A Framework for Fast Incremental Interpretation during Speech Decoding

William Schuler*
University of Minnesota

Stephen Wu*
University of Minnesota

Lane Schwartz*
University of Minnesota

This article describes a framework for incorporating referential semantic information from a world model or ontology directly into a probabilistic language model of the sort commonly used in speech recognition, where it can be probabilistically weighted together with phonological and syntactic factors as an integral part of the decoding process. Introducing world model referents into the decoding search greatly increases the search space, but by using a single integrated phonological, syntactic, and referential semantic language model, the decoder is able to incrementally prune this search based on probabilities associated with these combined contexts. The result is a single unified referential semantic probability model which brings several kinds of context to bear in speech decoding, and performs accurate recognition in real time on large domains in the absence of example in-domain training sentences.

1. Introduction

The capacity to rapidly connect language to referential meaning is an essential aspect of communication between humans. Eye-tracking studies show that humans listening to spoken directives are able to actively attend to the entities that the words in these directives might refer to, even while the words are still being pronounced (Tanenhaus et al. 1995; Brown-Schmidt, Campana, and Tanenhaus 2002). This timely access to referential information about input utterances may allow listeners to adjust their preferences among likely interpretations of noisy or ambiguous utterances to favor those that make sense in the current environment or discourse context, before any lower-level disambiguation decisions have been made. This same capability in a spoken language interface system could allow reliable human–machine interaction in the idiosyncratic language of day-to-day life, populated with proper names of co-workers, objects, and events not found in broad training corpora. When domain-specific training corpora are

* Department of Computer Science and Engineering, 200 Union St. SE, Minneapolis, MN 55455. E-mail: schuler@cs.umn.edu; swu@cs.umn.edu; lane@cs.umn.edu.

Submission received: 25 April 2007; revised submission received: 4 March 2008; accepted for publication: 2 June 2008.

not available, a referential semantic interface could still exploit its model of the world: the data to which it is an interface, and patterns characterizing these data.

This article describes a framework for incorporating referential semantic information from a world model or ontology directly into a statistical language model of the sort commonly used in speech recognition, where it can be probabilistically weighted together with phonological and syntactic factors as an integral part of the decoding process. Introducing world model referents into the decoding search greatly increases the search space, but by using a single integrated phonological, syntactic, and referential semantic language model, the decoder is able to incrementally prune this search based on probabilities associated with these combined contexts.

Semantic interpretation is defined dynamically in this framework, in terms of transitions over time from less constrained referents to more constrained referents. Because it is defined dynamically, interpretation in this framework can incorporate dependencies on referential context—for example, constraining interpretations to a presumed set of entities, or a presumed setting—which may be fixed prior to recognition, or dynamically hypothesized earlier in the recognition process. This contrasts with other recent systems which interpret constituents only given fixed inter-utterance contexts or explicit syntactic arguments (Schuler 2001; DeVault and Stone 2003; Gorniak and Roy 2004; Aist et al. 2007). Moreover, because it is defined dynamically, in terms of transitions, this context-dependent interpretation framework can be directly integrated into a Viterbi decoding search, like ordinary state transitions in a Hidden Markov Model. The result is a single unified referential semantic probability model which brings several kinds of referential semantic context to bear in speech decoding, and performs accurate recognition in real time on large domains in the absence of example domain-specific training sentences.

The remainder of this article is organized as follows: Section 2 will describe related approaches to interleaving semantic interpretation with speech recognition. Section 3 will provide definitions for world models used in semantic interpretation, and language models used in speech decoding, which will form the basis of a referential semantic language model, defined in Section 4. Then Section 5 will describe an evaluation of this model in a sample spoken language interface application.

2. Related Work

Early approaches to incremental interpretation (Mellish 1985; Haddock 1989) apply semantic constraints associated with each word in a sentence to progressively winnow the set of individuals that could serve as referents in that sentence. These incrementally constrained referents are then used to guide the syntactic analysis of the sentence, dispreferring analyses with empty interpretations in the current environment or discourse context. Similar approaches were applied to broad-coverage text processing, querying a large commonsense knowledge base as a world model (Martin and Riesbeck 1986). But this winnowing is done deterministically, invoking default assumptions and potentially exponential backtracking when default assumptions fail.

The idea of basing analysis decisions on constrained sets of referent individuals was later extended to pursue multiple interpretations at once by exploiting polynomial structure-sharing in a dynamic programming parser (Schuler 2001; DeVault and Stone 2003; Gorniak and Roy 2004; Aist et al. 2007). The resulting shared interpretation is similar to underspecified semantic representations (Bos 1996), except that the representation mainly preserves syntactic ambiguity rather than semantic (e.g., quanti-

fier scoping) ambiguity, and the size complexity of the parser chart representation is polynomially bounded. This approach was further extended to support hypothetical referents (DeVault and Stone 2003), domains with continuous relations (Gorniak and Roy 2004), and updates to the shared parser chart by components handling other levels of linguistic analysis in parallel, during real-time recognition (Aist et al. 2007).

The advantage of this use of the parser chart is that it allows a straightforward mapping between syntax and semantics using familiar compositional semantic representations. But the standard dynamic programming algorithm for parsing derives its complexity bounds from the fact that each recognized constituent can be analyzed independently of every other constituent. These independence assumptions must be relaxed if dynamic context dependencies are to be applied across sibling constituents (e.g., *in the package data directory, open ...*, where the files to be opened should be restricted to the contents of the package data directory). More importantly, from an engineering perspective, the dynamic programming algorithm for parsing runs in cubic time, not linear, which means this interpretation framework cannot be directly applied to continuous audio streams. Interface systems therefore typically perform utterance or sentence segmentation as a stand-alone pre-process, without integrating syntactic or referential semantic dependencies into this decision.

Finally, some speech recognition systems employ inter-utterance context-dependent language models that are pre-compiled into word *n*-grams for particular discourse or environment states, and swapped out between utterances (Young et al. 1989; Lemon and Gruenstein 2004; Seneff et al. 2004). But in some cases accurate interpretation will require spoken language interfaces to exploit context *continuously* during utterance recognition, not just between utterances. For example, the probability distribution over the next word in the utterance *go to the package data directory and get the ...* (or *in the package data directory get the ...*) will depend crucially on the linguistic and environment context leading up to this point: the meaning of *package data directory* in the first part of this directive, as well as the objects that will be available once this part of the directive has been carried out. Moreover, in rich environments pre-compilation to word *n*-grams can be expensive, since all referents in the world model must be considered to build accurate *n*-grams. This will not be practical if environments change frequently.

3. Background

In contrast to the approaches described in Section 2, this article proposes an incremental interpretation framework which is entirely contained within a single-pass probabilistic decoding search. Essentially, this approach directly integrates model theoretic semantics, summarized in Section 3.1, with conventional probabilistic time-series models used in speech recognition, summarized in Section 3.2.

3.1 Referential Semantics

Semantic interpretation requires a framework within which a speaker's intended meanings can be formalized. Sections 3.1.1 and 3.1.2 describe a model theoretic approach to semantic interpretation that will later be extended in Section 4.1. The referential states defined here will then be incorporated into a representation of nested syntactic constituents in a hierarchic time-series model in Section 4.2. Some of the notation introduced here is summarized later in Table 1 (Section 4).

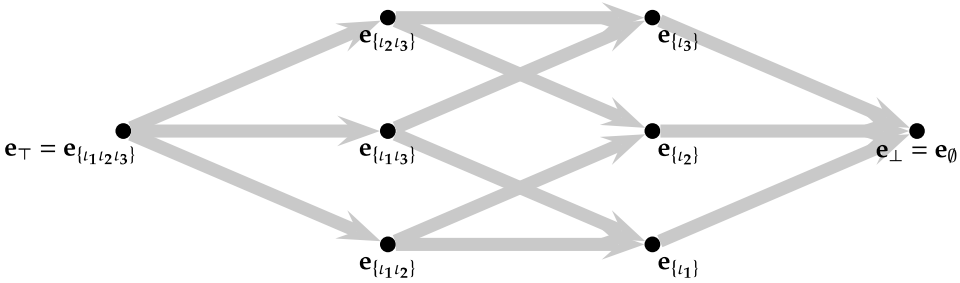

Figure 1
A subsumption lattice (laid on its side) over the power set of a domain containing three individuals: ι_1, ι_2, and ι_3. Subsumption relations are represented as gray arrows from supersets (or super-concepts) to subsets (or sub-concepts).

3.1.1 Model Theory. The language model described in this article defines semantic referents in terms of a world model $\mathcal{M}$. In model theory (Tarski 1933; Church 1940), a world model is defined as a tuple $\mathcal{M} = \langle \mathcal{E}, [\![\,]\!] \rangle$ containing a domain of individuals $\mathcal{E} = \{\iota_1, \iota_2, \dots\}$ and an interpretation function $[\![\,]\!]$ to interpret expressions in terms of those individuals. This interpretation function accepts expressions ϕ of various types: logical statements, of simple type $\mathbf{T}$ (for example, *the demo file is writable*) which may be true or false; references to individuals, of simple type $\mathbf{E}$ (for example, *the demo file*) which may refer to any individual in the world model; or functors of complex type $\langle \alpha, \beta \rangle$, which take an argument of type α and produce output of type β. Functor expressions ϕ of type $\langle \alpha, \beta \rangle$ can be applied to other expressions ψ of type α as arguments to yield expressions $\phi(\psi)$ of type β (for example, *writable* may take *the demo file* as an argument and return true). By nesting functors, complex expressions can be defined, denoting sets or properties of individuals: $\langle \mathbf{E}, \mathbf{T} \rangle$ (for example, *writable*), relations over individual pairs: $\langle \mathbf{E}, \langle \mathbf{E}, \mathbf{T} \rangle \rangle$ (for example, *contains*), or first-order functors over sets: $\langle \langle \mathbf{E}, \mathbf{T} \rangle, \langle \mathbf{E}, \mathbf{T} \rangle \rangle$ (for example, a comparative adjective like *larger*).

3.1.2 Ontological Promiscuity. First-order or higher models (in which functors can take sets as arguments) can be mapped to equivalent zero-order models (with functors defined only on entities). This is generally motivated by a desire to allow sets of individuals to be described in much the same way as individuals themselves (Hobbs 1985). Entities in a zero-order model $\mathcal{M}$ can be defined from individuals in a higher-order model $\mathcal{M}^*$ by mapping or *reifying* each set $S = \{\iota_1, \iota_2, \dots\}$ in $\mathcal{P}(\mathcal{E}_{\mathcal{M}^*})$ (or each set of sets in $\mathcal{P}(\mathcal{P}(\mathcal{E}_{\mathcal{M}^*}))$, etc.) as an entity $\mathbf{e}_S$ in a new domain $\mathcal{E}_{\mathcal{M}}$.[1] Relations l interpreted as zero-order functors in $\mathcal{M}$ can be defined directly from relations l^* interpreted as higher-order functors (over sets) in $\mathcal{M}^*$ by mapping each instance of $\langle S_1, S_2 \rangle$ in $[\![l^*]\!]_{\mathcal{M}^*} : \mathcal{P}(\mathcal{E}_{\mathcal{M}^*}) \times \mathcal{P}(\mathcal{E}_{\mathcal{M}^*})$ to a corresponding instance of $\langle \mathbf{e}_{S_1}, \mathbf{e}_{S_2} \rangle$ in $[\![l]\!]_{\mathcal{M}} : \mathcal{E}_{\mathcal{M}} \times \mathcal{E}_{\mathcal{M}}$. Set subsumption in $\mathcal{M}^*$ can then be defined on entities made from reified sets in $\mathcal{M}$, similar to 'IsA' relations over concepts in knowledge representation systems (Brachman and Schmolze 1985).

These subset or subsumption relations can be represented in a subsumption lattice, as shown in Figure 1, with supersets to the left connecting to subsets to the right. This representation will be used in Section 4 to define weighted transitions over first-order referents in a statistical time-series model of interpretation.

1 Here, $\mathcal{P}(X)$ is the power set of X, containing the set of all subsets.

3.2 Language Modeling for Speech Recognition

The referential semantic language model described in this article is based on Hierarchic Hidden Markov Models (HHMMs), an existing extension of the standard Hidden Markov Model (HMM) language modeling framework used in speech recognition, which has been factored to represent hierarchic information about language structure over time. This section will review HMMs (Section 3.2.1) and Hierarchic HMMs (Sections 3.2.2 and 3.2.3). This underlying framework will then be extended to include random variables over semantic referents in Section 4.2.

3.2.1 HMMs and Language Models. The model described in this article is a specialization of the HMM framework commonly used in speech recognition (Baker 1975; Jelinek, Bahl, and Mercer 1975). HMMs characterize speech as a sequence of hidden states h_t (which may consist of speech sounds, words, or other hypothesized syntactic or semantic information), and observed states o_t (typically finite, overlapping frames of an audio signal) at corresponding time steps t. A most-probable sequence of hidden states $\hat{h}_{1..T}$ can then be hypothesized given any sequence of observed states $o_{1..T}$, using Bayes' Law (Equation 2) and Markov independence assumptions (Equation 3) to define the full probability $\mathsf{P}(h_{1..T} \mid o_{1..T})$ as the product of a *Language Model (LM)* prior probability $\mathsf{P}(h_{1..T}) \stackrel{\text{def}}{=} \prod_t \mathsf{P}_{\Theta_{\text{LM}}}(h_t \mid h_{t-1})$ and an *Acoustic Model (AM)* likelihood probability $\mathsf{P}(o_{1..T} \mid h_{1..T}) \stackrel{\text{def}}{=} \prod_t \mathsf{P}_{\Theta_{\text{AM}}}(o_t \mid h_t)$:

$$\hat{h}_{1..T} = \underset{h_{1..T}}{\text{argmax}}\ \mathsf{P}(h_{1..T} \mid o_{1..T}) \tag{1}$$

$$= \underset{h_{1..T}}{\text{argmax}}\ \mathsf{P}(h_{1..T}) \cdot \mathsf{P}(o_{1..T} \mid h_{1..T}) \tag{2}$$

$$\stackrel{\text{def}}{=} \underset{h_{1..T}}{\text{argmax}} \prod_{t=1}^{T} \mathsf{P}_{\Theta_{\text{LM}}}(h_t \mid h_{t-1}) \cdot \mathsf{P}_{\Theta_{\text{AM}}}(o_t \mid h_t) \tag{3}$$

The initial hidden state h_0 may be defined as a constant.[2] HMM transitions can be modeled using Weighted Finite State Automata (WFSAs), corresponding to regular expressions. An HMM state h_t may then be defined as a WFSA state, or a symbol position in a corresponding regular expression.

3.2.2 Hierarchic HMMs. Language model transitions $\mathsf{P}_{\Theta_{\text{LM}}}(\sigma_t \mid \sigma_{t-1})$ over internally structured hidden states σ_t can be modeled using synchronized levels of stacked-up component HMMs in an HHMM (Murphy and Paskin 2001), generalized here as an abstract topology over unspecified random variables ρ and σ. In this topology, HHMM transition probabilities are calculated in two phases: a "reduce" phase (resulting in an intermediate, marginalized state ρ_t at time step t), in which component HMMs may terminate; and a "shift" phase (resulting in a modeled state σ_t), in which unterminated HMMs transition, and terminated HMMs are re-initialized from their parent HMMs. Variables over intermediate and modeled states are factored

2 It is also common to define a prior distribution over initial states at h_0, but this is not necessary here.

into sequences of depth-specific variables—one for each of D levels in the HHMM hierarchy:

$$\rho_t = \langle \rho_t^1 \dots \rho_t^D \rangle \tag{4}$$

$$\sigma_t = \langle \sigma_t^1 \dots \sigma_t^D \rangle \tag{5}$$

Transition probabilities are then calculated as a product of transition probabilities at each level, using level-specific "reduce" Θ_ρ and "shift" Θ_σ models:

$$\mathsf{P}_{\Theta_{\mathrm{LM}}}(\sigma_t \mid \sigma_{t-1}) = \sum_{\rho_t} \mathsf{P}(\rho_t \mid \sigma_{t-1}) \cdot \mathsf{P}(\sigma_t \mid \rho_t\ \sigma_{t-1}) \tag{6}$$

$$\stackrel{\text{def}}{=} \sum_{\rho_t^1 \dots \rho_t^D} \left(\prod_{d=1}^{D} \mathsf{P}_{\Theta_\rho}(\rho_t^d \mid \rho_t^{d+1} \sigma_{t-1}^d \sigma_{t-1}^{d-1}) \right) \cdot \left(\prod_{d=1}^{D} \mathsf{P}_{\Theta_\sigma}(\sigma_t^d \mid \rho_t^{d+1} \rho_t^d\ \sigma_{t-1}^d \sigma_t^{d-1}) \right) \tag{7}$$

with ρ_t^{D+1} and σ_t^0 defined as constants. In Viterbi (maximum likelihood) decoding, the marginals (sums) in this equation may be approximated using an argmax operator. A graphical representation of the dependencies in this model is shown in Figure 2.

3.2.3 Simple Hierarchic HMMs. The previous generalized definition can be considered a template for factoring HMMs into synchronized levels, using σ and ρ as parameters. The specific Murphy–Paskin definition of HHMMs can then be considered a "simple" instantiation of this template using FSA states for σ and switching variables for ρ. In Section 4, this instantiation will be augmented (or further factored) to incorporate additional variables over semantic referents at each depth and time step, without changing the overall topology of the model.

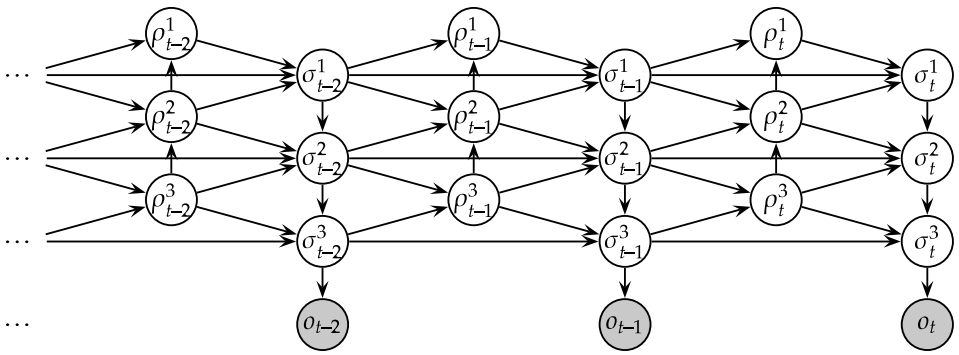

Figure 2
Graphical representation of a HHMM with $D = 3$ hidden levels. Circles denote random variables, and edges denote conditional dependencies. Shaded circles denote variables with observed values.

In simple HHMMs, each intermediate state variable ρ_t^d is a boolean switching variable $f_{\rho,t}^d \in \{\mathbf{0}, \mathbf{1}\}$ and each modeled state variable σ_t^d is a syntactic, lexical, or phonetic FSA state $q_{\sigma,t}^d$:

$$\rho_t^d = f_{\rho,t}^d \tag{8}$$

$$\sigma_t^d = q_{\sigma,t}^d \tag{9}$$

Instantiating Θ_ρ as $\Theta_{\text{Simple-}\rho}$, f^d is deterministic: true (equal to $\mathbf{1}$) with probability 1 if there is a transition at the level immediately below d and the stack element $q_{\sigma,t-1}^d$ is a final state, and false (equal to $\mathbf{0}$) with probability 1 otherwise:[3]

$$\mathsf{P}_{\Theta_{\text{Simple-}\rho}}(\rho_t^d \mid \rho_t^{d+1} \sigma_{t-1}^d \sigma_{t-1}^{d-1}) \stackrel{\text{def}}{=} \begin{cases} \text{if } f_{\rho,t}^{d+1} = \mathbf{0} & : \ [f_{\rho,t}^d = \mathbf{0}] \\ \text{if } f_{\rho,t}^{d+1} = \mathbf{1},\ q_{\sigma,t-1}^d \notin \text{Final} : \ [f_{\rho,t}^d = \mathbf{0}] \\ \text{if } f_{\rho,t}^{d+1} = \mathbf{1},\ q_{\sigma,t-1}^d \in \text{Final} : \ [f_{\rho,t}^d = \mathbf{1}] \end{cases} \tag{10}$$

where $f_{\rho,t}^{D+1} = \mathbf{1}$ and $q_{\sigma,t}^0 = \textbf{ROOT}$.

Shift probabilities at each level (instantiating Θ_σ as $\Theta_{\text{Simple-}\sigma}$) are defined using level-specific transition $\Theta_{\text{Simple-Trans}}$ and expansion $\Theta_{\text{Simple-Init}}$ models:

$$\mathsf{P}_{\Theta_{\text{Simple-}\sigma}}(\sigma_t^d \mid \rho_t^{d+1} \rho_t^d \ \sigma_{t-1}^d \sigma_t^{d-1}) \stackrel{\text{def}}{=} \begin{cases} \text{if } f_{\rho,t}^{d+1} = \mathbf{0},\ f_{\rho,t}^d = \mathbf{0} : \ [q_{\sigma,t}^d = q_{\sigma,t-1}^d] \\ \text{if } f_{\rho,t}^{d+1} = \mathbf{1},\ f_{\rho,t}^d = \mathbf{0} : \ \mathsf{P}_{\Theta_{\text{Simple-Trans}}}(q_{\sigma,t}^d \mid q_{\sigma,t-1}^d) \\ \text{if } f_{\rho,t}^{d+1} = \mathbf{1},\ f_{\rho,t}^d = \mathbf{1} : \ \mathsf{P}_{\Theta_{\text{Simple-Init}}}(q_{\sigma,t}^d \mid q_{\sigma,t}^{d-1}) \end{cases} \tag{11}$$

where $f_{\rho,t}^{D+1} = \mathbf{1}$ and $q_{\sigma,t}^0 = \textbf{ROOT}$. This model is conditioned on final-state switching variables at and immediately below the current HHMM level: If there is no final state immediately below the current level (the first case above), it deterministically copies the current FSA state forward to the next time step; if there is a final state immediately below the current level (the second case presented), it transitions the FSA state at the current level, according to the distribution $\Theta_{\text{Simple-Trans}}$; and if the state at the current level is final (the third case presented), it re-initializes this state given the state at the level above, according to the distribution $\Theta_{\text{Simple-Init}}$. The overall effect is that higher-level HMMs are allowed to transition only when lower-level HMMs terminate. An HHMM therefore behaves like a probabilistic implementation of a pushdown automaton (or "shift–reduce" parser) with a finite stack, where the maximum stack depth is equal to the number of levels in the HHMM hierarchy.

Like HMM states, the states at each level in a simple HHMM also correspond to weighted FSA (WFSA) states or symbol positions in regular expressions, except that some states can be *nonterminal states*, which introduce corresponding sub-expressions or sub-WFSAs governing state transitions at the level below. The process of expanding each nonterminal state $q_{\sigma,t}^{d-1}$ to a sub-expression or WFSA (with start state $q_{\sigma,t}^d$) is modeled in $\Theta_{\text{Simple-Init}}$. Transitions to adjacent (possibly final) states within each expression or WFSA are modeled in $\Theta_{\text{Simple-Trans}}$.

3 Here $[\cdot]$ is an indicator function: $[\phi] = 1$ if ϕ is true, 0 otherwise.

For example, a simple HHMM may factor a language model into word ($q^1_{\sigma,t}$), phone ($q^2_{\sigma,t}$), and subphone ($q^3_{\sigma,t}$) levels, where a word state may be a single word, a phone state may be a position in a sequence of phones corresponding to a word, and a subphone state may be a position in a sequence of subphone states (e.g., onset, middle, and end) corresponding to a phone. In this case, $\Theta_{\text{Simple-Init}}$ would define a prior model over words at level 1, a pronunciation model of phone sequences for each word at level 2, and a state-sequence model of subphone states for each phone at level 3; and $\Theta_{\text{Simple-Trans}}$ would define a word bigram model at level 1, and would deterministically advance along phone and subphone sequences at levels 2 and 3 (Bilmes and Bartels 2005).

This hierarchy of regular expressions may also be viewed as a probabilistic implementation of a cascaded FSA, used for modeling syntax in information extraction systems such as FASTUS (Hobbs et al. 1996).

4. A Referential Semantic Language Model

A referential semantic language model can now be defined as an instantiation of an HHMM (as described in Section 3.2), interpreting directives in a reified world model (as described in Section 3.1). This interpretation framework is novel in that it is defined dynamically in terms of transitions over *referential states*—evocations of entity referents from a (e.g., first-order) world model—stacked up in a Hierarchic HMM. This allows (1) a straightforward fast implementation of semantic interpretation (as transition) that is compatible with conventional time-series models used in speech recognition; and (2) a broader notion of semantic composition that exploits referential context in time order (from previous constituents to later constituents) as well as bottom-up (from component constituents to composed constituents).

First, Section 4.1 will describe a definition of semantic constraints as transitions in a time-series model. Then Section 4.2 will apply these transitions to nested referents in a Hierarchic HMM. Section 4.3 will introduce a state-based syntactic representation to link this semantic representation with recognized words. Finally, Section 4.4 will demonstrate the expressive power of this model on some common linguistic constructions.

Because this section combines notation from different theoretical frameworks (in particular, from formal semantics and statistical time-series modeling), a notation summary is provided in Table 1.

4.1 Dynamic Relations

Semantic interpretation may be easily integrated into a probabilistic time-series model if it is formulated as a type of *transition*, from source to destination referents of equivalent type at adjacent time steps. In other words, while relations in an ordinary Montagovian interpretation framework (Montague 1973) may be functions from entity referents to truth value referents, all relations in the world model defined here must be *transition functions* from entity referents to entity referents.

One-place properties l may be modeled in this system by defining transitions from preceding, unconstrained referents to referents constrained by l. The unconstrained referents can be thought of as *context* arguments: For example, in the context of the set of user-writable files, a property like EXECUTABLE evokes the subset of writable executables. In the subsumption lattice shown in Figure 1, this will define a rightward transition from each set referent to some subset referent, labeled with the traversed relation (see Figure 4 in Section 4.4).

Table 1
Summary of notation used in Section 4.

Model theory (see Section 3.1)
$\mathcal{M}$: a world model
$\mathcal{E}_{\mathcal{M}}$: the domain of individuals in world model $\mathcal{M}$
ι : an individual
$[\![\cdot]\!]_{\mathcal{M}}$: an interpretation function from logical symbols (e.g., relation labels) to logical functions over individuals, sets of individuals, etc.
variables with asterisks : refer to an initial world model prior to reification

Type theory (see Section 3.1.1)
E : the type of an individual
T : the type of a truth value
$\langle \alpha, \beta \rangle$: the type of a function from type α to type β (variables over types)

Set theory (see Section 4.1)
S : a set of individuals
R : a relation over tuples of individuals

Random variables (see Sections 3.2 and 4.2)
h : a hidden variable in a time-series model
o : an observed variable in a time-series model, (in this case, a frame of the acoustical signal)
ρ : a complex variable occurring in the reduce phase of processing; for example, composed of $\langle e_\rho, f_\rho \rangle$
σ : a complex variable occurring in the shift phase of processing; for example, composed of $\langle e_\sigma, q_\sigma \rangle$
f : a random variable over final state status; for example, with value **1** or **0**
q : a random variable over FSA (syntax) states, in this case compiled from regular expressions; for example, with value $\boldsymbol{q_1}$ or $\boldsymbol{q_2}$
e : a random variable over referent entities; for example, with value $\mathbf{e}_{\{\iota_1 \iota_2 \iota_3\}}$
l : a random variable over relation labels; for example, with value EXECUTABLE (see Section 4.1)

t : a time step, from 1 to the end of the utterance T
d : a depth level, from 1 to the maximum depth level D
Θ : a probability model mapping variable values to probabilities (real numbers form 0.0 to 1.0)
L : functions from FSA (syntax) states to relation labels

variables in **boldface** : instances or values of a random variable
non-bold variables with single subscripts : are specific to a time step; for example, ρ_t
non-bold variables with double subscripts : are specific to a reduce or shift phase within a time step; for example, $e_{\rho,t}$, $q_{\sigma,t}$
non-bold variables with superscripts : are specific to a depth level; for example, ρ_t^d, $e_{\rho,t}^d$

General n-ary semantic relations l in this framework are therefore formulated as a type of multi-source transition, distinguishing one argument of an original, ordinary relation l^* as an output (destination) and leaving the rest as input (source); then introducing a context referent as an additional input. Instead of defining simple transition arcs on a subsumption lattice, n-ary relations more accurately define *hyperarcs*, with *multiple* source referents: zero or more conventional arguments and one additional context referent, leading to a destination referent intersectively constrained to this context.

This model of interpretation as transition also allows referential semantic constraints to be applied that occur prior to hypothesized constituents, in addition to those that occur as arguments. For example, in the sentence *go to the package data directory and hide the executable file,* the phrase *go to the package data directory* provides a powerful constraint on the referent of *the executable file,* although it does not occur as an argument sub-constituent of this noun phrase. In this framework, the referent of *the package data directory* (as a set of files) can be passed as a context argument to intersectively constrain the interpretation of *the executable file.*

Recall the definition in Section 3.1.2 of a zero-order model $\mathcal{M}$ with referents $\mathbf{e}_{\{\iota_1,\iota_2,\dots\}}$ reified from sets of individuals $\{\iota_1, \iota_2, \dots\}$ in some original first- or higher-order model $\mathcal{M}^*$. The referential semantic language model described in this article interacts with this reified world model $\mathcal{M}$ through queries of the form $[\![l]\!]_{\mathcal{M}}(\mathbf{e}_{S_1}, \mathbf{e}_{S_2})$, where l is a relation, $\mathbf{e}_{S_1}$ is an argument referent, and $\mathbf{e}_{S_2}$ is a context referent (or $\mathbf{e}_{S_1}$ is a context referent if there is no argument). Each query returns a destination referent $\mathbf{e}_S$ such that S is a subset of the context set in the original world model $\mathcal{M}^*$. These context-dependent relations l in $\mathcal{M}$ are then defined in terms of corresponding ordinary relations l^* of various types in the original world model $\mathcal{M}^*$ as follows:

$$[\![l]\!]_{\mathcal{M}}(\mathbf{e}_{S_1}, \mathbf{e}_{S_2}) = \mathbf{e}_S \text{ s.t. } \begin{cases} \text{if } [\![l^*]\!]_{\mathcal{M}^*} \text{ is type } \langle \mathbf{E}, \mathbf{T} \rangle & : \; S = S_1 \cap [\![l^*]\!]_{\mathcal{M}^*} \\ \text{if } [\![l^*]\!]_{\mathcal{M}^*} \text{ is type } \langle \mathbf{E}, \langle \mathbf{E}, \mathbf{T} \rangle \rangle & : \; S = S_2 \cap (S_1 \cdot [\![l^*]\!]_{\mathcal{M}^*}) \\ \text{if } [\![l^*]\!]_{\mathcal{M}^*} \text{ is type } \langle \langle \mathbf{E}, \mathbf{T} \rangle, \langle \mathbf{E}, \mathbf{T} \rangle \rangle & : \; S = S_2 \cap [\![l^*]\!]_{\mathcal{M}^*}(S_1) \end{cases} \quad (12)$$

where relation products are defined to resemble matrix products:

$$S \cdot R = \{\iota'' \mid \iota' \in S, \; \langle \iota', \iota'' \rangle \in R\} \quad (13)$$

For example, a property like EXECUTABLE would ordinarily be modeled as a functor of type $\langle \mathbf{E}, \mathbf{T} \rangle$: given an individual, it would return true if the individual can be executed. The first case in Equation (12) casts this as a transition from an argument set S_1 to the set of individuals within S_1 that are executable. On the other hand, a relation like CONTAINS would ordinarily be modeled as $\langle \mathbf{E}, \langle \mathbf{E}, \mathbf{T} \rangle \rangle$: given an individual and then another individual, it would return true if the relation holds over the pair. The second case in Equation (12) casts this as a transition from a set of containers S_1, given a context set S_2, to the subset of this context that are contained by an individual in S_1. Finally, a first-order functor like LARGEST would ordinarily be modeled as $\langle \langle \mathbf{E}, \mathbf{T} \rangle, \langle \mathbf{E}, \mathbf{T} \rangle \rangle$: given a set of individuals and then another individual, it would return true if the individual belongs to the (singleton) set of things that are the largest in the argument set. The last case in Equation (12) casts this as a transition from a set S_1, given a context set S_2, to the (singleton) subset of this context that are members of S_1 and are larger than all other individuals in S_1. More detailed examples of each relation type in Equation (12) are provided in Section 4.4.

Relations in this world model have the character of being context-dependent in the sense that relations like CAPTAIN that are traditionally one-place (denoting a set of entities with rank captain) are now two-place, dependent on an argument superconcept in the subsumption lattice. Relations can therefore be given different meanings at different places in the world model: in the context of a particular football team, CAPTAIN will refer to a particular player; in the context of a different team, it will refer to someone else. One-place relations can still be defined using a subsumption lattice root concept

'⊤' as a context argument of course, but this will increase the perplexity (number of choices) at the root concept, making recognition less reliable.

In this definition, referents **e** are similar to the *information states* in Dynamic Predicate Logic (Groenendijk and Stokhof 1991), except that only limited working memory for information states is assumed, containing only one referent (or variable binding in DPL terms) per HHMM level.

4.2 Referential Semantic HHMM

Like the simple HHMM described in Section 3.2.3, the referential semantic language model described in this article (henceforth RSLM), is defined by instantiating the general HHMM "template" defined in Section 3.2.2. This RSLM instantiation incorporates both the switching variables $f \in \{\mathbf{0}, \mathbf{1}\}$ and FSA state variables q of the simple HHMM, and adds variables over semantic referents e to the "reduce" and "shift" phases at each level. Thus, the RSLM decomposes each HHMM reduce variable ρ_t^d into a joint variable subsuming an intermediate referent $e_{\rho,t}^d$ and a final-state switching variable $f_{\rho,t}^d$; and decomposes each HHMM shift variable σ_t^d into a joint variable subsuming a modeled referent $e_{\sigma,t}^d$ and an ordinary FSA state $q_{\sigma,t}^d$:

$$\rho_t^d = \langle e_{\rho,t}^d, f_{\rho,t}^d \rangle \tag{14}$$

$$\sigma_t^d = \langle e_{\sigma,t}^d, q_{\sigma,t}^d \rangle \tag{15}$$

A graphical representation of this referential semantic language model is shown in Figure 3.

The *intermediate referents* $e_{\rho,t}^d$ in this framework correspond to the traditional notion of compositional semantics (Frege 1892), in which meanings of composed constituents (at higher levels in the HHMM hierarchy) are derived from meanings of component constituents (at lower levels in the hierarchy). However, in addition to the referents

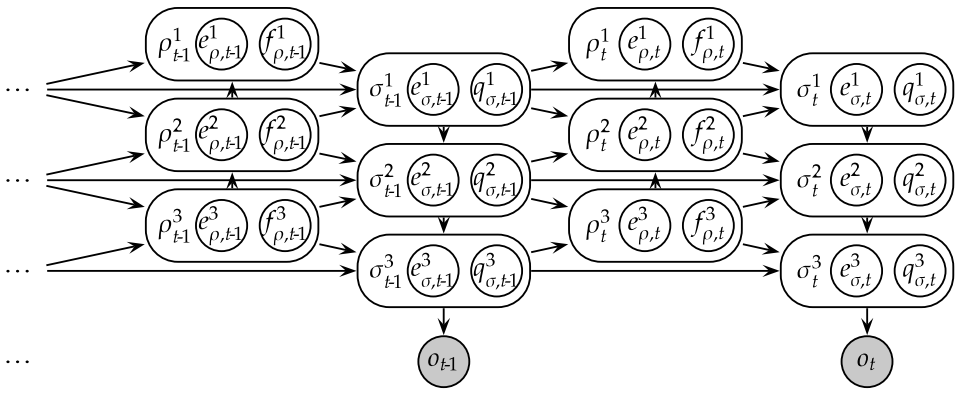

Figure 3
A graphical representation of the dependencies in the referential semantic language model described in this article (compare with Figure 2). Again, circles denote random variables and edges denote conditional dependencies. Shaded circles denote random variables with observed values.

of their component constituents, the intermediate referents in this framework are also constrained by the referents at the same depth in the previous time step—the referential context described in Section 4.1. The *modeled referents* $e^d_{\sigma,t}$ in this framework then correspond to a snapshot at each time step of the referential state of the recognizer, after all completed constituents have been composed (or reduced), and after any new constituents have been introduced (or shifted).

Both intermediate and modeled referents are constrained by labeled relations l in $[\![\cdot]\!]_{\mathcal{M}}$ associated with ordinary FSA states. Thus, relation labels are defined for "reduce" and "shift" HHMM operations via label functions L_ρ and L_σ, respectively, which map FSA states q to relation labels l.

Entity referents e^d_ρ at each reduce phase of this HHMM are constrained by the previous FSA state $q^d_{\hat{t}}$ using a reduce relation $l^d_{\rho,t} = L_\rho(q^d_{\sigma,\hat{t}})$, such that $e^d_\rho = [\![l^d_\rho]\!]_{\mathcal{M}}(e^{d+1}_\rho, e^d_{\hat{t}})$. Reduce probabilities at each level (instantiating Θ_ρ as $\Theta_{\text{RSLM-}\rho}$) are therefore:[4]

$$\mathsf{P}_{\Theta_{\text{RSLM-}\rho}}(\rho^d_t \mid \rho^{d+1}_t \sigma^d_{\hat{t}} \sigma^{d\hat{1}}_{\hat{t}}) \stackrel{\text{def}}{=} \begin{cases} \text{if } f^{d+1}_{\rho,t} = \mathbf{0} & : [f^d_{\rho,t} = \mathbf{0}] \cdot [e^d_{\rho,t} = e^d_{\sigma,t}] \\ \text{if } f^{d+1}_{\rho,t} = \mathbf{1},\ q^d_{\sigma,\hat{1}} \notin \text{Final} & : [f^d_{\rho,t} = \mathbf{0}] \cdot [e^d_{\rho,t} = e^{d+1}_{\rho,t}] \\ \text{if } f^{d+1}_{\rho,t} = \mathbf{1},\ q^d_{\sigma,\hat{1}} \in \text{Final} & : [f^d_{\rho,t} = \mathbf{1}] \cdot \\ & \quad [e^d_{\rho,t} = [\![l^d_{\rho,t}]\!]_{\mathcal{M}}(e^{d+1}_{\rho,t},\ e^{d\hat{1}}_{\sigma,\hat{t}})] \end{cases} \tag{16}$$

where $\rho^{D+1}_{\rho,t} = \langle e^D_{\sigma,\hat{1}}, \mathbf{1}\rangle$ and $\sigma^0_{\sigma,t} = \langle \mathbf{e}_\top, \mathbf{ROOT}\rangle$. Here, it is assumed that $L_\rho(q^d_{\sigma,\hat{1}})$ provides a non-trivial constraint only when $q^d_{\sigma,t}$ is a *final* state; otherwise it returns an IDENTITY relation such that $[\![\text{IDENTITY}]\!]_{\mathcal{M}}(e, e') = e$.

Entity referents $e^d_{\sigma,t}$ at each shift phase of this HHMM are constrained by the current FSA state $q^d_{\sigma,t}$ using a shift relation $l^d_{\sigma,t} = L_\sigma(q^d_{\sigma,t})$, such that $e^d_{\sigma,t} = [\![l^d_{\sigma,t}]\!]_{\mathcal{M}}(e^{d\hat{1}}_{\sigma,t}, \mathbf{e}_\top)$. Shift probabilities at each level (instantiating Θ_σ as $\Theta_{\text{RSLM-}\sigma}$) then generate relation labels using a "description" model $\Theta_{\text{Ref-Init}}$, with referents $e^d_{\sigma,t}$ and state transitions $q^d_{\sigma,t}$ conditioned on (or deterministically dependent on) these labels. The probability distribution over modeled variables is therefore

$$\mathsf{P}_{\Theta_{\text{RSLM-}\sigma}}(\sigma^d_t \mid \rho^{d+1}_t \rho^d_t\ \sigma^d_{\hat{t}} \sigma^{d\hat{1}}_t) \stackrel{\text{def}}{=} \begin{cases} \text{if } f^{d+1}_{\rho,t} = \mathbf{0}, f^d_{\rho,t} = \mathbf{0} : [e^d_{\sigma,t} = e^d_{\rho,t}] \cdot [q^d_{\sigma,t} = q^d_{\sigma,\hat{1}}] \\ \text{if } f^{d+1}_{\rho,t} = \mathbf{1}, f^d_{\rho,t} = \mathbf{0} : [e^d_{\sigma,t} = e^d_{\rho,t}] \cdot \mathsf{P}_{\Theta_{\text{Syn-Trans}}}(q^d_{\sigma,t} \mid q^d_{\sigma,\hat{1}}) \\ \text{if } f^{d+1}_{\rho,t} = \mathbf{1}, f^d_{\rho,t} = \mathbf{1} : \sum_{l^d_{\sigma,t}} \mathsf{P}_{\Theta_{\text{Ref-Init}}}(l^d_{\sigma,t} \mid e^{d\hat{1}}_{\sigma,t}\ q^{d\hat{1}}_{\sigma,t}) \\ \qquad \cdot [e^d_{\sigma,t} = [\![l^d_{\sigma,t}]\!]_{\mathcal{M}}(e^{d\hat{1}}_{\sigma,t}, \mathbf{e}_\top)] \\ \qquad \cdot \mathsf{P}_{\Theta_{\text{Syn-Init}}}(q^d_{\sigma,t} \mid l^d_{\sigma,t}\ q^{d\hat{1}}_{\sigma,t}) \end{cases} \tag{17}$$

where $\rho^{D+1}_{\rho,t} = \langle e^D_{\sigma,\hat{1}}, \mathbf{1}\rangle$ and $\sigma^0_{\sigma,t} = \langle \mathbf{e}_\top, \mathbf{ROOT}\rangle$. Here, it is assumed that $L_\sigma(q^d_{\sigma,t})$ provides a non-trivial constraint only when $q^d_{\sigma,t}$ is an *initial* state; otherwise it returns an IDENTITY relation such that $[\![\text{IDENTITY}]\!]_{\mathcal{M}}(e, e') = e$. The probability models $\Theta_{\text{Ref-Init}}$ and $\Theta_{\text{Syn-Init}}$ are induced from corpus observations or defined by hand.

The cases in this equation, conditioned on final-state switching variables $f^{d+1}_{\rho,t}$ and $f^d_{\rho,t}$, correspond to those in Equation (11) in Section 3.2.3. In the first case, where there is no final state immediately below the current level, referents and FSA states are simply propagated forward. In the second case, where there is a final state immediately below the current level, referents are propagated forward and the FSA state is advanced

4 Again, $[\cdot]$ is an indicator function: $[\phi] = 1$ if ϕ is true, 0 otherwise.

according to the distribution $\Theta_{\text{Syn-Trans}}$. In the third case, where the current FSA state is final and must be re-initialized, a new referent and FSA state are chosen by:

1. selecting, according to a "description" model $\Theta_{\text{Ref-Init}}$, a relation label $l^d_{\sigma,t}$ with which to constrain the current referent,
2. deterministically generating a referent $e^d_{\sigma,t}$ given this label and the referent at the level above, and
3. selecting, according to a "lexicalization" model $\Theta_{\text{Syn-Init}}$, an FSA state $q^d_{\sigma,t}$ that is compatible with this label (i.e., has $L_\sigma(q^d_{\sigma,t}) = l^d_{\sigma,t}$).

4.3 Associating Semantic Relations with Syntactic Expressions

In this framework, semantic referents are constrained over time by instances of semantic relations $\boldsymbol{l}_\sigma$ and $\boldsymbol{l}_\rho$. These relations are determined by instances of syntactic FSA states $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n$, themselves expanded from higher-level FSA states $\boldsymbol{q}$. These associations between syntactic and semantic random variable values can be represented in expansion rules of the form

$$\boldsymbol{q} \rightarrow \boldsymbol{q}_1 \ \ldots \ \boldsymbol{q}_n; \quad \text{with } \boldsymbol{l}_\sigma = L_\sigma(\boldsymbol{q}_1) \text{ and } \boldsymbol{l}_\rho = L_\rho(\boldsymbol{q}_n) \tag{18}$$

where $\boldsymbol{q}_1 \ \ldots \ \boldsymbol{q}_n$ may be any regular expression initiating at state $\boldsymbol{q}_1$ and culminating at (final) state $\boldsymbol{q}_n$. Note that regular expressions must therefore begin with shift relations and end with reduce relations. This is in order to keep the syntactic and referential semantic expansions synchronized.

These hierarchic regular expressions are defined to resemble expansion rules in a context free grammar (CFG). However, unlike CFGs, HHMMs have memory limits on nesting, in the form of a maximum depth D beyond which no expansion may take place. As a result, the expressive power of an HHMM is restricted to the set of regular languages, whereas CFGs may recognize the set of context-free languages; and HHMM recognition is worst-case linear on the length of an utterance, whereas CFG recognition is cubic.[5] Similar limits have been proposed on syntax in natural languages, motivated by limits on short term memory observed in humans (Miller and Chomsky 1963; Pulman 1986). These have been applied to obtain memory-limited parsers (e.g., Marcus 1980), and depth-limited right-corner grammars that are equivalent to CFGs, except that they restrict the number of internally recursive expansions allowed in recognition (Schuler and Miller 2005).

4.4 Expressivity

The language model described herein defines referential semantics purely in terms of HHMM shift and reduce operations over referent entities, made from reified sets of individuals in some original world model. This section will show that this basic model is sufficiently expressive to represent many commonly occurring linguistic phenomena,

5 When expressed as a function of the size of the grammar, HHMM recognition is asymptotically exponential on D, whereas CFG recognition is cubic regardless of depth. In practice, however, exact inference using either formalism is impractical, so approximate inference is used instead (e.g., maintaining a beam at each time step or at each constituent span in CFG parsing).

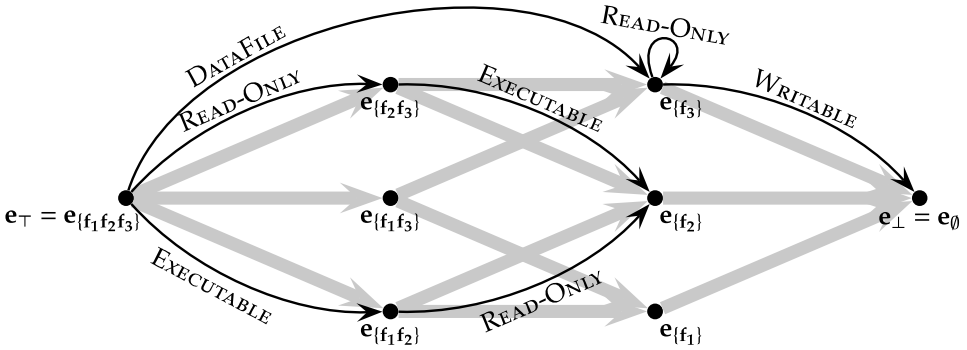

Figure 4
A subsumption lattice (laid on its side, in gray) over the power set of a domain containing three files: f_1 (a writable executable), f_2 (a read-only executable), and f_3 (a read-only data file). "Reference paths" made up of conjunctions of relations l (directed arcs, in black) traverse the lattice from left to right toward the empty set, as referents ($\mathbf{e}_{\{\ldots\}}$, corresponding to sets of files) are incrementally constrained by intersection with each $[\![l]\!]_{\mathcal{M}}$. (Some arcs are omitted for clarity.)

including intersective modifiers (e.g., adjectives like *executable*), multi-argument relations (e.g., prepositional phrases or relative clauses, involving trajector and landmark referents), negation (as in the adverb *not*), and comparatives over continuous properties (e.g., *larger*).

4.4.1 Properties. Properties (traditionally unary relations like EXECUTABLE or WRITABLE) can be represented in the world model as labeled edges l_t from supersets e_{t-1} to subsets e_t defined by intersecting the set e_{t-1} with the set $[\![l_t]\!]_{\mathcal{M}}$ satisfying the property l_t. Recall that a reified world model can be cast as a subsumption lattice as described in Section 3.1.2. The result of conjoining a property l with a context set e can therefore be found by downward traversal of an edge in this lattice labeled l and departing from e.[6]

Thus, in Figure 4, the set of *executables that are read-only* would be reachable by traversing a READ-ONLY relation from the set of executables, or by traversing an EXECUTABLE relation from the set of read-only objects, or by a composed path READ-ONLY∘EXECUTABLE or EXECUTABLE∘READ-ONLY from $\mathbf{e}_{\top}$. The resulting set may then serve as context for subsequent traversals. Property relations may also result in self-traversals (e.g., DATAFILE∘READ-ONLY in Figure 4) or traversals to the empty set $\mathbf{e}_{\perp}$ (e.g., DATAFILE∘WRITABLE). Property relations like EXECUTABLE can be defined using the dynamic relations in the first case of Equation (12) in Section 4.1, which simply ignore the non-context argument.

A general template for intersective nouns and modifiers can be expressed as a noun phrase (NP) expansion using the following regular expression (where $\boldsymbol{l}_{\sigma}$ and $\boldsymbol{l}_{\rho}$ indicate relation labels constraining referents at the beginning and end of the NP):

$$\text{NP} \rightarrow \text{Det } \big(\text{ Adj }\big)^{*} \text{ Noun } \big(\text{ PP} \,|\, \text{RC }\big)^{*}; \quad \text{with } \boldsymbol{l}_{\sigma} = \textsc{Identity} \text{ and } \boldsymbol{l}_{\rho} = \textsc{Identity} \quad (19)$$

6 Although properties (and later, n-ary relations) are defined in terms of an exponentially large subsumption lattice, this lattice need not be an actual data structure. If the world model is queried from a decoder trellis with a beam filter rather than from a complete search, only those lattice relations that are phonologically, syntactically, and semantically most likely (in other words, those that are on this beam) will be explored.

in which referents are successively constrained by the semantics of relations associated with adjective and noun expansions:

$$\text{Adj} \rightarrow \text{executable}; \quad \text{with } \boldsymbol{l}_\sigma = \textsc{Executable} \text{ and } \boldsymbol{l}_\rho = \textsc{Identity} \tag{20}$$

$$\text{Noun} \rightarrow \text{executable}; \quad \text{with } \boldsymbol{l}_\sigma = \textsc{Executable} \text{ and } \boldsymbol{l}_\rho = \textsc{Identity} \tag{21}$$

(and are also constrained by the prepositional phrase (PP) and relative clause (RC) modifiers, as described below). Here the relation EXECUTABLE traverses from referent $\mathbf{e}_{\{\mathbf{f}_1\mathbf{f}_2\mathbf{f}_3\}}$ to referent $\mathbf{e}_{\{\mathbf{f}_1\mathbf{f}_2\}}$, a subset of $\mathbf{e}_{\{\mathbf{f}_1\mathbf{f}_2\mathbf{f}_3\}}$ satisfying $[\![\textsc{Executable}^*]\!]_{\mathcal{M}^*}$.

4.4.2 n-ary Relations. Sequences of properties (traditionally unary relations) can be interpreted as simple nonbranching paths from referent to referent in a subsumption lattice, but higher-arity relations define more complex paths that fork and rejoin. For example, the referent of *the directory containing the executable* in Figure 5 would be reachable only by:

1. storing the original set of directories $\mathbf{e}_{\{\mathbf{d}_1\mathbf{d}_2\mathbf{d}_3\}}$ as a top-level referent in the HHMM hierarchy, then
2. traversing a CONTAIN relation departing $\mathbf{e}_{\{\mathbf{d}_1\mathbf{d}_2\mathbf{d}_3\}}$ to obtain the contents of those directories $\mathbf{e}_{\{\mathbf{f}_2\mathbf{f}_3\}}$, then
3. traversing an EXECUTABLE relation departing $\mathbf{e}_{\{\mathbf{f}_2\mathbf{f}_3\}}$ to constrain this set to the set of contents that are also executable: $\mathbf{e}_{\{\mathbf{f}_2\}}$, then
4. traversing the inverse CONTAIN′ of relation CONTAIN to obtain the containers of these executables, then constraining the original set of directories $\mathbf{e}_{\{\mathbf{d}_1\mathbf{d}_2\mathbf{d}_3\}}$ by intersection with this resulting set to yield the directories containing executables: $\mathbf{e}_{\{\mathbf{d}_2\}}$.

This 'forking' of referential semantic paths is handled via syntactic recursion: one path is explored by the recognizer while the other waits on the HHMM hierarchy (essentially

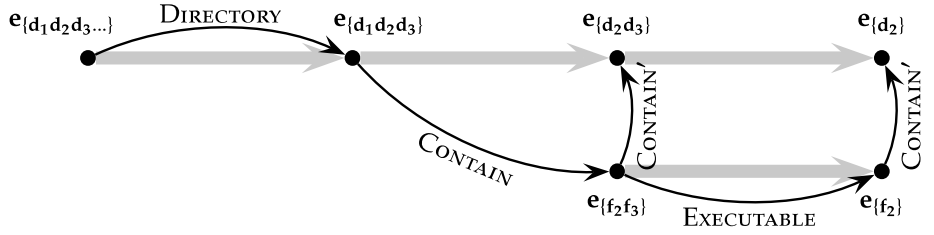

Figure 5
Reference paths for a relation *containing* in *the directory containing the executable file.* A reference path forks to specify referents using a two-place relation CONTAIN in a domain of directories d_1, d_2, d_3 and files f_1, f_2, f_3. Here, d_2 contains f_2 and d_3 contains f_3, and f_1 and f_2 are executable. The ellipsis in the referent set indicates the presence of additional individuals that are not directories. Again, subsumption is represented in gray and relations are represented in black. (Portions of the complete subsumption lattice and relation graph are omitted for clarity.)

functioning as a stack). A sample template for branching reduced relative clauses (or prepositional phrases) that exhibit this forking behavior can be expressed as below:

$$\text{RC} \rightarrow \text{containing NP}; \quad \text{with } \boldsymbol{l}_\sigma = \textsc{Contain} \text{ and } \boldsymbol{l}_\rho = \textsc{Contain}' \tag{22}$$

where the inverse relation CONTAIN′ is applied when the NP expansion concludes or reduces (when the forked paths are re-joined). Relations like CONTAIN are covered in the second case of Equation (12) in Section 4.1, which define transitions from sets of individuals associated with one argument of an original relation CONTAIN* to sets of individuals associated with the other argument of this relation, in the presence of a context set, which is a superset of the destination. The calculation of semantic transition probabilities for n-ary relations thus resembles that for properties, except that the probability term associated with the relation l_σ and the inverse relation l_ρ would depend on both context and argument referents (to its left and below it, in the HHMM hierarchy).

Note that there is ultimately a singleton referent $\{f_2\}$ of *the executable file* in Figure 5, even though there are two executable files in the world model used in these examples. This illustrates an important advantage of a dynamic context-dependent (three referent) model of semantic composition over the strict compositional (two referent) model. In a dynamic context model, *the executable file* is interpreted in the context of the files that are contained in a directory. In a strict compositional model, *the executable file* is interpreted only in the context of fixed constraints covering the entire utterance, and the constraints related to the relation *containing* are applied only to the directories. This means that a generative model based on strict composition will assign some probability to an infinitely recursive description *the directories containing executables contained by directories* ... In generation systems, this problem has been addressed by adding machinery to keep track of redundancy (Dale and Haddock 1991). But in this framework, a description model ($\Theta_{\text{Ref-Init}}$) which is sensitive to the sizes of its source referent and destination referent at the end of each departing labeled transition will be able to disprefer referential transitions that attempt to constrain already singleton referents, or that provide only trivial or vacuous (redundant) constraints in general. This solution is therefore more in line with graph-based models of generation (Krahmer, van Erk, and Verleg 2003), except that the graphs proposed here are over reified sets rather than individuals, and the goal is a generative probability model of language rather than generation per se.

4.4.3 Negation. Negation can be modeled in this framework as a relation between sets. Although it does not require any syntactic memory, negation does require referential semantic memory, in that the complement of a specified set must be intersected with some initial context set. *Files that are not writable* must still be files after all; only the *writable* portion of this description should be negated.

A regular expression for negation of adjectives is

$$\text{Adj} \rightarrow \text{not Adj}; \quad \text{with } \boldsymbol{l}_\sigma = \textsc{Identity} \text{ and } \boldsymbol{l}_\rho = \textsc{Not} \tag{23}$$

and is applied to a world model in Figure 6. Relations like NOT are covered in the third case of Equation (12) in Section 4.1, which define transitions between sets in an original relation NOT*.

4.4.4 Comparatives, Superlatives, and Subsective Modifiers. Comparatives (e.g., *larger*), superlatives (e.g., *largest*), and subsective modifiers (e.g., *large,* relative to some context

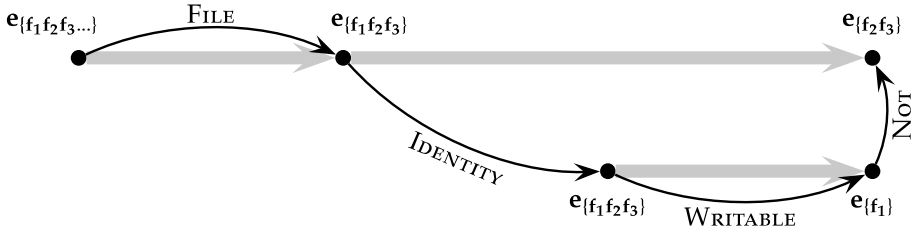

Figure 6
Reference paths for negation in *files that are not writable*, using a world model with files f_1, f_2, and f_3 of which only f_1 is writable. The recognizer first forks a copy of the set of files $\{f_1,f_2,f_3\}$ using the relation IDENTITY, then applies the adjective relation WRITABLE to yield $\{f_1\}$. The complement of this set $\{f_2,f_3,\dots\}$ is then intersected with the stored top-level referent set $\{f_1,f_2,f_3\}$ to produce the set of files that are not writable: $\{f_2,f_3\}$. Ellipses in referent sets indicate the presence of additional individuals that are not files.

set) define relations from sets to sets, or from sets to individuals (singleton sets). They can be handled in much the same way as negation. Here the context is provided from previous words and from sub-structure, in contrast to DeVault and Stone (2003), which define the context of a comparative either from fixed inter-utterance constraints or as the referent of the portion of the noun phrase dominated by the comparative (in addition to inter-utterance constraints). One advantage of dynamic (time-order) constraints is that implicit comparatives (*in the Clark directory, select the file that is larger,* with no complement) can be modeled with no additional machinery. If substructure context is not needed, then no additional HHMM storage is necessary.

A regular expression for superlative adjectives is

$$\text{Noun} \rightarrow \text{largest}\ \ \text{Noun}; \quad \text{with}\ \boldsymbol{l}_\sigma = \textsc{Identity}\ \text{and}\ \boldsymbol{l}_\rho = \textsc{Largest} \tag{24}$$

and is applied to a world model in Figure 7. Relations like LARGEST are also covered in the third case of Equation (12), which defines transitions between sets in an original relation LARGEST*.

5. Evaluation in a Spoken Language Interface

Much of the motivation for this approach has been to develop a human-like model of language processing. But there are practical advantages to this approach as well. One of the main practical advantages of the referential semantic language model described

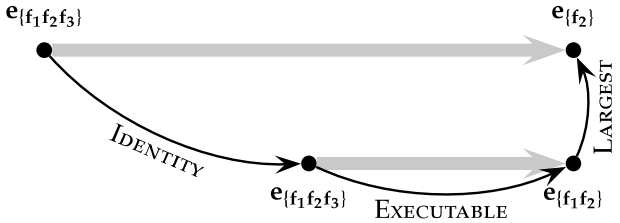

Figure 7
Reference paths for a comparative in *the largest executable*; this forks a copy of the referent set $\{f_1,f_2,f_3\}$ using the relation IDENTITY, applies EXECUTABLE to the forked set to obtain $\{f_1,f_2\}$, and returns the referent $\{f_2\}$ with the largest file size using LARGEST.

in this article is that it may allow spoken language interfaces to be applied to content-creation domains that are substantially developed by individual users themselves. Such domains may include scheduling or reminder systems (organizing items containing idiosyncratic person or event names, added by the user), shopping lists (containing idiosyncratic brand names, added by the user), interactive design tools (containing new objects designed and named by the user), or programming interfaces for home or small business automation (containing new actions, defined by the user). Indeed, computers are frequently used for content creation as well as content browsing; there is every reason to expect that spoken language interfaces will be used this way as well.

But the critical problem of applying spoken language interfaces to these kinds of content-creation domains is that the vocabulary of possible proper names that users may add or invent is vast. Interface vocabularies in such domains must allow new words to be created, and once they are created, these new words must be incorporated into the recognizer immediately, so that they can be used in the current context. The standard tactic of training language models on example sentences prior to use is not practical in such domains—except for relatively skeletal abstractions, example sentences will often not be available. Even very large corpora gleaned from Internet documents are unlikely to provide reliable statistics for users' made-up names with contextually appropriate usage, as a referential semantic language model provides.

Content-creation applications such as this may have considerable practical value as a means of improving accessibility to computers for disabled users. These domains also provide an ideal proving ground for a referential semantic language model, because directives in these domains mostly refer to a world model that is shared by the user and the interfaced application, and because the idiosyncratic language used in such domains makes it more resistant to domain-independent corpus training than other domains. In contrast, domains such as database query (e.g., of airline reservations), dictation, or information extraction are less likely to benefit from a referential semantic language model, because the world model in such domains is not shared by either the speaker (in database query) or by the interfaced application (in dictation or information extraction),[7] or because these domains are relatively fixed, so the expense of maintaining linguistic training corpora in these domains can often be justified.

This section will describe an evaluation of an implementation of the referential semantic language model as a spoken language interface in a very basic content-creation domain: that of a file organizer, similar to a Unix shell.[8] The performance of the model on this domain will be evaluated in large environments containing thousands of entities; more than will fit on the beam used in the Viterbi decoding search in this implementation.

The experiments described in Sections 5.1 through 5.8 were conducted to investigate the effect on recognition time and accuracy of using a referential semantic language model to recognize common types of queries, generated by an experimenter and read by several speakers. A thorough evaluation of the possible coverage of this kind of system on spontaneous input (e.g., in usability experiments) would require a rich syntactic representation and attention to disfluencies and speech repairs which are beyond the scope of this article (see Section 6).

7 Techniques based on abductive reasoning may mitigate this problem of incomplete model sharing (Hobbs et al. 1993), but this would require considerable extensions to the proposed model, and is beyond the scope of this article.

8 This is also similar to a spoken language version of Wilensky's Unix consultant (Wilensky, Arens, and Chin 1984).

5.1 Ontology Navigation Test Domain

To evaluate the contribution to recognition accuracy of referential semantics over that of syntax and phonology alone, a baseline (syntax only) and test (baseline plus referential semantics) recognizer were run on sample ontology manipulation directives in a "student activities" domain. This domain has the form of a simple tree-like taxonomy, with some cross-listings (for example, students may be listed in homerooms and in activities).

Taxonomic ontologies (e.g., for organizing biological classifications or computer file directories) can be mapped to reified world models of the sort described in Section 3.1.2. Concepts C in such an ontology define sets of individuals described by that concept: $\{\iota|C(\iota)\}$. Subconcepts C' of a concept C then define subsets of individuals: $\{\iota|C'(\iota)\} \subseteq \{\iota|C(\iota)\}$. These sets and subsets can be reified as referent entities and arranged on a subsumption lattice as described in Section 3.1.2. A sample taxonomic ontology is shown in Figure 8a (tilted on its side to match the subsumption lattices shown elsewhere in this article). Thus defined, such ontologies can be navigated using referent transitions described in Section 4.1 by entering concept referents via "downward" (rightward in the figure) transitions, and leaving concept referents via "upward" (leftward) transitions. For example, this ontology can be manipulated using directives such as:

(1) set Crookston campus homeroom two Clark to sports football captain

which are incrementally interpreted by transitioning down the subsumption lattice (e.g., from *sports* to *football* to *captain*) or forking to another part of the lattice (e.g., from *Clark* to *sports*).

As an ontology like this is navigated in spoken language, there is a sense in which other referents e' at the same level of the ontology as the most recently described referent e, or at higher levels of the ontology than the most recently described entity, should be semantically accessible without restating the ontological context (the path from the root concept $\mathbf{e}_\top$) shared by e' and e. Thus, in the context of having recently referred to someone in Homeroom 2 at a particular campus in a school activities database, other students in the same homeroom or other activities at the same campus should be accessible without giving an explicit *back up* directive at each branch in the ontology. To see the value of implicit upward transitions, compare Example (1) to a directive that makes upward transitions explicit using the keyword *back* (similar to '..' in the syntax of Unix paths) to exit the *homeroom two* and *Clark* folders:

(2) set Crookston campus homeroom two Clark to back back sports football captain

or if starting from the Duluth campus sports football directory:

(3) set back back back Crookston campus homeroom two Clark to back back sports football captain

Instead of requiring explicit *back* keywords, these upward transitions can be implicitly composed with downward transitions, resulting in transitions from source $\mathbf{e}_{\mathbf{S}_1}$ to destination $\mathbf{e}_\mathbf{S}$ via some ancestor $\mathbf{e}_{\mathbf{S}_0}$:

$$[\![\text{UP-}l]\!]_\mathcal{M}(\mathbf{e}_{\mathbf{S}_1}, \mathbf{e}_{\mathbf{S}_2}) = \mathbf{e}_\mathbf{S} \text{ s.t. } \exists \mathbf{e}_{\mathbf{S}_0}\ S_0 \supseteq S_1,\ S_0 \supseteq S,\ [\![l]\!]_\mathcal{M}(\mathbf{e}_{\mathbf{S}_0}, \mathbf{e}_\top) = \mathbf{e}_\mathbf{S} \tag{25}$$

The composed transition function finds a referent $\mathbf{e}_{\mathbf{S}_0}$ which subsumes both $\mathbf{e}_{\mathbf{S}_1}$ and $\mathbf{e}_\mathbf{S}$, then finds an ordinary (downward) transition l connecting $\mathbf{e}_{\mathbf{S}_0}$ to $\mathbf{e}_\mathbf{S}$. The result is a UP-l transition to every immediate child of an ancestor a referent (or in genealogical terms,

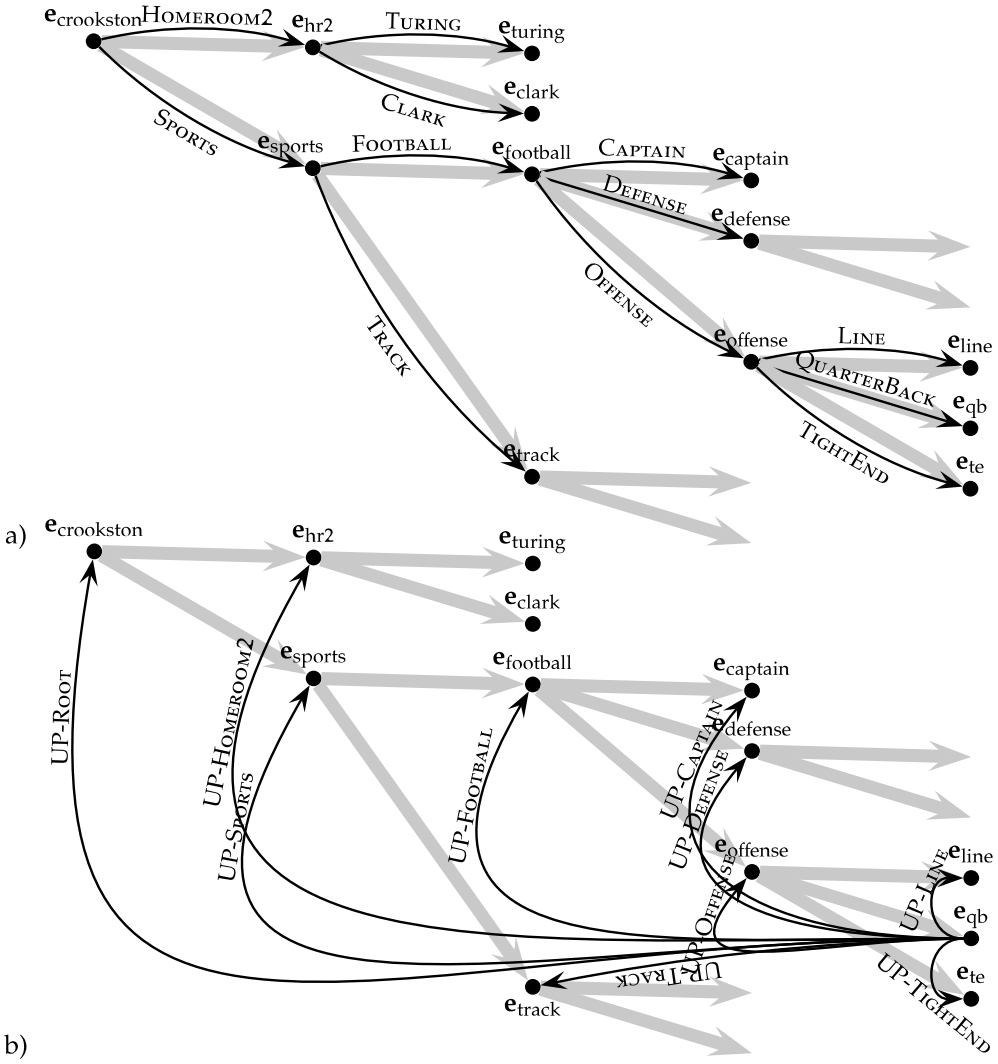

Figure 8
Upward and downward transitions in a sample student activities world model. Downward transitions (a) define basic sub-type relations. Upward transitions (b) relate sibling, ancestor, and (great-great-...-)aunt/uncle concepts. The entire model is reachable from any given referent via these two kinds of transitions.

to every sibling, ancestor, and sibling of ancestor), making these contextually salient concepts immediately accessible without explicit back-stepping (see Figure 8b).

Downward transitions are ordinary properties, as defined in the first case of Equation (12) in Section 4.1.

5.2 Scaling to Richer Domains

Although navigation in this domain is constrained to tree-like graphs, this domain tests all of the features of a referential semantic language model that would be required

in richer domains. As described in Section 4, rich domains (in particular, first-order domains, in which users can describe sets of individuals as referents) are mapped to transition edges on a simple graph, similar to the tree-like graphs used in this ontology. In first-order domains, the size of this graph may be exponential on the number of individuals in the world model. But once the number of referents exceeds the size of the decoder beam, the time performance of the recognizer is constrained not by the number of entities in the world model, but by the beam width and the number of outgoing relations (labels) that can be traversed from each hypothesis. In a first-order system, just as in the simple ontology navigation system evaluated here, this number of relations is constrained to the set of words defined by the user up to that point. In both cases, although the interface may be used to describe any one of an arbitrarily large set of referents, the number of referents that can be evoked *at the next time step* is bounded by a constant.

When this model is extended to first-order or continuous domains, the time required to calculate sets of individuals or hypothetical planner states that result from a transition may be nontrivial, because it may not be possible in such domains to retain the entire referent transition model in memory. In first-order domains, for example, this may require evaluating certain binary relations over all pairs of individuals in the world model, with time complexity proportional to the square of the size of the world model domain. Fortunately the model described herein, like most generative language models, hypothesizes words before recognizing them. This means a recognizer based on this model will be able to compute transitions that might follow a hypothesized word during the time that word is being recognized. If just the current set of possible transitions is known (say, these have already been pre-fetched into a cache), the set of outgoing transitions that will be required at some time *following* one of these current transitions can be requested as soon as the *beginning* of this transition is hypothesized—as soon as any word associated with this transition makes its way onto the decoder beam. From this point, the recognizer will have the entire duration of the word to compute (in parallel, in a separate thread, or on a separate server) the set of outgoing transitions that may follow this word. In other words, the model described herein may be scaled to richer domains because it is amenable to parallelization.

5.3 World Model

The student activities ontology used in this evaluation is a taxonomic world model defined with upward and downward transitions as described in Section 5.1. It organizes extracurricular activities under subcategories (e.g., offense $\subset$ football $\subset$ sports), and organizes students into homerooms, in which context they can be identified by a single (first or last) name. Every student or activity is an entity e in the set of entities $\mathcal{E}$, and relations l are subcategory labels or student names.

5.3.1 World Model $\mathcal{M}_{240}$. In the original student activities world model $\mathcal{M}_{240}$, a total of 240 entities were created in $\mathcal{E}$: 158 concepts (groups or positions) and 82 instances (students), each connected via a labeled arc from a parent concept.

Because a world model in this framework is a weighted set of labeled arcs, it is possible to calculate a meaningful perplexity statistic for transitions in this model, assuming all referents are equally likely to be a source. The perplexity of this world model (the average number of departing arcs) is 16.79, after inserting "UP" arcs as described in Section 5.1.

5.3.2 World Model $\mathcal{M}_{4175}$. An expanded version of the students ontology, $\mathcal{M}_{4175}$, includes 4,175 entities from 717 concepts and 3,458 instances. This model contains $\mathcal{M}_{240}$ as a subgraph, so that the same directives may be used in either domain; but it expands $\mathcal{M}_{240}$ from above, with additional campuses and schools, and below, with additional students in each class. The perplexity of this world model was 37.77, after inserting "UP" arcs as described in Section 5.1.

5.4 Test Corpus

A corpus of 144 test sentences (no training sentences) was collected from seven native English speakers (5 male, 2 female), who were asked to make specific edits to the student activities ontology described previously. The subjects were all graduate students and native speakers of English, from various parts of the United States. The edit directives were recorded as isolated utterances, not as part of an interactive dialogue, and the target concepts were identified by name in written prompts, so the corpus has much of the character of read speech. The average sentence length in this collection is 7.17 words.

5.5 Acoustic Model

Baseline and test versions of this system were run using a Recurrent Neural Network (RNN) acoustic model (Robinson 1994). This acoustic model performs competitively with multi-state triphone models based on multivariate Gaussian mixtures, but has the advantage of using only uniphones with single subphone states. As a result, less of the HMM trellis beam is occupied with subphone variations, so that a larger number of semantically distinct hypotheses may be considered at each frame.

Each model was evaluated using parameters trained from the TIMIT corpus of read speech (Fisher et al. 1987). This corpus yields several thousand examples for each of the relatively small set of single-state uniphones used in the RNN model. Read speech is also appropriate training data for this evaluation, because the test subjects are constrained to perform fixed edit tasks given written prompts, and the number of reasonable ways to perform these tasks is limited by the ontology, so hesitations and disfluencies are relatively rare.

5.6 Phone and Subphone Models

The language model used in these experiments is decomposed into five hierarchic levels, each with referent e and ordinary FSA state q components, as described in Section 4.2. The top three levels of this model represent syntactic states as q (derived from regular expressions defined in Section 4.3) and associated semantic referents as e. The bottom two levels represent pronunciation and subphone states as q, and ignore e.

Transitions across pronunciation states are defined in terms of sequences of phones associated with a word via a pronunciation model. The pronunciation model used in these experiments is taken from the CMU ARPABET dictionary (Weide 1998). Transitions across subphone states are defined in terms of sequences of subphones associated with a phone. Because this evaluation used an acoustic model trained on the TIMIT corpus (Fisher et al. 1987), the TIMIT phone set was used as subphones. In most cases, these subphones map directly to ARPABET phones, so each subphone HMM consists of a single, final state; but in cases of plosive phones (*B*, *D*, *G*, *K*, *P*, and *T*), the subphone HMM consists of a stop subphone (e.g., *bcl*) followed by a burst subphone (e.g., *b*).

Referents are ignored in both the phone and subphone models, and therefore do not need to be calculated.

State transitions within the phone level $\mathsf{P}_{\Theta_{Pron\text{-}Trans}}(q^4_{\sigma,t} \mid q^4_{\sigma,t-1})$ deterministically advance along a sequence of phones in a pronunciation; and initial phone sequences depend on words in higher-level syntactic states $q^3_{\sigma,t}$, via a pronunciation model $\Theta_{Pron\text{-}Init}$:

$$\mathsf{P}_{\Theta_{\text{RSLM-}\sigma}}(\sigma^4_t \mid \rho^5_t\ \rho^4_t\ \sigma^4_{t-1}\ \sigma^3_t) \stackrel{\text{def}}{=} \begin{cases} \text{if } f^5_{\rho,t}=\mathbf{0},\ f^4_{\rho,t}=\mathbf{0}: & [q^4_{\sigma,t}=q^4_{\sigma,t-1}] \\ \text{if } f^5_{\rho,t}=\mathbf{1},\ f^4_{\rho,t}=\mathbf{0}: & \mathsf{P}_{\Theta_{Pron\text{-}Trans}}(q^4_{\sigma,t} \mid q^4_{\sigma,t-1}) \\ \text{if } f^5_{\rho,t}=\mathbf{1},\ f^4_{\rho,t}=\mathbf{1}: & \mathsf{P}_{\Theta_{Pron\text{-}Init}}(q^4_{\sigma,t} \mid q^3_{\sigma,t}) \end{cases} \tag{26}$$

The student activities domain was developed with no synonymy—only one word describes each semantic relation. Alternate pronunciations are modeled using a uniform distribution over all listed pronunciations.

Initialization and transition of subphone sequences depend on the phone at the current time step and the subphone at the previous time step. This model was trained directly using relative frequency estimation on the TIMIT corpus itself:

$$\mathsf{P}_{\Theta_{\text{RSLM-}\sigma}}(\sigma^5_t \mid \rho^6_t\ \rho^5_t\ \sigma^5_{t-1}\ \sigma^4_t) \stackrel{\text{def}}{=} \tilde{\mathsf{P}}(q^5_{\sigma,t} \mid q^4_{\sigma,t}\ q^5_{\sigma,t-1}) \tag{27}$$

5.7 Syntax and Reference Models

The three upper levels of the HHMM comprise the syntactic and referential portion of the language model. Concept error rate tests were performed on three baseline and test versions of this portion of the language model, using the same acoustic, phone, and subphone models, as described in Sections 5.5 and 5.6.

5.7.1 Language Model $\Theta_{LM\text{-}Sem}$. First, the syntactic and referential portion of the language model was implemented as described in Section 4.2. A subset of the regular expression grammar appears in Figure 9. Any nondeterminism resulting from disjunction or Kleene-star repetition in the regular expressions was handled in $\Theta_{\text{Syn-Trans}}$ using uniform distributions over all available following states. Distributions over regular expression expansions in $\Theta_{\text{Syn-Init}}$ were uniform over all available expansions. Distributions over labels in $\Theta_{\text{Ref-Init}}$ were also uniform over all labels departing the entity referent condition that were compatible with the FSA state category generated by $\Theta_{\text{Syn-Init}}$.

S → set PNpath to PNpath;	$\boldsymbol{l}_\rho$ = SETTO
PNpath → PNup (PN)*	
PNup → homeroom two;	$\boldsymbol{l}_\sigma$ = UP-HOMEROOM2
PN → clark;	$\boldsymbol{l}_\sigma$ = CLARK
PNup → sports;	$\boldsymbol{l}_\sigma$ = UP-SPORTS
PN → football;	$\boldsymbol{l}_\sigma$ = FOOTBALL
PN → captain;	$\boldsymbol{l}_\sigma$ = CAPTAIN

Figure 9
Sample grammar for student activities domain. Relations $\boldsymbol{l}_\sigma, \boldsymbol{l}_\rho$ = IDENTITY unless otherwise specified.

5.7.2 Language Model $\Theta_{LM\text{-}NoSem}$. Second, in order to evaluate the contribution of referential semantics to recognition, a baseline version of the model was tested with all relations defined to be equivalent to NIL, returning $\mathbf{e}_\top$ at each depth and time step, with all relation labels reachable in $\mathcal{M}$ from $\mathbf{e}_\top$. This has the effect of eliminating all semantic constraints from the recognizer, while preserving the relation labels of the original model as a resource from which to calculate concept error rate. The decoding equations and grammar in Model $\Theta_{\text{LM-NoSem}}$ are therefore the same as in Model $\Theta_{\text{LM-Sem}}$; only the domain of possible referents is restricted.

Again, distributions over state transitions, expansions, and outgoing labels in $\Theta_{\text{Syn-Trans}}$, $\Theta_{\text{Syn-Init}}$, and $\Theta_{\text{Ref-Init}}$ are uniform over all available options.

5.7.3 Language Model $\Theta_{LM\text{-}Trigram}$. Finally, the referential semantic language model (Language Model $\Theta_{\text{LM-Sem}}$) was compiled into a word trigram model, in order to test how well the model would function as a pre-process to a conventional trigram-based speech recognizer. This was done by iterating over all possible sequences of hidden state transitions starting from every possible configuration of referents and FSA states on a stack of depth D (where $D = 3$):

$$h_t = \langle w_{t-1}, w_t \rangle \tag{28}$$

$$\mathsf{P}(h_t \mid h_{t-1}) = \mathsf{P}(w_{t-1}\ w_t \mid w_{t-2}\ w_{t-1}) = \mathsf{P}(w_t \mid w_{t-2}\ w_{t-1}) \tag{29}$$

$$\begin{aligned} &\stackrel{\text{def}}{=} \sum_{\sigma_{t-2..t}} \sum_{w_{t-2}, w_{t-1}} \mathsf{P}_{\Theta_{\text{Uniform}}}(\sigma_{t-2}) \cdot [w_{t-2} = W(q^{1..D}_{\sigma,t-2})] \\ &\qquad \cdot \mathsf{P}_{\Theta_{\text{LM-Sem}}}(\sigma_{t-1} \mid \sigma_{t-2}) \cdot [w_{t-1} = W(q^{1..D}_{\sigma,t-1})] \\ &\qquad \cdot \mathsf{P}_{\Theta_{\text{LM-Sem}}}(\sigma_t \mid \sigma_{t-1}) \cdot [w_t = W(q^{1..D}_{\sigma,t})] \end{aligned} \tag{30}$$

First, every valid combination of syntactic categories was calculated in a depth-first search using $\Theta_{\text{LM-NoSem}}$. Then every combination of three referents from $\mathcal{M}_{240}$ was hypothesized as a possible referent configuration. A complete set of possible initial values for σ_{t-2} was then filled with combinations from the set of syntactic category configuration crossed with the set of referent configurations. From each possible σ_{t-2}, $\Theta_{\text{LM-Sem}}$ was consulted to give a distribution over σ_{t-1} (assuming a word-level transition occurs, with $f^4_{\rho,t-1} = \mathbf{1}$), and then again from each possible configuration of σ_{t-1} to give a distribution over σ_t (again assuming a word-level transition). The product of these transition probabilities was then calculated and added to a trigram count, based on the words w_{t-2}, w_{t-1}, and w_t occurring in σ_{t-2}, σ_{t-1}, and σ_t. These trigram counts were then normalized over w_{t-2} and w_{t-1} to give $\mathsf{P}(w_t \mid w_{t-2}\ w_{t-1})$.

5.8 Results

The following results report Concept Error Rate (CER), as the sum of the percentages of insertions, deletions, and substitutions required to transform the most likely sequence of relation labels hypothesized by the system into the hand-annotated transcript, expressed as a percentage of the total number of labels in the hand-annotated transcript. Because there are few semantically unconstrained function words in this domain, this is essentially word error rate, with a few multi-word labels (e.g., *first chair, homeroom two*) concatenated together.

5.8.1 Language Model $\Theta_{LM\text{-}Sem}$ and World Model $\mathcal{M}_{240}$. Results using Language Model $\Theta_{\text{LM-Sem}}$ with the 240-entity world model ($\mathcal{M}_{240}$) show an overall 17.1% CER (Table 2).

Table 2
Per-subject results for Language Model $\Theta_{\text{LM-Sem}}$ with $\mathcal{M}_{240}$.

subject	% correct	% substitute	% delete	% insert	CER %
0	83.8	14.1	2.1	2.8	19.0
1	73.2	20.3	6.5	5.8	32.7
2	90.2	7.8	2.0	0.7	10.5
3	88.1	9.3	2.7	0.7	12.6
4	88.4	10.3	1.4	3.4	15.1
5	90.8	8.5	0.7	7.0	16.2
6	90.6	8.6	0.7	3.6	12.9
all	86.4	11.3	2.3	3.4	17.1

Table 3
Per-subject results for Language Model $\Theta_{\text{LM-Sem}}$ with $\mathcal{M}_{4175}$.

subject	% correct	% substitute	% delete	% insert	CER %
0	85.2	14.1	0.7	2.1	16.9
1	70.6	25.5	3.9	7.2	36.6
2	86.9	9.2	3.9	3.9	17.0
3	86.8	11.3	2.0	2.0	15.2
4	83.6	14.4	2.1	6.9	23.3
5	89.4	9.9	0.7	3.5	14.1
6	89.9	9.4	0.7	5.0	15.1
all	84.5	13.5	2.1	4.4	19.9

Here the size of the vocabulary was roughly equal to the number of referents in the world model. The sentence error rate for this experiment was 59.44%.

5.8.2 Language Model $\Theta_{LM\text{-}Sem}$ and World Model $\mathcal{M}_{4175}$. With the number of entities (and words) increased to 4,175 ($\mathcal{M}_{4175}$), the CER increases slightly to 19.9% (Table 3). Here again, the size of the vocabulary was roughly equal to the number of referents in the world model. The sentence error rate for this experiment was 62.24%. Here, the use of a world model (Language Model $\Theta_{\text{LM-Sem}}$) with no linguistic training data is comparable to that reported for other large-vocabulary systems (Seneff et al. 2004; Lemon and Gruenstein 2004), which were trained on sample sentences.

5.8.3 Language Model $\Theta_{LM\text{-}NoSem}$ with no World Model. In comparison, a baseline using only the grammar and vocabulary from the students domain $\mathcal{M}_{240}$ without any world model information and no linguistic training data (Language Model $\Theta_{\text{LM-NoSem}}$) scores 43.5% (Table 4).[9] The sentence error rate for this experiment was 93.01%.

Ignoring the world model significantly raises error rates compared to Model $\Theta_{\text{LM-Sem}}$ ($p < 0.01$ using pairwise t-test against Language model $\Theta_{\text{LM-Sem}}$ with $\mathcal{M}_{240}$, grouping scores by subject), suggesting that syntactic constraints are poor predictors of

9 Ordinarily a syntactic model would be interpolated with word *n*-gram probabilities derived from corpus training, but in the absence of training sentences these statistics cannot be included.

Table 4
Per-subject results for Language Model $\Theta_{\text{LM-NoSem}}$.

subject	% correct	% substitute	% delete	% insert	CER %
0	57.0	35.9	7.0	12.7	55.6
1	49.0	41.2	9.8	13.7	64.7
2	71.9	18.3	9.8	6.5	34.6
3	69.5	26.5	4.0	9.3	39.7
4	67.8	28.8	3.4	13.7	45.9
5	79.6	19.0	1.4	7.0	27.5
6	75.5	22.3	2.2	10.8	35.3
all	67.1	27.5	5.5	10.5	43.5

concepts without considering reference. But this is not surprising: because the grammar by itself does not constrain the set of ontology labels that can be used to construct a path, the perplexity of this model is 240 (reflecting a uniform distribution over nearly the entire lexicon), whereas the perplexity of $\mathcal{M}_{240}$ is only 16.79.

5.8.4 Language Model $\Theta_{\text{LM-Trigram}}$ and World Model $\mathcal{M}_{240}$. In order to test how well the model would function as a pre-process to a conventional trigram-based speech recognizer, the referential semantic language model (Language Model $\Theta_{\text{LM-Sem}}$) was compiled into a word trigram model. This word trigram language model (Language Model $\Theta_{\text{LM-Trigram}}$), compiled from the referential semantic model (in the 240-entity domain), shows a concept error rate of 26.6% on the students experiment (Table 5). The sentence error rate for this experiment was 66.43%.

Using trigram context (Language Model $\Theta_{\text{LM-Trigram}}$) similarly shows statistically significant increases in error over Language Model $\Theta_{\text{LM-Sem}}$ with $\mathcal{M}_{240}$ ($p = 0.01$ using pairwise t-test, grouping scores by subject), showing that referential context is also more predictive than word n-grams derived from referential context. Moreover, the compilation to trigrams required to build Language Model $\Theta_{\text{LM-Trigram}}$ is expensive (requiring several hours of pre-processing) because it must consider all combinations of entities in the world model. This would make the pre-compiled model impractical in mutable domains.

Table 5
Per-subject results for Language Model $\Theta_{\text{LM-Trigram}}$ with $\mathcal{M}_{240}$.

subject	% correct	% substitute	% delete	% insert	CER %
0	76.1	19.0	4.9	5.6	29.6
1	56.9	24.8	18.3	12.4	44.4
2	81.7	9.2	9.2	0.0	18.3
3	83.4	13.9	2.7	2.0	18.5
4	79.5	13.0	7.5	11.0	31.5
5	86.6	10.6	2.8	0.7	14.1
6	83.5	14.4	2.2	0.7	17.3
all	78.1	15.0	6.9	4.7	26.6

Table 6
Experimental results with four model configurations.

experiment	correct	substitute	delete	insert	CER
$\Theta_{\text{LM-Sem}}, \mathcal{M}_{240}$	86.4	11.3	2.3	3.4	17.1
$\Theta_{\text{LM-Sem}}, \mathcal{M}_{4175}$	84.5	13.5	2.1	4.4	19.9
$\Theta_{\text{LM-NoSem}}$	67.1	27.5	5.5	10.5	43.5
$\Theta_{\text{LM-Trigram}}, \mathcal{M}_{240}$	78.1	15.0	6.9	4.7	26.6

5.8.5 Summary of Results. Results in Table 6 summarize the results of the four experiments.

Some of the erroneously hypothesized directives in this domain described implausible edits: for example, making one student a subset of another student. Domain information or meta-data could eliminate some of these kinds of errors, but in content-creation applications it is not always possible to provide this information in advance; and given the subtle nature of the effect of this information on recognition, it is not clear that users would want to manage it themselves, or allow it to be automatically induced without supervision.[10] In any case, the comparison described in this section to a non-semantic model $\Theta_{\text{LM-NoSem}}$ suggests that the world model by itself is able to apply useful constraints in the absence of domain knowledge. This suggests that, in an interpolated approach, direct world model information may relieve some of the burden on authored or induced domain knowledge to perform robustly, so that this domain knowledge may be authored more sparsely or induced more conservatively than it otherwise might.

All evaluations ran in real time on a 4-processor dual-core 2.6GHz server, with a beam width of 1,000 hypotheses per frame. Differences in runtime performance were minimal, even between the simple trigram model and HHMM-based referential semantic language models. This was due to two factors:

1. All recognizers were run with the same beam width. Although it might be possible to narrow the beam width to produce faster than real-time performance for some models, widening the beam beyond 1,000 did not return significant reductions in CER in the experiments described herein.
2. The implementation of the Viterbi decoder used in these experiments was optimized to skip combinations of joint variable values that would result in zero probability transitions (which is a reasonable optimization for any factored time-series model), significantly decreasing runtime for HHMM recognition.

5.8.6 Statistical Significance vs. Magnitude of Gain. The experiments described in this article show a statistically significant increase in accuracy due to the incorporation of referential semantic information into speech decoding. But these results should not be interpreted to demonstrate any particular *magnitude* of error reduction (as might be claimed for the introduction of head words into parsing models, for example).

10 Ehlen et al. (2008) provide an example of a user interface for managing imperfect automatically-induced information about task assignments from meeting transcripts, which is much more concrete than the kind of domain knowledge inference considered here.

First, this is because the acoustic model used in these experiments was trained on a relatively small corpus (6,000 utterances), which introduces the possibility that the acoustic model was under-trained. As a result, the error rates for both baseline and test systems may be greater here than if a larger training corpus had been used, so the performance gain due to the introduction of referential semantics may be overstated.

Second, these experiments were designed with relatively strong referential constraints (a tree-like ontology, with a perplexity of about 17 for $\mathcal{M}_{240}$) and relatively weak syntactic constraints (allowing virtually any sequence of relation labels, with a much higher perplexity of about 240), in order to highlight differences due to referential semantics. In general use, recognition accuracy gains due to the incorporation of referential semantic information will depend crucially on the relative perplexity of the referential constraints combined with syntactic constraints, compared to that of syntactic constraints alone. This paper has argued that in content-creation applications this difference can be manipulated and exploited—in fact, by reorganizing folders into a binary branching tree (with perplexity 2), a user could achieve nearly perfect speech recognition—but in applications involving fixed ontologies and purely hypothetical directives, as in database query applications, gains may be minimal or nonexistant.

6. Conclusion and Future Work

This article has described a referential semantic language model that achieves recognition accuracy favorably comparable to a pre-compiled trigram baseline in user-defined domains with no available domain-specific training corpora, through the use of explicit hypothesized semantic referents. This architecture requires that the interfaced application make available a queryable world model, but the combined phonological, syntactic, and referential semantic decoding process ensures the world model is only queried when necessary, allowing accurate real time performance even in large domains containing several thousand entities.

The framework described in this article is defined over first-order sets (of individuals), making transition functions over referents equivalent to expressions in first-order logic. This framework can be extended to model other kinds of references (e.g., to time intervals or events) by casting them as individuals (Hobbs 1985).

The system as defined herein also has some ability to recognize referents constrained by quantifiers: for example, *the directory containing two files.* Because its referents are reified sets, the system can naturally model relations that are sensitive to cardinality (self-transitioning if the set has N or greater individuals, transitioning to $\mathbf{e}_\perp$ otherwise). But a dynamic view of the referential semantics of nested quantifiers requires referents to be indexed to particular iterations of quantifiers at higher levels of nesting in the HHMM hierarchy (corresponding to higher-scoping quantifiers). Extending the system to dynamically interpret nested quantifiers therefore requires that all semantic operations preserve an "iteration context" of nested outer-quantified individuals for each inner-quantified individual. This is left for future work.

Some analyses of phenomena like intensional or non-inherent adjectives—for example, *toy* in *toy guns,* which are not actually guns; or *old* in *old friends,* who are not necessarily elderly (Peters and Peters 2000)—involve referents corresponding to second-order sets (this allows these adjectives to be composed before being applied to a noun: *old but casual friend*). Unfortunately, extending the framework described in this article to use a similarly explicit representation of second- or higher-order sets would be impractical. Not only would the number of possible second- or higher-order sets be exponentially larger than the number of possible first-order sets (which is already

exponential on the number of individuals), but the length of the description of each referent itself would be exponential on the number of individuals (whereas the list of individuals describing a first-order referent is merely linear).

The definition of semantic interpretation as a transition function does support interesting extensions to hypothetical reasoning and planning beyond the standard closed-world model-theoretic framework, however. Recall the sentence *go to the package data directory and hide the executable files,* or equivalently, *in the package data directory, hide the executable files,* exemplifying the continuous context-sensitivity of the referential semantic language model. Here, the system focuses on the contents of this directory because a sequence of transitions resulting from the combined phonological, syntactic, and referential semantic context of the sentence led it to this state. One may characterize the referential semantic transitions leading to this state as a hypothetical sequence of *change directory* actions moving the active directory of the interface to this directory (for the purpose of understanding the consequences of the first part of this directive). The hypothesized context of this directory is then a *world state* or *planning state* resulting from these actions. Thus characterized, the referential semantic decoder is performing a kind of statistical plan recognition (Blaylock and Allen 2005). By viewing referents as world states, or as having world-state components, it would then be possible to use logical conclusions of other types of actions as implicit constraints—e.g., *unpack the tar file and hide the executable [which will result from this unpacking]*—without adding extra functionality to the recognizer implementation. Similarly, referents for hypothetical objects like the noun phrase *a tar file* in the directive *create a tar file,* are not part of the world model when the user describes them.

Recognizing references to these hypothetical states and objects requires a capacity to dynamically generate referents not in the current world model. The domain of referents in this extended system is therefore unbounded. Fortunately, as mentioned in Section 5.2, the number of referents that can be generated *at each time step* is still bounded by a constant, equal to the recognizer's beam width multiplied by the number of traversable relation labels. This means that distributions over outgoing relation labels are still well-defined for each referential state. The only difference is that, when modeling hypothetical referents, these distributions must be calculated dynamically.

Finally, this article has primarily focused on connecting an explicit representation of referential semantics to speech recognition decisions. Ordinarily this is thought of as being mediated by syntax, which is covered in this article only through a relatively simple framework of bounded recursive HHMM state transitions. However, the bounded HHMM representation used in this paper has been applied (without semantics) to rich syntactic parsing as well, using a transformed grammar to minimize stack usage to cases of center-expansion (Schuler et al. 2008). Coverage experiments with this transformed grammar demonstrated that over 97% of the large syntactically annotated Penn Treebank (Marcus, Santorini, and Marcinkiewicz 1994) could be parsed using only three elements of stack memory, with four elements giving over 99% coverage. This suggests that the relatively tight bounds on recursion described in this paper might be expressively adequate if syntactic states are defined using this kind of transform.

This transform model (again, without semantics) was then further applied to parsing speech repairs, in which speakers repeat or edit mistakes in their directives: for example, *select the red, uh, the blue folder* (Miller and Schuler 2008). The resulting system models incomplete disfluent constituents using transitions associated with ordinary fluent speech until the repair point (the *uh* in the example), then processes the speech repair using only a small number of learned repair reductions. Coverage results for the same transform model on the Penn Treebank Switchboard Corpus of transcribed

spontaneous speech showed a similar three- to four-element memory requirement. If this HHMM speech repair model were combined with the HHMM model of referential semantics described in this article, referents associated with ultimately disfluent constituents could similarly be recognized using referential transitions associated with ordinary fluent speech until the repair point, then reduced using a repair rule that discards the referent. These results suggest that an HHMM-based semantic framework such as the one described in this article may be psycholinguistically plausible.

Acknowledgments

The authors would like to thank the anonymous reviewers for their input. This research was supported by National Science Foundation CAREER/PECASE award 0447685. The views expressed are not necessarily endorsed by the sponsors.

References

Aist, Gregory, James Allen, Ellen Campana, Carlos Gallo, Scott Stoness, Mary Swift, and Michael Tanenhaus. 2007. Incremental understanding in human–computer dialogue and experimental evidence for advantages over nonincremental methods. In *Proceedings of DECALOG*, pages 149–154, Trento.

Baker, James. 1975. The Dragon system: an overivew. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 23(1):24–29.

Bilmes, Jeff and Chris Bartels. 2005. Graphical model architectures for speech recognition. *IEEE Signal Processing Magazine*, 22(5):89–100.

Blaylock, Nate and James Allen. 2005. Recognizing instantiated goals using statistical methods. In *IJCAI Workshop on Modeling Others from Observations (MOO-2005)*, pages 79–86, Edinburgh.

Bos, Johan. 1996. Predicate logic unplugged. In *Proceedings of the 10th Amsterdam Colloquium*, pages 133–143, Amsterdam.

Brachman, Ronald J. and James G. Schmolze. 1985. An overview of the kl-one knowledge representation system. *Cognitive Science*, 9(2):171–216.

Brown-Schmidt, Sarah, Ellen Campana, and Michael K. Tanenhaus. 2002. Reference resolution in the wild: Online circumscription of referential domains in a natural interactive problem-solving task. In *Proceedings of the 24th Annual Meeting of the Cognitive Science Society*, pages 148–153, Fairfax, VA.

Church, Alonzo. 1940. A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5(2):56–68.

Dale, Robert and Nicholas Haddock. 1991. Content determination in the generation of referring expressions. *Computational Intelligence*, 7(4):252–265.

DeVault, David and Matthew Stone. 2003. Domain inference in incremental interpretation. In *Proceedings of ICoS*, pages 73–87, Nancy.

Ehlen, Patrick, Matthew Purver, John Niekrasz, Stanley Peters, and Kari Lee. 2008. Meeting adjourned: Off-line learning interfaces for automatic meeting understanding. In *Proceedings of the International Conference on Intelligent User Interfaces*, pages 276–284, Canary Islands.

Fisher, William M., Victor Zue, Jared Bernstein, and David S. Pallet. 1987. An acoustic–phonetic data base. *Journal of the Acoustical Society of America*, 81:S92–S93.

Frege, Gottlob. 1892. Uber sinn und bedeutung. *Zeitschrift fur Philosophie und Philosophischekritik*, 100:25–50.

Gorniak, Peter and Deb Roy. 2004. Grounded semantic composition for visual scenes. *Journal of Artificial Intelligence Research*, 21:429–470.

Groenendijk, Jeroen and Martin Stokhof. 1991. Dynamic predicate logic. *Linguistics and Philosophy*, 14:39–100.

Haddock, Nicholas. 1989. Computational models of incremental semantic interpretation. *Language and Cognitive Processes*, 4:337–368.

Hobbs, Jerry R. 1985. Ontological promiscuity. In *Proceedings of ACL*, pages 61–69, Chicago, IL.

Hobbs, Jerry R., Douglas E. Appelt, John Bear, David Israel, Megumi Kameyama, Mark Stickel, and Mabry Tyson. 1996. Fastus: A cascaded finite-state transducer for extracting information from natural-language text. In Yves Schabes, editor, *Finite State Devices for Natural Language Processing*. MIT Press, Cambridge, MA, pages 383–406.

Hobbs, Jerry R., Mark Stickel, Douglas E. Appelt, and Paul Martin. 1993. Interpretation as abduction. *Artificial Intelligence*, 63:69–142.

Jelinek, Frederick, Lalit R. Bahl, and Robert L. Mercer. 1975. Design of a linguistic

statistical decoder for the recognition of continuous speech. *IEEE Transactions on Information Theory*, 21:250–256.
Krahmer, Emiel, Sebastiaan van Erk, and Andre Verleg. 2003. Graph-based generation of referring expressions. *Computational Linguistics*, 29(1):53–72.
Lemon, Oliver and Alexander Gruenstein. 2004. Multithreaded context for robust conversational interfaces: Context-sensitive speech recognition and interpretation of corrective fragments. *ACM Transactions on Computer-Human Interaction*, 11(3):241–267.
Marcus, Mitch. 1980. *A Theory of Syntactic Recognition for Natural Language*. MIT Press, Cambridge, MA.
Marcus, Mitchell P., Beatrice Santorini, and Mary Ann Marcinkiewicz. 1994. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.
Martin, Charles and Christopher Riesbeck. 1986. Uniform parsing and inferencing for learning. In *Proceedings of AAAI*, pages 257–261, Philadelphia, PA.
Mellish, Chris. 1985. *Computer Interpretation of Natural Language Descriptions*. Wiley, New York.
Miller, George and Noam Chomsky. 1963. Finitary models of language users. In R. Luce, R. Bush, and E. Galanter, editors, *Handbook of Mathematical Psychology*, volume 2. John Wiley, New York, pages 419–491.
Miller, Tim and William Schuler. 2008. A unified syntactic model for parsing fluent and disfluent speech. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL '08)* pages 105–108, Columbus, OH.
Montague, Richard. 1973. The proper treatment of quantification in ordinary English. In J. Hintikka, J. M. E. Moravcsik, and P. Suppes, editors, *Approaches to Natural Language*. D. Riedel, Dordrecht, pages 221–242. Reprinted in R. H. Thomason ed., *Formal Philosophy*, Yale University Press, New Haven, CT, 1994.
Murphy, Kevin P. and Mark A. Paskin. 2001. Linear time inference in hierarchical HMMs. In *Proceedings of NIPS*, pages 833–840, Vancouver.
Peters, Ivonne and Wim Peters. 2000. The treatment of adjectives in simple: Theoretical observations. In *Proceedings of LREC*, paper # 366, Athens.
Pulman, Steve. 1986. Grammars, parsers and memory limitations. *Language and Cognitive Processes*, 1(3):197–225.
Robinson, Tony. 1994. An application of recurrent nets to phone probability estimation. In *IEEE Transactions on Neural Networks*, 5:298–305.
Seneff, Stephanie, Chao Wang, Lee Hetherington, and Grace Chung. 2004. A dynamic vocabulary spoken dialogue interface. In *Proceedings of ICSLP*, pages 1457–1460, Jeju Island.
Schuler, William. 2001. Computational properties of environment-based disambiguation. In *Proceedings of ACL*, pages 466–473, Toulouse.
Schuler, William, Samir AbdelRahman, Tim Miller, and Lane Schwartz. 2008. Toward a psycholinguistically-motivated model of language. In *Proceedings of COLING*, pages 785–792, Manchester, UK.
Schuler, William and Tim Miller. 2005. Integrating denotational meaning into a DBN language model. In *Proceedings of the 9th European Conference on Speech Communication and Technology / 6th Interspeech Event (Eurospeech/ Interspeech'05)*, pages 901–904, Lisbon.
Tanenhaus, Michael K., Michael J. Spivey-Knowlton, Kathy M. Eberhard, and Julie E. Sedivy. 1995. Integration of visual and linguistic information in spoken language comprehension. *Science*, 268:1632–1634.
Tarski, Alfred. 1933. *Prace Towarzystwa Naukowego Warszawskiego, Wydzial III Nauk Matematyczno-Fizycznych*, 34. Translated as 'The concept of truth in formalized languages', in J. Corcoran, editor, *Logic, Semantics, Metamathematics: Papers from 1923 to 1938*. Hackett Publishing Company, Indianapolis, IN, 1983, pages 152–278.
Weide, R. L. 1998. Carnegie Mellon University Pronouncing Dictionary v0.6d. Available at `www.speech.cs.cmu.edu/cgi-bin/cmudict`.
Wilensky, Robert, Yigal Arens, and David Chin. 1984. Talking to UNIX: An overview of UC. *Communications of the ACM*, 27(6):574–593.
Young, S. L., A. G. Hauptmann, W. H. Ward, E. T. Smith, and P. Werner. 1989. High level knowledge sources in usable speech recognition systems. *Communications of the ACM*, 32(2):183–194.