the Frame Map, as explained in Section 3.4.3. Nevertheless, sometimes INSOMNET is unable to learn this distinction. Fortunately this problem is not propagated further up in the semantic dependency graph in encoding other handles because the slots in the frames higher up in the MRS graph are populated with handles generated by pointer quantification, the effect of which is to clean up the handles from nodes referenced by these frames.

The **Handle** and **Argument** pointer components are evaluated over the entire Frame Map. Therefore, due to the clustering effect of semantic self-organization, there is a greater probability for the pointer components to be confounded by similar non-target Frame Nodes that are activated in the Frame Map than there is for the non-pointer **Word**, **Semantic-Relation**, **Subcategorization**, and **Feature** components that comprise Frame Nodes and have static random representations.

The overall performance of the network is only slightly worse on the Test Set than on the Training Set. There are relatively few false positives, but those that do occur have a noticeable impact on the



(a) Training **(b) Test**

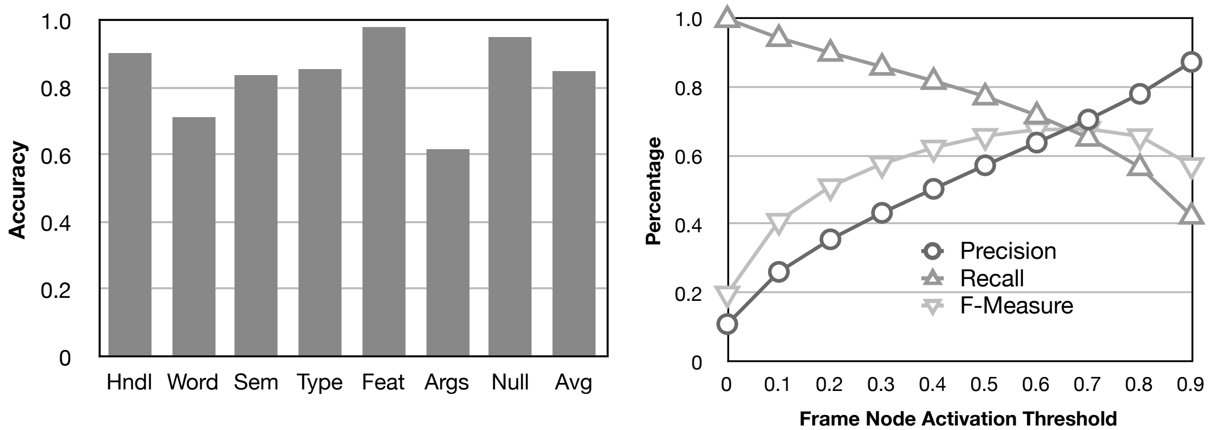
Figure 16: **Parsing Performance.** These graphs show how well InSomNet identifies those nodes that are part of the final interpretation. Accordingly, precision and recall are used to find an optimal F-measure. InSomNet performs well on the Training Set (a) and also generalizes on the Test Set (b) well, identifying those frames that belong to the sentence interpretation with a maximum F-measure of 0.81 at $x = 0.7$.

accuracy of the **Argument** field. The reason is that the false positives are very similar to some of the true positives and they can therefore confuse the network, indicating that the network is generalizing perhaps too well.

Figure 16 shows the basic parsing performance for both Training and Test Sets in terms of precision, recall, and F-measure. The F-Measure for the Training Set (a) is 0.9188 at $x = 0.8$, and for the Test Set (b) is 0.8128 at $x = 0.7$. The CCPM, then, for the two sets is (a) 0.9058 and (b) 0.7369. The combined overall measure provides a useful means of comparing performance between datasets, as well as generalization.

4.5 Error Analysis

More than half of the errors InSomNet makes are due to inaccurate **Argument**s. This result is not surprising since **Argument**s occur multiple times in most frames and range over the entire Frame Map, where they can be confounded by similar pointers. In order to generate Arguments, a Raam network needs to be trained to ensure that handles are encoded accurately, that they reference the correct Frame Node, and that they decode correctly into a complete frame, as described in detail in Section 3.4.3. Many erroneous quantized Arguments point to Frame Nodes that decode into almost exactly the same frame, but for one field (indeed, there are a few that do replicate the information in the targeted Frame Node). To get an idea of how close the misses are, consider that roughly 35% of the **Argument**s point to neighboring nodes. Because the Frame Map is based on a SardNet, most of these nodes would have been viable targets had they been trained to represent the frame filling the role of the **Argument** pointing to them. Yet, training these nodes would probably lead to more false positives, while substantially increasing training time. At any rate, improving the accuracy of the **Argument** component would improve the network's performance across the board. How this may be done will be discussed in Section 7.



(a) Comprehension **(b) Parsing**

Figure 17: **The Benefit of Semantic Self-Organization.** Compared with InSomNet's average test (b) performance as shown in Figures 15 and 16, the advantange of semantic self-organization is clear. The difference in the **Word**, **Semantic-Relation**, **Subcategorization**, and **Argument** components for comprehension (a) are significant, because semantic self-organization increases the distinction between dissimilar frames, reducing conflicts among frames. Also, when the map is not self-organized, the maximum F-measure (b) is 0.67 at $x = 0.7$, versus 0.81 at $x = 0.7$ as a result of self-organization.

4.6 The Benefit of Semantic Self-Organization

An interesting question is whether organizing the compressed semantic representations on a map confers any advantage over simply using an unorganized grid. Figure 17 shows that self-organization indeed results in a significant performance gain. Figure 17 (a) shows a component breakout of the frame representations for both an unorganized and an organized map on the test set. A two-tailed paired t-test shows these results are statistically significant ($p = 0.0092$, $t = 3.7811$). The most significant improvements are visible in the **Word**, **Semantic-Relation**, **Subcategorization**, and **Argument** components. Nevertheless, the network performs perhaps better than would have been expected without self-organization. The reason is that the compressed frame representations do preserve similarities when the frame constituents are similar, and similar frames will still tend to map to the same (random) unit in an unorganized map. The topological properties of the semantic input space are lost, however, resulting in an increase in conflicts among dissimilar frames being mapped to the same Frame Node. Figure 17 (b) also shows that precision and recall decrease significantly, with a maximum F-measure of 0.67 at $x = 0.7$.

4.7 Model Comparison

Head-to-head performance comparisons with statistical models are difficult to do for two reasons: (1) sentence processing with InSomNet results in a graded semantic parse, unlike parse-ranking models, and (2) the rich Mrs representation is difficult for other parsers to handle because it is a DAG rather than a tree. Nevertheless, a qualitative comparison was run, upon our request, with the state-of-the-art grammar-based conditional log-linear parsing model described in Toutanova, Manning, Shieber, Flickinger, and Oepen (2002). For this comparison, InSomNet was trained on a subset of the Mrs dataset that contained representations of *elementary semantic dependency graphs.* These graphs were *we can go until say one or two*

```
| h0 prpstn_rel        SA         e0  __  __  __  __  __ |
| e0 _can_rel          A0A4DMEV   __  e2  __  __  __  __ |
| e1 until_rel         A0A3DMEV   e2  e3  __  __  __  __ |
| e2 _go_rel           A0A1DMEV   __  x0  __  __  __  __ |
| e3 _or_rel           CALHLIRHRI e2  __  x1  __  x1  __ |
| h1 def_rel           BVDMRESC   x0  __  __  __  __  __ |
| x0 pron_rel          IX         __  __  __  __  __  __ |
| h2 def_rel           BVDMRESC   x1  __  __  __  __  __ |
| x1 number_hour_rel   APDMHRIXMI __  __  __  x1  __  __ |
```

Figure 18: **Dataset Comparison.** This figure shows an example of the differences between the elementary semantic dependency graphs used in the model comparison to be contrasted with the full MRS dependency graph representation for the sentence *we can go until say one or two* shown in Figure 13. The full graph contains more than twice as many frames, including modifiers (such as the frame for the word *say*), extra feature information like gender, person and number for nominals (nouns and pronouns; mood, tense, and aspect for verbal elements (verbs and adjectives), as well as basic feature information for other categories (such as adverbs). The full graph also has more argument roles filled in (e.g., the additional *restriction* (**RE**) in the **def_rel** frames) and uses handle-sharing. As this comparison makes clear, the full MRS dependency representation is considerably more complex than the elementary MRS dependency graphs used in the statistical model. Yet, INSOMNET performs as well on the full MRS dataset as the elementary semantic dependency graphs, suggesting that the performance is at least comparable to state-of-the-art statistical parsers.

designed to capture the basic predicate argument structure while omitting details such as the verbal and nominal features (Figure 18). In order to parse the data, Toutanova's model was trained with the following kinds of information: (1) features derived from the argument labels for each semantic relation in the graphs, (2) the roles the semantic relation governs, and (3) semantic relations the roles typically take. Toutanova et al. (2002) used maximum entropy to estimate the probability of licensed graph structures based on the feature set, which have the important advantage that they need not be independent. Using ten-fold cross-validation, they reported an overall accuracy of 74.9% (Toutanova, personal communication). Also using ten-fold cross-validation, INSOMNET was trained on the elementary semantic dependency graphs rendered as sets of frames, and achieved an overall F-measure of 0.75 at $x = 0.7$ using the CCPM. INSOMNET does not have access to a grammar, but must approximate the grammar implicit in the semantic annotations of sentences. Nevertheless, INSOMNET performs as well on the simplified dataset as it does on the full MRS dependency graphs.

4.8 Performance Conclusions

In this section, INSOMNET was evaluated on the MRS representations of approximately 5000 sentences from the Redwoods Treebank. The network generalized well and benefited significantly from self-organization. INSOMNET was also trained and tested on the same dataset as the conditional log-linear model by Toutanova et al. (2002) to allow a rough comparison between INSOMNET and a state-of-the-art statistical model. Overall, these results demonstrate that the subsymbolic approach is at least comparable in performance to statistical parsers, scaling up beyond toy grammars to parse sentences with realistic complexity. Moreover, it does so while retaining its power as a cognitive model, as will be discussed in Section 6.

5 Model Robustness

One of the most alluring properties of connectionist models is their automatic robustness to noise, both in the form of errors in the input and lesioning of the network itself. In this section, InSomNet is first evaluated on how robustly it is able to process sentences that contain a variety of dysfluencies and speech errors. These dysfluencies and speech errors were a part of the original Verbmobil corpus transcriptions. Then, InSomNet is evaluated on how well it tolerates Gaussian noise injected into the network's weights.

5.1 VerbMobil Transcription

The sentences that make up the dataset used in Section 4.1 were originally derived from the audio transcriptions of the VerbMobil project (Wahlster, 2000) and cleaned up to yield well-formed English sentences for the Redwoods Treebank. Those sentences that were not given at least one grammatical analysis by the Erg were not included. The original sentences in the Redwoods Treebank are distributed across four VerbMobil datasets: CDs 6, 13, 31, and 32. Each dataset records a number of dialogues between two people whose spoken *sentences* are divided into *turns* for each person (Alexandersson, Reithinger, & Maier, 1997). The turns are hand-segmented and transcribed to give as faithful a textual representation of the audio-taped dialogue as possible.

All but CD 13 use the VerbMobil-II transcription convention, which is a more detailed annotation scheme than the VerbMobil-I convention. In order to regenerate a transcribed test dataset of sentences for InSomNet that correspond to the cleaned-up sentences, the four VerbMobil datasets were analyzed to find the original sentences, and the VerbMobil-II transcriptions were simplified to be consistent with the VerbMobil-I annotation of CD 13. Table 4 shows the VerbMobil transcription symbols, their meaning, and the simplified conventions used for InSomNet. This simplified set consisted of noises (#), hesitations and interjections (variations on *uh*), and speech repairs, where `X` in the *VerbMobil Transcription* column in Table 4 denotes extrasentential sounds that are preserved in the *Simplified Transcription* column used for the InSomNet annotations. Thus, for example, Table 5 shows one of the original VerbMobil transcribed sentences located on dataset CD 31 that corresponds to the cleaned-up Redwoods sentence *I am free the first through the fourth.* The original sentence features a false start such as `-/the fir/-`. Using Table 4, this transcription is simplified to *the fir* (i.e., without the enclosing `-/` and `/-` annotations).

This process yielded a total of 5068 sentences for the 4817 sentences in the Mrs dataset. The 251 extra sentences resulted from the different annotations derived from the same cleaned-up sentence. For example, each of the original annotated sentences *hm let us see*, *uh let us see*, and *# let us see # #* mapped to the ubiquitous Redwoods sentence *let us see* when cleaned of their annotations. Figure 19 shows the percentage of sentences in the transcribed test dataset that differed from the cleaned-up versions according to the number of extra annotation symbols in each sentence. Of the 5068 sentences, 4543 had four or fewer extra symbols, and 904 of these had no extra annotations. Only eight sentences had twelve or more added annotations. Each new symbol was given a random

VerbMobil Transcription	*Definition*	*Simplified Transcription*
`<Smack>`	human noise	#
`<Swallow>`	human noise	#
`<Throat>`	human noise	#
`<Cough>`	human noise	#
`<Laugh>`	human noise	#
`<Noise>`	human noise	#
`<\#Squeak>`	mechanical noise	#
`<\#Rustle>`	mechanical noise	#
`<\#Knock>`	mechanical noise	#
`<hes>`	hesitation	*uh*
`<uh>`	hesitation	*uh*
`<\"{a}h>`	hesitation	*uh*
`<uhm>`	hesitation	*uhm*
`<\"{a}hm>`	hesitation	*uhm*
`<hm>`	hesitation	*hm*
`<uh-huh>`	affirmative interjection	*uh huh*
`<mhm>`	affirmative interjection	*mhm*
`<uh-uh>`	negative interjection	*uh uh*
`<mm>`	negative interjection	*mm*
`+/X/+`	repetition/correction	X
`-/X/-`	false start	X
`<*X>`	foreign word	X
`~X`	proper name	X
`*X`	neologism	X
`\#X`	number annotation	X
`\$X`	spelling out	X
`[<:...>X:>]`	noise interference	X
`<!X>`	pronunciation comment	X
`<;...>`	commentary	(Not Used)
`\_`	articulatory interruption	(Not Used)
`;`	global comment	(Not Used)
`<B>`	breathing	(Not Used)
`<P>`	empty pause	(Not Used)
`<*T>`	technical interruption	(Not Used)
`<*T>t`	turn break	(Not Used)
`<T\_>`	beginning of turn or word missing	(Not Used)
`<\_T>`	end of word missing	(Not Used)
`\%`	difficult to hear	(Not Used)
`<\%>`	incomprehensible	(Not Used)

Table 4: **VerbMobil Transcriptions.** The VerbMobil annotations are designed to provide a faithful textual representation of all elements of a recorded dialogue. The symbol `X` in the *VerbMobil Transcription* column is simply a variable that stands for extrasentential sounds in the transcription that are preserved in the *Simplified Transciption*, but without the original punctuation. The entry (Not Used) in the last column means that the transciption was not represented. This simplified set of transcriptions makes it possible to represent sentences in all four original VerbMobil datasets consistently.

representation that was effectively treated as noise by InSomNet.

5.2 Robustness Evaluation

InSomNet's robustness was evaluated in two ways on the transcribed dataset using the CCPM

Original VerbMobil Transcribed Sentence:

<Laugh> <B> <uh> -/the fir=/- <!2 I'm> free <uh> the #first through the #fourth

Simplified Transcriptions of Original VerbMobil annotations:

Original	*Simplified*
<Laugh>	#
<B>	(not transcribed)
<uh>	*uh*
-/the fir/-	*the fir*
<!2 I'm>	*i m*
#first	*first*
#fourth	*fourth*

Simplified Transcribed Sentence:

 # uh the fir I m free uh the first through the fourth.

Table 5: **Example Transcribed Sentence.** To demonstrate InSomNet's robustness to a variety of speech errors, the transcriptions of sentences from the four VerbMobil datasets were simplified for the sake of consistency and to present these sentences as a sequence of tokens to InSomNet. For example, the false start -/the fir/- is simplified to *the fir*, with the novel "word" *fir* given a random representation.

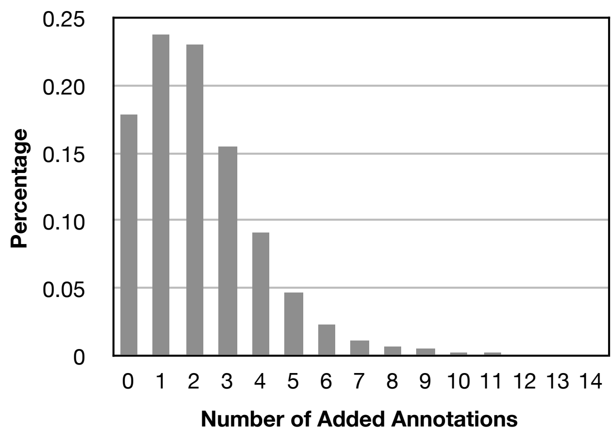
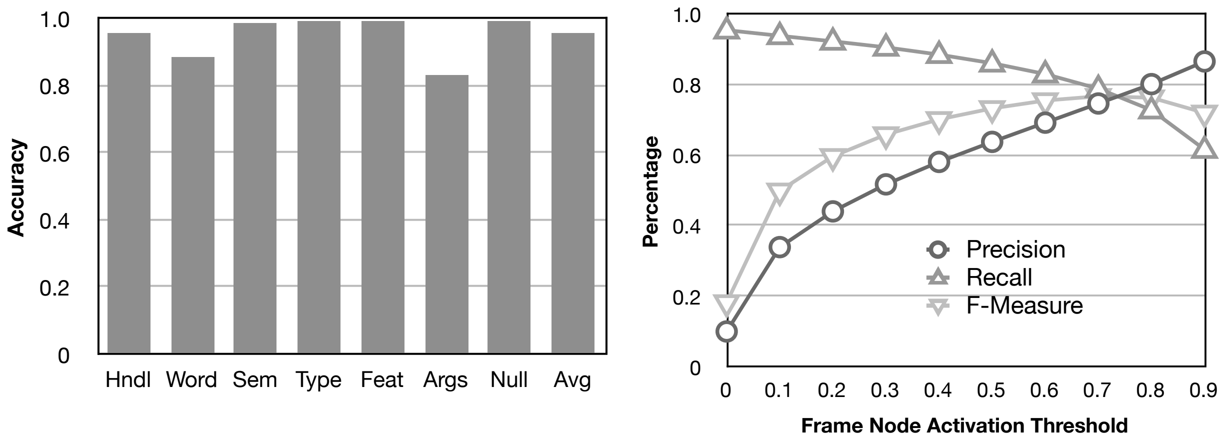


Figure 19: **Extra Transcription Symbols.** The x-axis denotes the difference between the length of sentences in the transcribed dataset and their cleaned-up counterparts in the Redwoods Treebank. The y-axis gives the percentage of sentences from the corpus that differed by these amounts. Thus, $x = 0$ means that 904 sentences, or approximately 18% of the total number of sentences in the transcribed dataset, were left unchanged. One sentence had as many as 14 extra transcription tokens added. Most (90%) had four or fewer additional tokens. Thus, 82% of the sentences InSomNet was tested on for robustness included extra transcriptions.

introduced in Section 4.3 using one of the trained systems from Section 4.4. In the first evaluation, the trained model was tested on the transcribed dataset described above to measure how well the model tolerates the extra annotations. In the second evaluation, Gaussian noise with a mean of 0 and standard deviation ranging from 0.0001 to 0.1 was injected into all of the network weights to measure how well InSomNet tolerates damage to the network.



(a) Comprehension **(b) Parsing**

Figure 20: **Robust Performance With Transcribed Spoken Language.** InSomNet's comprehension (a) and parsing (b) performance is very good on the robust dataset. The F-measure is highest at $x = 0.7$ with a value of 0.77. These results show that InSomNet is very robust to dysfluencies in the input.

5.3 Robustness Results

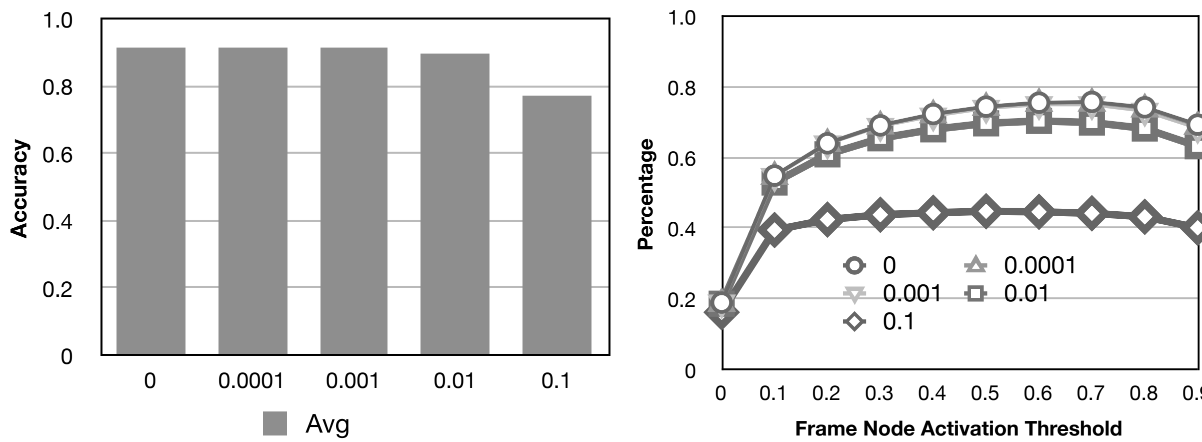
Figure 20 shows that InSomNet's comprehension performance (a) is very similar to its performance on the unaltered test data shown in Figure 15 (b). The highest parsing F-measure (b) is 0.77 at a threshold of $x = 0.7$, yielding an CCPM of 0.7344, which is also comparable to InSomNet's parsing performance on the unannotated test sentences shown in Figure 16 (b). These results show that InSomNet tolerates the extra annotation symbols very well. Note that InSomNet was *not* trained on the robust dataset: Its performance is simply extremely robust. Even though the system was only trained on clean data, it could still process sentences annotated with transcriptions for numerous effects, including

- ungrammatical input: *here is some clues*
- accents: *uhm that ist no good*
- contractions: *# # uh how 'bout some time #*
- dysfluency: *I am free # from ah , that , # from nine , until , # oh # six*

Figure 21 shows the effect of different levels of noise in the network's weights on the Test Set. Figure 21 (a) shows average comprehension performance on the Test Set, and Figure 21 (b) shows the corresponding F-measures for each noise level. A noise level of 0.0001 is virtually indistiguishable from no noise, while 0.001 degrades accuracy very slightly. Performance at a noise level of 0.01 falls off more clearly, but InSomNet still tolerates it rather well. Finally, at a noise level of 0.1 performance decreases substantially.

5.4 Robustness Conclusions

InSomNet demonstrates graceful degradation in performance, both to errors in the input and noise added to the weights. Such robustness is a hallmark characteristic of subsymbolic systems that have made them the models of choice for researchers investigating psycholinguistic phenomena related



(a) Comprehension **(b) Parsing**

Figure 21: **Robustness to Gaussian Noise Lesioning.** Performance of InSomNet is evaluated on the test data for different levels of noise added to the network weights. In all cases, the mean of the noise was 0, and standard deviations ranged from 0.0001 to 0.1. The average overall comprehension (a) and parsing (b) performance show the graceful degradation that is a hallmark of subsymbolic models such as InSomNet. A noise level of 0.0001 shows no real effect on the network's performance, but when increased to 0.1, the effect is much more pronounced. These results show that InSomNet is not only robust to dysfluent input, but also to damage to the network weights.

to issues of gradience, such as grammaticality, as well as the effects of aphasia. Robust sentence processing is a behavior that emerges naturally from parallel computation across massively interconnected units that comprise subsymbolic models. Such behavior would be very difficult to elicit from rule-based systems.

6 Model Cognitive Validity

In Section 2.1, several characteristics of human sentence processing as revealed by psycholinguistic research were discussed that a cognitively plausible model should exhibit. First, InSomNet should build and maintain multiple interpretations in parallel in the face of *ambiguity*. Second, *incremental sentence* processing should keep the potential combinatorial explosion of multiple interpretations in check: each word read in the input imposes linguistic and semantic constraints on *viable* interpretations, as well as the broader context in which a sentence is situated, including language experience among other factors such as discourse and environment. Third, the model should seamlessly *adapt to* and *integrate* these constraints into the developing interpretations. Fourth, the model should exhibit *anticipatory* behavior in the form of *expectations* and *semantic priming*. Fifth, the model should be able to *nonmonotonically revise* interpretations as disambiguating information is processed.

In order for the network to learn that a given sentence may be interpreted in different ways, the sentence must occur with more than one interpretation during training. If those interpretations also occur with different frequencies, then the network will become sensitive to these statistics and prefer one interpretation over another, without necessarily pruning the less likely interpretation out completely. As an example, the frameset in Figure 8 shows two interpretations of a sentence with

Template	Sentence Frame	Case Roles
1	The **human** ate.	agent
2	The **human** ate the **food**.	agent/patient
3	The **human** ate the **food** with the **food**.	agent/patient/modifier
4	The **human** ate the **food** with the **utensil**.	agent/patient/instrument
5	The **animal** ate.	agent
6	The **predator** ate the **prey**.	agent/patient
7	The **human** broke the **fragileObject**.	agent/patient
8	The **human** broke the **fragileObject** with the **breaker**.	agent/patient/instrument
9	The **breaker** broke the **fragileObject**.	instrument/patient
10	The **animal** broke the **fragileObject**.	agent/patient
11	The **fragileObject** broke.	patient
12	The **human** hit the **thing**.	agent/patient
13	The **human** hit the **human** with the **possession**.	agent/patient/modifier
14	The **human** hit the **human** with the **hitter**.	agent/patient/instrument
15	The **hitter** hit the **thing**.	instrument/patient
16	The **human** moved.	agent/patient
17	The **human** moved the **object**.	agent/patient
18	The **animal** moved.	agent/patient
19	The **object** moved.	patient

Figure 22: **Sentence Templates.** Each template is used to generate sentences by filling in categories (shown in bold type) with nouns from that category as listed in Table 23. Because a noun may belong to more than one category, ambiguous sentences are also generated. For example, the word *bat* may be both an **animal** and an **object**. Replacing these categories in templates 18 and 19 yield the sentence *the bat moved* with two case-role interpretations, either agent or patient. This data can be used to test the cognitive validity of InSomNet with ambiguous input.

prepositional phrase attachment. The two senses will generally be encoded in separate Frame Nodes in InSomNet, although not necessarily (to the extent that it does so depends on model parameters such as learning rate, the size of the Frame Map, and how the Frame Node Indicator Map is self-organized). How strongly these two senses are activated depends on where in the sentence the network is and how strongly the processed words correlate with each sense.

In this section, the cognitive validity of InSomNet is evaluated with respect to how it handles ambiguity in terms of coactivation of multiple interpretations, expectations, semantic priming, and nonmonotonic revision of interpretations in the face of disambiguating information.

6.1 Training Data

While the results thus far demonstrate that InSomNet scales up to realistic language as provided by the Mrs representations of sentences from the Redwoods Treebank, there are too few sentences in the corpus exhibiting the psycholinguistically interesting phenomena listed above, nor do they appear systematically to allow for a meaningful demonstration of cognitive plausibility. Accordingly, in order to evaluate its cognitive behavior, InSomNet was trained on the dataset originally created by McClelland and Kawamoto (1986) and modified by Miikkulainen (here called the "psych dataset"). In McClelland and Kawamoto's study, a connectionist system was trained to map syntactic constituents to thematic roles using a corpus of 152 sentences over 34 nouns and verbs for which semantic features (e.g., animacy) were prespecified. In Miikkulainen's task, the network learned to develop its own word representations and a portion of the original McClelland and Kawamoto

Category	Nouns
thing	**human animal object**
human	*man woman boy girl*
animal	*bat chicken dog sheep wolf lion*
predator	*wolf lion*
prey	*chicken sheep*
food	*chicken cheese pasta carrot*
utensil	*fork spoon*
fragileObject	*plate window vase*
hitter	*bat ball hatchet hammer vase paperweight rock*
breaker	*bat ball hatchet hammer paperweight rock*
possession	*bat ball hatchet hammer vase dog doll*
object	*bat ball hatchet hammer paperweight rock vase plate window fork spoon pasta cheese chicken carrot desk doll curtain*

Figure 23: **Noun Categories.** Each category in the templates from Figure 22 can be replaced by the nouns to its right. Notice that the categories overlap and comprise a basic ontology with **thing** as the root category.

dataset was therefore expanded to 1475 sentences over 30 words, using the 19 sentence templates and 12 noun categories shown in Figures 22 and 23. The categories define a basic ontology with **thing** as the root. The 1475 sentences were generated by replacing each category in each template with a noun belonging to that category.

Of those 1475 sentences, roughly half ($\frac{728}{1475} = 49.4\%$) have an instrumental reading, generated by templates 4, 8, and 14. Only 112 (7.6%) have a modifier sense, given by templates 3 and 13. Thus, the instrumental interpretation is more than six times more frequent than the modifier interpretation. Templates 9, 11, 15, and 19 generate 221 sentences with a non-agent subject, and templates 7, 10, 12, 16, and 18 generate 144 sentences with the same syntactic structure as the sentences with a non-agent subject, but with an agent subject. For example, the sentence *the bat broke the window* has a non-agent subject when the word *bat* is interpreted as an inanimate object, but an agent subject when *bat* is interpreted as animate. A total of 85 sentences in the dataset were globally ambiguous (i.e., had two or more interpretations for the sentence as a whole, such as the sentence *the boy hit the girl with the ball*, where *ball* may be an instrument or modifier).

To train InSomNet, an Mrs frameset for each sentence in the psych dataset was constructed. The dataset was divided into a Training Set of 1438 sentences and Test Set of 37 sentences. The sentence *the boy hit the girl with the ball* will illustrate the encoding scheme used. This sentence has the same syntactic structure as the sentence *the boy hit the girl with the doll* that has been used as a running example throughout the article, but is more ambiguous in that people do not have as strong a preference for the modifier reading for *ball* as they do for *doll*. This preference is reflected in the psych dataset by giving both the instrument and modifer interpretations for *the boy hit the girl with the ball*, but only the modifier reading for *the boy hit the girl with the doll* (as shown in Figure 24). For sentences that have an instrument interpretation for the prepositional phrase, the frame for the preposition *with* has the same handle **h1** as the verb to which it attaches, and the **A0** role of *with* will have the same representation as the **EV** role of *hit*. The modifier reading is indicated by the *with* frame sharing the same handle **h5** as the noun *girl* it modifies, while its **A0** role will match the *girl*'s **IX** role. The sequence *the boy hit* occurs 182 times in the Training Set with the instrument sense, but only 27 times with the modifier sense, and *the boy hit the girl* occurs with both senses 7 times. Therefore, InSomNet should have a strong preference for the instrument

Prefix	*Instrument*	*Modifier*
the boy hit	182	27
the boy hit the girl	7	7
the boy hit the girl with the doll	0	1
the boy hit the girl with the ball	1	1

Figure 24: **Sentence Prefix Frequencies in the Psych Training Dataset.** This table summarizes the tallies for the instrumental and modifier interpretations of the sentences *the boy hit the girl with the doll* and *the boy hit the girl with the ball* as each content word is processed. Initially, InSomNet should presume the instrumental interpretation because it occurs almost seven times as frequently as the modifier when *the boy hit* is read in. Once *the girl* is processed, the two interpretations occur equally often. The final prepositional phrase *with the doll* only occurs with the modifier interpretation, so InSomNet should learn to suppress the instrumental interpretation completely at this point. Because *with the ball* remains globally ambiguous, InSomNet should maintain both interpretations activated. In this manner, the frequencies of different ambiguous interpretations are expressed in the data. InSomNet should be able to generate and weight expectations based on these frequencies.

reading until it reads in the word *girl*, at which point both senses should be approximately the same. Once the end-of-sentence marker is read, the activation of the Frame Node with the strongest sense should approach the value 1.0, and the activation of the Frame Node with the opposite sense should fall off to 0.0. In the case of the sentence *the boy hit the girl with the ball*, these two activations strengths should be roughly equal, whereas for *the boy hit the girl with the doll*, only the modifier interpretation should be strongly activated. This is indeed the case, as will be demonstrated in the next section.

6.2 Results of Psycholinguistics Experiments

InSomNet learned the case-role mapping task well, with an average performance on the Test Set of 95.5%. The system of Miikkulainen (1997), which induced its own representations, achieved approximately 96% accuracy on the data, and the original, more limited McClelland and Kawamoto system achieved an accuracy of 85% in identifying thematic relations.

In order to demonstrate cognitive validity, InSomNet's performance needs to be analyzed on actual ambiguous sentences, such as *the boy hit the girl with the doll* and *the boy hit the girl with the ball* shown in Figure 25. How it coactivates multiple senses, resolves ambiguities, primes semantically related words, generate expectations of potential future inputs, and revises its sentence interpretations nonmonotonically, will be analyzed. Because these two sentences occur with slightly different statistics, InSomNet should process them in a manner consistent with these statistics.

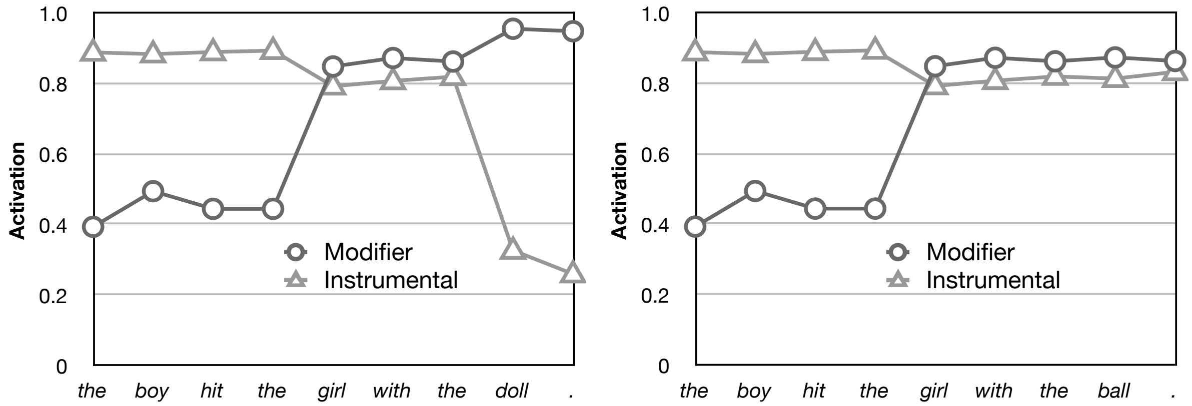
6.2.1 Coactivation of Multiple Senses and Ambiguity Resolution

InSomNet is trained to represent ambiguity by coactivating the Frame Node patterns on the Frame Map that encode the different sense distinctions (in this case) of the preposition *with*. Figure 25 illustrates this idea with the prepositional phrase attachment ambiguity in the sentence *the boy hit the girl with the ball*. In the interpretation with *ball* as an instrument, a distinct Frame Node decodes into an Mrs frame in which *with* has the same handle representation as the verb *hit* frame (**h1**) and the same filler representation in its **A0** role as the verb's **EV** role in the *hit* verbal feature

Figure 25: **Representing Ambiguity in InSomNet.** The prepositional phrase attachment ambiguity in the sentence *the boy hit the girl with the ball* allows the two interpretations of *ball* as an instrument or as a modifier to be represented in the *with* frames shown in Figure 8. Both interpretations are activated in InSomNet: Two separate Frame Nodes are decoded into the two distinct *with* frames. In the interpretation with *doll* as an instrument, the *with* frame has the same handle **h1** as the verb *hit* to represent verb-attachment, and its **A0** role has the same filler as the verb's **EV** role **e0**. In the interpretation with *ball* as a modifier, the *with* frame has the same handle **h5** as the noun *girl* frame to represent noun-attachment and its **A0** role has the same filler as the noun's **IX** role **x1**. Allowing such multiple representations to be explicitly activated is one of the main advantages of the Frame Map component of InSomNet.

frame (**e0**). In the interpretation with *ball* as a modifier of *girl*, another distinct Frame Node decodes into an Mrs frame in which *with* has the same handle representation as the *girl* frame (**h5**) and the same representation in its **A0** role as the **IX** role in the *girl* nominal feature frame (**x1**). The two distinct *with* frames are coactivated in the Frame Node Modulator Map with activations that depend on the sequence of words InSomNet has read in. In this manner, InSomNet maintains both interpretations explicitly until it encounters disambiguating information that, in this case, relies on the semantics of the object of the prepositional phrase, not just structural information.

InSomNet should be able to not only represent ambiguity, but disambiguate sentences given appropriate context with respect to the frequency of a given interpretation in the training set. The sentence *the boy hit the girl with the ball* is globally ambiguous in the corpus because both modifier (M) and instrument (I) senses are acceptable, whereas *the boy hit the girl with the doll* only occurs with the modifier sense. Figure 26 shows the sense activation of the *with* Frame Node that decodes into the modifier representation and the *with* Frame Node that decodes into the instrumental representation. Following the relative frequencies of the two interpretations, InSomNet demonstrates an *a priori*



(a) Unambiguous Interpretation **(b) Ambiguous Interpretation**

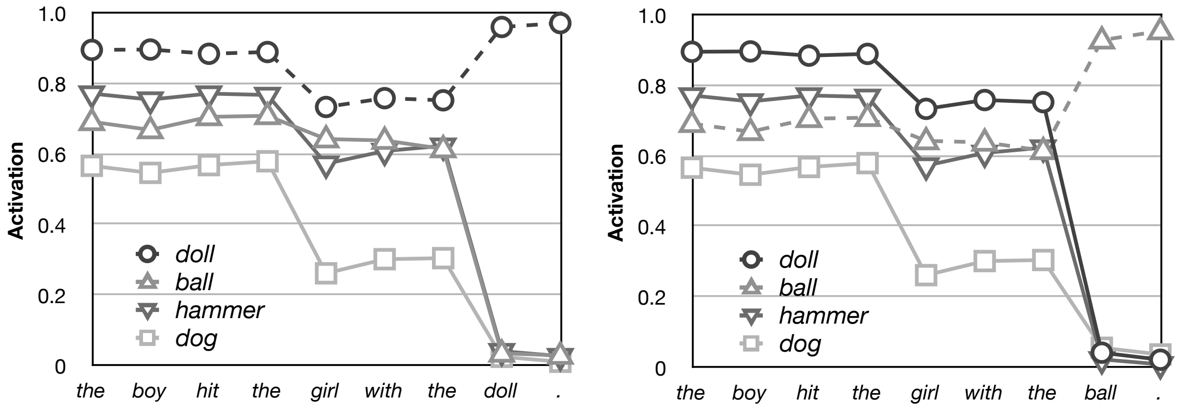
Figure 26: **Activation and Resolution of Alternative Interpretations.** This figure demonstrates how InSomNet interprets the prepositional phrase attachment ambiguity in the unambiguous sentence (a) *the boy hit the girl with the doll* and globally ambiguous sentence (b) *the boy hit the girl with the ball*. The activation of the two *with* nodes in the Frame Map (one representing the modifier sense, and the other, the instrumental sense) is plotted over time, showing how the interpretation varies as each word is read in. In the unambiguous sentence (a), InSomNet is able to correctly identify the modifier sense of *with* when it reads in the disambiguating noun *doll*. In the globally ambiguous sentence (b), both senses of *with* remain activated. In this manner, InSomNet demonstrates sensitivity to the likely thematic role of the object of the preposition, and, thus, to attachment preference.

preference for the instrumental reading until the word *girl* is encountered. Once *the girl* is processed, both interpretations are roughly equal. On reading the final prepositional phrase *with the doll*, InSomNet strongly prefers the modifier reading, while the instrumental reading is suppressed. On reading *with the ball*, both interpretations remain activated. This approach to representing multiple coactivated interpretations until disambiguated means that InSomNet need not prune away initially unlikely interpretations that may later prove correct. Rather, traces of dispreferred interpretations remain in the Hidden Layer, which allows the network to recover them later with appropriate context. Indeed, it is unusual for licensed interpretatons to be lost completely, as contrasted with statistical parsers that employ a beam width, in which an unlikely parse is completely pruned away if it falls outside the beam.

6.2.2 Expectations, Semantic priming, and Nonmonotonocity.

Figure 27 shows how expectations and semantic priming arise during sentence processing in InSomNet. Plots of the activations of the frame nodes encoding *doll*, *ball*, *hammer*, and *dog* are provided as the words of the sentence *the boy hit the girl with the doll/ball* are read in. It is important to to keep in mind that the nodes do not encode the roles of the nouns (cf. frames **h3**, **h5**, and **h7** in Figure 6). That information is only encoded in the verb frame for *hit* (**h1**) and the *with* frames (either **h1** for the instrumental interpretation or **h5** for the modifier interpretation for *ball*). For this reason, these nouns are activated to varying degrees because they could fill the **Agent**, **Patient** of the verb, or **Object** of the preposition *with*.

Initially, all of the words are activated as possible continuations of the sentence, i.e., as expecta-



(a) ... *doll* **(b)** ... *ball*

Figure 27: **Expectations, Semantic Priming, and Nonmonotonicity.** When either the sentence (a) *the boy hit the girl with the doll* or (b) *the boy hit the girl with the ball* is processed, other potential role fillers such as *hammer* and *dog* are also initially activated. The dashed line shows the trajectory for the actual target word. By the end of the sentence, the other words are suppressed once the target word is read in. The initial activation of other words demonstrates that the network has learned to develop expectations about possible upcoming role fillers, including those semantically related to the words in the sentence as it is processed. In (a), the nonmonotonic revision of the sentence is particularly apparent because the activation of *doll* starts highly activated, then subsides somewhat, and finally increased substantially when the word is read in at the end of the sentence, exhibiting the cognitive phenomenon of semantic flipping.

tions. The activations of the target objects *doll* and *ball*, as well as the non-target words *hammer* and *dog*, fluctuate as the sentence is processed. In Figure 27 (a) and (b), the word *doll* and *ball* are highly activated once the final noun is processed, while the activations of the other nouns fall off.

Figure 27 also illustrates the how InSomNet interprets sentences nonmonotonically, as described in Section 2.1 with respect to the phenomena of *semantic flipping*. In Figure 27 (a), the noun *doll* is initially strongly activated, then its activation decreases when *girl* is read in, and again increased when the final word *doll* is processed. This ability to revise an interpretation on-line and in parallel with other interpretations avoids the back-tracking required in monotonic interpretation.

6.3 Cognitive Validity Conclusions

The results in this section show that InSomNet retains the hallmark cognitively plausible behavior of connectionist models. It is able to maintain multiple coactivated word senses as well as sentence interpretations. The model also automatically develops expectations about possible sentence continuations, based on frequency, and primes semantically licensed words. Moreover, InSomNet exhibits the interesting behavior of nonmonotonic revision of its interpretation as it processes a sentence. All these behaviors have been demonstrated in human sentence processing, and they constitute an important validation of the InSomNet model.

7 Discussion and Future Work

In this article, a subsymbolic sentence processing model, InSomNet, was presented that is able to parse a corpus of real-world sentences into richly annotated Mrs representations. InSomNet retains those qualities that have made subsymbolic models indispensable in psycholinguistics research: It is robust to input errors and network damage, coactivates multiple interpretations, expects and primes future inputs, parses incrementally, and revises its interpretation nonmonotonically.

InSomNet breaks new ground in the connectionist modeling field because it is the first model to scale up to processing a real-world corpus of sentences of realistic complexity that have not been generated by a predefined grammar nor been tailored to the network with *a priori* suppositions about how predicates are to be represented given the constraints of subsymbolic models (e.g., as trees encoded with Raam). Rather, the representations hew very closely to the theory-independent representations of Mrs, and InSomNet's architecture was designed to accommodate the resulting directed acyclic graphs. It not only produces *explicit* semantic representations, but is remarkably flexible in how the semantic graph nodes and arcs are allocated to the nodes in the Frame Map.

Furthermore, InSomNet instantiates the notion of parsing as a *graded* process: That is, in the most general case, there simply is no one ideal semantic interpretation of a sentence that can be ranked above all others. The nodes in the semantic dependency graph are activated in a graded manner depending on how likely they are to be a part of the sentence interpretation in the context of the network's training experience. It is particularly this innovative aspect of the model that underlies its cognitively plausible behavior as reflected in the psycholinguistic literature that shows that people also seem to activate parts of an interpretation in a graded manner according to their language experience. This approach stands in sharp contrast to symbolic and probabilistic parsers in which sentence constituents are either part of a parse representation or not, and the parse representation as a whole is ranked according to its likelihood. Trained and evaluated on the Mrs representations of approximately 5000 sentences from the Redwoods Treebank, InSomNet was able to represent their complex semantics and generalize well to novel sentences.

A crucial innovation of the InSomNet model was the use of a self-organizing map to represent directed acyclic graphs as flat semantic-dependency structures. In this manner, it avoided the memory problems associated with repeated compression of nested structures, as is typical in previous Raam-based connectionist sentence-processing systems. Accordingly, the system does not have to commit to one structure that would be difficult to revise with further sentence context. The self-organizing process captured regularities in the semantic encodings so that InSomNet could learn to decode compressed representations of those frames back into their components. Because Mrs handles were represented as predicate labels, it was possible to use them as context-sensitive pointers that fill the argument slots of frames. By representing the handles as compressed frame patterns, InSomNet was able to learn to associate arguments with the semantic features of their fillers.

This approach to semantic representation suggests an interesting interpretation of the nature of how linguistic meaning may be realized in human cognition. The traditional symbolic Predicate Logic approach posits abstract predicates that are specified at the lexical level, such as **love**(x_0,x_1), or at the level of semantic relations such as **transitive_verb**(v_0,x_0,x_1). The predicate arguments are instantiated from sets of role fillers. Because the variable bindings are arbitrary, this approach can be used to model human language competence. In contrast, the representations in InSomNet suggests stronger constraints on the predicates. For example, a given node in the Frame Map may represent a set of predicates that are semantically similar based on the arguments they tend to take. Moreover, these arguments are not completely arbitrary, but also tend to be semantically similar. Thus, both predicates and arguments for a given node are weighted according to how frequent they are in the training corpus, and how they correlate with each other. The context-sensitive pointers in the semantic representation in the Frame Map allows the network to represent a greater variety of sentences because arguments are not encoded with their predicates, but rather are represented in other nodes in the Frame Map. Because a given node can represent more than one predicate depending on the sentence input, InSomNet implements a subset of second-order Predicate Logic, where variables (Frame Nodes) can range over predicates as well as arguments.

To verify that InSomNet is a robust system, the model was evaluated on the original sentences from the VerbMobil project that had been annotated with dialogue transcriptions. These sentences included a variety of speech errors, such as pauses, repairs, dysfluencies, and hesitations. A network that had been trained only on the cleaned-up versions of these sentences performed nearly as well on these messy sentences as on the cleaned up versions. Such robust processing has long been a hallmark of connectionist systems. It emerges automatically from their representations, whereas symbolic and probabilistic systems are characteristically brittle under these same conditions.

Finally, in order to demonstrate the model's cognitive plausibility, InSomNet was trained and evaluated on the case-role assignment task using a variant of the McClelland and Kawamoto corpus. The results on the McClelland and Kawamoto corpus, however, showed that InSomNet develops an interpretation of a sentence incrementally, and revises it nonmonotonically during processing. The network was also able to coactivate ambiguous interpretations, and choose one interpretation or another based on disambiguating context. In the course of developing a semantic interpretation, InSomNet demonstrated expectations and defaults, as well as semantic priming in accordance with its training experience.

The current version of InSomNet does have certain limitations. As is typical of neural networks, training takes a long time, and the network's performance degrades as sentences become longer. Also, the number of nodes in the Frame Map sets an upper bound on the number of frames that can constistute an interpretation of a sentence. Interestingly, when sentences from the Redwoods corpus are input in reverse order, the system parses them nearly as well as when they are presented in their natural order. This behavior suggests that the network treats the sentences more like a bag of words rather than a sequence. However, analysis of the network did show that InSomNet was sensitive to the order of crucial words like *not* and determiners, and indeed even showed evidence of local coherence effects (Tabor, Galantucci, & Richardson, 2004; Konieczny, 2005). Furthermore, on the psych dataset, the model is highly sensitive to word order, because the training set covered most of the sentence possibilities. Thus, the bag of words behavior is an artifact of the sparsity of data of the corpus that is a significant source of errors for any statistical model. Another interesting aspect of InSomNet is that compressed representations of frames may include "dangling pointers", i.e., the argument filler of a role may point to a weakly activated frame node that is not in the target set representing the preferred interpretation of a sentence. A side-effect of the self-organization process, however, is that some of these non-targeted frame nodes sometimes are decoded into the same components of a neighboring node which is in the target set. The results reported in Section 4 did not count such frames as correct because they strictly did not belong to the target set, on which precision and recall were calculated. In addition, many of these weakly activated nodes corresponded to semantically primed frames based on the network's still rather limited training experience of roughly 5000 sentences.

InSomNet also suggests a number of directions that could be taken up in future research. Instead of random representations for lexical items, feature-based phonological or semantic representations could be used, or they could be induced automatically with FGReP (Miikkulainen, 1993). Such representations should facilitate InSomNet's ability to handle out-of-vocabulary items, and thus alleviate some of the problems with sparse data.

In addition, the approach taken with respect to InSomNet's architecture and semantic representation provides a useful starting point for processing more varied linguistic structures, such as multiple sentences in a discourse, question answering, and the incorporation of information from multiple sources, such as a visual environment in which sentences may be situated: e.g., extending the system to model covert attention to objects and depicted actions in a co-present scene (Mayberry & Crocker, in press). The Mrs representations could be extended with back links (i.e., two-way pointers), making it possible to traverse the DAG semantic representation directly, and thereby recover the complete sentence interpretation without having to poll each Frame Node to determine if it is above threshold and therefore a part of the interpretation. Furthermore, more research is required to improve the ability of the network to have nominal and verbal frames keep track of their constituent feature frames, which are easily confused because many are identical or very nearly so. Such an approach might include an ordering component in the feature frames to indicate their natural order in the input sentence.

A demo of InSomNet, along with supplementary material about Minimal Recursion Semantics, the English Resource Grammar, and the VerbMobil dataset used in the article is available at `http://nn.cs.utexas.edu/?INSOMNet`.

8 Conclusion

InSomNet is a subsymbolic sentence processing system that produces explicit and graded semantic graph representations. The novel technique of semantic self-organization allows the network to learn typical semantic dependencies between nodes in a graph that helps the InSomNet process novel sentences. The technique makes it possible to assign case roles flexibly, while retaining the cognitively plausible behavior that characterizes connectionist modeling. InSomNet was shown to scale up to sentences of realistic complexity, including those with dysfluencies in the input and damage in the network. The network also exhibits the crucial cognitive properties of incremental processing, expectations, semantic priming, and nonmonotonic revision of an interpretation during sentence processing. InSomNet therefore constitutes a significant step towards building a cognitive parser that work with the everyday language that people use.

Acknowledgements

Profound appreciation is extended to Drs. Matthew Crocker, Igor Farkas, Pia Knoeferle, John Hale, Shravan Vasishth, Lars Konieczny, and Christopher Kello for excellent suggestions and constructive critiques during the course of writing this article.

References

Alexandersson, J., Reithinger, N., & Maier, E. (1997). *Insights into the dialogue processing of VerbMobil* (Tech. Rep. No. 191). Saarbrücken, Germany: DFKI.

Altmann, G. T. M. (1998). Ambiguity in sentence processing. *Trends in Cognitive Science*, *2*(4), 146–152.

Altmann, G. T. M., & Kamide, Y. (1999). Incremental interpretation at verbs: Restricting the domain of subsequent reference. *Cognition*, *73*, 247–264.

Barton, G. E., Berwick, R. C., & Ristad, E. S. (1987). *Computational complexity and natural language*. Cambridge, MA: MIT Press.

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, *5*(2), 157–166.

Berg, G. (1992). A connectionist parser with recursive sentence structure and lexical disambiguation. In W. Swartout (Ed.), *Proc. of the Tenth National Conference on Artificial Intelligence* (pp. 32–37). Cambridge, MA: MIT Press.

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

Chrisman, L. (1991). Learning recursive distributed representations for holistic computation. *Connection Science*, *3*, 345–366.

Church, K., & Patil, R. (1982). Coping with syntactic ambiguity or how to put the block in the box on the table. *American Journal of Computational Linguistics*, *8*(3–4), 139-149.

Collins, M. J. (1999). *Head-driven statistical models for natural language parsing.* Unpublished doctoral dissertation, University of Pennsylvania.

Copestake, A., & Flickinger, D. (2000). An open source grammar development environment and broad-coverage English grammar using HPSG. In *Proc. of the 2nd International Conference on Language Resources and Evaluation* (p. 591-600). Athens, Greece.

Copestake, A., Flickinger, D., Sag, I. A., & Pollard, C. (2005). Minimal recursion semantics: An introduction. *Research on Language and Computation*, *3(2–3)*, 281–332.

Crocker, M. W. (2005). Rational models of comprehension: Addressing the performance paradox. In A. Cutler (Ed.), *Twenty-first century psycholinguistics: Four cornerstones* (pp. 363–380). Hillsdale, NJ: Erlbaum.

Crocker, M. W., & Keller, F. (2006). Probabilistic grammars as models of gradience in language processing. In G. Fanselow, C. Féry, R. Vogel, & M. Schlesewsky (Eds.), *Gradience in grammar: Generative perspectives* (pp. 227–245). Oxford, UK; New York: Oxford University Press.

Dahan, D., & Gaskell, M. G. (2007). The temporal dynamics of ambiguity resolution: Evidence from spoken-word recognition. *Journal of Memory and Language*, *57*(4), 483–501.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*, 179–211.

Elman, J. L. (1991). Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning*, *7*, 195–225.

Federmeier, K. D. (2007). Thinking ahead: The role and roots of prediction in language comprehension. *Psychophysiology*, *44*, 491–505.

Fillmore, C. J. (1968). The case for case. In E. Bach & R. T. Harms (Eds.), *Universals in linguistic theory* (pp. 0–88). New York: Holt, Rinehart and Winston.

Flickinger, D. (2000). On building a more efficient grammar by exploiting types. *Natural Language*

Engineering, 6(1) (Special Issue on Efficient Processing with HPSG), 15–28.
Fodor, J. A. (1983). *Modularity of mind*. Cambridge, MA: MIT Press.
Garrett, M. F. (1990). Sentence processing. In D. N. Osherson & H. Lasnik (Eds.), *Language (vol.1): an invitation to cognitive science* (pp. 133–175). Cambridge, MA: MIT Press.
Henderson, J. (1994). *Description based parsing in a connectionist network*. Unpublished doctoral dissertation, University of Pennsylvania, Philadelphia, PA. (Technical Report MS-CIS-94-46)
Hindle, D., & Rooth, M. (1993). Structural ambiguity and lexical relations. *Computational Linguistics*, *19*, 103–120.
Hinton, G. E. (1981). Implementing semantic networks in parallel hardware. In G. E. Hinton & J. A. Anderson (Eds.), *Parallel models of associative memory* (pp. 161–187). Hillsdale, NJ: Erlbaum.
Ho, K. S. E., & Chan, L. W. (1997). Confluent preorder parsing of deterministic grammars. *Connection Science*, *9*, 269–293.
Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.
Hudson, S. B., & Tanenhaus, M. K. (1984). Ambiguity resolution in the absence of contextual bias. In *Proc. of the Sixth Annual Conference of the Cognitive Science Society* (pp. 188–192). Hillsdale, NJ: Erlbaum.
Jain, A. N. (1991). *PARSEC: A connectionist learning architecture for parsing spoken language*. Unpublished doctoral dissertation, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA. (Technical Report CMU-CS-91-208)
James, D. L., & Miikkulainen, R. (1995). SARDNET: A self-organizing feature map for sequences. In G. Tesauro, D. S. Touretzky, & T. K. Leen (Eds.), *Advances in Neural Information Processing Systems 7* (pp. 577–584). Cambridge, MA: MIT Press.
John, M. F. S., & McClelland, J. L. (1990). Learning and applying contextual constraints in sentence comprehension. *Artificial Intelligence*, *46*, 217–258.
Jurafsky, D. (1996). A probabilistic model of lexical and syntactic access and disambiguation. *Cognitive Science*, *20*, 137–194.
Kohonen, T. (1990). The self-organizing map. *Proc. of the IEEE*, *78*, 1464–1480.
Kohonen, T. (1995). *Self-organizing maps*. Berlin; New York: Springer.
Konieczny, L. (2005). The psychological reality of local coherences in sentence processing. In *Proc. of the 27th Annual Conference of the Cognitive Science Society*. Mahawah, NJ: Erlbaum.
Lane, P. C. R., & Henderson, J. B. (2001). Incremental syntactic parsing of natural language corpora with simple synchrony networks. *IEEE Transactions on Knowledge and Data Engineering*, *13*(2), 219–231.
Lawrence, S., Giles, C. L., & Fong, S. (2000). Natural language grammatical inference with recurrent neural networks. *IEEE Transactions on Knowledge and Data Engineering*, *12*(1), 126–140.
MacDonald, M. C. (1993). The interaction of lexical and syntactic ambiguity. *Journal of Memory and Language*, *32*, 692–715.
MacDonald, M. C., Just, M. A., & Carpenter, P. A. (1992). Working memory constraints on the processing of syntactic ambiguity. *Cognitive Psychology*, *24*, 56–98.
Marcus, M. P., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, *19*, 313–330.

Marslen-Wilson, W. D. (1975). Sentence perception as an interactive parallel process. *Science*, *189*, 226–228.

Marslen-Wilson, W. D. (Ed.). (1989). *Lexical representation and process.* Cambridge, MA: MIT Press.

Mayberry, M. R., & Crocker, M. W. (in press). Learning to attend: A connectionist model of situated language comprehension. *Cognitive Science*.

Mayberry, M. R., & Miikkulainen, R. (1994). Lexical disambiguation based on distributed representations of context frequency. In *Proc. of the 16th Annual Conference of the Cognitive Science Society.* Hillsdale, NJ: Erlbaum.

Mayberry, M. R., & Miikkulainen, R. (1999a). SARDSRN: A neural network shift-reduce parser. In *Proc. of the 16th International Joint conference on Artificial Intelligence* (pp. 820–825). San Francisco, CA: Morgan Kaufmann.

Mayberry, M. R., & Miikkulainen, R. (1999b). Using a sequential SOM to parse long-term dependencies. In *Proc. of the 21st Annual Conference of the Cognitive Science Society* (pp. 367–372). Hillsdale, NJ: Erlbaum.

Mayberry, M. R., & Miikkulainen, R. (2000). Combining maps and distributed representations for shift-reduce parsing. In S. Wermter & R. Sun (Eds.), *Hybrid neural systems* (pp. 144–157). Berlin; New York: Springer-Verlag.

Mayberry, M. R., & Miikkulainen, R. (2005). Broad-coverage parsing with neural networks. *Neural Processing Letters*, *21*, 121–132.

McClelland, J. L., & Kawamoto, A. H. (1986). Mechanisms of sentence processing: Assigning roles to constituents. In J. L. McClelland & D. E. Rumelhart (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition, Volume 2: Psychological and biological models* (pp. 272–325). Cambridge, MA: MIT Press.

McRae, K., Spivey-Knowlton, M. J., & Tanenhaus, M. K. (1998). Modeling the influence of thematic fit (and other constraints) in on-line sentence comprehension. *Journal of Memory and Language*, *38*(3), 283–312.

Miikkulainen, R. (1993). *Subsymbolic natural language processing: An integrated model of scripts, lexicon, and memory.* Cambridge, MA: MIT Press.

Miikkulainen, R. (1996). Subsymbolic case-role analysis of sentences with embedded clauses. *Cognitive Science*, *20*, 47–73.

Miikkulainen, R. (1997). Natural language processing with subsymbolic neural networks. In A. Browne (Ed.), *Neural network perspectives on cognition and adaptive robotics* (pp. 120–139). Bristol, UK; Philadelphia, PA: Institute of Physics Publishing.

Miikkulainen, R., & Dyer, M. G. (1991). Natural language processing with modular PDP networks and distributed lexicon. *Cognitive Science*, *15*, 343–399.

Mitchell, D. C., Cuetos, F., Corley, M., & Brysbaert, M. (1995). Exposure-based models of human parsing: Evidence for the use of coarse-grained (nonlexical) statistical records. *Journal of Psycholinguistic Research*, *24*, 469–488.

Mooney, R. (1999). Learning for semantic interpretation: Scaling up without dumbing down. In J. Cussens (Ed.), *Proc. of the 1st Workshop on Learning Language in Logic* (pp. 7–15). Morristown, NJ, USA: Association for Computational Linguistics.

Oepen, S., Flickinger, D., Toutanova, K., & Manning, C. (2002). LinGO redwoods: A rich and dynamic treebank for HPSG. In *Proc. of the First Workshop on Treebanks and Linguistic Theories (TLT02).* Sozopol, Bulgaria: TLT.

Onifer, W., & Swinney, D. A. (1981). Accessing lexical ambiguities during sentence comprehension: Effects of frequency of meaning and contextual bias. *Memory and Cognition*, *9*, 225–226.

Pollack, J. B. (1990). Recursive distributed representations. *Artificial Intelligence*, *46*, 77–105.

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

Reilly, R. (1992). *A connectionist technique for on-line parsing.* (Neuroprose)

Rohde, D. L. T. (2002). *A connectionist model of sentence comprehension and production.* Unpublished doctoral dissertation, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA. (Technical Report CMU-CS-02-105)

Rohde, D. L. T., & Plaut, D. C. (2003). Connectionist models of language processing. *Cognitive Studies*, *10*(1), 10–28.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition, Volume 1: Foundations* (pp. 318–362). Cambridge, MA: MIT Press.

Sells, P. (1985). *Lectures on contemporary syntactic theories.* Stanford, CA: Center for the Study of Language and Information.

Sharkey, N. E., & Sharkey, A. J. C. (1992). A modular design for connectionist parsing. In M. F. J. Drossaers & A. Nijholt (Eds.), *Twente Workshop on Language Technology 3: Connectionism and Natural Language Processing* (pp. 87–96). Enschede, the Netherlands: Department of Computer Science, University of Twente.

Shastri, L., & Ajjanagadde, V. (1993). From simple associations to systematic reasoning: A connectionist representation of rules, variables and dynamic bindings using temporal synchrony. *Behavioral and Brain Sciences*, *16*, 417–494.

Small, S. L., Cottrell, G. W., & Tanenhaus, M. K. (Eds.). (1988). *Lexical ambiguity resolution: Perspectives from psycholinguistics, neuropsychology and artificial intelligence.* San Francisco, CA: Morgan Kaufmann.

Spark-Jones, K., & Willett, P. (Eds.). (1997). *Readings in information retrieval.* San Francisco, CA: Morgan Kaufmann.

St. John, M. F., & McClelland, J. L. (1988). Learning and applying contextual constraints in sentence comprehension. In *Proc. of the 10th Annual Conference of the Cognitive Science Society.* Hillsdale, NJ: Erlbaum.

Tabor, W., Galantucci, B., & Richardson, D. (2004, May). Effects of merely local syntactic coherence on sentence processing. *Journal of Memory and Language*, *50*(4), 355–370.

Toutanova, K., Manning, C. D., Shieber, S. M., Flickinger, D., & Oepen, S. (2002). Parse disambiguation for a rich HPSG grammar. In *First Workshop on Treebanks and Linguistic Theories (TLT2002)* (pp. 253–263). Sozopol, Bulgaria.

Wahlster, W. (Ed.). (2000). *Verbmobil. Foundations of speech-to-speech translation.* Berlin; New York: Springer-Verlag.

Williams, R. J., & Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, *1*, 270–280.

Appendix

This appendix provides a description of the components of the Mrs representations used in the article. A more detailed account can be found in Copestake et al. (2005); Copestake and Flickinger (2000), and Flickinger (2000).

Recall that semantic frames are represented in the form:

| **Handle Word Semantic-Relation Subcategorization <Arguments>** |.

As an example, the node labeled **h1:*hit*** in the middle of the Mrs graph in Figure 8 was represented by the frame | **h1 *hit* _arg13_rel A0A1A3DMEV _ x0 x1 _ e0 _** | with the following frame field fillers:

Handle:	**h1**
Word:	**hit**
Semantic-Relation:	**_arg13_rel**
Subcategorization-Type:	**A0A1A3DMEV**
<Arguments>:	**_ x0 x1 _ e0 _**

Abbreviation	Mrs *Subcategorization*	*Abbreviation*	Mrs *Subcategorization*
A0	ARG	A0CVDM	ARG CONST_VALUE DIM
A0A1A2A3DMEV	ARG ARG1 ARG2 ARG3 DIM EVENT	A0DM	ARG DIM
A0A1A2DMEV	ARG ARG1 ARG2 DIM EVENT	A0DMEV	ARG ARG3 DIM EVENT
A0A1DMEV	ARG ARG1 DIM EVENT	A0DMEVRL	ARG DIM EVENT ROLE
A0A1A3DMEV	ARG ARG1 ARG3 DIM EVENT	A0DMT1T2	ARG DIM TERM1 TERM2
A0A1A4DMEV	ARG ARG1 ARG4 DIM EVENT	A0EX	ARG EXCL
A0A2A4DMEV	ARG ARG2 ARG4 DIM EVENT	A4IX	ARG4 INST
A0A1A3A4DMEV	ARG ARG1 ARG3 ARG4 DIM EVENT	CALHLIRHRI	C-ARG L-HANDEL L-INDEX R-HANDEL R-INDEX
A0A3DGDM	ARG ARG3 DARG DIM	APDMHRIXMI	AM-PM DIM HOUR INST MIN
A0A3DMEV	ARG ARG3 DIM EVENT	BVDMRESC	BV DIM RESTR SCOPE
A0A3DMEVIX	ARG ARG3 DIM EVENT INST	DGDM	DARG DIM
A0A4DMEV	ARG ARG4 DIM EVENT	DMIX	DIM INST
A0DMF1F2	ARG DIM FACTOR1 FACTOR2	DV	DIVISIBLE
DMHDIXMI	DIM HOUR-IND INST MINUTE-IND	EVPRPS	EVENT PROP PRPSTN
DVASMOTN	DIVISIBLE ASPECT MOOD TENSE	HXNX	HINST NHINST
DVGNPNPT	DIVISIBLE GEN PN PRONTYPE	IX	INST
G1G2CXDGDM	ARG-1 ARG-2 C-INST DARG DIM	IXND	INST NAMED
A3IXNDYI	ARG3 INST NAMED YEAR-IND	IXTL	INST TITLE
A3IXND	ARG3 INST NAMED	MNSB	MAIN SUBORD
A3DMIX	ARG3 DIM INST	A3IX	ARG3 INST
A3IXSN	ARG3 INST SEASON	SA	SOA

Table A1: **Subcategorization types.**

Table A1 lists the Mrs Subcategorizations and their abbreviations used in this article. For example, **A0A1A3DMEV** denotes the Mrs Subcategorization for a transitive verb: [ARG ARG1 ARG3 DIM EVENT]. Accordingly, the ARG1 slot is filled with **x0**, the ARG3 slot, with **x1**, and the EVENT slot, with **e0**. The ARG and DIM slots are left blank. The three instantiated roles, ARG1, ARG3, and EVENT denote an external argument, a patient/theme argument, and a basic predicative index, respectively, as shown in Table A2 that lists the semantic roles, their abbreviations, and linguistic meaning.

Abbr	*Semantic Role*	*Meaning*	*Abbr*	*Semantic Role*	*Meaning*
A0	ARG	predicative argument	IX	INST	basic nominal relation index
A1	ARG	external argument	LH	L-HANDEL	discourse left branch handle
A2	ARG2	raising argument	LI	L-INDEX	discourse left branch index
A3	ARG3	patient argument	MD	MINUTE-IND	minute index
A4	ARG4	propositional argument	MI	MIN	minute
AP	AM-PM	a.m./p.m.	MN	MAIN	main clause
BV	BV	bound variable	ND	NAMED	named entity
CA	C-ARG	discource argument	NX	NHINST	non-head of phrasal relation
CV	CONST_VALUE	constant value	PR	PROP	property
CX	C-INST	comparative index	PS	PRPSTN	proposition
DG	DARG	degree argument	RE	RESTR	restriction
DM	DIM	dimension	RH	R-HANDEL	discourse right branch handle
DV	DIVISIBLE	divisible	RI	R-INDEX	discourse right branch index
EV	EVENT	basic predicative index	RL	ROLE	role for ellipsis construction
EX	EXCL	exclamation	SA	SOA	state of affairs
F1	FACTOR1	first addend	SB	SUBORD	subordinate clause
F2	FACTOR2	second addend	SC	SCOPE	scope
G1	ARG-1	1st comparative argument	SN	SEASON	season
G2	ARG-2	2nd comparative argument	T1	TERM1	first multiplicand
HA	HANDEL	predicate label	T2	TERM2	second multiplicand
HI	HOUR-IND	hour index	TL	TITLE	title
HR	HOUR	hour	YI	YEAR-IND	year index
HX	HINST	head of phrasal relation			

Table A2: **Semantic Roles and Meanings.**

Feature Frames. As described in Section 4.1, the leaf nodes of the MRS dependency graphs constitute an important subset of the semantic frames called feature frames because they describe the nouns or verbs whose **IX** or **EV** argument slots they fill. For example, the verbal features for the *hit* frame above is given in the feature frame

$$|\ \ \mathbf{e0}\ \ \boldsymbol{\textit{-d}}\ \ \mathbf{EVT}\ \ \mathbf{DVASMOTN}\ \ \mathbf{BOOL}\ \ \mathbf{\text{-}ASP}\ \ \mathbf{IND}\ \ \mathbf{PAST}\ \ \_\ \_\ |.$$

As above, the frame has the following correspondences:

Handle:	**e0**
Word:	**-d**
Semantic-Relation:	**EVT**
Subcategorization:	**DVASMOTN**
<Arguments>:	**BOOL -ASP IND PAST _ _**

Abbreviation	MRS *Semantic Feature Type*
FRI	full referential index
RI	referential index
DI	degree index
CI	conjunctive index
CRI	conjunctive referential index
CFRI	conjunctive full referential index
E|I	event or referential index
EVT	event feature structure
CE	conjunctive event feature structure

Table A3: **Semantic Features Types.**

The **DVASMOTN** subcategorization from Table A1 is an abbreviaion for the MRS event roles, [DIVISIBLE ASPECT MOOD TENSE] that characterize the **Semantic-Relation EVT** (listed in Table A3 along with the rest of the semantic feature types that occurred in the MRS Dataset used to train and test InSomNet). In Mrs representations, there are three main categories of features: general, nominal and verbal (Tables A4, A5, and A6). In this example, the **DVASMOTN** subcategorization gives the semantic features for the verb *hit* via the **e0** filler of the EVENT (EV) slot. The event roles corresponding to the **Semantic-Relation EVT** were instantiated by the following semantic features:

DIVISIBLE	**BOOL**
ASPECT	**-ASP**
MOOD	**IND**
TENSE	**PAST**

The first field, DIVISIBLE (DV) is a general feature that leaves its Boolean value underspecified as BOOL (i.e., neither + nor -). The other three fields are verbal, and in Table A6 indicate that the verb *hit* is non-aspectual (-ASP), indicative (IND), and in the past tense (PAST).

Abbreviation	Mrs *Semantic Feature*	*Meaning*
G918	GLBTYPE918	tag question
BOOL	BOOL	unspecified Boolean
!BOOL	STRICT_BOOL	strictly Boolean
+	+	positive Boolean
-	-	negative Boolean
+&-	+_AND_-	both positive and negative Boolean

Table A4: **General Semantic Features.**

Abbreviation	Mrs *Semantic Feature*	*Meaning*
GEN	GENDER	unspecified gender
MASC	MASC	masculine gender
FEM	FEM	feminine gender
NEU	NEUT	neuter gender
ANDR	ANDRO	grammatical masculine gender
PERNUM	PERNUM	unspecified person and number
2PER	2PER	second person number
!2PER	STRICT_2PER	strictly second person
1SG	1SG	first person singular
2SG	2SG	second person singular
3SG	3SG	third person singular
1PL	1PL	first person plural
3PL	3PL	third person plural
-1SG	NON1SG	not first singular
!-1SG	STRICT_NON13SG	strictly non-singular
3PLSG	3PL_AND_3SG	third person plural and singular
!-3SG	STRICT_NON3SG	strictly non-third singular
PRN	PRONTYPE	unspecified pronoun type
STDPN	STD_PRON	standard pronoun
STD1SG	STD_1SG	stardard first singular
STD1PL	STD_1PL	standard first plural
STD2	STD_2	standard second person
STD3	STD_3	standard third person
0PN	ZERO-PRON	null pronoun
RECP	RECIP	reciprocal pronoun
REFL	REFL	reflexive pronoun

Table A5: **Nominal Features**

Abbreviation	MRS *Semantic Feature*	*Meaning*
ASP	ASPECT	generic aspect
-ASP	NO_ASPECT	non-aspectual
-ASP+PRG	NOASP+PROGR	non-aspectual and progressive
!-PRF	STRICT_NONPRF	strictly non-perfective
-PRF	NONPRF	non-perfective aspect
-PRG+-PRF	NONPRG+NONPRF	non-progressive and non-perfective
PRF	PERF	perfective aspect
PRG+PRF	PROGR+PERF	progressive and perfective
MOOD	MOOD	generic mood
IND	INDICATIVE	indicative mood
SBJ	SUBJ	subjunctive mood
MODSBJ	MOD_SUBJ	modal subjunctive
IND+MODSBJ	IND+MODSUBJ	indicative and modal-subjunctive mood
!IND|MODSBJ	STRICT_iND_oR_MOD_SUBJ	strictly indicative or modal subjunctive mood
IND|MODSBJ	IND_oR_MOD_SUBJ	indicative or modal subjunctive mood
TNS	TENSE	generic tense
-TNS	NO_TENSE	non-tensed
PRES	PRESENT	present tense
PAST	PAST	simple past tense
FUT	FUTURE	future tense
PRG	PROGR	progressive tense
PRES+PAST	PRES+PAST	present and past tense
PRES+FUT	PRES+FUT	present and future tense

Table A6: **Verbal Features.**

The presentation of these features here are somewhat simplified; the MRS features themselves actually derive from the HPSG type hierarchy, and, accordingly, should strictly be presented as a graph of feature derivations (e.g., PRES+PAST combines tenses PRES and PAST in Table A6).