Positive Effects of Redundant Descriptions in an Interactive Semantic Speech Interface

Lane Schwartz lane@cs.umn.edu
Luan Nguyen lnguyen@cs.umn.edu
Andrew Exley exley@cs.umn.edu
William Schuler schuler@cs.umn.edu
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN, USA

ABSTRACT

Spoken language interfaces based on interactive semantic language models [16, 14] allow probabilities for hypothesized words to be conditioned on the semantic interpretation of these words in the context of some interfaced application environment. This conditioning may allow users to avoid recognition errors in an intuitive way, by adding extra, possibly redundant description. This paper evaluates the effect on error reduction of redundant descriptions in an interactive semantic language model. In order to evaluate the effect in natural use, the model is run on rich domains, supporting references to sets of individuals (instead of just individuals themselves) arranged in multiple continuous dimensions (a 2-D floorplan scene). Results of these experiments suggest that an interactive semantic language model allows users to achieve significantly higher recognition accuracy by providing additional redundant spoken description.

INTRODUCTION

Recognition accuracy remains a limiting factor in spoken language interfaces, particularly for content-creation applications such as organizers, reminder systems, or immersive design applications, in which new entities are introduced and named by users. This is because most speech recognizers estimate probabilities for hypothesized words using word co-occurrence statistics derived from fixed corpora, which naturally will not include names or novel words introduced by users.

Psycholinguistic studies [15] suggest human language processing bases its hypotheses not only on past word frequencies, but also on referential semantic information about likely referents for spoken descriptions. For example, a directive like *select the diff file in the eval folder* will be very likely to be recognized if there is known to be such a file in such a folder. Experiments on language models that allow recognition to interact with semantic interpretation in this way [16, 14] show them to be more accurate than syntax-only models or trigram word co-occurrence models compiled from referential semantic information in simulated content-creation applications.

This paper explores whether interactive models[1] can additionally allow users to improve recognition by adding redundant descriptions (e.g. directing the system to *select the diff file in the eval folder* when there is only one diff file anywhere).

Implicit in the standard word error rate statistic used in evaluating speech recognition systems is the assumption that recognition errors occur on a word-by-word basis. In fact, the independence assumptions in most word co-occurrence based speech recognizers means that word choice – and therefore word error – are indeed defined by a function on individual words in some local context of preceding words, so that the total number of errors and the overall sentence error rate (the percent of sentences with no word errors) will naturally increase as sentence length increases. Results using an interactive semantic interface, however, indicate that users can indeed significantly *decrease* sentence error rate by using redundant descriptions.

Even more encouragingly, interviews with test subjects suggest that the process of selecting redundant descriptions was conscious — even strategic: if the redundant phrase followed a non-redundant description that resulted in an error, subjects typically chose redundant descriptions that excluded the erroneous selection.

The remainder of this paper is organized as follows: The background section describes a basic implementation of an interactive semantic language model as an HMM-like probabilistic time-series model. The following section explores related work in interactive speech interfaces. The next section describes how this model was extended to two-dimensional scenes, allowing references to sets of individuals with continuous-valued attributes. The section after that gives results showing that the model can be ported to rich domains (in which redundancy is likely to be more natural) without substantial loss of recognition speed or accu-

[1]The term 'interactive model' [7] refers to a model in which semantics and the state of the world interact with other components to influence the recognition process. This contrasts with other possible uses of the term, where (for example) an interactive system could request additional clarifying information from the user.

IUI'09, February 8 - 11, 2009, Sanibel Island, Florida, USA.

racy. The following two sections describe how this model was used to evaluate the effects of redundant descriptions in a simulated design application.

RELATED WORK

Most existing spoken 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 used in this paper employs a speech decoding strategy – namely, an interactive semantic language model – designed especially for custom interface tasks 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 precompiled for particular discourse or environment states, and swapped out between utterances [6, 2]. 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 [16]. The approach used 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 [12]. 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 [3, 1] have sophisticated sensitivity to referential context, but are not defined to integrate efficiently into the speech decoding process. The approach used in this paper is able to exploit arbitrarily large environments,both concrete and abstract, including complex conditional program scripts, in order to improve recognition accuracy during real-time speech decoding.

BACKGROUND: INTERACTIVE SEMANTIC LANGUAGE MODEL

The model used in this paper is a factored Hidden Markov Model (HMM). HMMs characterize speech or text as sequences of hidden states s_t (in this case, stacked-up syntactic categories and referents) and observed states o_t (in this case, 10ms frames of audio input) at corresponding time steps t. A most likely sequence of hidden states $\hat{s}_{1..T}$ 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 a full $\mathsf{P}(s_{1..T} \mid o_{1..T})$ probability as the product of a *Transition Model (*Θ_A*)* probability $\mathsf{P}(s_{1..T}) \stackrel{\text{def}}{=} \prod_t \mathsf{P}_{\Theta_\text{A}}(s_t \mid s_{t-1})$ and an *Observation Model (*Θ_B*)* probability $\mathsf{P}(o_{1..T} \mid s_{1..T}) \stackrel{\text{def}}{=} \prod_t \mathsf{P}_{\Theta_\text{B}}(o_t \mid s_t)$:

$$\hat{s}_{1..T} = \underset{s_{1..T}}{\text{argmax}}\, \mathsf{P}(s_{1..T} \mid o_{1..T}) \quad (1)$$

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

$$\stackrel{\text{def}}{=} \underset{s_{1..T}}{\text{argmax}} \prod_{t=1}^{T} \mathsf{P}_{\Theta_\text{A}}(s_t \mid s_{t-1}) \cdot \mathsf{P}_{\Theta_\text{B}}(o_t \mid s_t) \quad (3)$$

This basic HMM framework is then extended to a Hierarchic Hidden Markov Model (HHMM) [9] in order to incorporate syntactic and semantic recursion into this process. This model first divides Θ_A transitions into two phases (Equation 4): a 'reduce' phase (resulting in an intermediate state r_t, which is marginalized or summed out), and a 'shift' phase (resulting in a modeled state s_t). These phases are then factored into hierarchies of depth-specific variables $r_t^1 ... r_t^D$ and $s_t^1 ... s_t^D$ at each time step (Equation 5):

$$\mathsf{P}_{\Theta_\text{A}}(s_t \mid s_{t-1}) = \sum_{r_t} \mathsf{P}(r_t \mid s_{t-1}) \cdot \mathsf{P}(s_t \mid r_t\ s_{t-1}) \quad (4)$$

$$\stackrel{\text{def}}{=} \sum_{r_t^{1..D}} \prod_{d=1}^{D} \mathsf{P}_{\Theta_\text{R}}(r_t^d \mid r_t^{d+1}\ s_{t-1}^d\ s_{t-1}^{d-1}) \cdot \mathsf{P}_{\Theta_\text{S}}(s_t^d \mid r_t^{d+1}\ r_t^d\ s_{t-1}^d\ s_t^{d-1}) \quad (5)$$

with r_t^{D+1} and s_t^0 defined as constants. In Viterbi decoding, the sums are replaced with argmax operators in order to extract a most likely sequence hypothesis.

In an ordinary HHMM, shift and reduce probabilities are defined in terms of finitely recursive Finite State Automata (FSAs) with probability distributions over transition, recursive expansion, and final-state status of states at each hierarchy level. Each intermediate variable is a boolean variable over final-state status $f_{r_t^d} \in \{\mathbf{0}, \mathbf{1}\}$ and each modeled state variable is a syntactic, lexical, or phonetic state $q_{s_t^d}$.

Syntactic states

Figure 1 shows a sample HHMM hypothesis for the sentence *select the diff in the eval folder*. In the interactive semantic model described in this paper, the syntactic states ($q_{s_t^d}$) at each depth and time step are derived from a context-free grammar, annotated with relation labels (such as IN or FOLDER) at the beginning and end of each expansion. The example in Figure 1 is derived from the following grammar G:

S → (SELECT) select NP (SELECT′)
NP → NP PP
NP → the N
N → (DIFF) diff
N → (FOLDER) folder
N → (EVAL) eval N
PP → (IN) in NP (IN′)

This grammar may be augmented by the user through the interface, for example by adding nouns to describe newly created classes of objects. After every such addition, the grammar is compiled using a right-corner transform [5, 13] into

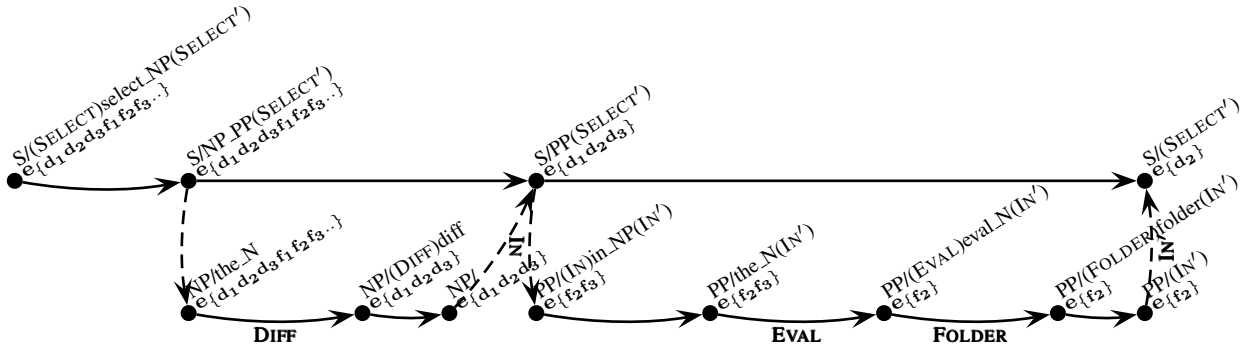

Figure 1. Bounded recursive state transitions among states derived from the following grammar (relations in parentheses): S → (SELECT) select NP (SELECT′), NP → NP PP, NP → the N, N → (DIFF) diff, N → (FOLDER) folder, N → (EVAL) eval N, PP → (IN) in NP (IN′). Solid lines indicate allowable transitions. Dashed lines indicate allowable recursive expansions or reductions. Reduce states — syntactic states without syntactic symbols following the slash — are shown for completeness, but are not explicitly calculated during the reduce phase.

a set of state transitions over states of the form $\alpha/\beta\_\gamma\_...$, defining incomplete instances of categories α lacking instances of categories $\beta, \gamma, ...$ yet to come. This transform defines within-level transition and cross-level expansion operations in a syntactic state model Θ_Q. In particular, this transform is designed to 'unroll' head or tail recursion of rules in an input grammar G as much as possible into transitions within a single stack level. This within-level recursion allows rich syntactic constructions to be recognized within the bounded memory store of an HHMM.[2]

The transitions and expansions allowed by Θ_Q are defined as follows:

- Within-level tail recursion (a rightward transition in Θ_Q) is licensed when expanding a right child ϵ of a right child γ:

  $$\begin{aligned}&\text{if } \alpha \rightarrow ... \ \gamma \ ... \in G\\ &\quad \gamma \rightarrow ... \ \delta \ \epsilon \ (l') \in G\\ &\quad \epsilon \rightarrow (l) \ \zeta \ \eta \ (-) \in G \text{ where } \zeta \rightarrow ... \notin G\\ &\text{then } \alpha/\delta\_\epsilon(l') \text{ may transition to } \alpha/(l)\zeta\_\eta(l') \text{ in } \Theta_Q \quad (6)\end{aligned}$$

  For example, the following rules in G:

  PP → (IN) in NP (IN′)
  NP → the N
  N → (EVAL) eval N

  license the following transition in Θ_Q:

  PP/the_N(IN′) → PP/(EVAL)eval_N(IN′)

  from a syntactic state PP/the_N (in which a prepositional phrase has been recognized lacking the article *the* followed by a noun category), to a syntactic state PP/eval_N (in which a prepositional phrase has been recognized lacking the noun *eval* followed by a noun category), essentially recognizing the symbol *the* in the first state, then expanding the next symbol (N) within-level using the rule N → eval N.

[2]In fact, evidence from large syntactically annotated corpora suggest that a large majority of English sentences can be recognized in this transformed representation using only a three or four element memory store [13]. Phrase structure trees can then be recovered via an inverse transform, though this is not necessary for interpretation.

- Cross-level head recursion (a downward expansion in Θ_Q) is licensed when expanding a left descendant δ' of a right child γ':

  $$\begin{aligned}&\text{if } \alpha \rightarrow ... \ \beta \ \gamma \ ... \in G\\ &\quad \gamma \stackrel{*}{\rightarrow} (-) \ ... \ \gamma' \ (-) \in G\\ &\quad \gamma' \rightarrow (-) \ \delta \ \epsilon \ ... \in G\\ &\quad \delta \stackrel{*}{\rightarrow} (-) \ \delta' \ ... \ (-) \in G\\ &\quad \delta' \rightarrow (l) \ \zeta \ \eta \ (l') \in G \text{ where } \zeta \rightarrow ... \notin G\\ &\text{then } \alpha/\delta\_\epsilon \text{ may expand to } \delta'/(l)\zeta\_\eta(l') \text{ in } \Theta_Q \quad (7)\end{aligned}$$

  where $\stackrel{*}{\rightarrow}$ indicates repeated application of a grammar rule. For example, the following rules in G:

  S → (SELECT) select NP (SELECT′)
  NP → NP PP
  NP → the N

  license the following expansion in Θ_Q:

  S/NP_PP(SELECT′) → NP/the_N

- Within-level head recursion (a rightward transition in Θ_Q) is licensed when expanding a right child ϵ of a left child β:

  $$\begin{aligned}&\text{if } \alpha \rightarrow ... \ \beta \ \gamma \ ... \in G\\ &\quad \beta \rightarrow ... \ \delta \ \epsilon \ (l') \in G\\ &\quad \delta \stackrel{*}{\rightarrow} \ ... \ \delta' \ (-) \in G\\ &\quad \epsilon \rightarrow (l) \ \zeta \ \eta \ (-) \in G \text{ where } \zeta \rightarrow ... \notin G\\ &\text{then } \delta/\delta' \text{ may transition to } \beta/(l)\zeta\_\eta(l') \text{ in } \Theta_Q \quad (8)\end{aligned}$$

  where $\stackrel{*}{\rightarrow}$ indicates repeated application of a grammar rule. For example, the following rules in G:

  S → S PP
  S → NP VP
  NP → the N
  N → (DIFF) diff
  VP → is PP

  license the following transition in Θ_Q:

  NP/diff → S/is_PP

  (This transition is not used in Figure 1, however.)

It is important to note that the order of the semantic relations in these transitions is preserved from the original grammar (labeled on the arcs in the figure). This ensures that if these

relations are applied in the order they occur in a tree (e.g. as chains of matrix products that fork and join with the tree structure), the result using the transformed tree-like HHMM transitions will match that obtained using the original tree.

Referential states

In order to incorporate referential semantic interpretation into this model, the intermediate (reduce) r_t^d and modeled (shift) s_t^d variables at each depth and time step are then further factored to include variables over referential states $e_{r_t^d}$ and $e_{s_t^d}$, following Wu et al. [16], in addition to the syntactic $q_{s_t^d}$ and final $f_{r_t^d}$ states from the ordinary HHMM:

$$r_t^d = \langle e_{r_t^d}, f_{r_t^d} \rangle \tag{9}$$

$$s_t^d = \langle e_{s_t^d}, q_{s_t^d} \rangle \tag{10}$$

In this paper, referential states will be constrained to first-order sets of individuals from some world model domain.[3] The model therefore behaves like a probabilistic version of an incremental interpreter [8, 4], sequentially applying constraints associated with each hypothesized word to sets of individuals hypothesized as the speaker's intended referents. As recognition progresses, these referent sets are winnowed down (or replaced, depending on the defined relations); and some ('trajector') referents may be shifted onto higher levels of a stack while other ('landmark') referents are described, to be composed or reduced together after this description has finished.

Referential states introduced into HHMM reduce and shift variables are constrained by labeled relations l (e.g. IN, FOLDER) associated with syntactic states q. Relation labels used during reduce and shift phases are defined using label functions L' and L, respectively.

Hypothesized referents $e_{r_t^d}$ at each reduce phase of this HHMM are constrained by the previous syntactic state $q_{s_{t-1}^d}$ using a reduce relation $l' = L'(q_{s_{t-1}^d})$, such that $e_{r_t^d} = l'(e_{r_t^{d+1}}, e_{s_{t-1}^{d-1}})$. In a first-order world model, this means the relation l' with the set $e_{r_t^{d+1}}$ as an argument, constrains the set $e_{s_{t-1}^d}$ to $e_{r_t^d}$. Reduce probabilities at each level are therefore:[4]

$$\mathsf{P}_{\Theta_{\mathrm{R}}}(r_t^d \mid r_t^{d+1} s_{t-1}^d s_{t-1}^{d-1}) \stackrel{\text{def}}{=} \begin{cases} \text{if } f_{r_t^{d+1}} = \mathbf{0} \;:\; [e_{r_t^d} = e_{s_t^d}] \cdot [f_{r_t^d} = \mathbf{0}] \\ \text{if } f_{r_t^{d+1}} = \mathbf{1} \;:\; [e_{r_t^d} = l'(e_{r_t^{d+1}}, e_{s_{t-1}^{d-1}})] \\ \qquad\qquad \cdot \mathsf{P}_{\Theta_{\mathrm{F}}}(f_{r_t^d} \mid d\; e_{r_t^d}\; q_{s_{t-1}^d}\; q_{s_t^{d-1}}) \end{cases} \tag{11}$$

where $r_t^{D+1} = \langle e_{s_{t-1}^D}, \mathbf{1} \rangle$ and $s_t^0 = \langle \mathbf{e}_\top, \mathbf{q}_\top \rangle$, and $l'(e_{r_t^{d+1}}, e_{s_{t-1}^{d-1}})$ is a semantic function indexed by l' applied to referents $e_{r_t^{d+1}}$ and $e_{s_{t-1}^{d-1}}$. Here $\mathbf{q}_\top$ is a start state and $\mathbf{q}_\perp$ is a null state.

Hypothesized referents $e_{s_t^d}$ at each shift phase of this HHMM are constrained by the current syntactic state $q_{s_t^d}$ using relations $l = L(q_{s_t^d})$ and $l' = L'(q_{s_t^d})$, such that $\mathsf{P}_{\Theta_{\mathrm{LL}}}(l \;\; l' \mid d, l(e_{r_t^{d+1}}), e_{r_t^{d+1}}, q_{s_{t-1}^d}, q_{s_t^{d-1}}) \neq 0$ (or $\mathsf{P}_{\Theta_{\mathrm{LL}}}(l\, l' \mid d, l(e_{s_t^{d-1}}), e_{s_t^{d-1}}, \mathbf{q}_\perp, q_{s_t^{d-1}}) \neq 0$, depending on the values of $f_{r_t^{d+1}}$ and $f_{r_t^d}$). Shift probabilities at each level are therefore:

$$\mathsf{P}_{\Theta_{\mathrm{S}}}(s_t^d \mid r_t^{d+1} r_t^d\; s_{t-1}^d s_t^{d-1}) \stackrel{\text{def}}{=} \begin{cases} \text{if } f_{r_t^{d+1}} = \mathbf{0},\; f_{r_t^d} = \mathbf{0} : [e_{s_t^d} = e_{r_t^d}] \cdot [q_{s_t^d} = q_{s_{t-1}^d}] \\ \text{if } f_{r_t^{d+1}} = \mathbf{1},\; f_{r_t^d} = \mathbf{0} : \sum_{l, l'} \mathsf{P}_{\Theta_{\mathrm{LL}}}(l\; l' \mid d\; e_{r_t^{d+1}}\; q_{s_{t-1}^d}\; q_{s_t^{d-1}}) \\ \qquad\qquad \cdot [e_{s_t^d} = l(e_{r_t^{d+1}})] \\ \qquad\qquad \cdot \mathsf{P}_{\Theta_{\mathrm{Q}}}(q_{s_t^d} \mid d\; l\; l'\; q_{s_{t-1}^d}\; q_{s_t^{d-1}}) \\ \text{if } f_{r_t^{d+1}} = \mathbf{1},\; f_{r_t^d} = \mathbf{1} : \sum_{l, l'} \mathsf{P}_{\Theta_{\mathrm{LL}}}(l\; l' \mid d\; e_{s_t^{d-1}}\; \mathbf{q}_\perp\; q_{s_t^{d-1}}) \\ \qquad\qquad \cdot [e_{s_t^d} = l(e_{s_t^{d-1}})] \\ \qquad\qquad \cdot \mathsf{P}_{\Theta_{\mathrm{Q}}}(q_{s_t^d} \mid d\; l\; l'\; \mathbf{q}_\perp\; q_{s_t^{d-1}}) \end{cases} \tag{12}$$

where $r_t^{D+1} = \langle e_{s_{t-1}^D}, \mathbf{1} \rangle$ and $s_t^0 = \langle \mathbf{e}_\top, \mathbf{q}_\top \rangle$, and $l(e)$ is a semantic function indexed by l applied to referent e.

The cases in the above equation are conditioned on final-state boolean variables $f_{r_t^{d+1}}$ and $f_{r_t^d}$. In the first case, where there is no final state immediately below the current level, referential and syntactic states are simply copied forward. The second and third cases correspond to (rightward) transition and (downward) expansion respectively, as defined in the previous section. In these cases, referential and syntactic states are chosen by:

1. selecting, according to a 'description' model Θ_{LL}, a relation label l with which to constrain the current referent, and a referent set $e_{s_t^d}$ resulting from this constraint,
2. selecting, according to a 'lexicalization' model Θ_{Q}, a syntactic state $q_{s_t^d}$ that is compatible with this label (i.e. has $L(q_{s_t^d}) = l$).

In this definition, traditionally one-place relations like FOLDER are represented as referential semantic transitions over labeled edges l from supersets (referential states) $e_{s_{t-1}^d}$ to subsets (other referential states) $e_{r_t^{d+1}}$, defined by intersecting the superset $e_{s_{t-1}^d}$ with the set of individuals satisfying the property l (see Figure 1).

Higher-arity relations like IN define more complex paths that fork and rejoin. For example, the referent of *the diff in the eval folder* in Figure 1 would be reachable only by:

1. storing the original set of diff files $\mathbf{e}_{\{\mathbf{d_1 d_2 d_3}\}}$ as a top-level referent in the HHMM hierarchy, then
2. traversing an IN relation departing $\mathbf{e}_{\{\mathbf{d_1 d_2 d_3}\}}$ to obtain the containers of those diffs $\mathbf{e}_{\{\mathbf{f_2 f_3}\}}$, then

[3] But in principle there is nothing to prevent arbitrary descriptions from serving as referents. The number of possible first-order sets (of individuals) is exponential on the size of the domain of individuals. The techniques applied in this paper to efficiently estimate sequences of referents can therefore be applied equally well to referents with continuous (i.e. infinite) domains.

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

3. traversing an EVAL relation departing $\mathbf{e}_{\{\mathbf{f_2 f_3}\}}$ to constrain this set to the set of eval folders that are also containers: $\mathbf{e}_{\{\mathbf{f_2}\}}$, then

4. traversing the inverse IN′ of relation IN to obtain the contents of these folders, then constraining the original set of diff files $\mathbf{e}_{\{\mathbf{d_1 d_2 d_3}\}}$ by intersection with this resulting set to yield the diff files in eval folders: $\mathbf{e}_{\{\mathbf{d_2}\}}$.

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

$$\text{PP} \rightarrow (\text{IN}) \text{ in NP } (\text{IN}') \tag{13}$$

where the inverse relation IN′ is applied when the NP expansion concludes or reduces (when the forked paths are rejoined), as shown in Figure 1.

Negation and comparatives can also be modeled in this framework as special types of relations between sets [16].

PRAGMATICS IN TRANSITION PROBABILITIES

One advantage of a first-order reference model (which allows references to sets of individuals as well as to individuals themselves) is that it allows transition probabilities to be based not only on properties of the individuals in the source or destination sets, but also on properties of these sets taken as a whole. For example, transition probabilities can be made to reflect pragmatic constraints on felicitous reference by conditioning on the cardinalities of the source and destination sets. Observations of these cardinalities in a graphical interface domain, where the world model is shared with the user via a display, suggest that:

1. referential transitions to empty sets are very unlikely – users seem to naturally constrain themselves to refer only to things on the display;

2. referential self-transitions (or redundant transitions, which have no winnowing effect) are somewhat less likely in this domain, but still possible (e.g. *the diff file in the eval folder* when the eval folder contains only one file); and

3. referential transitions that apply substantial constraints from the source referent to destination are the most common.

A simple statistic based on these observations can be used to constrain decoding. The statistic used in this paper starts with uniform probabilities over departing labels l (where each label is associated with the first expanded state q_{dst} of some syntactic expansion rule $q_{src} \rightarrow q_{dst}...$), then augments these uniform probabilities with weights $w(e_{src}, e_{dst})$ for the three classes of referential transition described above (where e_{src} is the source referent and e_{dst} is the destination), normalized over all destinations:[5]

$$\mathsf{P}_{\Theta_{\text{LL}}}(l\, l' \mid e_{src}\, q_{src}\, ...) = \frac{[q_{src} \rightarrow .../(l)...(l') \in \Theta_{\text{Q}}] \cdot w(e_{src}, l(e_{src}))}{\sum_{l,l'} [q_{src} \rightarrow .../(l)...(l') \in \Theta_{\text{Q}}] \cdot w(e_{src}, l(e_{src}))} \tag{14}$$

Ideally these weights would be empirically determined, but in the absence of an appropriate training set they can be set by hand, e.g.:

$$w(e_{src}, e_{dst}) = \begin{cases} \text{if } 0 < |e_{dst}| = |e_{src}| : .1 \\ \text{if } 0 < |e_{dst}| < |e_{src}| : 1 \\ \text{if } 0 = |e_{dst}| : 0 \end{cases} \tag{15}$$

The result is a simple probabilistic model of the pragmatic intuition that people use language to provide meaningful reference.

RICH DOMAINS

The purpose of the experiments described in this paper was to evaluate the effect of redundant descriptions using an interactive semantic language model in a simulated design task where redundant descriptions are a natural option for the users. Earlier studies using interactive semantic language models involve purely discrete structures such as file directories [16]. Since these use only one relation (CONTAIN), redundant descriptions do not risk introducing very much ambiguity, potentially exaggerating the advantage of this strategy.

The present study was therefore conducted in the context of a two-dimensional spatial design application in which referents can be sets of individuals with at least two continuous-valued attributes (the x and y coordinates of each individual object). The individuals used in this evaluation are rectangles or ovals of varying size, at various locations.

The introduction of continuous-valued location and size coordinates allows a generalization of the basic IN and CONTAIN relations based on location of the content individual and the size of the container individual. Specifically the CONTAIN relation is satisfied if the centroid of the content is within the bounding box of the container; and IN is the inverse of CONTAIN.

The introduction of continuous-valued attributes also allows relative spatial relations ABOVE, BELOW, LEFTOF, and RIGHTOF to be defined. These relations are satisfied when the centroid of the trajector (the first referent) lies within $\pm 45°$ of the appropriate cardinal direction.[6]

Relations in this framework are also designed to be augmented by the user, whenever the grammar is modified. For example, when nouns are added, usually an individual object instantiating this noun is also added, with a corresponding

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

[6] Regier and Carlson [10] propose a much more sophisticated model, but the present definition is adequate for the experiment described in this paper.

```
CHAIR :
  (set-of-all i in (source-set) s-t
    ((type of i) is (chair)))

ROOM :
  (set-of-all i in (source-set) s-t
    ((type of i) is (room)))

RIGHTOF :
  (product-of (source-set) with
    (matrix-from-each i to-each j s-t
      ((i is-not j)
       and ((abs-of ((y of i)
                     minus (y of j)))
            l-t-eq ((x of i)
                    minus (x of j))))))
RIGHTOF′ :
  (intersection-of (context-set) with
    (product-of (source-set) with
      (matrix-from-each i to-each j s-t
        ((i is-not j)
         and ((abs-of ((y of i)
                       minus (y of j)))
              l-t-eq ((x of j)
                      minus (x of i)))))))
```

```
CONTAIN :
  (product-of (source-set) with
    (matrix-from-each i to-each j s-t
      ((i is-not j)
       and (((abs-of ((y of i)
                      minus (y of j)))
             l-t-eq (yradius of i))
            and ((abs-of ((x of i)
                          minus (x of j)))
                 l-t-eq (xradius of i))))))

CONTAIN′ :
  (intersection-of (context-set) with
    (product-of (source-set) with
      (matrix-from-each i to-each j s-t
        ((i is-not j)
         and (((abs-of ((y of i)
                        minus (y of j)))
               l-t-eq (yradius of j))
              and ((abs-of ((x of i)
                            minus (x of j)))
                   l-t-eq (xradius of j)))))))
```

Figure 2. Sample relation definitions, as LISP-like scripts. Constants `source-set` **and** `context-set` **refer to the first and second arguments provided when the relation is called as a function (see Equations 11 and 12). Functions** `product-of` **and** `matrix-from` **apply relations to sets by casting sets as vectors and relations as matrices. Keywords** `s-t` **abbreviate *such that,*** `l-t-eq` **abbreviate *less than or equal to,* and** `abs-of` **abbreviate *absolute value of*.**

object type. A new relation must therefore also be defined to pick out individuals of this new type.

More complex relations can be added as well. The current implementation uses a simple LISP-like scripting language to abstractly specify relations based on discrete or continuous attribute values (see Figure 2).[7] From time to time the implementation evaluates these scripts over the world model and stores the result in a set of relation matrices. Compositional semantics in this system can therefore be cast as matrix multiplication chains which fork and join, as referents are shifted and reduced in the HHMM memory store.

EVALUATION OF PERFORMANCE IN RICH DOMAINS

The performance of this framework was evaluated on a testbed 2-D scene design domain. In this domain, subjects were shown randomly-generated 2-D scenes containing 110 colored shapes designated as rooms, tables, chairs, dividing walls, etc (see Figure 3).

Subjects were then told to direct the system to select predefined goal sets of these shapes, using one or more spatial relations (e.g. 'in,' 'on,' 'above,' 'to the left'). Five subjects each described 20 such goal sets, with an average sentence error rate of 13% (see Table 1). Users were allowed to retry goals which were not correctly selected; in all cases the system correctly selected the goal set after 1 or 2 retries.

These experiments used an off-the-shelf RNN acoustic model [11] for Θ_B, which provides a frame rate of 62.5Hz (so T=62.5 per sec). At this frame rate, the system was observed to run in approximately real time on a 2.4 GHz 64-bit dual quad core server and a 2.4 GHz 64-bit dual core client, using a beam width B of 100 hypotheses, a world model domain I with 110 individuals (so 2^{110} possible set referents), and a relation label vocabulary L of size 30 (mapping to a word vocabulary of size 50).

This experiment was then repeated for a large vocabulary ($L = 1000$) with similar results (see Table 2). This is not surprising, since the interactive semantic model constrains the set of usable words to those that describe the available set of individuals (the pragmatics described in the previous section assigns zero probability to words that yield empty referents).

EVALUATION OF REDUNDANT DESCRIPTIONS

To evaluate the contribution of redundant descriptions, users were again presented with a scene design application and were again asked to manipulate items in a scene.

Scene design task

The scene design task used in these experiments allows users to select individual items and sets of items in a continu-

[7] In principle, new syntactic expressions and associated relations could be added using speech. In the current implementation, this is not supported.

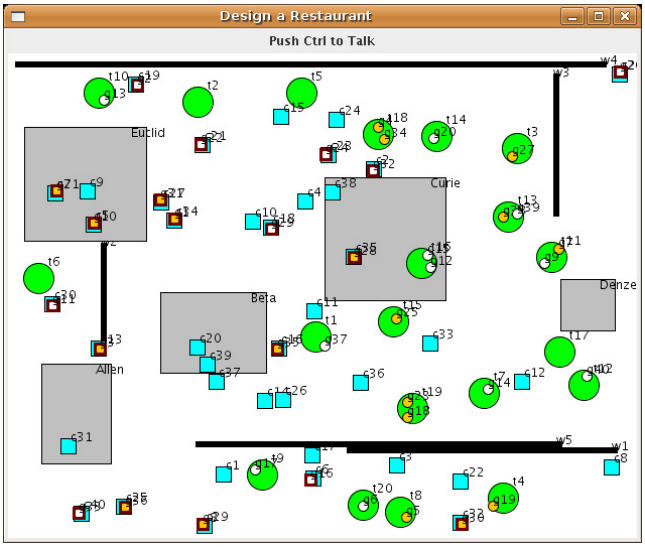

Figure 3. A sample world model scene resulting from the directive 'select the glasses on a chair.' Selected objects are outlined in red (black in print). Such sets are difficult to select with a mouse.

ous two-dimensional scene. Each scene consists of several named areas. Items are located throughout the scene, and may be located in a named area or may be outside any named area.

The sample scene used in this experiment contained 100 items, each with a unique name (see Figure 4). Each item in the scene was randomly selected from a list of single-syllable nouns. The resulting list contained many pairs of nouns with similar pronunciations. In isolation, the nouns in this list are easily confusable by a speech recognizer.

This experiment sought to determine whether redundant descriptions in spoken commands produce a positive effect on recognition accuracy in a semantic speech interface. Subjects directed the system to select predefined items from the scene using simple commands with no redundant information (e.g. *select the bed*) and using more complex commands with redundant information (e.g. *select the bed to the left of the chair*). Each subject attempted to select each of 100 predefined items first using simple commands with no redundant information. After attempting to select an item, each subject was directed to select the item using a more complex commands with redundant information, regardless of whether or not the prior utterance (with no redundant information) was correctly recognized. Subjects were free to choose the prepositions and landmarks in the redundant commands.

Empirical Results

A corpus of 1000 test sentences (no training sentences) was collected from 5 native English speakers who were asked to select items in the sample scene described above. For each of 100 predetermined items, each subject attempted to select the item, first using a simple command with no redundant information, then using a more complex command with redundant information.

Subject	Sentence error rate	Corrected on 1st retry	Corrected on 2nd retry
1	2 / 20	2	-
2	2 / 20	1	1
3	3 / 20	2	1
4	4 / 20	4	-
5	2 / 20	1	1
Total	13%	10	3

Table 1. Sentence error rate: number of times the system incorrectly selected the set of individuals described by the user, using I=110, B=100, L=30.

Subject	Sentence error rate
1	2 / 20
2	1 / 20
3	5 / 20
Total	13%

Table 2. Sentence error rate: number of times the system incorrectly selected the set of individuals described by the user, using I=110, B=100, L=1000.

Results are shown in Table 3. For sentences with no redundant information, the overall sentence error rate was 32.6%. This high error rate is to be expected, as this part of the task is largely equivalent to isolated single word recognition of monosyllabic words. Adding redundant information using a single prepositional phrase results in a substantial 37% reduction in sentence error rate, from 32.6 to 20.6. The reduction in error rate is statistically significant to P=0.0035 by pairwise per subject Student's *t* test (two-tailed).

DISCUSSION

Intuitive User Strategies for Error Reduction

Adding redundant information reduced sentence error rates by more than a third, and for one subject by over a half. Users were free to select prepositional phrases that they felt would be most useful. In informal interviews, test subjects indicated that this enabled relatively natural error recovery; if a non-redundant description resulted in a recognition error, subjects tended to simply chose a redundant description that would exclude the erroneous selection.

For example, if the original utterance was *select the nut* and the system incorrectly selected the net, the user might follow up with *select the nut in the field*. The scene contains only one nut and one net; but because the nut is in the field and the net is not, the additional semantic information provided by the redundant prepositional phrase is able to usefully constrain the speech recognition to recognize the correct utterance. Users were also observed to make explicit use of erroneously selected items in formulating followup utterances. An example of this phenomenon could be seen when *select the track* was erroneously recognized as *select the trap.* A successful followup used *trap* in the redundant description: *select the track to the right of the trap*.

This suggests that interactive language models provide users

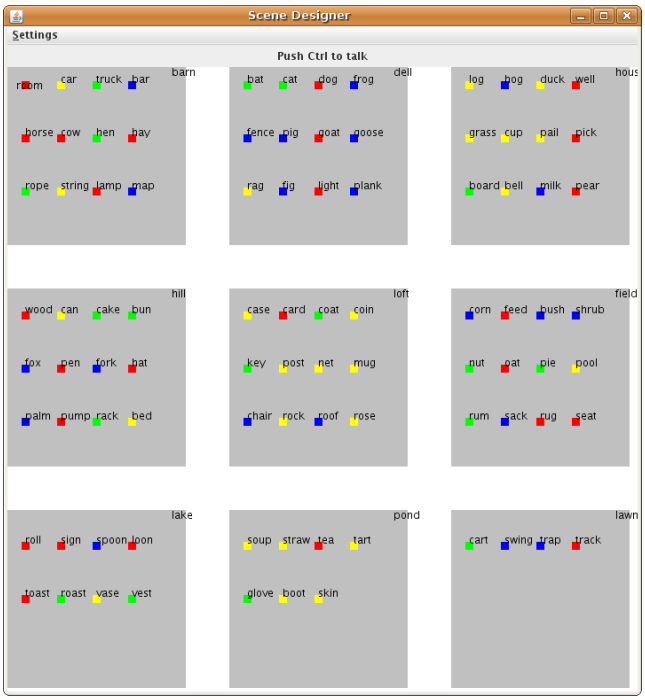

Figure 4. Sample scene in the scene design task

not only with better basic accuracy than conventional trigram models for content-creation domains, but also with a natural means to explicitly trade speaking time for recognition accuracy in cases where errors are more likely or more difficult to repair. The implementation described in this experiment can even be employed to let users explicitly negate misrecognized (or potentially misrecognizable) analyses: e.g. *select the cart in the loft not in the lawn*, making the probability of the originally misrecognized description extremely low.

CONCLUSION

This paper has described an experiment to determine whether interactive semantic language models can allow users to intuitively improve recognition accuracy of a spoken language interface by providing redundant descriptions. Interfaces based solely on word co-occurrences will typically *increase* their sentence error rate as sentence length increases, since sentence error rate in co-occurrence-based models is usually a result of largely independent errors on individual words. The results described in this paper show that the use of redundant phrases in an interactive semantic speech interface instead results in a substantial and statistically significant *decrease* in error rate.

The use of interactive language models not only shows better basic accuracy than conventional trigram models for content-creation domains, it suggests an opportunity for users to explicitly trade speaking time for recognition accuracy in cases where errors are more likely or more difficult to repair.

Subject	Sentence error rate without redundancy	Sentence error rate with redundancy
1	54 / 100	37 / 100
2	32 / 100	21 / 100
3	25 / 100	18 / 100
4	28 / 100	12 / 100
5	24 / 100	15 / 100
All	32.6%	20.6%

Table 3. Sentence error rate. Users attempted to select items in a scene using simple descriptions (without descriptive prepositional phrase) and then with a redundant descriptive prepositional phrase.

ACKNOWLEDGEMENTS

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

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. G. Chung, S. Seneff, C. Wang, and I. Hetherington. A dynamic vocabulary spoken dialogue interface. In *Proc. ICSLP*, pages 1457–1460, 2004.
3. D. DeVault and M. Stone. Domain inference in incremental interpretation. In *Proc. ICoS*, pages 73–87, 2003.
4. N. Haddock. Computational models of incremental semantic interpretation. *Language and Cognitive Processes*, 4:337–368, 1989.
5. M. Johnson. Finite state approximation of constraint-based grammars using left-corner grammar transforms. In *Proceedings of COLING/ACL*, pages 619–623, 1998.
6. O. Lemon and A. Gruenstein. 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.
7. J. L. McClelland and D. E. Rumelhart. An interactive activation model of context effects in letter perception: Part 1. an account of basic findings. *Psychological Review*, 88:375–407, 1981.
8. C. Mellish. *Computer interpretation of natural language descriptions*. Wiley, New York, 1985.
9. K. P. Murphy and M. A. Paskin. Linear time inference in hierarchical HMMs. In *Proc. NIPS*, pages 833–840, 2001.

10. T. Regier and L. Carlson. Grounding spatial language in perception: An empirical and computational investigation. *Journal of Experimental Psychology: General*, 130:273–298, 2001.

11. T. Robinson. An application of recurrent nets to phone probability estimation. In *IEEE Transactions on Neural Networks*, volume 5, pages 298–305, 1994.

12. D. Roy and N. Mukherjee. Towards situated speech understanding: visual context priming of language models. *Computer Speech & Language*, 19(2):227–248, 2005.

13. W. Schuler, S. AbdelRahman, T. Miller, and L. Schwartz. Toward a psycholinguistically-motivated model of language. In *Proceedings of COLING*, Manchester, UK, August 2008.

14. W. Schuler, S. Wu, and L. Schwartz. A framework for fast incremental interpretation during speech decoding. *Computational Linguistics*, in press.

15. M. K. Tanenhaus, M. J. Spivey-Knowlton, K. M. Eberhard, and J. E. Sedivy. Integration of visual and linguistic information in spoken language comprehension. *Science*, 268:1632–1634, 1995.

16. S. Wu, L. Schwartz, and W. Schuler. Exploiting referential context in spoken language interfaces for data-poor domains. In *Proc. International Conference on Intelligent User Interfaces (IUI'08)*, Canary Islands, Spain, January 2008.