Neural Architectures for Learning to Answer Questions

Derek Monner[a,*], James A. Reggia[a,b]

[a] *Department of Computer Science, University of Maryland, College Park, MD 20742, USA*
[b] *Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742, USA*

Abstract

Two related and integral parts of linguistic competence are the ability to comprehend incoming utterances and the ability to respond to them appropriately. In this context, we present two biologically inspired recurrent neural network models, based on the long short-term memory (LSTM) architecture, each of which develops a functional grasp of a small language by participating in a question-answering game. Both models receive questions in English, presented as temporal streams of speech sound patterns. As a further input, the models receive a set of symbolic facts about a simulated visual environment. The models learn to correctly answer input questions by observing question-answer pairs produced by other participants. The first of our two models produces its answers as symbolic facts, demonstrating an ability to ground language. The second model learns by observation to produce its answers as full English sentences. This latter task in particular is closely analogous to that faced by human language learners, involving segmentation of morphemes, words, and phrases from observable auditory input, mapping of speech signals onto intended environmental referents, comprehension of questions, and content-addressed search capabilities for discovering the answers to these questions. Analysis of the models shows that their performance depends upon highly systematic learned representations that combine the best properties of distributed and symbolic representations.

Keywords: question answering, language comprehension, speech production, recurrent neural network, long short term memory

Introduction

In human discourse, it is often the case that the speech acts of the participants alternate between requests and responses, with one party seeking information and the other providing it, followed by one of the parties expressing another request, and so on. There is a considerable literature on how one could

*Corresponding author. Telephone: +1 (301) 586-2495. Fax: (301) 405-6707.
Email addresses: dmonner@cs.umd.edu (Derek Monner), reggia@cs.umd.edu (James A. Reggia)

construct a program capable of participating in the request/response paradigm (e.g., Diederich & Long, 1991; Graesser & Franklin, 1990; Miikkulainen, 1993, 1998; Williams & Miikkulainen, 2006; St. John & McClelland, 1990; Rohde, 2002). One of the most recent and well-known examples of such a question-answering system is IBM's Watson (Ferrucci et al., 2010), which can often best experienced players in the quiz show *Jeopardy!*. While Watson and systems like it are undoubtedly impressive, their errors are often baffling and inscrutable to onlookers, suggesting that the strategies they employ differ greatly from those that humans use. While question-answering systems have been well-studied in natural language processing domains, little research has been done on how the question/answer style of interaction might influence the ways in which humans acquire language. In human language modeling, much interest has been paid to the study of language comprehension (the transformation of an auditory signal into a meaning; see, e.g., Markert et al., 2009) and of language production (the inverse problem of transforming a meaning into speech sounds; see, e.g., Dell, 1993), or both simultaneously (e.g., Plaut & Kello, 1999; Chang et al., 2006), but there is little human language modeling research that focuses specifically on learning to produce appropriate responses to questions. This is an interesting subject in light of the fact that, when listening to language, learners are constantly confronted with these request/response, question/answer pairs. Particularly interesting is the question of how the language faculties of a learner in this situation could be implemented solely by a complex neural network like the human brain.

Here we investigate the extent to which a biologically inspired neural network model of a human learner can develop a grasp of a small language by listening to simple question/answer pairs. The model is situated in a simulated visual environment along with two other speakers, referred to as Watson and Sherlock. Watson asks simple questions about the shared environment in a subset of English, and Sherlock responds to these questions with the information Watson seeks. The model's task is to learn to emulate Sherlock. To do this effectively, the model must listen to the speech sounds of Watson's questions and learn to segment them into morphemes, words, and phrases, and then interpret these components with respect to the common surroundings, thereby grounding them in visual experience. The model must then recognize what information Watson is asking for and provide that information in a form that aligns with an answer that Sherlock would give.

We examine two related models that differ in how the answers are provided. The first model learns to provide the answer to a question from Watson as a raw meaning—a series of predicates describing properties of, and relations between, objects in the environment. As such, it is called the **meaning-answer model**. The answers this model provides are meant to be analogous to the learner's internal representations of meaning, though the representational form the model uses is determined *a priori* instead of learned. Teaching a model to answer this way is useful because it demonstrates explicitly that such a model can ground its answers by referring to concrete objects in the environment, rather than simply rearranging the question and guessing a plausible answer.

However, for this to be a reasonable model of human language learning, it would need to learn entirely based on data that are readily available to any language learner. Thus, the model would need to have direct access to Sherlock's internal meaning representations, which is, of course, not generally possible in human language learning situations (though there is some evidence that the listener may often be able to infer meanings from context; see Tomasello, 2003). However, examining this limited model can still be useful, as its predicate-based outputs provide direct evidence that neural models can learn to produce a fully grounded representation of an answer to a question.

A second model addresses this limitation of the meaning-answer model by providing its answers much like the input questions—as sequences of speech sounds. This second model is termed the **spoken-answer model**. The representation of an answer in this case is unambiguously observable whenever Sherlock responds to Watson, placing the spoken-answer model a step closer to the reality of human language learning. The problem of learning to answer questions is more difficult in this case, since the network is trained using only the observable linguistic stimuli produced by Sherlock, which are not only more verbose than Sherlock's intended meanings but also potentially lossy translations thereof. Nonetheless, analysis of this model provides evidence that this approach to training offers a tractable path to both language comprehension and production.

Methods

Both the meaning-answer model and the spoken-answer model have the same general structure, shown in Figure 1. The remainder of this section explores in detail the tasks performed by the models and the representations for their input and output signals before comparing and contrasting the neural network architectures of each.

Tasks

As mentioned in the introduction, the models are faced with the task of learning to comprehend and produce language in the context of answering questions. The overview of the task is simple: One of the models is placed in a shared visual environment with two other speakers, Watson and Sherlock. A training trial begins with Watson asking a question, the answer to which is directly observable in the environment. Sherlock surveys his surroundings and gives the answer, which the model observes and attempts to mimic. A testing trial, in contrast, is one in which Watson asks a similar question, but Sherlock is silent, relying on the model to produce the desired answer.

During a trial, the model first takes stock of the environment, receiving a stream of visual input and compressing it into a sort of visual gestalt representation for later reference. It then listens for Watson's question as a temporal sequence of speech sounds (phonemes), processing these into an auditory gestalt of sorts. After hearing the totality of the question, the model attempts to combine the visual and auditory gestalts, internally mapping the references in the

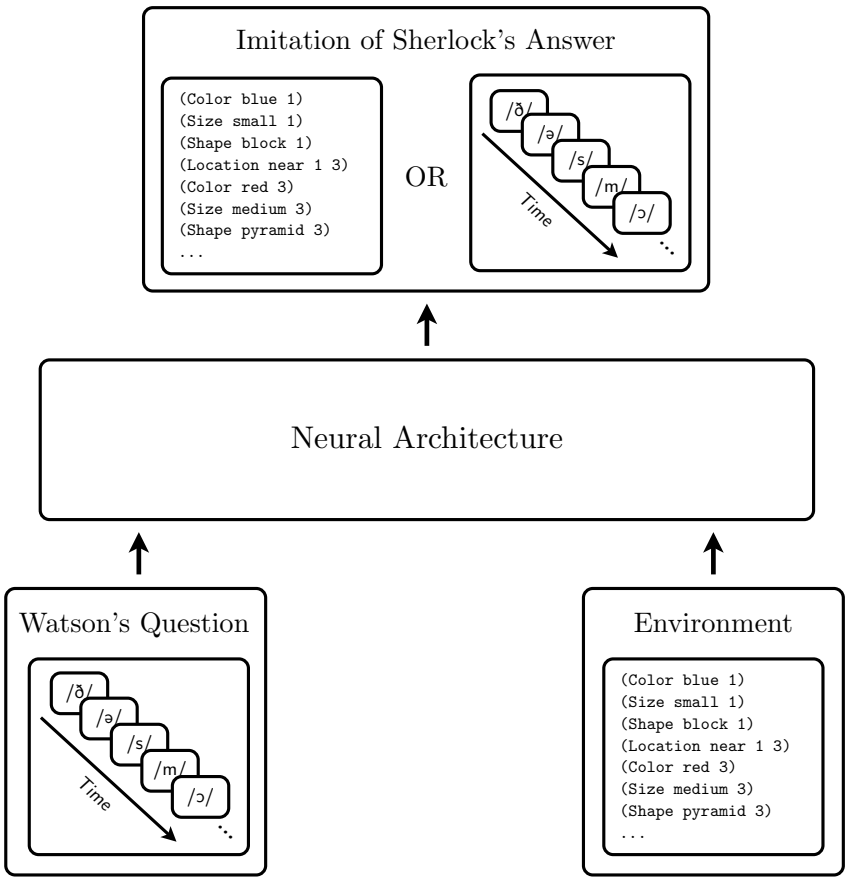

Figure 1: Information flow in the models. A model receives Watson's question as a sequence of phonemes in time, and information about the environment as a set of predicates, also presented temporally but in random order. The meaning-answer model would then attempt to produce an answer to the question as a set of predicates, while the spoken-answer model would answer in the form of a speech stream.

sentence to objects in the environment and subsequently figuring out what it is that Watson wants to know.

If the model in question is the meaning-answer model, it then provides the answer in the form of a sequence of grounded predicates that could serve as the meaning for a complete-sentence answer to Watson's question. The meaning-answer model must learn this behavior by imitating Sherlock, who always knows the answers to Watson's questions. In real human language learning, Sherlock's raw meaning would not necessarily be available for the model to imitate. The motivation for the meaning-answer model is to transparently demonstrate that a neural network model is capable of learning to produce this type of precise answer by grounding its auditory input in the visual scene.

In contrast to the meaning-answer model, the spoken-answer model answers questions by generating sequences of speech sounds (phonemes) that form a complete English sentence. The model again learns to produce these sounds by imitating Sherlock, and since these sounds are observable, this addresses the concern with the meaning-answer model and positions the spoken-answer model much closer to the reality of a human language learner. Because the model's internal meaning representations are learned, they are also much harder to interpret than the meaning-answer model's hand-designed predicates, and as such, it is more difficult to demonstrate robust grounding in the spoken-answer model than in the meaning-answer model. However, the spoken-answer model makes a different point: Mimicking observed speech, which is generally a lossy translation of the internal meaning of the speaker, is sufficient to learn question-answering behavior.

The following subsections examine in detail the ways in which the auditory inputs, visual inputs, and both predicate- and speech-based model outputs are constructed, and also describe the vocabularies and grammars to which Watson and Sherlock restrict themselves.

Environment input

The visual environment shared by the three participants consists of a number of objects placed in relation to each other; an example environment configuration is depicted in Figure 2. Each object has values for the three attributes (`Size`, `Shape`, and `Color`), and each attribute has three possible values: `small`, `medium`, and `large` for `Size`; `block`, `cylinder`, and `pyramid` for `Shape`; and `red`, `green`, and `blue` for `Color`. Thus, 27 distinct objects are possible. In addition, each object has a label that identifies it uniquely in the environment, which is a useful handle for a specific object and is necessary in the event that the participants need to distinguish between two otherwise identical objects.

Each object is represented as three predicates, each of which binds an attribute value to a unique object label. We use integers as the labels in the text, though the labels are arbitrary and the objects they represent are not ordered in any way. For example, a small red block with label 2 is completely described by the predicates `(Size small 2)`, `(Color red 2)`, and `(Shape block 2)`, which are presented to the models in a temporal sequence. An environment consists of two to four randomly generated objects, and the predicates describing all of

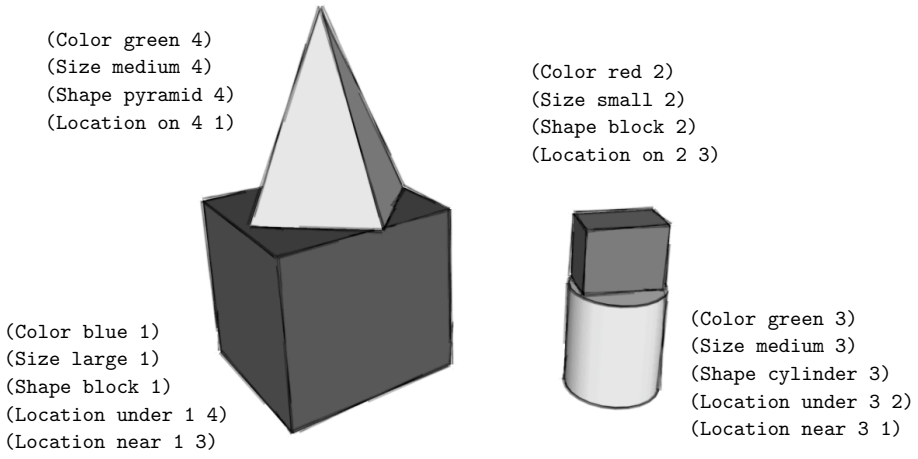

Figure 2: Example visual environment. This simulated environment contains four objects, along with the complete set of predicates that describe the environment. The predicates are the environmental input to the model; the visual depiction of the objects here is simply for reference.

these objects are presented at the visual input layer of the model as a randomized temporal sequence at the start of a trial. Each predicate is presented as a set of input unit activations. Input units are generally inactive, except for single active units corresponding to the type of attribute (e.g., `Color`), the value of that attribute (e.g., `blue`), and the label (e.g., `3`). See Figure 3 for a depiction of example predicates and their representations in terms of neural activity.

Additional predicates are used to describe spatial relations between the objects. One object may be near, on, or underneath another. For example, if the small red block (with label 2) is on top of a medium-sized green cylinder (label 3), that fact would be presented to the model as the predicate `(Location on 2 3)`. In the simulated environment, the `on` and `under` relations are complementary (meaning that `(Location on 2 3)` implies `(Location under 3 2)`), and the `near` relation is symmetric (such that `(Location near 1 3)` implies `(Location near 3 1)`). The location predicates are presented along with the attribute predicates and at the same visual input layer.

Though this space of possible environments may seem small at first, the number of unique environmental configurations is quite large. Using only two, three, or four objects at a time provides approximately 2.48×10^{10} distinct possible environment configurations.

The visual representation just described is a drastic simplification of real visual input that the models adopt for reasons of computational tractability. At the cost of moving away from sensory-level visual input, the models gain enough computational efficiency to study the language acquisition phenomenon of primary interest. This type of high-level visual representation can be viewed as the equivalent of a representation that might be produced by the later stages of the human visual system, though probably not in this precise form. Much

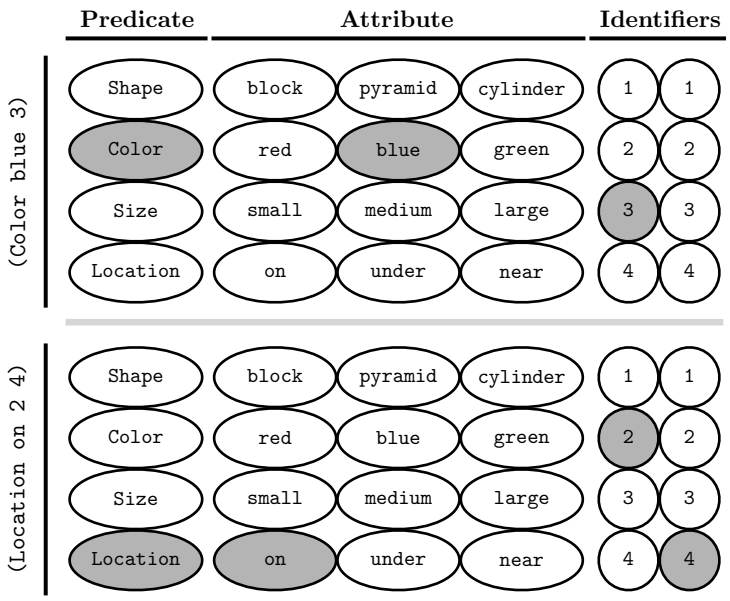

Figure 3: Neural representation of predicates. These two example predicates could be used as visual input or as meaning output. The left-most group of 4 cells in each case denotes the predicate type, of which only one cell will be active in a well-formed predicate. Cells in the middle group of 12 each stand for an attribute value; again, in well-formed predicates, only one of these is active at a time, and the attribute value will correspond to the correct predicate type. The right-most cells correspond to unique labels for environmental objects; in most cases, only a label from the first column is active, binding the active predicate-attribute pair to that label, as in the first example above representing `(Color blue 3)`. Location predicates, such as the second example `(Location on 2 4)`, activate two label nodes to establish a relationship between them.

like hypothesized human visual representations, this scheme requires the models to bind several types of information together to form a coherent representation of a seen object. The inclusion of additional predicates describing the spatial relations between the objects in the environment allows questions and answers to be conditioned on the locations of objects, so the models can distinguish, for example, two otherwise identical small red blocks by the fact that one of them rests on top of a cylinder and the other does not.

Question input

Watson produces complete English sentences that ask questions about the current shared environment. There are many possible questions for each environment; for the example environment in Figure 2, Watson might ask: *What color block is the green pyramid on?*, *What thing is under the small block?*, *What color are the medium things?*, or *Where is the pyramid?*.

At the start of each trial, Watson examines the environment and then begins deriving a question beginning at the **Question** nonterminal in the mildly context-sensitive grammar of Figure 4. The derivation is constrained not only by the grammar itself, but also by the environment. Specifically, any objects that Watson refers to must be present in the environment and, to a sophisticated observer, unambiguously identified by the full context of the question. For example, Watson could not ask *What color is the block?* because it is not clear which block is being referred to, and thus any answer would be poorly defined. Watson can, however, ask questions about groups of objects that share a property, such as *What color are the medium things?*; in this case, the medium things are the cylinder and pyramid, which are both green, so the answer is well defined. Questions posed to the models are required to have well-defined answers to maintain the simplicity of evaluation and training; after all, if the answer is ambiguous, how can one tell whether the model produced the correct answer or not? An important future research task will be to relax this requirement and see if one can train a model such that, when it is given an ambiguous question, it produces either an answer that is plausible, or an answer indicating its uncertainty.

The words in Watson's question are phonetically transcribed, and the resulting phoneme sequences are appended to produce one uninterrupted temporal sequence corresponding to the spoken sentence. Word and morpheme boundaries are not marked, leaving the model to discover those on its own, just as with real-world speech signals. When the speech sequence for a question is presented as input to the model, individual phonemes are given one at a time. Phonemes are represented using vectors of binary acoustic features based on observable phonetic features thought to play a role in human language recognition; a full listing of phonemes and their feature mappings can be found in Table 1.

Answer output

After Watson has finished asking a question, the model attempts a response. On training trials, Sherlock then gives the correct response, and the model

Question	→	where **Is Object** | **What Is Property**
Answer	→	**Object Is Property**
Object	→	the [**Size**] [**Color**] **Shape**
What	→	what color [**Size**] **Shape** | what size [**Color**] **Shape** | what [**Size**] [**Color**] thing
Property[a]	→	**Location** | **Color** | **Size**
Location[b]	→	on **Object** | under **Object** | near **Object**
Is[c]	→	is | are
Size	→	small | medium | large
Color	→	red | blue | green
Shape	→	things | pyramid[s] | block[s] | cylinder[s]

Figure 4: Grammar used with the meaning-answer model. This mildly context-sensitive grammar is used to train the meaning-answer model on the question-answering task. For simplicity, the grammar is shown as context-free, with the context-sensitivies indicated and explained by the footnotes. Terminals begin with a lowercase letter, while nonterminals are in boldface and begin with an uppercase letter. The symbol | separates alternative derivations, and terms in brackets are optional. Watson begins derivations from the **Question** nonterminal to generate a question that can be given as input to the model, and Sherlock begins from the **Answer** nonterminal to generate an answer to such a question.

[a] When the parent nonterminal is **Question**, the evaluation chosen for **Property** is constrained so as not to reveal the missing property from the preceding **What** expansion. For example, questions such as *What color pyramid is red?* are disallowed.

[b] Pyramid objects are disallowed as subjects of *under* and objects of *on* because, in the simulated environment, pyramids cannot support other objects due to their pointed tops.

[c] The evaluation chosen for **Is** must agree in number with the **Object** expansion that serves as its subject.

Table 1: Binary Acoustic Features of Heard Phonemes

	Phoneme																																				
Feature	b	d	e	f	g	h	i	j	k	l	m	n	o	p	s	t	u	v	w	z	æ	ð	ŋ	ɑ	ɔ	ə	ɚ	ɛ	ɪ	ɹ	ʃ	ʊ	ʌ	ʒ	ʤ	ʧ	θ
Consonantal	+	+	−	+	+	+	−	+	+	+	+	+	−	+	+	+	−	+	+	+	−	+	+	−	−	−	−	−	−	+	+	−	−	+	+	+	+
Vocalic	−	−	+	−	−	−	+	−	−	+	−	−	+	−	−	−	+	−	−	−	+	−	−	+	+	+	+	+	+	+	−	+	+	−	−	−	−
Compact	−	−	−	−	+	−	−	−	+	−	−	−	−	−	−	−	−	−	−	−	−	−	+	−	−	−	−	−	−	−	+	−	−	+	+	+	−
Diffuse	+	+	−	+	−	−	−	−	−	−	+	+	−	+	+	+	−	+	−	+	−	+	−	−	−	−	−	−	−	−	−	−	−	−	−	−	+
Grave	+	−	−	+	−	−	−	−	−	−	+	−	−	+	−	−	−	+	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−
Acute	−	+	−	−	−	−	−	−	−	−	−	+	−	−	+	+	−	−	−	+	−	+	−	−	−	−	−	−	−	−	−	−	−	−	−	−	+
Nasal	−	−	−	−	−	−	−	−	−	−	+	+	−	−	−	−	−	−	−	−	−	−	+	−	−	−	−	−	−	−	−	−	−	−	−	−	−
Oral	+	+	−	+	+	+	−	+	+	+	−	−	−	+	+	+	−	+	+	+	−	+	−	−	−	−	−	−	−	+	+	−	−	+	+	+	+
Tense	−	−	+	+	−	+	+	−	+	−	−	−	+	+	+	+	+	−	−	−	−	−	−	+	−	−	+	−	−	−	+	−	−	−	−	+	+
Lax	+	+	−	−	+	−	−	−	−	−	−	−	−	−	−	−	−	+	−	+	+	+	−	−	+	+	−	+	+	−	−	+	+	+	+	−	−
Continuant	−	−	−	+	−	−	−	−	−	−	−	−	−	−	+	−	−	+	+	+	−	+	−	−	−	−	−	−	−	−	+	−	−	+	−	−	+
Interrupted	+	+	−	−	+	−	−	−	+	−	−	−	−	+	−	+	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	+	+	−
Strident	−	−	−	−	−	−	−	−	−	−	−	−	−	−	+	−	−	−	−	+	−	−	−	−	−	−	−	−	−	−	−	−	−	−	+	+	−
Mellow	−	−	−	−	+	−	−	−	+	−	−	−	−	−	−	−	−	−	−	−	−	+	−	−	−	−	−	−	−	−	−	−	−	−	−	−	+
+Voicing	+	+	+	−	+	−	+	+	−	+	+	+	+	−	−	−	+	+	+	+	+	+	+	+	+	+	+	+	+	+	−	+	+	+	+	−	−
−Voicing	−	−	−	+	−	+	−	−	+	−	−	−	−	+	+	+	−	−	−	−	−	−	−	−	−	−	−	−	−	−	+	−	−	−	−	+	+
+Duration	−	−	−	−	−	−	−	−	−	−	−	−	−	−	+	−	−	−	−	+	−	−	−	−	−	−	−	−	−	−	+	−	−	+	−	−	−
−Duration	+	+	−	+	+	+	−	+	+	+	+	+	−	+	−	+	−	+	+	−	−	+	+	−	−	−	−	−	−	+	−	−	−	−	+	+	+
+Frication	−	−	−	+	−	+	−	−	−	−	−	−	−	−	+	−	−	+	−	+	−	+	−	−	−	−	−	−	−	−	+	−	−	+	+	+	+
−Frication	+	+	−	−	+	−	−	+	+	+	+	+	−	+	−	+	−	−	+	−	−	−	+	−	−	−	−	−	−	+	−	−	−	−	−	−	−
Liquid	−	−	−	−	−	−	−	−	−	+	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	+	−	−	−	−	−	−	−
Glide	−	−	−	−	−	−	−	+	−	−	−	−	−	−	−	−	−	−	+	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−
Retroflex	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	+	−	−	+	−	−	−	−	−	−	−
$F_{2,VH}$	−	−	+	−	−	−	+	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	+	−	−	−	−	−	−	−	−
$F_{2,H}$	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	+	−	−	−	−	−	−	+	−	−	−	−	−	−	−	−	−
$F_{2,HM}$	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	+	+	−	−	−	−	−	+	−	−	−	−
$F_{2,LM}$	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	+	−	−	−	−	−	−	−	−	−	−	−	−	−
$F_{2,L}$	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	+	−	−	−	−	−	−	+	−	−	−	−	−
$F_{2,VL}/F_{1,VH}$	−	−	−	−	−	−	−	−	−	−	−	−	+	−	−	−	+	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	+	−	−	−	−
$F_{1,H}$	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	+	−	−	+	−	+	−	−	−	−	−	−	−	−	−	−	−
$F_{1,HM}$	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	+	−	−	+	−	−	−	−	−	−	−	−	−
$F_{1,LM}$	−	−	+	−	−	−	−	−	−	−	−	−	+	−	−	−	−	−	−	−	−	−	−	−	−	−	+	−	−	−	−	−	−	−	−	−	−
$F_{1,L}$	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	+	−	−	−	−	−	−	−	−	−	−	−	+	−	−	−	−	−	−	−	−
$F_{1,VL}$	−	−	−	−	−	−	+	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	+	−	−	−	−	−

adjusts its connection weights to better imitate this response in the future. Training is supervised in the sense that complete training targets are always provided, but it is a type of self-supervision where the model need only observe the naturally occurring dialog between Watson and Sherlock and attempt—perhaps covertly—to imitate the latter.

To answer Watson's question, the model must examine the environment and determine the set of object attributes and relations Watson is referring to. If the question was *What color block is the green pyramid on?*, in the context of the environment of Figure 2, the model must first determine that `(Color green 4)` and `(Shape pyramid 4)` are of interest, and then find that the on-relation only makes sense with the object labeled 1, as `(Location on 4 1)`. `(Shape block 1)` shows that object 1 fits with the clue that Watson has a block in mind. Finally, the model must deduce that Watson asked about the color, and so must retrieve the predicate `(Color blue 1)` from its working-memory representation of the environment.

When training the meaning-answer model, Sherlock gives precisely these five predicates as its answer, and the meaning-answer model must learn to do the same. On the other hand, when training the spoken-answer model, Sherlock gives the answer as a speech sequence. Specifically, Sherlock derives a complete English sentence, starting from the **Answer** nonterminal in the grammar of Figure 4. Sherlock constrains the derivation to conform to the contents of the predicate-based answer. For the question *What color block is the green pyramid on?* and the environment shown in Figure 2, Sherlock will invariably answer *The green pyramid is on the blue block.* The answer sentence is phonetically transcribed, just as Watson's questions are, to form an unsegmented temporal sequence of phonemes. The spoken-answer model uses Sherlock's speech stream as a temporal sequence of training targets to be produced in response to the question input.

Previous models (e.g., Plaut & Kello, 1999) have proposed that a learner run its internal representation of the answer's meaning forward to create a series of articulations for the sounds of the sentence. By feeding these representations through a forward model mapping articulatory features to auditory features, the learner could generate predictions about the speech stream. This would enable the learner to compare Sherlock's speech stream with its own predictions, working backward to turn auditory prediction errors into a training signal for articulatory features, and thus learning how to produce the desired speech. The spoken-answer model, for computational expediency, assumes that this process has already taken place, training directly on phonemes as bundles of binary articulatory features. The complete list of binary articulatory features and associated phonemes can be found in Table 2.

Table 2: Binary Articulatory Features of Spoken Phonemes

	Phoneme																																				
Feature	b	d	e	f	g	h	i	k	l	m	n	o	p	s	t	u	v	v	w	z	æ	ð	ɳ	ɑ	ɔ	ə	ɚ	ɛ	ɪ	ɹ	ʃ	ʊ	ʌ	ʒ	ʤ	ʧ	θ
Consonantal	+	+	−	+	+	+	−	+	+	+	+	−	+	+	+	−	+	+	+	+	−	+	+	−	−	−	−	−	−	+	+	−	−	+	+	+	+
Vocalic	−	−	+	−	−	−	+	−	+	−	−	+	−	−	−	+	−	−	−	−	+	−	−	+	+	+	+	+	+	+	−	+	+	−	−	−	−
Anterior	+	+	−	+	−	−	−	−	+	+	+	−	+	+	+	−	−	+	−	+	−	+	−	−	−	−	−	−	−	−	−	−	−	−	−	−	+
Coronal	−	+	−	−	−	−	−	−	+	−	+	−	−	+	+	−	−	−	−	+	−	+	−	−	−	−	−	−	−	+	+	−	−	+	+	+	+
+Voicing	+	+	+	−	+	−	+	−	+	+	+	+	−	−	−	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	−	+	+	+	+	−	−
−Voicing	−	−	−	+	−	+	−	+	−	−	−	−	+	+	+	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	+	−	−	−	−	+	+
Continuant	−	−	+	+	−	+	+	−	+	−	−	+	−	+	−	+	+	+	+	+	+	+	−	+	+	+	+	+	+	+	+	+	+	+	−	−	+
Stop	+	+	−	−	+	−	−	+	−	+	+	−	+	−	+	−	−	−	−	−	−	−	+	−	−	−	−	−	−	−	−	−	−	−	+	+	−
Nasal	−	−	−	−	−	−	−	−	−	+	+	−	−	−	−	−	−	−	−	−	−	−	+	−	−	−	−	−	−	−	−	−	−	−	−	−	−
Strident	−	−	−	+	−	−	−	−	−	−	−	−	−	+	−	−	−	+	−	+	−	−	−	−	−	−	−	−	−	−	+	−	−	+	+	+	−
Very High	−	−	−	−	−	−	+	−	−	−	−	−	−	−	−	+	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−
High	−	−	−	−	+	−	−	+	−	−	−	−	−	−	−	−	+	−	+	−	−	−	+	−	−	−	−	−	+	−	+	+	−	+	+	+	−
Middle	+	+	+	+	−	−	−	−	+	+	+	+	+	+	+	−	−	+	−	+	−	+	−	−	−	+	+	−	−	+	−	−	−	−	−	−	+
Low	−	−	−	−	−	+	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	+	−	−	+	−	−	−	−	+	−	−	−	−
Very Low	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	+	−	−	+	−	−	−	−	−	−	−	−	−	−	−	−	−
Front	−	−	+	−	−	−	+	−	−	−	−	−	−	−	−	−	−	−	−	−	+	−	−	−	−	−	−	+	+	−	−	−	−	−	−	−	−
Front Center	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	+	+	−	−	−	−	−	−	−	−	−	−
Center	+	+	−	+	−	+	−	−	+	+	+	−	+	+	+	−	+	+	−	+	−	+	−	−	−	−	+	−	−	+	+	−	−	+	+	+	+
Back Center	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	+	−	−	−	−
Back	−	−	−	−	+	−	−	+	−	−	−	+	−	−	−	+	−	−	+	−	−	−	+	+	+	−	−	−	−	−	−	+	−	−	−	−	−

In both the predicate-output and speech-output scenarios, the answers to the questions differ, based on the current environment, such that the mapping from questions to answers is never one-to-one. Indeed, both predicate-based and speech-based answers to Watson's question *What color block is the green pyramid on?* would change if the block under the green pyramid were red instead of blue; the predicate-based answer would also change (albeit slightly) if the scene were visually identical but the object labeling was altered. Though this example suggests that the number of possible answer variations is small, some questions admit much greater environment-based variation than others. Questions that are more open-ended, such as *Where is the block?* could have literally hundreds of possible answers stemming from the diversity of environments where the question makes sense. Thus, neither the model nor Sherlock could answer questions reliably without integrating information from the visual environment.

Neural architectures

To learn the tasks described in the previous section, the neural networks that comprise both the meaning-answer model and the spoken-answer model need to accurately recognize and produce sequences that are as long as 20 predicates or 40 phonemes. They are built[1] using the long short-term memory architecture (LSTM; Hochreiter & Schmidhuber, 1997; Gers & Cummins, 2000; Gers

[1] The models were implemented using our open-source Java library XLBP, which stands for eXtensible Layered Back-Propagation and is capable of creating and training classic back-

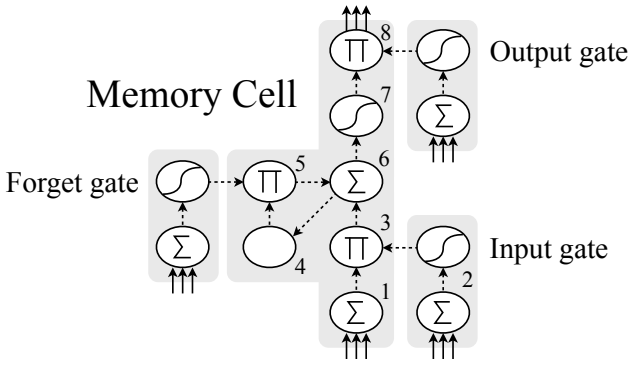

Figure 5: Diagram of an LSTM memory cell. Each oval represents a locus of computation within the cell. Solid arrows represent trainable weighted links into or out of the memory cell, while dashed arrows represent non-weighted input/output relationships among the cell-internal computations. Like traditional neural network units, the memory cell forms a weighted sum of its inputs (1). The input gate similarly aggregates its own inputs and squashes the result with a logistic function (2), putting it in the [0, 1] range. The resulting input gate activation multiplicatively modulates the input coming into the memory cell (3), allowing none, some, or all of the input to enter. The state of the cell from the previous time step is retained (4) and modulated by the activation of the forget gate (5). The resulting state is added to the current input to form the new state (6), which is then passed through a logistic function (7). The memory cell's outward-facing activation is calculated by multiplying in the activation of the output gate (8). The resulting activation can be passed downstream to other units via further weighted connections.

& Schmidhuber, 2001), which has been previously shown to perform very well on long temporal sequences. Hidden layers of an LSTM network are similar in many ways to those in a simple recurrent network (SRN; Jordan, 1986; Elman, 1990), with the chief difference being that, instead of a layer-level context, where the contents of a layer are copied to serve as input for the next step in the sequence, LSTM networks are comprised of special hidden layer units, called memory cells, that each utilize a unit-level context. In other words, each memory cell's current activation depends directly upon its state during the previous time-step. In addition, these self-recurrent memory cells sport multiplicative input and output gates that learn to protect the unit's stored activation from unhelpful input and prevent the unit from prematurely interfering with downstream processing. A third multiplicative gate learns to provide a gain on each unit's previous state, allowing it to forget this state when it is no longer useful. Figure 5 depicts and further explains a typical memory cell and its gates. For a detailed account of the activation and learning properties of these memory cells, see Monner & Reggia (2012).

This type of network can be trained by gradient descent methods similar

propagation networks, SRNs, LSTMs, and many other types. A link to the library is available at the author's website, `http://www.cs.umd.edu/~dmonner/`.

to those utilized by SRNs. Our models adopt the LSTM-g training algorithm (Monner & Reggia, 2012), which focuses on maintaining information locality as a means of approaching biological plausibility. Like many gradient-descent based algorithms, LSTM-g utilizes back-propagating errors, which apportions errors at the outputs and propagates error information to upstream units in proportion to each unit's responsibility for causing the error. This granular blame assignment allows individual units to modify the strengths of their incoming connections to help prevent similar errors in the future. Since real neural networks have no know mechanism for passing signals backwards through a network, many consider the use of back-propagation to be a detriment to the biological realism of any model. While we do not completely disagree, recent work shows how such error propagation is closely related to the more plausible family of Hebbian training methods (Xie & Seung, 2003), potentially minimizing this objection.

Although the basic components of the two models are the same, the network architectures differ slightly, based on the demands of each task. The network architecture of the meaning-answer model is shown in Figure 6. The network has an input layer for visual predicates (bottom right), which are presented temporally and accumulated into the visual gestalt in a visual accumulation layer. Internal layers such as this one are composed of the self-recurrent LSTM memory cells described above and are identified by the small recurrent arc on the top-left side. The network also has an input layer for auditory features, which feed through two successive layers of memory cells, forming an auditory gestalt on the second such layer. The two pathways differ in serial length because pilot experiments showed that the visual and auditory gestalt representations are best formed with one and two layers of intermediary processing, respectively. After both input gestalts have formed, they are integrated by another layer of memory cells, which is then used to sequentially generate predicates that specify the grounded output that answers the input question. The network's previous output is fed back to the integration layer in the style of a simple recurrent network (SRN; Jordan, 1986) to provide a more specific context for sequential processing.

The network that implements the spoken-answer model, detailed in Figure 7, differs from that of the meaning-answer model in only a few key respects. The main change is that the meaning-answer model's predicate-output layer is replaced by an output layer representing articulatory features used to create output speech sequences. The other change is the addition of another layer of memory cells between the integration layer and the output. Pilot experiments showed that network architectures lacking this additional layer had more trouble converting the intended speech stream into articulatory features. A key feature of the spoken-answer model architecture is that, unlike the meaning-answer model, it does not prespecify any semantic representations. This forces the network to learn its own internal meaning representations, which have the benefit of being distributed while supporting the type of systematic generalization that language use requires (Fodor & Pylyshyn, 1988; Hadley, 1994).

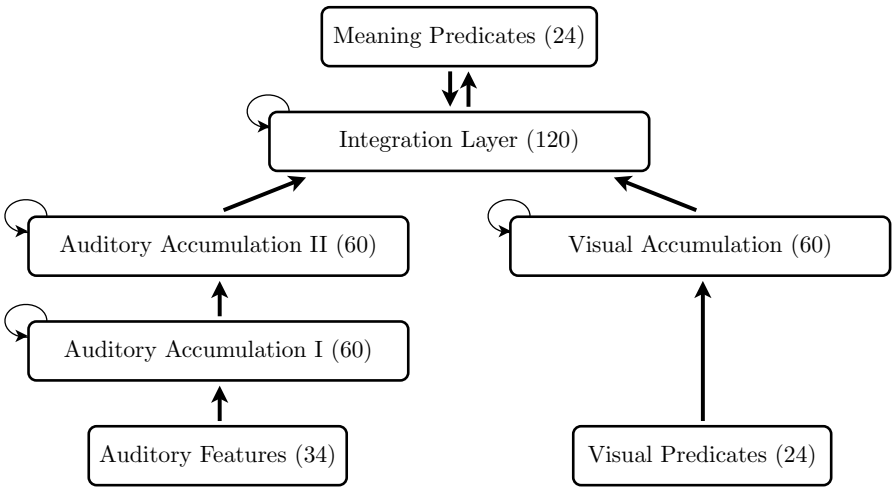

Figure 6: Neural network architecture of the meaning-answer model. Boxes represent layers of units (with number of units in parentheses), and straight arrows represent banks of trainable connection weights between units of the sending layer and units of the receiving layer. Layers with a curved self-connecting arrow are composed of LSTM memory cells. This architecture receives visual input from the bottom-right layer, which is temporally accumulated in the downstream layer of memory cells. Auditory input is received from the bottom-left layer, to be accumulated in two layers of memory cells, which helps the model represent two levels of structure (words and phrases). The network's representations of its auditory and visual inputs are integrated in a final layer of memory cells, from which the network produces the desired sequence of output predicates.

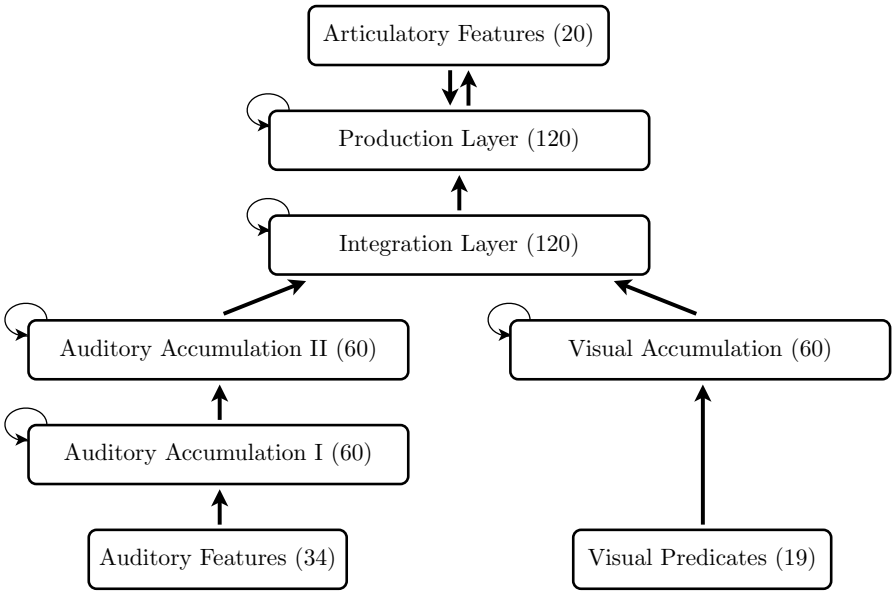

Figure 7: Neural network architecture of the spoken-answer model. Depicted in the style described in Figure 6, this architecture has an additional production layer that helps the network learn to generate the output phonemes from the internal meaning representations of the integration layer. The number of units in the visual predicate input layer decreased from 24 to 19 to accommodate the slightly simplified grammar that the spoken-answer model learns (see Figure 12).

Results

The meaning-answer model

We trained 10 independent instances of the meaning-answer model, each with randomly chosen initial connection weights, on the question-answering task for 5 million randomly generated trials. This duration of training may seem long, but it represents a mere fraction of a percent of the input space, leaving the model to generalize across the rest. An arrow between a pair of layers in Figure 6 indicates that a given pair of units from these layers possesses a trainable weighted connection with a probability of 0.7. The networks have 300 memory cells in all internal layers combined and about 60 thousand trainable connection weights in total. The networks use a learning rate of 0.01.

Reported accuracy results are based only on sets of 100 trials that the model had never encountered before evaluation time, and as such, always reflect generalization and never memorization. For each such trial, the model must produce a sequence of predicates, where a predicate is counted as correct only if it occurs in the correct sequential position and has all unit activations on the correct side of 0.5. On average, the trained meaning-answer models were able to produce a perfect sequence of grounded predicates—one which contains all necessary predicates in the proper order and no others—to answer a novel input question 92.5% of the time, strongly suggesting that this style of observe-and-imitate training is sufficient even for the difficult task of grounded question comprehension and answering. The 7.5% of incorrect answers skew towards the most difficult trials—those involving the longest questions, or with a large number of similar objects in the environment. Most incorrect answers can be traced to a single incorrect object label.

To evaluate the relative difficulty of the components of the task, one can decompose the output predicates into three categories: attribute values corresponding to linguistic descriptions that are present in the question, referent identifications made by grounding the question in the environment, and attributes that are inferred from the environment to answer the question. In other words, we examine the output units that the model activates, dividing them into three categories and scoring the model's accuracy at activating each. The referent identification category corresponds to all object label units, since each such output represents the identification of a referent from the question via the environment. The remaining output units represent attributes, which are divided into two classes: those that were mentioned in the input question (linguistic descriptions), and those that were not in the question but are required in the answer (attribute inferences). Comparing the time-course of accuracy over these three categories, Figure 8 shows that the model learns to understand spoken descriptions fairly quickly, with accuracy leveling off above 95% after the first million trials. At this point, accuracy on referent identifications and question-answers is relatively weak, at 80% and 70%, respectively; however, accuracy on these areas continues to slowly improve as training progresses, providing evidence that grounding the language in the environment is much more

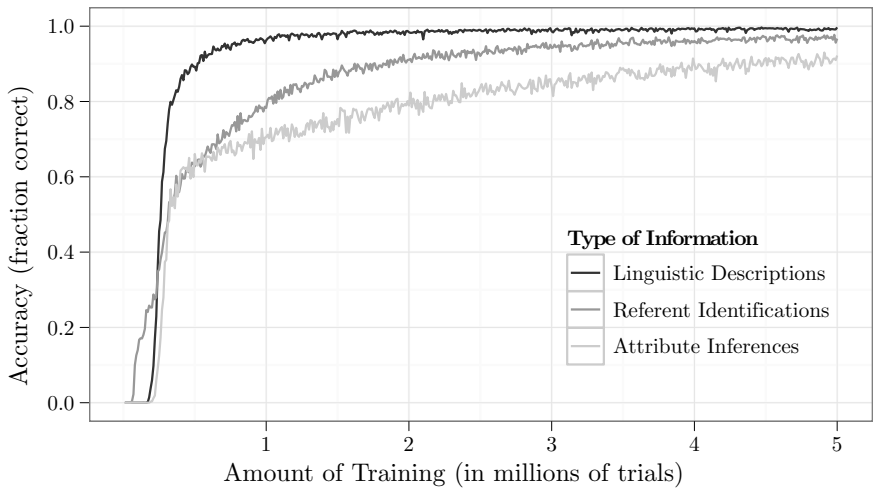

Figure 8: Time-course of three metrics during training. The three lines represent accuracy recovering linguistic descriptions, referent identifications, and attribute inferences, averaged over all ten training runs. That referent identifications precede linguistic descriptions at the far left of the graph is to be expected, since chance accuracy at picking a referent 25%.

difficult than recognizing a sequence of words and the basic outline of the meaning it entails. It also shows that using the environment to infer the answers to questions is harder still.

To begin to gain an understanding of how the model acquires this systematic ability to answer novel questions in novel environments, one must examine the internal representations that the model learns during training. These are obtained from snapshots of one trained model's integration layer taken immediately after it finishes hearing Watson's question; thus, the integration layer representation takes into account the network's working memory representations of both the visual environment and the auditory input. For certain very simple questions, such as *What color is the block?*, there are enough instances in the test data such that each possible answer—in this case, *red*, *green*, or *blue*—is represented multiple times. Looking for systematic differences between related sets of questions—and between multiple instances of the same question where the different environments would suggest differing answers—reveals what the model knows at the instant immediately after it hears the complete question.

The representations of three related questions are examined first. Each question asks about the color of a different type of object—a *block*, a *pyramid*, or a *cylinder*—and can have any of the three possible colors as its answer, depending on the environment. Applying principal component analysis (PCA) to the representations aids in analysis by removing some of the redundancy and distributedness that is characteristic of learned representations in neural networks. The input to the PCA in this case is a collection of activation vectors, each

of length 120, from the integration layer, where each vector is collected from a different trial, immediately after the inputs have finished and before the outputs begin. The output of the PCA, then, is a collection of the same size consisting of different 120-dimensional vectors in the transformed PC-space. We then consider the first 10 or so PC dimensions, since those account for the most variation in the integration layer's representations. While variation along the principal components (PCs) need not necessarily correspond to interpretable changes in the internal representation, this is in fact the case for many PCs. This indicates that the models are learning representations that can vary systematically in many dimensions. The following figures only involve those PCs for which the variation is easily interpretable. Though lower PC numbers represent larger amounts of variation in general, the PC number is essentially immaterial here, since the goal is merely to point out systematic variation in *some* PC.

Figure 9 graphs the representations in PCA-space, grouping them by question type, and subdividing those groups based on the expected answer. Here, each shaded polygon corresponds to a group of questions that are identical, though they may have different answers. For example, a polygon might represent the question *What color is the pyramid?*. Valid answers to this question involve a color attribute: `red`, `green`, or `blue`. Each vertex of such a polygon is labeled with the answer Watson is looking for and represents an average over all such questions that have this answer. So, to complete the example, the vertex labeled `blue` (the top-left-most vertex in Figure 9) represents the average representation over all instances where Watson asked *What color is the pyramid?* and had a blue pyramid in mind.

In this figure and those that follow, the representations for some of the shaded groups may seem to overlap, which one might expect to cause the model to mistake one group for another. These representations, however, are always well-separated by other principal components that are not shown, leaving the model with no such ambiguity.

The results show a remarkably systematic pattern. First, the model's internal representations differ consistently, based on the expected answer to the question, which was not present in the auditory input and has not yet been produced by the network as output predicates. This systematic difference implies that the model is aware of the expected answer to the question as soon as it is asked. While the model possessing the answer is not surprising, since all the information needed to deduce it is present in the visual input, the clarity of the answer's presence in the internal representation just after the question input suggests that the model processed the question online, deriving the answer immediately from its working memory representation of the environment. Additional systematicity is apparent in the respective orientations of the answer-groups for each question. Regardless of which question was asked, a *blue* answer is always a positive shift along PC7 and a *red* answer is always a negative shift, with *green* answers falling in the middle and along a positive shift in PC6. This type of organization serves as evidence of compositional, symbol-like representations for the various color concepts.

This analysis was repeated for a collection of similar *What size?* questions,

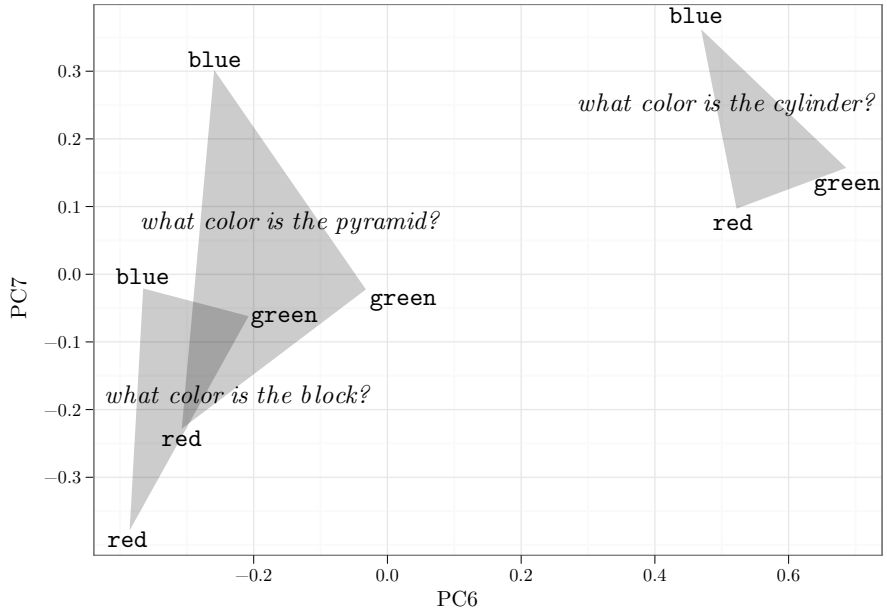

Figure 9: *What color?* question representations by expected answer. This plot shows the learned internal representations generated by the model for a set of *What color?* questions, visualized in PCA-space and separated according to the expected answer to the question. Shaded polygons correspond to sets of questions that are identical but might nonetheless have different answers. Each possible answering attribute corresponds to a labeled vertex of the polygon.

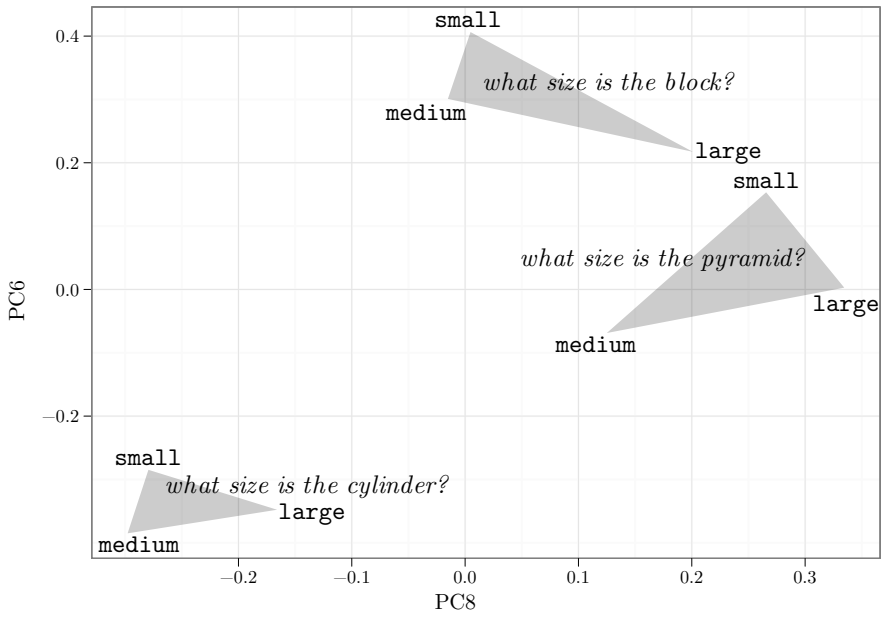

Figure 10: *What size?* question representations by expected answer. This plot shows the learned internal representations generated by the model for a set of *What size?* questions, visualized in PCA-space and separated according to the expected answer to the question, in the style of Figure 9.

with very similar results as shown in Figure 10. Again, the model displays clear representational differences that reveal its knowledge of the question's answer and further show it represening this information using systematic variations that are largely independent of the question being asked.

One might next inquire about the extent to which the model's internal representations reflect the environmental referents present in the question and answer. This can be gauged by analyzing the same set of *What color?* and *What size?* questions, again partitioning the trials into groups by question, but this time subdividing these groups based on the unique label that the simulated environment assigned to the referent. These labels are never represented in auditory input. Thus, the network's reliable knowledge about object labels is direct evidence that the model's internal sentence representations are grounded in the visual input. The results in Figure 11 show a remarkably systematic representation that makes clear several distinctions at once. First, the two question types *What color?* and *What size?* are separated along PC1. Second, the shape of the object in each question has an observable effect on its representation regardless of question type, with *pyramid* questions producing markedly smaller shaded regions in this projection than the other questions, while the *cylinder* and *block* questions produce regions of similar size, with the former shifted down along

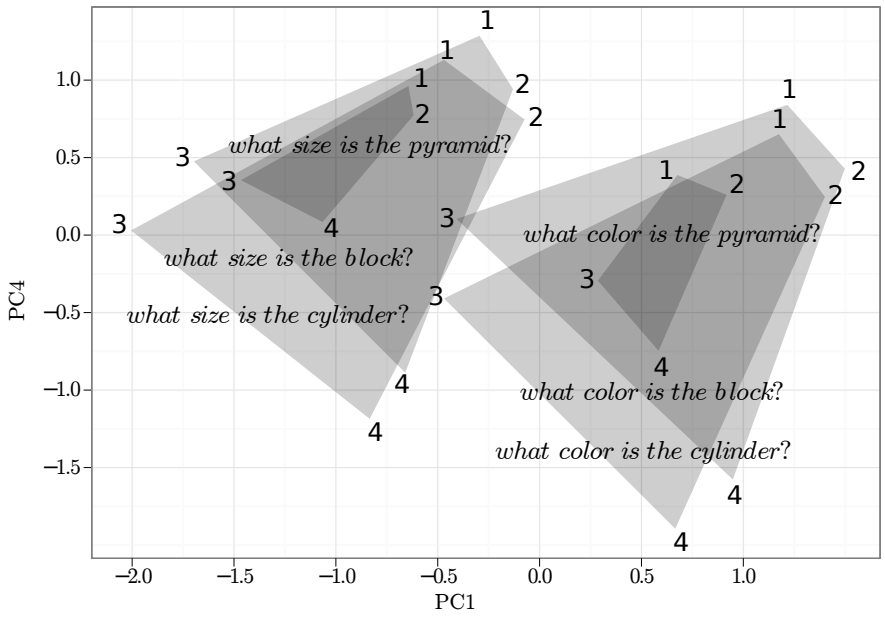

Figure 11: *What color/size?* question representations by object label. This plot shows the learned internal representations generated by the model for a set of *What color?* and *What size?* questions, visualized in PCA-space and separated according to the object label of the sentence's referent, in the style of Figure 9.

PC4 across both question types. Finally and most importantly, each question's representation reveals clear knowledge of the label assigned to the question's referent, with questions about objects 1 and 4 being distinguished along PC4, while questions about objects 2 and 3 differ along PC1. This figure definitively shows the systematic independence of the model's knowledge about the type of answer being sought (*size* or *color*), the shape of the object in question (*block*, *pyramid*, or *cylinder*), and the identifying label of the referent.

The spoken-answer model

Because of the additional size and complexity of the spoken-answer model, it trains on a slightly smaller grammar for computational expediency. This grammar is a subset of the one in Figure 4, arrived at by disallowing cylindrical objects and removing the size attribute altogether. For reference, the resulting grammar is shown in Figure 12. Pilot experiments indicated that the spoken-answer model has no trouble learning the full grammar, and scales from smaller grammars at approximately the same rate as the meaning-answer model, which appears to scale linearly in the number of words in the grammar; this is a non-trivial observation given that each additional word creates a new combinatorial dimension of variation in the space of the grammar. Computational resource

Question	→	where **Is Object** | **What Is Property**
Answer	→	**Object Is Property**
Object	→	the [**Color**] **Shape**
What	→	what color **Shape** | what [**Color**] thing
Property[1]	→	**Location** | **Color**
Location[2]	→	on **Object** | under **Object** | near **Object**
Is[3]	→	is | are
Color	→	red | blue | green
Shape	→	things | pyramid[s] | block[s]

Figure 12: Grammar used with the spoken-answer model. This slightly simpler grammar is used to train the spoken-answer model. For comparison, and for the text of the footnotes, see the original grammar in Figure 4.

constraints impeded replicated experiments with the spoken-answer model at a comparable size to the meaning-answer model.

We trained 10 individual instances of the spoken-answer model for up to 5 million randomly generated trials to learn the question-answering task, using the grammar from Figure 12. The networks are connected as shown in Figure 7, with each pair of units in connecting layers having a probability of 0.7 of sharing a weighted connection. This probability, combined with the 420 total memory cells across all internal layers, results in networks with approximately 120 thousand connection weights. The learning rate is 0.002.

Trained models are assessed on their ability to produce a perfect sequence of phonemes comprising a full-sentence answer to an input question on a novel trial. Each phoneme in a sequence is considered correct if each feature unit has an activation on the correct side of 0.5. On average, the spoken-answer models are able to accomplish this for 96.9% of never-before-seen trials. For example, in an environment containing a blue block on top of a red block, as well as a red pyramid nearby, Watson asked the model *What red thing is the blue block on?*, to which the model responded with [ðəblʊblɑkɪzɑnðəɹɛdblɑk], which translates as *The blue block is on the red block.* This response was correct in the sense that the phoneme sequence the model chose represents the desired answer to the question; additionally, the model produced these phonemes in the correct sequence, and each had exactly the right articulatory features.

The occasional (3.1%) errors made by the networks fall into roughly three categories. The first and most common is a slight mispronunciation of an otherwise correct answer. For example, the model was asked *What color pyramid is on the red block?*, and it produced the answer [ðəbl?pɪəmədɪzɑnðəɹɛdblɑk], where the "?" indicates that the network produced a pattern of articulatory features that does not precisely correspond to one of the language's phonemes. However,

it is clear from context that the model meant to say *The blue pyramid is on the red block*, which is the correct answer to the question in the environment provided during that trial. Despite the fact that the network clearly knew the correct answer and came extremely close to producing it, the reported statistics count every trial like this as an error.

The other type of error occurs in cases where the model seems unsure of the expected answer to the question. Sometimes, this takes the form of a direct and confident substitution of an incorrect answer, as was the case when the model was asked *What color is the block?* and confidently answered *The block is green* when the block was in fact blue. Other times, the model muddles two possible answers when speaking. For example, when asked *What color block is the blue pyramid under?*, the model responded with [ðəblʊpɹəmədɪzəndɚ·ðəgl?blɑk], which glosses roughly as *The blue pyramid is under the glih block*. The correct answer for the malformed word here would have been *blue*, but the model was apparently confused by a preponderance of green objects present in the environment on that trial, producing this hybrid of the two words in its answer. In other instances, the model trails off or "mumbles" when it does not know which word to say in its answer. A trial where the model was asked *Where is the blue pyramid?* provides an example of this behavior. The pyramid in question was on top of a green block, which required the model to produce three salient pieces of information that were not present in the question (i.e., *on*, *green*, and *block*) as part of its answer. The model came back with [ðəblʊpɹəmədɪzɑnðəgɻinb??], roughly *The blue pyramid is on the green buhhh....* Though the model produced the first two components of the expected answer, it was apparently unsure enough about *block* that it trailed off into unrecognizable phonemes and, in fact, stopped producing phonemes short of where the utterance would normally end.

Looking at the spoken-answer model's learned distributed representations reveals the same sorts of patterns that were present in those of the meaning-answer model. Figures 13–15 examine the spoken-answer model's internal representations by analyzing snapshots of the integration layer activations immediately after a question is presented. As before, PCA strips some of the redundancy out of the representations, identifying the main components for productive visualization, two at a time. This time, the investigation focuses on a more involved set of *Where?* questions, which each require the network to produce three pieces of information that were not present in the question. In response to the example question *Where is the red block?*, the model would need to respond by placing the red block in relation to a reference object, as in *The red block is under the blue pyramid*. Figures 13–15 test the internal representations for the presence of information about the location (*under*), color (*blue*), and shape (*pyramid*) of the reference object immediately after the model hears the question and before it begins its response.

Figure 13 shows a view of internal representations from PCs 2 and 3, depicting not only a clear separation of three variations of the *Where?* question, but also systematic manipulations of PC3 to distinguish the *on* and *near* relationals, while PC2 separates these from *under*.

Figure 14 shows the color of the reference objects in PCs 4 and 8. While

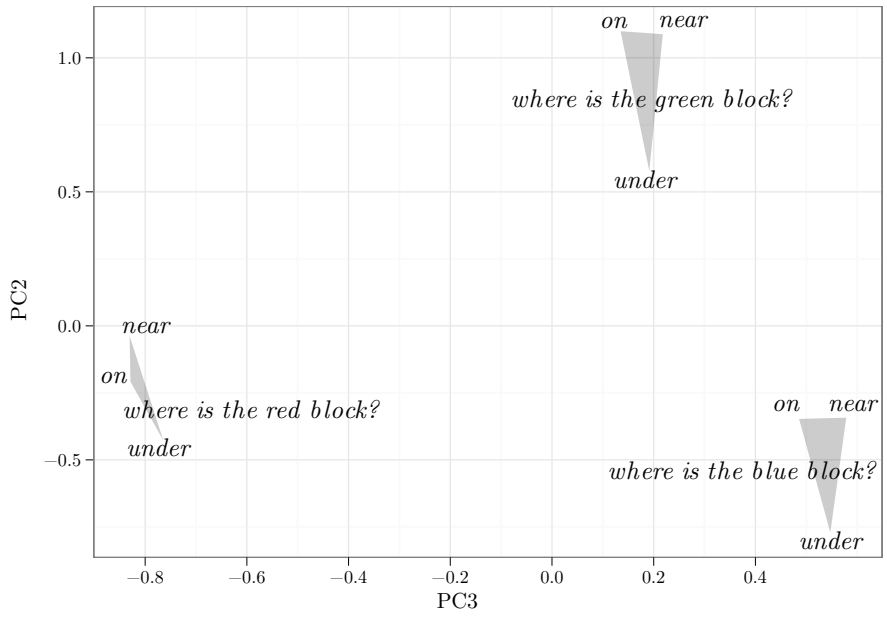

Figure 13: *Where?* question representations by answer location. This plot shows the learned internal representations generated by the spoken-answer model for a set of *Where?* questions, visualized in PCA-space and separated according to the relational word that positions the reference object in relation to the question's subject. Shaded polygons correspond to sets of questions that are identical but might nonetheless have different answers. Each possible word representing a valid answer corresponds to a labeled vertex of the polygon.

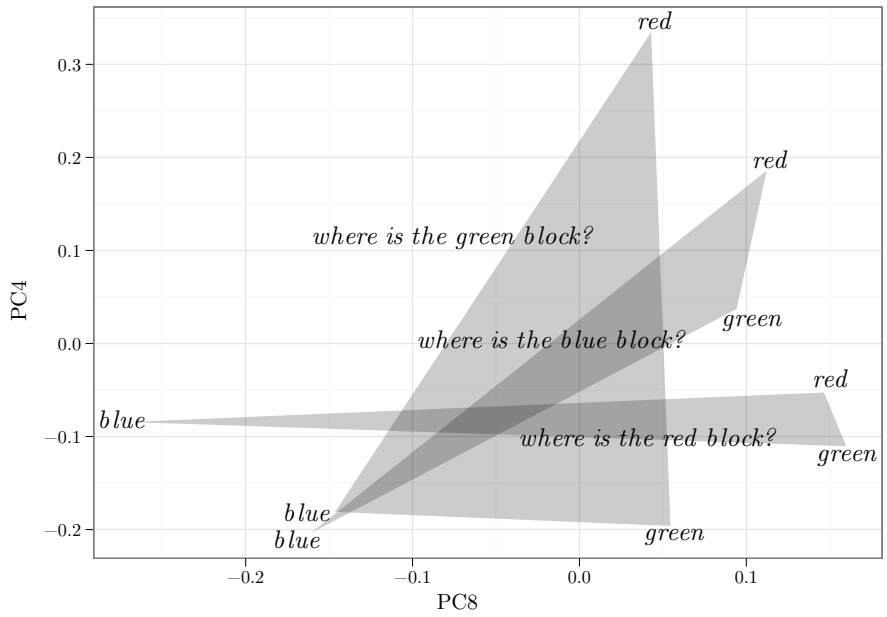

Figure 14: *Where?* question representations by answer color. This plot shows the learned internal representations generated by the model for a set of *Where?* questions, visualized in PCA-space and separated according to the color of the reference object used to locate the question's subject, in the style of Figure 13.

other PCs not depicted here demarcate representations of the different question types, this figure shows PC4 separating *red* from *green* and PC8 distinguishing *blue* from either of these.

Finally, Figure 15 breaks out the question representations in PCs 1 and 3, showing that the identity of the reference object as a *block* or a *pyramid* is primarily represented along the first principal component.

In total, Figures 13–15 present convincing evidence that the spoken-answer model learns internal representations much like those of the meaning-answer model. These representations quantize the input space and vary systematically along a number of principal dimensions to represent complex knowledge.

This section on the spoken-answer model, unlike the previous section on the meaning-answer model, does not contain a figure depicting information about the remaining property of the reference objects—the identifying label assigned to each. This is because an analysis of the representations showed no underlying systematicity to the representations when broken down by object label. However, this is not surprising. Being more speech-like, the responses that the spoken-answer model produces differ from those of the meaning-answer model in that they do not involve explicit specification of object labels. Object labels only exist in the visual input to the spoken-answer model, and to perform

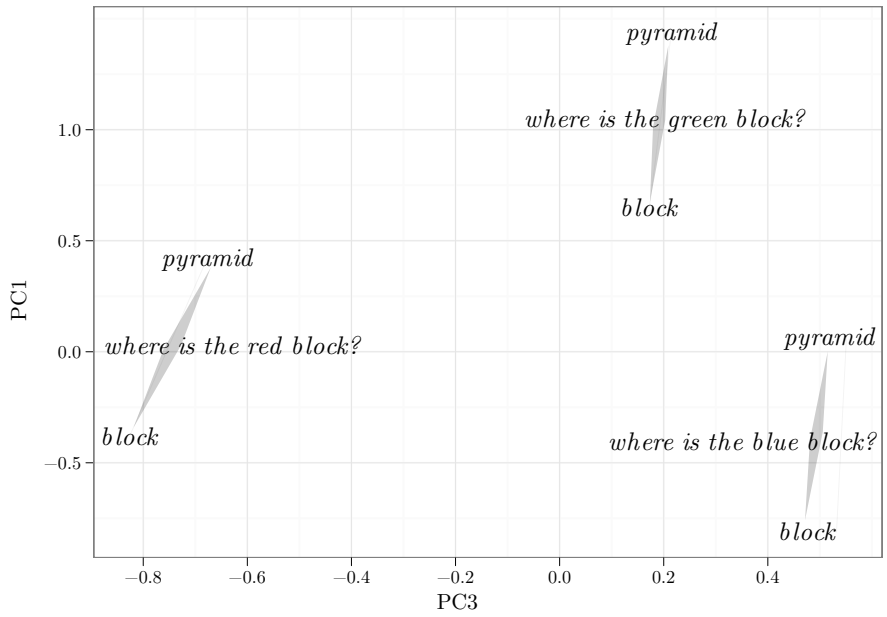

Figure 15: *Where?* question representations by answer shape. This plot shows the learned internal representations generated by the model for a set of *Where?* questions, visualized in PCA-space and separated according to the shape of the reference object used to locate the question's subject, in the style of Figure 13.

the question-answering task, the model must use them to bind attributes together to form coherent objects—that is, binding `(Color blue 2)` with `(Shape pyramid 2)` and `(Location near 2 3)` to produce the conception of a blue pyramid that is near some other object. Once this binding is complete for the entire environment, the model has no reason to retain the object label that was used to perform the binding; the label has served its function and is henceforth superfluous since it is not needed as part of the response. Therefore, one should not expect the high-level representations at the integration layer to involve object label at all, and indeed, the variation due to object label is small and unsystematic compared to the meaning-answer model.

Discussion

The results from the previous section suggest that for both the meaning-answer model and the spoken-answer model, observation and imitation are realistic methods by which linguistic question-answering behavior can be learned. The meaning-answer model learned to answer questions by directly mimicking the intended meaning of another speaker, Sherlock. While such imitation may occasionally be possible in situations where the learner can readily infer the speaker's meaning, such situations might not be frequent enough to facilitate full language learning. In any event, such a model could only hope to explain how the answers are derived, but not how they are communicated. Besides serving as a simplified proof-of-concept and a stepping stone to the spoken-answer model, the meaning-answer model provides clear evidence that a purely neural network model is capable of integrating two separate sensory streams to produce a coherent, grounded whole. The meaning-answer model is the only known neural network model able to successfully learn to map sentence-level questions, represented at sub-lexical resolution, onto complex, composable symbolic meanings representing their answers.

The spoken-answer model improves on the meaning-answer model in two ways. First, it learns to perform the question-answering task not by mimicking Sherlock's meaning, but by mimicking Sherlock's speech. Even though speech is often a lossy translation of the meaning, it has the virtue of always being observable—a property that places this model closer to real-world plausibility. Second, the spoken-answer model's performance encompasses not only answer derivation but also response generation, giving it an extra level of explanatory power over the meaning-answer model. This final model is the only known neural network able to map sub-lexical representations of natural language questions to natural language answers, devising its own semantic representations along the way.

While the spoken-answer model is a step in the right direction, it still relies on a symbol-like representation of the simulated environment. This fact limits its ability to scale, since each new property or relation added to the environment immediately requires a linear increase in the size of the network. In contrast, new words require no such increases up front, since new words can be created out of the model's current inventory of phonemes. The eventual solution to this

scalability issue will be to create a model that replaces the symbol-like environment representations with much more general features capable of generating a variety of rich visual spaces without modification. Such a model could potentially grow sublinearly as new words, attributes, and relations are added, thereby lending itself more readily to large-scale modeling efforts. Developing such a model and studying its scaling properties are important tasks left for future work.

Both of the models presented here appear to devise internal representations that are approximately compositional, imbuing them with the generative power of symbolic approaches to cognition. At the same time, these learned representations are distributed, conferring beneficial properties like redundancy and graceful degradation.

These models suggest that simple observation and imitation might be sufficient tools for learning to solve question-answering tasks. As discussed at the outset, question answering—or more generally, the request/response mode of communication—is fundamental to language. Since the principles in question are so elementary, computational models that observe and imitate seem to be ideal for application to complex language learning tasks.

Acknowledgements

This work was supported in part by an IC Postdoctoral Fellowship (2011-11071400011) awarded to DM.

Chang, F., Dell, G., & Bock, K. (2006). Becoming syntactic. *Psychological Review*, *113*, 234–272.

Dell, G. (1993). Structure and content in language production: A theory of frame constraints in phonological speech errors. *Cognitive Science*, *17*, 149–195.

Diederich, J., & Long, D. L. (1991). Efficient question answering in a hybrid system. In *Proceedings of the International Joint Conference on Neural Networks* (pp. 479–484).

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

Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A. A., Lally, A., Murdock, J. W., Nyberg, E., Prager, J., Schlaefer, N., & Welty, C. (2010). Building Watson: An overview of the DeepQA project. *AI Magazine*, *31*, 59–79.

Fodor, J. A., & Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition*, *28*, 3–71.

Gers, F. A., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. *Neural Computation*, *12*, 2451–2471.

Gers, F. A., & Schmidhuber, J. (2001). LSTM recurrent networks learn simple context-free and context-sensitive languages. *IEEE Transactions on Neural Networks*, *12*, 1333–1340.

Graesser, A. C., & Franklin, S. P. (1990). QUEST: A cognitive model of question answering. *Discourse Processes*, *13*, 279–303.

Hadley, R. (1994). Systematicity in connectionist language learning. *Mind & Language*, *9*, 247–272.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*, 1735–1780.

Jordan, M. I. (1986). Attractor dynamics and parallelism in a connectionist sequential machine. In *Proceedings of the Conference of the Cognitive Science Society* (pp. 531–546).

Markert, H., Kaufmann, U., Kara Kayikci, Z., & Palm, G. (2009). Neural associative memories for the integration of language, vision and action in an autonomous agent. *Neural Networks*, *22*, 134–143.

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

Miikkulainen, R. (1998). Text and discourse understanding: The DISCERN system. In R. Dale, H. Moisl, & H. Somers (Eds.), *A handbook of natural language processing: Techniques and applications for the processing of language as text*. New York: Marcel Dekker.

Monner, D., & Reggia, J. A. (2012). A generalized LSTM-like training algorithm for second-order recurrent neural networks. *Neural Networks*, *25*, 70–83.

Plaut, D., & Kello, C. (1999). The emergence of phonology from the interplay of speech comprehension and production: A distributed connectionist approach. In B. MacWhinney (Ed.), *The emergence of language* (pp. 381–415). Mahwah, NJ: Lawrence Erlbaum Associates.

Rohde, D. L. T. (2002). *A connectionist model of sentence comprehension and production*. Ph.D. dissertation, Carnegie Mellon University.

St. John, M. F., & McClelland, J. L. (1990). Learning and applying contextual constraints in sentence comprehension. *Artificial Intelligence*, *46*, 217–257.

Tomasello, M. (2003). *Constructing a language: A usage-based theory of language acquisition*. Cambridge, MA: Harvard University Press.

Williams, P., & Miikkulainen, R. (2006). Grounding language in descriptions of scenes. In *Proceedings of the Conference of the Cognitive Science Society* (pp. 2381–2386).

Xie, X., & Seung, H. S. (2003). Equivalence of backpropagation and contrastive Hebbian learning in a layered network. *Neural Computation*, *15*, 441–454.