Incremental Semantic Models for Continuous Context-Sensitive Speech Recognition*

Tim Miller, Lane Schwartz, and William Schuler

University of Minnesota, Twin Cities
{tmill,lschwar,schuler}@cs.umn.edu

Abstract. Context-sensitive speech recognizers use environment or discourse information to influence language model probabilities used in speech decoding. This is usually done by switching language models between utterances. This paper explores the use of a *continuously* context-sensitive language model that uses incremental interpretation to update context at every time step in decoding. Because it only explores the world model incrementally, this semantic model does not need to be pre-computed, raising the possibility of representing continuously-variable concepts as semantic referents (such as time points and measurements, or real numbers themselves), and supporting dynamic reasoning about consequences of actions during speech decoding.

1 Introduction

It is now common 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 [1, 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 For example, the probability distribution over the next word in the utterance 'go to the distribution source directory and open . . . ' 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 distribution source directory,' as well as the files that will be available once this part of the directive has been carried out.

This paper will describe semantic representation strategies in a statistical language model that incrementally interprets spoken directives during Viterbi decoding. Unlike earlier constraint-based incremental interpreters [3, 4], the approach described in this paper pursues multiple interpretations at once, ranked probabilistically. Finally, since the language model performs interpretation based on the left-to-right sharing of a Viterbi dynamic programming algorithm instead of the bottom-up sharing of a CKY-like parsing algorithm [5–8], this approach can constrain semantics at the beginning of recognition using the context of prior utterances, avoiding the large, relatively unconstrained sets of referents which arise at the bottom of a parser chart.

* This research was supported by National Science Foundation CAREER/PECASE award 0447685. The views expressed are not necessarily endorsed by the sponsors.

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, words w_t, and speech phones p_t to verbalize these relations:[1]

$$\hat{\mathsf{P}}(e_t\, c_t\, w_t\, p_t \mid e_{t-1}\, c_{t-1}\, w_{t-1}\, p_{t-1}) \stackrel{\text{def}}{=}$$
$$\sum_{l_t} \hat{\mathsf{P}}_{\Theta_{\text{Sem}}}(l_t\, e_t \mid e_{t-1}\, c_{t-1}) \cdot \hat{\mathsf{P}}_{\Theta_{\text{Syn}}}(c_t \mid c_{t-1}\, l_t) \cdot \hat{\mathsf{P}}_{\Theta_{\text{Lex}}}(w_t \mid w_{t-1}\, c_t) \cdot \hat{\mathsf{P}}_{\Theta_{\text{Phon}}}(p_t \mid p_{t-1}\, w_t) \quad (1)$$

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 not vary across environments, and so can be pre-compiled into a static syntactic model $\hat{\mathsf{P}}_{\Theta_{\text{Syn}}}(c_t \mid c_{t-1}, l_t)$. 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 will be introduced consisting of a vector of pointers to referents in e_{t-1} for each referent in e_t:

$$\hat{\mathsf{P}}_{\Theta_{\text{Sem}}}(l_t, e_t \mid e_{t-1}, c_{t-1}) = \sum_{v_t} \hat{\mathsf{P}}_{\Theta_{\text{SemCoind}}}(v_t \mid c_{t-1}, e_{t-1}) \cdot \hat{\mathsf{P}}_{\Theta_{\text{SemRef}}}(l_t, e_t \mid v_t, e_{t-1}) \quad (2)$$

These coindexation patterns are derived from referent variables specified in the same grammar rules as those used to derive $\hat{\mathsf{P}}_{\Theta_{\text{Syn}}}(c_t \mid c_{t-1}, l_t)$ (see Section 2). Relation labels l_t and referents e_t are then chosen according to the pattern specified in v_t, so that probabilities $\hat{\mathsf{P}}_{\Theta_{\text{SemRef}}}(l_t, e_t \mid v_t, e_{t-1})$ will be well defined for relations l_t of any arity.

In Viterbi decoding, the above probability (1) must be calculated for every possible combination of e_{t-1}, c_{t-1}, w_{t-1}, and p_{t-1}, on the trellis at time $t-1$, for every possible transition available to the model. Of the terms in this equation, the referent transition $\hat{\mathsf{P}}_{\Theta_{\text{SemRef}}}(l_t, e_t \mid v_t, e_{t-1})$ is potentially the most expensive, because it requires a query to a possibly very large world model. But this expense is mitigated by the following factors:

1. Referents will only transition when a word at the bottom of the stack changes (which will be possible only for the fraction of trellis entries that are on their final phone);
2. When referents do transition, these transitions will be localized to particular stack elements (in most cases transitions only affect the stack element above the most recent stack pop); and
3. The same semantic referent transitions will be queried many times across neighboring time steps, or within the same time step (by syntactic or pronunciation variants of a predicate, or synonyms), so considerable savings can be achieved through caching query results.

[1] This model is an instance of a Hierarchic Hidden Markov Model [9], and as such also includes binary 'final state' variables between levels to synchronize transitions at each level with those below. Terms for these final state variables have been omitted from the equation, since they are a standard topology and not germane to this paper.

Overall, unless words are very short or referents are heavily interconnected, the combination of acoustical and referential constraints in this model keeps the number of semantic transitions explored on the trellis fairly low. A version of this system maintaining the top 255 trellis hypotheses at each time step runs in real time on a dual processor 2.6GHz desktop, in tests using a world model queried in real time from a network connection, with an average fanout of 18 reachable referents from each context referent. With these runtime parameters the system was able to maintain a significant improvement in concept accuracy due to incremental semantic interpretation.[2]

2 Semantic Representation

One ostensible disadvantage of this incremental semantic model is that incrementally recognized entity descriptions will be incompletely specified during most of their existence on the Viterbi trellis. This means they must either be represented as *distributions* over individual entities, each of which consumes a trellis entry from the available beam width, or as *sets* (reified first-order referents), which exponentially increases the size of the world model over one which represents only individual entities (zero-order referents).

Fortunately, a key advantage of this incremental semantic model is that the world model can be queried only as needed by the recognizer. For example, the real-time server implementation of this interface queries the world model for an initial context referent at the start of each utterance (which may simply be a null referent). Then, when any referent makes it onto the *n*-best trellis, the referents immediately reachable from it (connected by one relation) are pre-fetched from the client world model and cached in a local world model, before any word describing one of these relations has concluded. This 'just in time' exploration means that the set of referents defined in the world model can be extremely large (e.g. the power set of a large set of entities), or even infinite, without affecting the run-time performance of the interface, as long as the level of interconnectivity among referents remains controlled.

Relations (e.g. subsumption) among referents corresponding to sets can be navigated as a graph, just like relations over individual entities. Properties (unary relations like 'jar file' or 'write-protected') can be represented in the referent transition model $\hat{\mathsf{P}}_{\Theta_{\text{SemRef}}}(l_t, e_t \,|\, v_t, e_{t-1})$ 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 $[\![l_t]\!]$ satisfying the property l_t. The world model can therefore be cast as a subsumption lattice with the set of all entities at the top $\top$ (the result of intersecting an empty set of properties) and the empty set of entities at the bottom $\bot$ (resulting from an intersection of properties denoting disjoint sets).[3] 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 '*jar files* (property *J*) that are *write-protected* (property *W*)' would be reachable by traversing a '*W*' relation from the set of jar files, or by traversing a '*J*' relation from the

[2] Achieving a concept accuracy of 73%, vs. 46% for a baseline using only syntax and acoustics, on a corpus of 160 spoken shell directives ('go to . . . ', 'add . . . to . . . ', etc.).

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

set of write-protected files, or by either path WJ or path JW from $\top$. The resulting set may then serve as context for subsequent traversals. A general template for intersective adjectives can be expressed as a grammar rule:[4]

$$\mathrm{NP}(g,h) \rightarrow \mathrm{Adj}{:}l(g,k)\ \ \mathrm{NP}(k,h)$$

where g, h and k are variables over referent sets: g is the referential context of the parent noun phrase (in some sense, the set of entities the agent is thinking about before interpreting the noun phrase), k is the result of interpreting the relation l associated with the adjective 'Adj' in context g, and h is the result of interpreting the noun phrase that follows this in context k. These referents g,h,k are similar to the *information states* in Dynamic Predicate Logic [10] if an information state is taken to be a set of entities to which the system is currently paying attention (similar to entities being 'selected' in a graphical interface). The world model may thus define semantic transition probabilities for properties $\hat{\mathrm{P}}_{\Theta_{\mathrm{SemRef}}}(l_t, e_t \,|\, v_t, e_{t-1})$ to be (say) uniform over l_t, then deterministic on e_t: equal to one if $e_t = e_{t-1} \cap [\![l_t]\!]$, zero otherwise.

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 directories (set g) that '*contain* (relation C) files that are *write-protected* (property W)' would be reachable only by:

1. pushing the original set of directories g onto the referent stack e_t, then
2. traversing a C relation departing g to obtain the contents of those directories j, then
3. traversing a W relation departing j to obtain the set of contents that are write-protected k, then
4. traversing the inverse C' of relation C to obtain the containers of these write-protected contents, then intersecting this set with the original set of directories g to get h: the directories containing write-protected files.

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,h,j,k defined above:

$$\mathrm{RR}(g,h) \rightarrow \mathrm{Verb}{:}l(g,j)\ \ \mathrm{NP}(j,k)\ \ {-}{:}l'(k,g,h)$$

where the inverse relation l' at the last, empty constituent '$-$' is intended to apply when this rule is reduced (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 inverse relation l' would depend on both referents k and g on the stack e_t.

The ability to efficiently represent first-order world models, with sets of entities as referents, is also important for modeling quantifiers, but this topic is beyond the scope of this paper.

[4] A statistically weighted grammar of such rules is used to define $\hat{\mathrm{P}}_{\Theta_{\mathrm{Syn}}}(c_t \,|\, c_{t-1}, l_t)$, where c_t and c_{t-1} are stacks of syntactic categories.

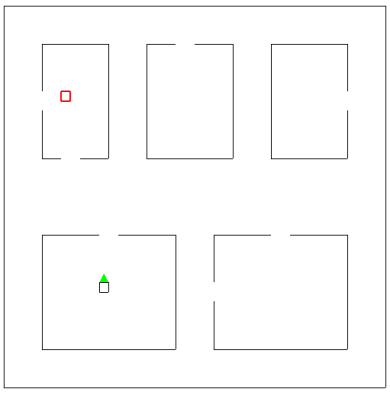

Fig. 1. A sample randomly generated SLAM-like environment. The small box with the triangle attached represents a robot (with the triangle denoting orientation). The other small box (colored red on the actual interface) represents the target to which the subject should direct the robot. Subjects were told that the target did not represent a tangible object in the world so that they did not say, for example 'Go to the red box'. Subjects were also shown these environments rotated randomly between 20 and 70 degrees to discourage use of screen-centered directions (e.g. 'Go to the top left room').

3 Continuous-Valued World States as Referents: A Robot Navigation Application

Spoken directives can refer not only to discrete objects in a model of the *current* state of the world; they can also refer to hypothetical consequences (world states) in a branching model of *possible* worlds, of the sort commonly used in AI planning. Such states may be comprised of arbitrarily large sets of discrete or continuous properties, and as such may be uncountably infinite. Modeling an infinite set of referents may seem daunting, but continuous-valued referents can be calculated only as needed by the recognizer in the context-sensitive interface system described in the previous section.

This system is therefore currently being evaluated in a robot navigation application, in which referent states have continuous location, orientation, and velocity properties. This extends the set of modeled referents from an exponentially large power set of entities in a first-order world model to an uncountably infinite set of points in a continuous multidimensional space. But since there is still a finite number of words that can be recognized at any point in a directive, the incremental semantic language model described above will still be able to track these referents and use them to constrain this search.

3.1 Map Navigation Task

This interface has been implemented in a system for giving spoken language directives for robot navigation in a simulated mobile robot with SLAM (Simultaneous Localization and Mapping) [11] capabilities. The output of a mobile robot performing SLAM

indoors with a laser range finder is a collection of points that can be regressed to a set of line segments representing the walls and doors of the environment. The goal of this application is to allow a user to direct the robot through this continuous environment without needing to micromanage the movements (such as with a joystick or other form of tele-operation). Similar work in this area includes [12], in which a human user gives navigation directions to a robot with a camera. This differs from the work described here in that recognition at the phone level is not informed by semantics, and therefore may converge on analyses that are acoustically possible but not semantically likely.

For the current work, environments representing SLAM maps were randomly generated, and subjects were asked to verbally guide a robot from the starting position and orientation inside one room in the environment to a final destination room in the environment (See Figure 1 for illustration of a sample environment). Each randomly generated environment consists of two separate rectangular building segments, placed either side by side or at right angles. Each segment contains between one and three inner blocks representing offices or rooms, and each inner block contains between one and four doors. The representation of the environment available to the simulated robot consists only of the set of line segments making up the walls and rooms.

The starting position is the center of a randomly chosen room within a randomly chosen segment, and the destination is the center of a randomly chosen room within the other segment. Spontaneous spoken descriptions were collected from a few subjects with the goal of determining what semantic modeling would be most beneficial. The collected utterances were generally either directing the robot to turn ninety degrees in either direction, or go forward until some condition was met. Thus, the main source of complexity in this domain is in the variety of different conditions that could exist for stopping forward motion. Some of the stopping conditions encountered include:

- a doorway appears on the right
- you get into the hallway
- there's a corner on your right
- there is a wall
- you hit the wall

3.2 Semantic Representation of Navigation Directives

Incremental interpretation requires a model of what referents are accessible from the current context state via the defined semantic relations. The referents in this world model represent the state of the robot, and are potentially infinite. The state of the robot consists of x and y coordinates in the environment, an orientation θ in degrees relative to the environment, and a velocity v.

As mentioned above, the robot is only aware of the environment as a collection of line segments. Using this information, the robot is able to use some simple rules to figure out where there are doorways, corners, and walls. This information can be used to predict what relations and states are reachable from the current state.

For a simple example, the robot can turn right or left from any state. So, if the robot's world model is queried for reachable states from a state with orientation $\theta = 90$, two of the returned reachable states will be the state with the new theta value $\theta = 0$ (via the

relation TURNLEFT), and the state with the new theta value $\theta = 180$ (via the relation TURNRIGHT). In both cases the relation is defined not to change any state variable other than orientation.

An example of a more difficult utterance to model is 'Go straight until you hit the wall.' Presumably the subjects did not intend to send the robot crashing into the wall, but to go straight until the path forward was blocked by a wall, in anticipation of a turning command. The first part, 'Go straight...,' is modeled with the relation GOSTRAIGHT, which is available in any state, and has the effect of setting $v = 1$, meaning the robot will be moving.

The second part of the phrase, '... until you hit the wall,' is modeled with the relation EXTRAPOLATETO with the entity referent for the wall as an argument. This relation is only available when the robot is in a moving state, as it is a stopping criterion. The effect of the relation EXTRAPOLATETO is to transition the current state of the robot to a new state in which the x and y values are changed to reflect the robot moving forward along its current trajectory as far as possible until it gets close to the entity referent of its argument (in this case the wall). This new referent state can be computed quickly on the fly by the world model from any location by finding the closest intersection point between the current trajectory of the robot and all the walls in the environment.

3.3 Semantic Representation of Number

Representation of number is a potential problem for systems such as this, because the system may be required to choose between referents that differ in the values of continuous variables. In a continuous environment such as robot navigation, the robot could have a state which includes real numbers for its location, meaning there are an infinite number of successive states. This can be a problem because there is a potentially infinite branching factor for future states, and thus potentially infinite referents for the system to evaluate at each time step.

The system described here solves this problem by modeling numbers explicitly as referents, which can be taken as arguments of relations. Rather than infinitely branching, the reference path forks to recognize this number, then rejoins to provide the number as an argument. The argument fork has on the order of ten possible branches at each time step (assuming decimal numbers).

In the robot domain described above, numerical reference could be exemplified by the sentence 'Go straight three feet.'[5] After the words 'Go straight,' the system will have hypothesized the relation GOSTRAIGHT as described above, and a resulting robot state. From the resulting state, the recognizer will query the world model for possible relations departing this state. Among these departing relations, the option DISTANCE will be available, transitioning to a real-valued measure. This relation transitions from any state referent to a referent e_N, an initially unspecified number. This referent can then transition to the referents $e_{N=1}$, $e_{N=2}$, ..., $e_{N=9}$, via the relations ONE, TWO, ..., NINE, depending on subsequent recognized words. Additional numbers will allow

[5] Subjects did not actually use any distance terms like this in our corpus, since there was no indication of scale in the environments. However, in a world model representing a real environment, it is easy to imagine speakers using this kind of directive.

transitions from (say) $e_{N=3}$ to $e_{N=3.7}$ permitting arbitrarily long number descriptions without ever encountering branching factors significantly greater than ten. At any point in the recognition process, the recognizer will be exploring hypotheses that terminate the description of this number, filling in the number referent as a distance argument to another relation.

With this framework, the world model may use non-uniform probability distributions over numbers (e.g. defining a Gaussian over likely distances for various actions), using different context referents e_N, e'_N, ... for different distributions. This continuously context-sensitive recognition of numbers also allows users to refer to real-valued referents already in the world model (e.g. 'the three-foot wall') as well as new referents.

4 Conclusion

Context-sensitive interfaces have been shown to be more accurate than interfaces that are not sensitive to context. The research described in this paper shows that context-sensitive recognition can be performed continuously during decoding, in real time, with modest hardware requirements, and can still be flexible enough to represent complex and even continuous environments.

References

1. Lemon, O., Grunstein, A.: Multithreaded context for robust conversational interfaces: Context-sensitive speech recognition and interpretation of corrective fragments. ACM Transactions on Computer-Human Interaction **11**(3) (2004) 241–267
2. Chung, G., Seneff, S., Wang, C., Hetherington, I.: A dynamic vocabulary spoken dialogue interface. In: Proc. ICSLP. (2004)
3. Haddock, N.: Computational models of incremental semantic interpretation. Language and Congnitive Processes **4** (1989) 337–368
4. Mellish, C.: Computer interpretation of natural language descriptions. Wiley, New York (1985)
5. Schuler, W.: Computational properties of environment-based disambiguation. In: Proc. ACL. (2001) 466–473
6. DeVault, D., Stone, M.: Domain inference in incremental interpretation. In: Proc. ICoS. (2003)
7. Gorniak, P., Roy, D.: Grounded semantic composition for visual scenes. Journal of Artificial Intelligence Research **21** (2004) 429–470
8. Aist, G., Allen, J., Campana, E., Gallo, C., Stoness, S., Swift, M., Tanenhaus, M.: Incremental understanding in human-computer dialogue and experimental evidence for advantages over nonincremental methods. In: Proc. DECALOG. (2007) 149–154
9. Murphy, K.P., Paskin, M.A.: Linear time inference in hierarchical HMMs. In: Proc. NIPS. (2001) 833–840
10. Gronendijk, J., Stokhof, M.: Dynamic predicate logic. Linguistics and Philosophy **14** (1991) 39–100
11. Leonard, J.J., Durraant-Whyte, H.F.: Simultaneous map building and localization for an autonomous mobile robot. In: IEEE/RSJ International Workshop on Intelligent Robots and Systems IROS '91, Osaka, Japan (1991)
12. Bugmann, G., Klein, E., Lauria, S., Kyriacou, T.: Corpus-based robotics : A route instruction example. In: Proceedings of Intelligent Autonomous Systems. (2004) 96–103