

Exploiting Referential Context in Spoken Language Interfaces for Data-Poor Domains*

Stephen Wu

Department of Computer
Science and Engineering
University of Minnesota
Minneapolis, MN
swu@cs.umn.edu

Lane Schwartz

Department of Computer
Science and Engineering
University of Minnesota
Minneapolis, MN
lschwar@cs.umn.edu

William Schuler

Department of Computer
Science and Engineering
University of Minnesota
Minneapolis, MN
schuler@cs.umn.edu

ABSTRACT

This paper describes an implementation of a shell-like programming interface that utilizes *referential context* (that is, information about the current state of an interfaced application) in order to achieve accurate recognition – even in user-defined domains with no available domain-specific training corpora. The interface incorporates a knowledge of context into its model of syntax, yielding a referential semantic language model. Interestingly, the referential semantic language model exploits context *dynamically*, unlike other recent systems, by using incremental processing and the limited stack memory of an HMM-like time series model.

INTRODUCTION

The development of general-purpose artificial assistants could have a transformative effect on society from early education to elder care. But to be useful, these assistants will need to communicate with the people they assist in the mutable and idiosyncratic language of day to day life, populated with proper names of co-workers, objects, and local events not found in broad corpora. The fundamental lack of appropriately detailed spoken language training corpora for interfaces to general-purpose assistants places this application beyond the reach of conventional corpus-based speech recognition strategies, which have been developed for applications with established linguistic conventions and plentiful corpora, e.g. dictating formal documents or performing specific call routing tasks.

But assistants can exploit another source of information for accurate speech recognition: a model of the world with which they are expected to assist – a model which, crucially, is mostly shared with the user. This is an extremely valuable

*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.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
IUI'08, January 13-16, 2008, Maspalomas, Gran Canaria, Spain.
Copyright 2008 ACM 978-1-59593-987-6/ 08/ 0001 \$5.00.

source, since hypothesized directives that do not describe entities in this world model are very likely to be incorrect.

The interpretation of hypothesized directives can be expensive, so the challenge in using this information to guide recognition lies in using it promptly. This paper 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 weighed together with phonological and syntactic factors as an integral part of the decoding process. Introducing world model referents into the decoding search greatly increases this 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 referential semantic language model described in this paper is also interesting in that, unlike other recent systems which interpret grammatical constituents bottom-up, in the context of sub-constituents in a parser chart [19, 8, 11, 1], it instead exploits context *dynamically*, using incremental processing and limited stack memory of an HMM-like time series model. This allows interpretations of constituents at the bottom of a phrase structure tree (which would be relatively unconstrained in bottom-up interpretation) to be constrained by interpretations of constituents occurring earlier in the utterance, without overriding (and thereby weakening) the structure-sharing of the recognition algorithm. The result is a single unified referential semantic probability model which achieves significant recognition accuracy gains over non-semantic alternatives and runs in real time on large domains.

SAMPLE INTERACTION

A sample interaction is shown below, in which a user defines portions of a student activities ontology, then navigates this newly-defined ontology to make further edits. The user is not expected to provide training sentences or generalizations about the domain. The topology of the ontology that the user defines, using the referential semantic language model described in this paper, is enough to reliably constrain the

recognition of navigation directives.¹

The interface accepts simple spoken imperative sentences as directives. Some of these are editor commands for creating and navigating taxonomic (tree-like) ontologies which reside in the world model of the interfaced application. This is done by navigating the interface context to specific locations and defining identifiers (world model relation labels) for those locations or the objects found there. For example, the user may utter:

- (1) ‘go to MUSIC, ORCHESTRA’
- (2) ‘add new label CELLO’

Here, the word ‘CELLO’ is pronounced naturally, not spelled out. This user-defined object is understood by the system as a sequence of phonemes or speech sounds, e.g. ‘CH.EH.L.OW’, denoted here using ARPABET phone symbols [9]. Pronunciations of new words may be frequently misrecognized, since the semantic context for these words has yet to be defined. In such cases, the interface will echo back an incorrect pronunciation of the word, e.g. CH.EH.R.OW.IH. The user can then navigate this incorrect pronunciation to edit the appropriate part:²

- (3) ‘change phone three to L’
- (4) ‘delete phone five’

When the system echoes back the correct pronunciation CH.EH.L.OW, the user can define the rest of the ontology incorporating this update into its recognition constraints:

- (5) ‘go to CELLO’
- (6) ‘add new label FIRST CHAIR’
- (7) ‘add HOMEROOM ONE, BEN, to FIRST CHAIR’

RELATED WORK

Spoken language interfaces to agents that recognize immediate go/move commands have been well studied [4]. In some studies, these commands are augmented with shell-like scripting capabilities [18]. But most existing spoken

¹It is important to note that since the interface is user-configurable, the navigation commands used here are not hard-wired into the system. Lowercase words are therefore not keywords per se, they are simply words that the system has already been taught. Uppercase words are user-defined words (also already known by the system) which exist as relation labels in the agent’s world model ontology. Phones connected with underscores are newly-defined (unknown) words, which will be introduced into the world model ontology as relation labels.

²Users of pure audio interfaces (with no video display) may prefer to navigate pronunciations which have been divided into syllables as well as phonemes, e.g. via directives like ‘go to syllable two and change the beginning to L’ or ‘change syllable two to LOW.’ Unfortunately, syllabification in English depends to some extent on etymological information (about where a word came from) – information which will not be available for new words. A user attempting to define the word ‘sportswear,’ for example, would not expect it to be decomposed into syllables ‘sport’ and ‘swear.’

language interface architectures rely on off-the-shelf speech decoding strategies developed for tasks like dictation or database querying, with mostly fixed vocabularies and plentiful training corpora. The approach described in this paper is novel in that it employs a speech decoding strategy – namely, a referential semantic language model – designed especially for the shell interface task, in which vocabularies are user-defined and training corpora are scarce, but world model information is readily available.

It is also not uncommon for spoken language interfaces to employ *context-sensitive* language models that are pre-compiled for particular discourse or environment states, and swapped out between utterances [14, 6]. But to approach human levels of recognition accuracy, spoken language interfaces will also need 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 music orchestra directory and set ...’ will depend crucially on the linguistic and environment context leading up to this point: the meaning of the first part of this directive ‘go to the music orchestra directory,’ as well as the objects that will be available once this part of the directive has been carried out. The approach described in this paper can be described as continuously context-sensitive.

Similar interfaces have been proposed that perform referential semantics continuously during speech decoding for the purpose of improving the accuracy of human-robot interfaces [17]. But these lack a linguistically rich semantic framework permitting complex nested references, and have not been scaled to abstract environments or concrete environments larger than a few dozen objects on a tabletop. Other approaches [8, 1] have sophisticated sensitivity to referential context, but are not defined to integrate efficiently into the speech decoding process. The approach described in this paper is able to exploit arbitrarily large environments,³ both concrete and abstract, including complex conditional program scripts, in order to improve recognition accuracy during real-time speech decoding.

BACKGROUND

Referential Semantics

Model Theory

The language model described in this paper defines semantic referents in terms of a world model \mathcal{M} . In model theory [20, 7], a world model is defined as a tuple $\mathcal{M} = \langle \mathcal{E}, \llbracket \cdot \rrbracket \rangle$ containing a domain of entity constants \mathcal{E} (e.g. persons or events in a scheduling application, files in a directory structure, etc.) and an interpretation function $\llbracket \cdot \rrbracket$ to define how (spoken) expressions can refer to these constants. Here, $\llbracket \cdot \rrbracket$ is quite versatile, accepting expressions ϕ that are equivalent to logical statements (simple type **T**), references to entities (simple type **E**), or functors (complex type $\langle \alpha, \beta \rangle$, e.g. defining sets) that take an argument of type α (e.g. an entity) and produce output of type β (e.g. a truth value, true if the entity is in the set). These functor expressions ϕ can then be

³As long as there is some notion of local context in the world model which limits the accessibility of referents, as described in the following section.

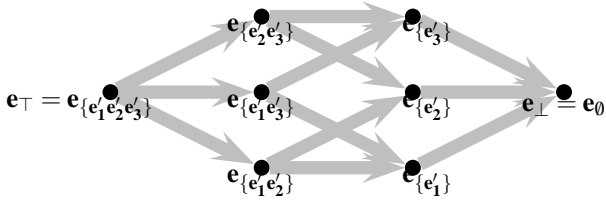


Figure 1. A subsumption lattice (laid on its side) over the power set of a domain containing three individuals: e'_1 , e'_2 , and e'_3 . Subsumption relations are represented as grey arrows from supersets (or super-concepts) to subsets (or sub-concepts).

applied to other expressions ψ of type α as arguments to yield expressions $\phi(\psi)$ of type β . By nesting functors, complex expressions can be defined, denoting sets or properties of entities: $\langle \mathbf{E}, \mathbf{T} \rangle$, relations over entity pairs: $\langle \mathbf{E}, \langle \mathbf{E}, \mathbf{T} \rangle \rangle$, or higher-order functors over sets: $\langle \langle \mathbf{E}, \mathbf{T} \rangle, \langle \mathbf{E}, \mathbf{T} \rangle \rangle$.

Ontological Promiscuity

First order or higher models can be mapped to equivalent zero order models. This is generally motivated by a desire to allow sets of entities to be described in much the same way as individual entities [12]. Entities in a zero order model \mathcal{M} can be defined from entities in a higher order model \mathcal{M}' by mapping (or *reifying*) each set $S = \{e'_1, e'_2, \dots\}$ in $\mathcal{P}(\mathcal{E}_{\mathcal{M}'})$ (or set of sets in $\mathcal{P}(\mathcal{P}(\mathcal{E}_{\mathcal{M}'}))$, etc.) as an entity $e_{\mathbf{S}}$ in $\mathcal{E}_{\mathcal{M}}$. Zero order functors in the interpretation function of \mathcal{M} can be defined directly from higher order functors (over sets) in \mathcal{M}' by mapping each instance of $\langle S_1, S_2 \rangle$ in $[[\cdot]]_{\mathcal{M}'}$: $\mathcal{P}(\mathcal{E}_{\mathcal{M}'}) \times \mathcal{P}(\mathcal{E}_{\mathcal{M}'})$ to a corresponding instance of $\langle e_{\mathbf{S}_1}, e_{\mathbf{S}_2} \rangle$ in $[[\cdot]]_{\mathcal{M}}$: $\mathcal{E}_{\mathcal{M}} \times \mathcal{E}_{\mathcal{M}}$. Set subsumption \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 [3]. These subset or subsumption relations can be represented in a subsumption lattice, as shown in Figure 1.

Language Modeling for Speech Recognition

The referential semantic language model described in this paper is based on the standard HMM-based language modeling framework commonly used in speech recognition.

HMMs and Language Models

The model described in this paper is a specialization of the Hidden Markov Model (HMM) framework commonly used in speech recognition [2, 13]. HMMs are probabilistic sequence models, or ‘time series’ models. They 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 short, overlapping frames of an audio signal) at corresponding time steps t . A most probable sequence of hidden states $\hat{h}_{1..T}$ can then hypothesized given any sequence of observed states $o_{1..T}$, using Bayes’ Law (Equation 2) and Markovian independence assumptions (Equation 3) to define the full $P(h_{1..T} | o_{1..T})$ probability as the product of a *Language Model (LM)* prior probability $P(h_{1..T}) \stackrel{\text{def}}{=} \prod_t \hat{P}_{\Theta_{\text{LM}}}(h_t | h_{t-1})$ and an *Acoustical Model (AM)* likelihood

probability $P(o_{1..T} | h_{1..T}) \stackrel{\text{def}}{=} \prod_t \hat{P}_{\Theta_{\text{AM}}}(o_t | h_t)$:

$$\hat{h}_{1..T} = \underset{h_{1..T}}{\operatorname{argmax}} P(h_{1..T} | o_{1..T}) \quad (1)$$

$$= \underset{h_{1..T}}{\operatorname{argmax}} P(h_{1..T}) \cdot P(o_{1..T} | h_{1..T}) \quad (2)$$

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

Language Model Components

The hidden variable values h_t at each time step t in an HMM are then usually divided into super-phonetic s_t (usually word), phone-level p_t , and sub-phonetic q_t (‘subphone’ or ‘state’) components.

$$h_t = \langle s_t, p_t, q_t \rangle \quad (4)$$

Superphone s_t components are usually words (or consecutive tuples of words) in most speech recognition systems, but can also include parsing information such as stacks or histories of phrase labels [5]. Phone-level p_t components define sequences of speech sounds (or consecutive tuples of speech sounds) associated with each word (e.g. ‘CH EH L OW’ for the word ‘cello’, using the ARPABET phone set [10]). Sub-phonetic q_t components define sub-states of each phone (e.g. the stop ‘kcl’ and burst ‘k’ phases of a plosive phone ‘K’, using the TIMIT subphone set⁴), which may vary depending on the immediately previous subphone.

These components must transition in order: super-phonetic units (e.g. words) can transition only when phones transition, and phones can transition only when subphones do. This behavior can be defined through the introduction of boolean *switching variables* to indicate whether subphones (or phones) have transitioned [21, 15], allowing phones (superphones) above them to transition only if true. Language model probabilities over these factored hidden states can be defined as a product of superphone, phone, and subphone transition probabilities and switching variable probabilities, with the switching variables then marginalized out:

$$\hat{P}_{\Theta_{\text{LM}}}(h_t | h_{t-1}) = \hat{P}_{\Theta_{\text{LM}}}(s_t p_t q_t | s_{t-1} p_{t-1} q_{t-1}) \quad (5)$$

$$\stackrel{\text{def}}{=} \sum_{f_t^Q f_t^P} \hat{P}_{\Theta_{\text{Subphone-Switch}}}(f_t^Q | p_{t-1} q_{t-1}) \cdot \hat{P}_{\Theta_{\text{Phone-Switch}}}(f_t^P | f_t^Q s_{t-1} p_{t-1}) \cdot \hat{P}_{\Theta_{\text{Superphone}}}(s_t | f_t^P s_{t-1}) \cdot \hat{P}_{\Theta_{\text{Phone}}}(p_t | f_t^Q f_t^P p_{t-1} s_t) \cdot \hat{P}_{\Theta_{\text{Subphone}}}(q_t | f_t^Q q_{t-1} p_t) \quad (6)$$

Superphones transition only when the phone sequence (word) below it finishes (switching $f_t^P = 1$), otherwise they deterministically propagate forward:

$$\hat{P}_{\Theta_{\text{Superphone}}}(s_t | f_t^P s_{t-1}) \stackrel{\text{def}}{=} \begin{cases} \hat{P}_{\Theta_{\text{Superphone-Tr}}}(s_t | s_{t-1}) & \text{if } f_t^P = 1 \\ 1 & \text{if } f_t^P = 0 \text{ if } s_t = s_{t-1}, 0 \text{ oth.} \end{cases} \quad (7)$$

⁴Essentially, the ARPABET with the addition of plosive closure symbols.

If the superphone above a phone sequence does transition (switching $f_t^P = 1$), a new phone sequence (word) is generated from the resulting superphone s_t using a pronunciation model $\Theta_{\text{Pronunciation}}$. If there is no superphone transition, each phone sequence p_{t-1} deterministically advances to the next phone $next(p_{t-1})$ when the subphone below it finishes (switching $f_t^Q = 1$), otherwise the phone deterministically propagates forward unchanged.

$$\hat{P}_{\Theta_{\text{Phone}}}(p_t | f_t^Q f_t^P p_{t-1} s_t) \stackrel{\text{def}}{=} \begin{cases} \text{if } f_t^Q = 1, f_t^P = 1 : \hat{P}_{\Theta_{\text{Pronunciation}}}(p_t | s_t) \\ \text{if } f_t^Q = 1, f_t^P = 0 : 1 \text{ if } p_t = next(p_{t-1}), 0 \text{ oth.} \\ \text{if } f_t^Q = 0, f_t^P = 0 : 1 \text{ if } p_t = p_{t-1}, 0 \text{ oth.} \end{cases} \quad (8)$$

The pronunciation model $\Theta_{\text{Pronunciation}}$ is defined in a pronunciation lexicon, which can be edited by the user using the SLUSH interface, and the subphone model Θ_{Subphone} is estimated directly from domain-independent corpora. The superphone transition model $\Theta_{\text{Superphone-Tr}}$ will be defined in terms of referential semantics in the following section.

A consequence of this hierarchy of transition models is that the highest-level superphone transition model is only consulted when a word transitions to a different word. This means that, although the variable t does in fact count time steps (corresponding to 10ms or so speech frames in this paper), when a word or other syntactic or syntactic configuration s transitions, the variables s_{t-1} and s_t at time steps $t-1$ and t do indeed contain the distinct previous and current values of s .

REFERENTIAL SEMANTIC DECODING

Referential Semantic Components

This basic language model is then extended to allow referential contexts to influence HMM transition probabilities, and thereby guide decoding. In particular, the super-phone units s_t of the factored language model described in the previous section will be expanded to include hypothesized referents e_t to concepts or entities in some world model, as well as words and syntactic components c_t to facilitate parsing:

$$s_t = \langle e_t, c_t \rangle \quad (9)$$

Viewed as a generative process, this model represents language at the top level as a random walk through a world model of referents (entities or sets of entities) connected by relations (logic predicates). The model first chooses semantic relation labels l_t and referents e_t , at each time step t , that are reachable from the semantic referents e_{t-1} at the previous time step. The model then chooses syntactic categories c_t , phone sequences p_t , and subphone sequences q_t to verbalize these relations.

During the course of processing, categories and referents may need to be stored and retrieved, so c_t and e_t will in fact consist of *stacks* (or vectors) of categories and referents, most of which will simply be propagated forward from time step to time step. Propagation of category labels in c_t will

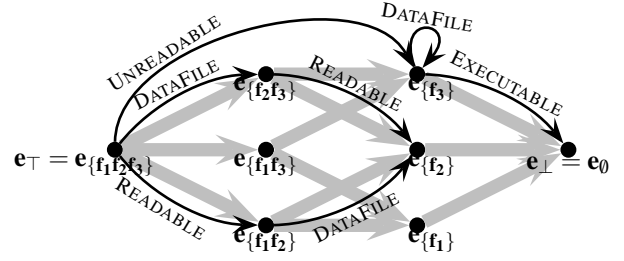


Figure 2. A subsumption lattice (laid on its side, in gray) over the power set of a domain containing three files: f_1 (a readable executable), f_2 (a readable data file), and f_3 (an unreadable data file). ‘Reference paths’ made up of conjunctions of relations l (directed arcs, in black) traverse the lattice from left to right toward the empty set, as referents $\{e_{\dots}\}$, corresponding to sets of files) are incrementally constrained by intersection with each $\llbracket l \rrbracket_{\mathcal{M}}$. (Some arcs are omitted for clarity.)

not vary across environments, and so can be pre-compiled into a static syntactic model $\hat{P}_{\Theta_{\text{Syn}}}(c_t | v_t l_t c_{t-1})$. But propagation of referents in e_t will vary across environments. To account for this propagation without pre-compiling environment information into the language model, a coindexation pattern v_t is introduced consisting of a vector of pointers to referents in e_{t-1} for each referent in e_t .

$$\hat{P}_{\Theta_{\text{Superphone-Tr}}}(s_t | s_{t-1}) = \hat{P}_{\Theta_{\text{Superphone-Tr}}}(e_t c_t | e_{t-1} c_{t-1}) \quad (10)$$

$$\stackrel{\text{def}}{=} \sum_{v_t, l_t} \hat{P}_{\Theta_{\text{Coind}}}(v_t | c_{t-1} e_{t-1}) \cdot \hat{P}_{\Theta_{\text{Ref}}}(l_t e_t | v_t e_{t-1}) \cdot \hat{P}_{\Theta_{\text{Syn}}}(c_t | v_t l_t c_{t-1}) \quad (11)$$

This coindexation pattern can then be compiled into the syntactic model instead.

Reference Transitions on a Subsumption Lattice

Relations (e.g. subsumption) among referents corresponding to sets can be navigated as a graph, just like relations over individuals. Properties (unary relations like READABLE or DATAFILE) can be represented in the referent transition model $\hat{P}_{\Theta_{\text{Ref}}}(l_t, e_t | v_t, e_{t-1})$ 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 $\llbracket l_t \rrbracket_{\mathcal{M}}$ satisfying the property l_t . The world model can therefore be cast as a subsumption lattice with the set of all individuals at the top e_{\top} (the result of intersecting an empty set of properties) and the empty set of individuals at the bottom e_{\perp} (resulting from an intersection of properties denoting disjoint sets).⁵ 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 . Thus, the set of ‘user-readable objects (property READABLE) that are data files (property DATAFILE)’ would be reachable by traversing a DATAFILE relation from the set of user-readable objects, or by traversing a READABLE relation from the set of data files, or by either path DATAFILE \circ READABLE or path READABLE \circ DATAFILE from e_{\top} . The resulting set may then serve as context for subsequent traversals.

⁵This lattice need not be an actual data structure. Since the world model is queried incrementally, the lattice relations may be calculated as needed.

A general template for intersective adjectives can be expressed as a noun phrase (NP) expansion using the following regular expression:

$$\text{NP}(g) \rightarrow \text{Det} \left(\text{Adj}(g) \right)^* \text{Noun}:l(g) \left(\text{PP}(g) \mid \text{RC}(g) \right)^*$$

where g is a variable over referential contexts (in this case, sets of individuals that are considered potential referents while the noun phrase is being interpreted), which is successively constrained by the semantics of the adjective and noun relation l , followed by prepositional phrase (PP) and relative clause (RC) subconstituents.

Reference Transitions with Relation Arguments

Sequences of properties (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 set of rooms (set g) that ‘contain (relation CONTAIN) objects that are *user-readable objects* (property READABLE)’ would be reachable only by:

1. pushing the original set of directories g onto the referent stack e_t , then
2. traversing a CONTAIN relation departing g to obtain the contents of those directories h , then
3. traversing a READABLE relation departing h to constrain this set to the set of contents that are also user-readable objects, then
4. traversing the inverse CONTAIN^l of relation CONTAIN to obtain the containers of these user-readable objects, then constraining the original set of directories g by intersection with this resulting set to yield the directories containing user-readable objects.

Forking is therefore handled via syntactic recursion: one path is explored by the recognizer while the other waits on a stack. A general template for branching reduced relative clauses (or prepositional phrases) that exhibit this forking behavior can be expressed as below, using the variables g and h defined above:

$$\text{RC}(g) \rightarrow \text{Verb}:l(h, g) \text{NP}(h) \text{---}^l(g, h)$$

where the inverse or transpose relation l^l at the last, empty constituent ‘---’ is intended to apply when the NP expansion concludes or reduces (when the forked paths are re-joined). 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 and the inverse or transpose relation l^l would depend on both referents g and h on the stack e_t .

Training

Phone and subphone models can be trained from corpora of transcribed utterances from other domains. But if in-domain corpora are not available, higher-level transition models will have to come from elsewhere.

The reference model defines probabilities over transitions from referents to referents (from e_{t-1} to e_t via relation l_t) in a

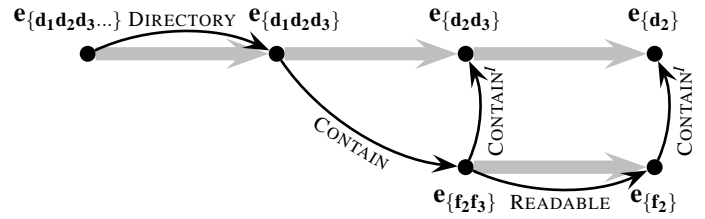


Figure 3. A reference path forks to specify referents using a two-place relation ‘Contain’ in a domain of directories d_1, d_2, d_3 . Here, d_2 contains f_2 and d_3 contains f_3 , and f_2 is readable. Again, subsumption is represented in grey and relations are represented in black. (Portions of the complete subsumption lattice and relation graph are omitted for clarity.)

world model graph. Generally these transitions will progress from vague referents to more specific referents: from large sets to small sets or individuals, or from root directories to sub-directories or files in a shell domain. These probabilities are therefore instantiated as uniform distributions over target referents e_t .

If available, this model could be trained on non-speech use data in a new domain, but even with no use data at all, these transition are still tightly constrained by the data in the world model itself. Sub-directories that do not exist cannot be traversed, and properties that do not apply to a vague referent cannot be used to designate more specific referents.⁶

Vocabulary (here modeled as phone sequences p_t) and to some extent syntax (category transitions c_t) can be expected to vary greatly across domains – particularly in design or programming applications, which are by definition concerned with creating new objects or new behaviors. One of the main advantages of the referential semantic language model described here is that it conditions these diverse surface phenomena on smaller sets of ‘local’ or accessible relations in an underlying world model. Conditioning syntax and word choice on world model relations can result in a nearly deterministic model in many cases (assuming each relation can be described using only a small set of possible words).

As mentioned above, referents may need to be stored and retrieved, in order to describe them in terms of other referents. For example, ‘go to X and set Y to ... and set Z to ...’, where X must be stored and used as context for Z. This can be handled using an explicit recursive syntax, which can be represented as complex (vector-valued) c_t random variables, associated with complex (vector-valued) referents e_t via complex (vector-valued) coindexation patterns v_t . Referents in e_t are simply determined by coindexation patterns in v_t , but these coindexation patterns v_t and complex categories c_t must come from somewhere.

⁶It is important to note that this assumes the user and system have shared knowledge of a world model. This can safely be assumed in design applications, where the user has created the world model, but in applications such as database query systems, it may be common for users to describe objects that do not exist.

If users are permitted to create new syntax patterns, these vectors must originate in a somewhat intuitive form, similar to grammar rules. In this implementation, category and coindexation vectors are defined in nestable regular expressions within a Hierarchic Hidden Markov Model [15], similar to that used to coordinate superphone, phone, and subphone transitions in the previous section.

ONTOLOGY NAVIGATION

In the case where a world model \mathcal{M} is zero-order, and relations defined in \mathcal{M} are binary over entities, then \mathcal{M} can be defined as a finite state automaton (FSA), whose states correspond to entities in \mathcal{E} , and whose transition function is defined by the (ordered, or directed) relations in $\mathbb{R}_{\mathcal{M}}$. This kind of configuration, when combined with the ‘ISA’ relations defined above, will resemble a hierarchic ontology with labeled arcs from concepts to subconcepts or instances. As an ontology, the FSA can be navigated downward, from concept to subconcept, by traversing labeled arcs in the usual way. It can also be navigated upward, from concept to superconcept, using unlabeled arcs or ε -transitions (making the FSA nondeterministic) in the reverse direction of each arc in the original downward-navigable deterministic FSA. This upward- and downward-navigable ontology will be used as a world model in the spoken language understanding shell evaluated in the following section.

In a stochastic model, probabilities can be associated with ε -transition functions as well as labeled- (or l -) transition functions so that if any concept and ancestor entities have outgoing arcs with the same label, these ε -transition probabilities can be combined with those for l -transitions and renormalized to prefer the arcs departing the lower-level concept, but still include as possible those departing an ancestor concept. Labels not explicitly defined at a given concept entity (as outgoing arcs) are implicitly assumed to exist with a ‘sink’ state destination \mathbf{e}_{\perp} , so that interpretation probabilities will be well defined for all label and entity conditions. The sink state is then constrained not to be generable by any lexical rule in Θ_{Syn} , and therefore cannot be described in directives. Labels that are not explicitly defined (whose destination is the sink state) are therefore referential ‘dead ends.’

As navigation of an ontology proceeds in the context of a particular entity e , there is a sense in which other entities e' at the same level of the ontology as the most recently described entity e , or at higher levels of the ontology than the most recently described entity, are semantically reachable without restating the ontological context (the path from the root concept \mathbf{e}_{\top} shared by e' and e). Thus, in the context of an activity like the wide receiver position in the sport football, other positions in the same sport, or other sports in the same school should be accessible without giving an explicit ‘back up’ directive at every level. Using a closure operation over the ε labels used as ontological back-pointers in the previous section, these sibling, ancestor, and (great-)aunt/uncle concepts e' can be connected to e via ε^*l -transitions (Figure 4b). These ε^*l -transitions are added to the world model definition prior to recognition, by composing any number of

$$\begin{aligned} S(g) &\rightarrow \text{set:SETTO}(h,g) \text{ PNpath}(g) \text{ to PNpath}(h) \\ \text{PNpath}(g) &\rightarrow \text{PNup}(g) \text{ PNsubpath}(g) \\ \text{PNpath}(g) &\rightarrow \text{PNup}(g) \\ \text{PNsubpath}(g) &\rightarrow \text{PN}(g) \text{ PNsubpath}(g) \\ \text{PNsubpath}(g) &\rightarrow \text{PN}(g) \\ \text{PNup}(g) &\rightarrow \text{homeroom:UP-HOMEROOM0}(g) \text{ zero} \\ \text{PN}(g) &\rightarrow \text{bell:BELL}(g) \\ \text{PNup}(g) &\rightarrow \text{sports:UP-SPORTS}(g) \\ \text{PN}(g) &\rightarrow \text{football:FOOTBALL}(g) \\ \text{PN}(g) &\rightarrow \text{captain:CAPTAIN}(g) \end{aligned}$$

Table 1. Sample grammar for student activities domain.

FSA ε -transitions, followed by a single l -transition:

$$\mathcal{E}' = \{ x \overset{\varepsilon^*l}{\curvearrowright} y \mid x \overset{\varepsilon^*}{\curvearrowright} z \in \mathcal{E}, z \overset{l}{\curvearrowright} y \in \mathcal{E} \} \quad (12)$$

where $x \overset{\varepsilon^*l}{\curvearrowright} y$ indicates that entity y is accessible from entity x by relation ε^*l ; and $x \overset{\varepsilon^*}{\curvearrowright} z$ indicates that z is reachable from x via any number of ε -transitions (in which case z is an ancestor zero or more levels above x in the ontology).

EVALUATION

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.

A Student Activities Database

The student activities ontology 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. A fragment of this ontology is shown in Figures 4, where every student or activity is an entity e in the set of entities \mathcal{E} , and all relations are l -transitions (in the case of 4a) or ε^*l -transitions (in the case of 4b).

A total of 240 entities were created in \mathcal{E} : 158 concepts (groups or positions) and 82 instances (students), each connected via a label (l -transition) to a parent concept. These l -transitions were then expanded to create ε^*l -transitions as shown in Figure 4b to give a total of 4704 transitions. On average, each node had a fanout of 18.7 outgoing transitions.

This ontology is manipulated using directives such as:

(8) ‘set homeroom two Bell to sports football captain’

which are incrementally interpreted by traversing l -relations from superconcept to subconcept (e.g. from ‘sports’ to ‘football’ to ‘captain’) or traversing ε^*l -transitions between arguments (e.g. from ‘Bell’ to ‘sports’). Recognition therefore requires syntax rules to be annotated with lexical semantic relations (no prefix for l -transitions, ‘UP-’ for ε^*l -transitions) and coindices g, h, \dots (which are translated di-

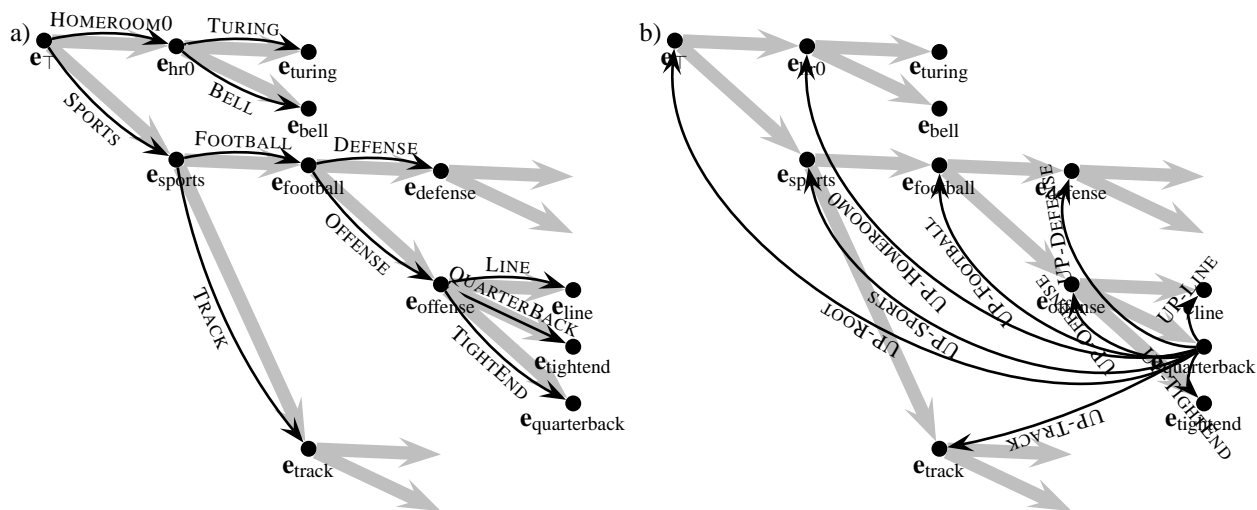


Figure 4. Upward and downward transitions in a sample student activities world model. Downward transitions (a) or l -transitions define basic subtype relations. Upward transitions (b) or ϵ^*/l -transitions relate sibling, ancestor, and (great-great-...)aunt/uncle concepts, are created from closure over ϵ -transitions followed by an l -transition. The entire model is reachable from any given referent via these two kinds of transitions.

rectly to \vec{v} parameters) that specify how the denotations of each constituent are passed to sub-constituents.

A sample set of grammar rules is shown in Table 1. These are implemented as nested regular expressions in a Hierarchic HMM, as described above.

Empirical Results

A corpus of 144 test sentences (no training sentences) was collected from 7 native English speakers (5 male, 2 female), who were asked to make specific edits to the student activities ontology described above. The average length of the sentences in this collection is 7.17 words, where all words are assumed to have pronunciations previously defined by the user.

Baseline and test versions of this system were run using a RNN acoustical model [16] trained on the TIMIT corpus of read speech [10]. Results below report concept error rate, where concepts correspond to relation labels in the world model.

Results using world model (ontology) information, according to reference links specified in a directive grammar (assuming no in-domain training sentences are available) show an overall 17.1% concept error rate, which is comparable to that reported for other dialogue systems trained on sample sentences [6, 14]:

subj	correct	subst	delete	insert	error
0	83.8%	14.1%	2.11%	2.82%	19.0%
1	73.2%	20.3%	6.54%	5.88%	32.7%
2	90.2%	7.84%	1.96%	0.65%	10.5%
3	88.1%	9.27%	2.65%	0.66%	12.6%
4	88.4%	10.3%	1.37%	3.42%	15.1%
5	90.8%	8.45%	0.70%	7.04%	16.2%
6	90.6%	8.63%	0.72%	3.60%	12.9%
all	86.4%	11.3%	2.34%	3.41%	17.1%

The overall sentence error rate was 59.44%. But many sentences had recognition errors in the last word only. Such errors are relatively easy to correct with additional user input (these words are local to the hypothesized context at the end of the utterance). The sentence error rate ignoring these errors in last word was 34.27%.

On the other hand, the results using the directive grammar alone, ignoring world model information and reference links in the grammar (and again, assuming no in-domain training sentences are available) show a much higher concept error rate of 43.5% (significant to $p = 1.1 \times 10^{-19}$ using pairwise t-test):

subj	correct	subst	delete	insert	error
0	57.0%	35.9%	7.04%	12.7%	55.6%
1	49.0%	41.2%	9.80%	13.7%	64.7%
2	71.9%	18.3%	9.80%	6.54%	34.6%
3	69.5%	26.5%	3.97%	9.27%	39.7%
4	67.8%	28.8%	3.42%	13.7%	45.9%
5	79.6%	19.0%	1.41%	7.04%	27.5%
6	75.5%	22.3%	2.16%	10.8%	35.3%
all	67.1%	27.5%	5.46%	10.5%	43.5%

with a sentence error rate of 93.01% (81.12% ignoring errors in the last word). This is because the grammar is *syntactically* relatively unconstrained, allowing any sequence of concept labels.

One interesting result of this experiment was that many of the erroneously hypothesized directives in both the baseline and test evaluations described edits that would have violated a ‘domain model’ of this task, had one existed: for example, some hypothesized directives would set the one student to another. Ordinarily this kind of information might be gleaned from training sentences. But if no training sentences are available, this information could explicitly be provided as restrictions on the actions associated with the words

‘set’ and ‘add’. It is difficult to determine how much of this kind of information can reliably be expected of nontechnical users, so the effect of incorporating this kind of domain knowledge was not evaluated.

Both test and baseline evaluations ran in real time on an 8-processor 2.6GHz server, with a beam width of 1000 hypotheses per frame.

CONCLUSION

This paper has described an implementation of a shell-like programming interface that achieves accurate recognition in user-defined domains with no available domain-specific training corpora, through the use of a referential semantic language model. This architecture requires that the agent make available a world model via a direct API or network connection, but even through a socket connection the combined phonological, syntactic, and referential semantic decoding process ensures the world model is only queried when necessary, so the interface runs in real time with modest hardware requirements.

This interface, including server and sample client source code and data files, is free for research purposes. Contact the authors for more information.

REFERENCES

1. G. Aist, J. Allen, E. Campana, C. Gallo, S. Stoness, M. Swift, and M. Tanenhaus. Incremental understanding in human-computer dialogue and experimental evidence for advantages over nonincremental methods. In *Proc. DECALOG*, pages 149–154, 2007.
2. J. Baker. The dragon system: an overview. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 23(1):24–29, 1975.
3. R. J. Brachman and J. G. Schmolze. An overview of the kl-one knowledge representation system. *Cognitive Science*, 9(2):171–216, Apr. 1985.
4. G. Bugmann, E. Klein, S. Lauria, and T. Kyriacou. Corpus-based robotics : A route instruction example. In *Proceedings of Intelligent Autonomous Systems*, pages 96–103, 2004.
5. C. Chelba and F. Jelinek. Exploiting syntactic structure for language modeling. In *Proc. COLING/ACL*, pages 225–231, Montreal, Canada, 1998.
6. G. Chung, S. Seneff, C. Wang, and I. Hetherington. A dynamic vocabulary spoken dialogue interface. In *Proc. ICSLP*, 2004.
7. A. Church. A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5(2):56–68, 1940.
8. D. DeVault and M. Stone. Domain inference in incremental interpretation. In *Proc. ICoS*, 2003.
9. W. M. Fisher, G. R. Doddington, and K. M. Goudie-Marshall. The darpa speech recognition research database: Specifications and status. In *Proceedings of DARPA Workshop on Speech Recognition*, pages 93–99, Feb. 1986.
10. W. M. Fisher, V. Zue, J. Bernstein, and D. S. Pallet. An acoustic-phonetic data base. *Journal of the Acoustical Society of America*, 81:S92–S93, 1987.
11. P. Gorniak and D. Roy. Grounded semantic composition for visual scenes. *Journal of Artificial Intelligence Research*, 21:429–470, 2004.
12. J. R. Hobbs. Ontological promiscuity. In *Proc. ACL*, pages 61–69, 1985.
13. F. Jelinek, L. R. Bahl, and R. L. Mercer. Design of a linguistic statistical decoder for the recognition of continuous speech. *IEEE Transactions on Information Theory*, 21:250–256, 1975.
14. O. Lemon and A. Grunstein. 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, 2004.
15. K. P. Murphy and M. A. Paskin. Linear time inference in hierarchical HMMs. In *Proc. NIPS*, pages 833–840, 2001.
16. T. Robinson. An application of recurrent nets to phone probability estimation. In *IEEE Transactions on Neural Networks*, 1994.
17. D. Roy and N. Mukherjee. Towards situated speech understanding: visual context priming of language models. *Computer Speech & Language*, 19(2):227–248, 2005.
18. P. E. Rybski, K. Yoon, J. Stolarz, and M. M. Veloso. Interactive robot task training through dialog and demonstration. In *Proceedings of HRI 2007*, pages 49–56, 2007.
19. W. Schuler. Computational properties of environment-based disambiguation. In *Proc. ACL*, pages 466–473, 2001.
20. A. Tarski. The concept of truth in the languages of the deductive sciences (polish). *Prace Towarzystwa Naukowego Warszawskiego, Wydział III Nauk Matematyczno-Fizycznych*, 34, 1933. translated as ‘The concept of truth in formalized languages’, in: J. Corcoran (Ed.), *Logic, Semantics, Metamathematics: papers from 1923 to 1938*, Hackett Publishing Company, Indianapolis, IN, 1983, pp. 152–278.
21. G. Zweig and S. J. Russell. Speech recognition with dynamic bayesian networks. In *AAAI/IAAI*, pages 173–180, 1998.