Exploiting a Sensed Environment to Improve Human-Agent Communication

Shana Watters, Tim Miller, Praveen Balachandran, William Schuler, Richard Voyles
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55345
{watters,tmill,pkbalac,schuler,voyles}@cs.umn.edu

ABSTRACT

This paper describes an implemented robotic agent architecture in which the environment, as sensed by the agent, is used to guide the recognition of spoken and gestural directives given by a human user. The agent recognizes these directives using a probabilistic language model that conditions probability estimates for possible directives on visually-, proprioceptively-, or otherwise-sensed properties of entities in its environment, and updates these probabilities when these properties change. The result is an agent that can discriminate against mis-recognized directives that do not 'make sense' in its representation of the current state of the world.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: Natural Language Processing; I.2.9 [**Artificial Intelligence**]: Robotics

General Terms

Algorithms

Keywords

language modeling, spoken language interfaces, robotics, multimodal interfaces, sensor fusion

1. INTRODUCTION

The capacity to rapidly connect language to referential meaning is an essential aspect of communication between humans. Eye-tracking studies show that humans listening to spoken directives are able to actively attend to the entities in the environment that the words in these directives might refer to or '*denote*', even while the words are still being pronounced [26, 5]. This timely access to sensory information about what input utterances might refer to in the environment may allow listeners to adjust their preferences among likely interpretations of noisy or ambiguous utterances to favor those that make sense in this context, before any lower-level recognition decisions have been made.

AAMAS'05, July 25-29, 2005, Utrecht, Netherlands.

If provided early enough in the recognition process, it is conceivable that this sensory information could significantly improve recognition accuracy of spoken language interfaces, particularly for robotic applications in which users and interfaced agents share the same environment. Moreover, a recognizer that estimates probabilities of input analyses based on the entities or relations they denote may be significantly easier to train and port across applications than existing recognizers based only on word co-occurrences in text corpora, since the associations between words and entities that would be required in order to recognize input directives would be identical to those required in order to understand and execute these directives once they have been recognized. This re-use of training data could save considerable expense in applications where task requirements are relatively mutable and trained programmers are scarce, and could facilitate the development of broadly portable artificial agents for assisting elderly or disabled users, who may have difficulty operating other kinds of controls.

This paper describes an implementation of this kind of 'environment-sensitive' interface architecture, in which the environment as sensed by a particular agent (including properties of the agent itself), is used to guide the recognition of spoken and gestural directives to that agent. Moreover, unlike existing multi-sensor interface architectures such as [12], which align input modalities on a word lattice representation after complete utterances have been recognized, the proposed approach performs referential interpretation at every frame *during* the recognition process, so that an interfaced system would be able to provice incremental feedback (gazing or gesturing at hypothesized referents) while a user is still speaking.

The remainder of this paper is organized as follows:

- Section 2 describes other recent approaches to linguistic interfaces for agents and how they relate to the current 'environment-sensitive' approach.
- Section 3 describes an extensible agent architecture which takes information about the agent's environment from whatever sensors it has, and makes it available to the agent's directive recognizers.
- Then Section 4 describes a language model used inside the agent's directive recognizers that uses the sensor information to guide the recognition of its input directives.
- Finally, Section 5 presents an evaluation of this general approach in an interactive mobile robot direction task, using wheel tachometer sensors as context.

2. BACKGROUND

Until recently, most research on using denotational meaning to guide incremental recognition has focused on purely symbolic analyses of semantic composition, using lexically-associated logical expressions as hard constraints in an effort to find a unique satisfying variable binding for each hypothesized derivation of an input utterance [11, 15]. This approach does not scale up well to the kind of spatial and temporal applications that are most likely to elicit referential descriptions however, because these hard constraints do not provide appropriate definitions for graded concepts like 'near' or 'large' or spatial relations like 'above' or 'in front of' that become graded at their boundaries. As a result, the constraints become arbitrary, and there is no guarantee that applying them will result in a unique variable binding in the correct derivation.

More recent approaches [25, 21, 2] focus on modeling continuously-graded atomic concepts such as color, shape, and motion (e.g. resulting from visual perception), but do not incorporate these into a model of deriving complete utterances from intended denotations. Recognizers have been proposed that apply continuous models of word meaning to filter the output of a purely corpus-based language model, but these are either based on finite-state grammars [22] and are unable to derive arbitrarily complex descriptions involving multiple entities, or are based on incomplete context-free derivations for utterances [10], and therefore do not define complete probability estimates for hypothesized analyses of input utterances, as conventional speech recognizers do.

The implementation described in this paper aims to fill the gap between these two general approaches by developing a complete probability model that derives entire utterances from denotational meanings, which is naturally able to incorporate continuous probabilities for graded concepts and spatial relations.

Similar approaches have been developed for successfully integrating sensory data into autonomous robot architectures, for purposes other than human-robot communication. An attentive stereo vision system was integrated with a multi-tier agent architecture onboard a mobile robot [27], and both auditory and visual perception was integrated to develop mobile robot soccer players for the RoboCup competition [17]. With these types of successes, proposals are being put forth for unified cognitive architectures for mobile robots [3], which attempt to endow a robot agent with the full range of cognitive abilities, including perception, use of natural language, learning, and the ability to solve complex problems. However, no currently implemented approach integrates sensor data into the process of recognizing users' directives, as does the system described in this paper.

3. AGENT ARCHITECTURE

As [9] states, "With seemingly no effort, the human brain reconstructs the environment from the incoming stream of - often ambiguous - sensory information and generates unambiguous interpretations of the world. To do so many different sources of sensory information are constantly processed, analysed, and combined." Similar to humans in that a number of sensory percepts are often available to make decisions, robotic agents must have an architecture built which can not only collect the sensory data but also integrate it for use in an appropriate manner (in this case, in order to more accurately recognize users' spoken and gestural directives). It's often the case where there are plenty of available sensors to collect data but there is no way for the system to integrate the data in order to make a useful decision or action determination.

We propose an agent architecture called MuSICA (Multi-Sensor Integration for Communicative Agents) to build an autonomous robotic agent capable of using both exteroceptive (visual, audio/speech) and proprioceptive (in this case, robot wheel speed) percepts obtained from the environment to more accurately recognize noisy or ambiguous directives from a human user. Instead of relying on a conventional static language model (trained on word pairs or word triples in a corpus of example sentences) to help recognize unclear utterances, the proposed robotic agent architecture is able to use its sensory data to dynamically update a probabilistic language model, based on its current environment context. This model conditions probability estimates for possible directives on the properties of entities in its environment and updates these probabilities when these properties change.

Figure 1 shows a graphical representation of the components that make up the MuSICA architecture. The components are grouped into classes relating to perception, integration, or effectors of the robot. Components in each class can therefore interact with components of other classes through a standardized interface. This simplifies the task of adding new sensors, effectors, or new types of directives to the system.

3.1 Perception

The role of the perception components are to collect data directly from the environment. These may include proprioceptive (e.g. wheel speeds of robot) and/or exteroceptive sensors (e.g. microphones, cameras, magnetic gloves). The sensors detect changes in the environment and relay these changes to low-level processing modules that translate the raw data into recognizable formats that the integration components can use.

The types of sensors that are available for use in the current implementation include:

1. A microphone using the front end and acoustical model from the CMU Sphinx 4 speech recognizer. The Sphinx 4 component is used in 'live' mode (speech is processed as quickly as a person begins talking). The raw voice data is translated by the front end and acoustical model into a probability distribution over *subphone* units (representing the beginning, middle, or end sounds of *phonemes*, which roughly correspond to sounds of the alphabet letters in spoken English).

2. 16 Polaroid 6500 sonar ranging modules (on the robotic agent: a Nomad Super Scout). The Polaroid 6500 is an acoustic range finding device that can measure distances from 6 inches to 35 feet with a typical absolute accuracy of +1 or -1 percent over the entire range.

3. Two wheel sensors (also on the Super Scout robot) that can provide the velocity of the wheels in 1/10s of inches per second, the integrated x- and y- coordinate of the robot in 1/10s of inches with respect to the start position, and the orientation of the steering in 1/10s of degrees with respect to the start orientation in the range [0, 3600].

4. An Ascension Technologies Corp. Flock of Birds sensor on a CyberGlove (Immersion Corp.) is used for gross pointing gestures. The Flock of Birds sensor provides 6-degree-of-freedom positioning of the wearer's wrist with respect to a magnetic base unit. These gestures are used to provide direction and focus attention on objects, and may occur contemporaneously with spoken directives.

5. A Sensoray frame grabber card, which grabs frames at 30Hz per second through a color camera. The color mode is RGB where there are 3 bytes for every pixel (number of bits for every pixel is 24). The video capture is done in NTSC mode.

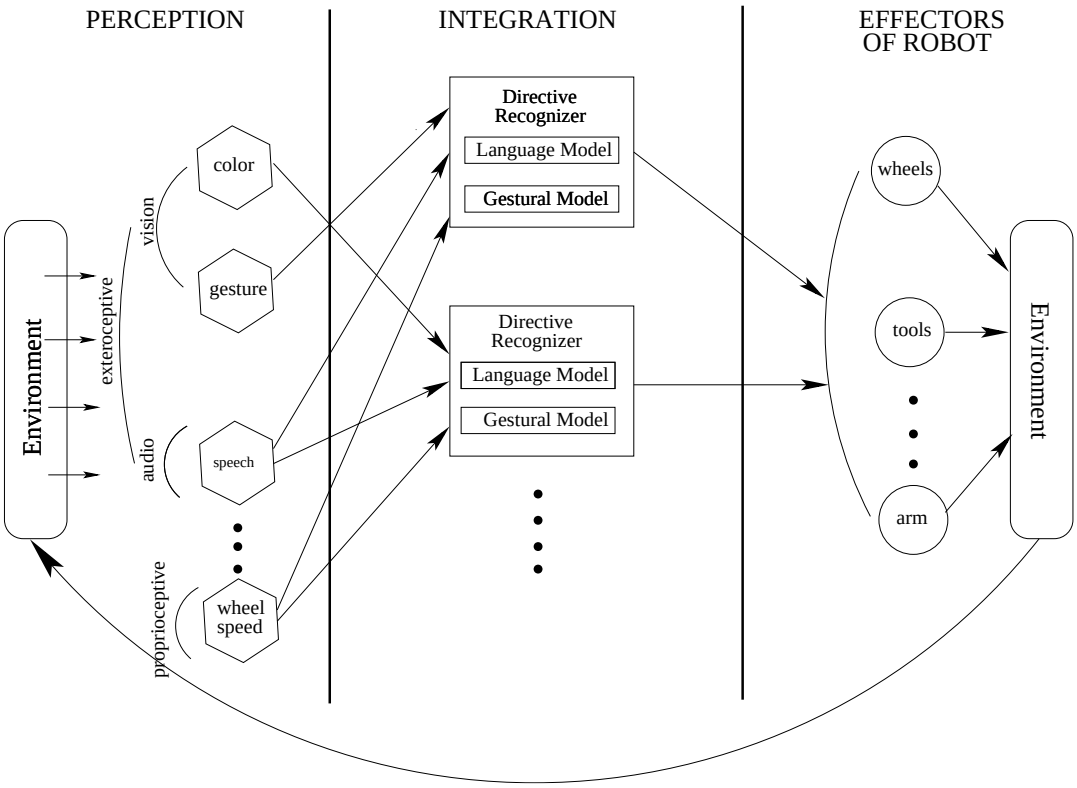

Figure 1: Agent Architecture

Pre-processors then detect likely candidates for entities in the environment using blob coloring and color histogramming; background subtraction is used to detect motion. The program controlling the frame grabber translates the raw data into a format used by the directive recognizer, which contains the number of entities, the location of the entities, and the color information of the entities.

3.2 Integration

The role of the integration components is to incorporate the agent's diverse sensor data into their internal (spoken language or gestural) recognition models. The architecture supports multiple recognizer components (running as independent threads on separate machines), each of which may be tuned to a different type of directive. For example, there may be separate components for recognizing directives for manipulatory actions, for movement actions, for directing sensors to attend to a particular area or phenomenon, etc. Distributing recognition among independent task-specific recognizers in this way allows each recognizer to consider fewer possible interpretations in its analysis, and thereby allows even complex multi-purpose agents to communicate efficiently.

Each of the recognizers can also be viewed as an autonomous agent since they are completely independent from one another and make decisions without interaction from other recognizers. Each recognizer communicates with the robot which in turn effects change to the environment.

The internal modules that make up the directive recognizers are where the actual integration of the data occurs. The types of modules used in the current implementation include:

1. Denotational Language Model: The Language Model in the current implementation receives pre-processed speech input from a perceptual component containing the front end and acoustical model from the CMU Sphinx 4 speech recognizer [13]. This component accepts acoustical signals from a (headset) microphone and sends the data to a directive recognizer as the probabilities of sub-phonetic units (beginning, middle, or end of *phonemes*, which roughly correspond to sounds of the alphabet letters in spoken English). The language model in the directive recognizer then builds a probability distribution over hypothesized directives using the environment information it receives from the agent's sensors. This environment-sensitive language model is described in further detail in Section 4.

2. Gestural Model: The thread corresponding to the gestural model communicates with a simplified gesture preprocessor that in turn interprets various constrained pointing gestures from data from the Flock of Birds. The gesture model in this directive recognizer builds a probability distribution over positions and velocities indicating valid and invalid pointing gestures based on recognized relational utterances (e.g. "that").

3.3 Effectors of the Robot

The role of the effectors of the robot is to act upon the commands the robot has received from the directive recognizers and has chosen to act upon. Effectors are devices that the robot uses to effect change in the environment. In the current implementation the only effectors that are used are the Nomad Super Scout mobile robot's two motor-driven wheels. Although the current system only utilizes the wheels, it is capable of directing other types of effectors such as robot arms, grippers, and tools.

3.4 Component Interfaces

The architecture is built so components in each class can interact with components of other classes through standardized communication interfaces. This allows new sensors, effectors and directive recognizers to be easily accommodated since the only specifications a new component must provide is how the data is sent, where the data is sent/received, and the required format structure. The two basic interfaces built into the system are the 1) perception components to integration components and the 2) integration components to effectors of the robot. Both layers are built using TCP/IP connections which provides a reliable, point-to-point communication channel using sockets. This allows any program component to be written in either C, C++, or Java.

4. AN ENVIRONMENT-SENSITIVE ('DENOTATIONAL') LANGUAGE MODEL

Most modern spoken language interfaces recognize spoken directives using probabilistic models, which assign a probability estimate $\mathsf{P}(Y \mid X)$ for each possible message Y that might have been intended by the user, given an uncertain observed acoustical signal X. This probability is then decomposed, using Bayes law, into a *language model*, consisting of a prior probability $\mathsf{P}(Y)$ of the intended message Y, and an *acoustical model*, consisting of a posterior probability $\mathsf{P}(X \mid Y)$ of the observation X given the message Y (divided by an additional prior probability of the observation $\mathsf{P}(X)$, which can be eliminated because it is constant across all possible analyses):

$$\mathsf{P}(Y \mid X) = \frac{\mathsf{P}(X,Y)}{\mathsf{P}(X)} \quad (1)$$

$$= \frac{\mathsf{P}(X \mid Y) \cdot \mathsf{P}(Y)}{\mathsf{P}(X)} \quad (2)$$

$$\propto \mathsf{P}(X \mid Y) \cdot \mathsf{P}(Y) \quad (3)$$

The interface then assumes the message Y which generates the observation X with the highest estimated probability must be the user's intended message, and substitutes it for the observation in later processing. Most language models used in automatic speech recognition systems generate word strings as intended messages in the prior model $\mathsf{P}(Y)$ using '*n-gram*' probabilities of n-word sequences (for example, 'trigrams' of three-word sequences) in a set of training sentences. However, these purely-word-based models do not provide any notion of phrasal or clausal constituent structure, which will be necessary in order to distinguish the different environment entities that may be involved in a directive (e.g. those denoted by the subject and object of a directive) if the model is expected to be sensitive to these denoted entities.

These phrasal or clausal constituents can be learned from a general corpus of transcribed directives that are annotated with brackets delimiting noun phrases, verb phrases, relative clauses, etc. (see Figure 2.a.). Although this is a very detailed kind of annotation, it can be partially automated using existing broad-coverage parsers. Moreover, since it is likely that the syntactic patterns encoded in these rules will be generally applicable, this annotation need only be done once (the resulting rules can then be ported to different environments).

These phrase-structure trees are then mapped to a Dynamic Bayes Net (DBN) representation [7] via a variant of the left-corner grammar transformation [20, 1], which preserves an explicit representation of constituents while minimizing the number of stack position required to recognize the string using an incremental recognizer

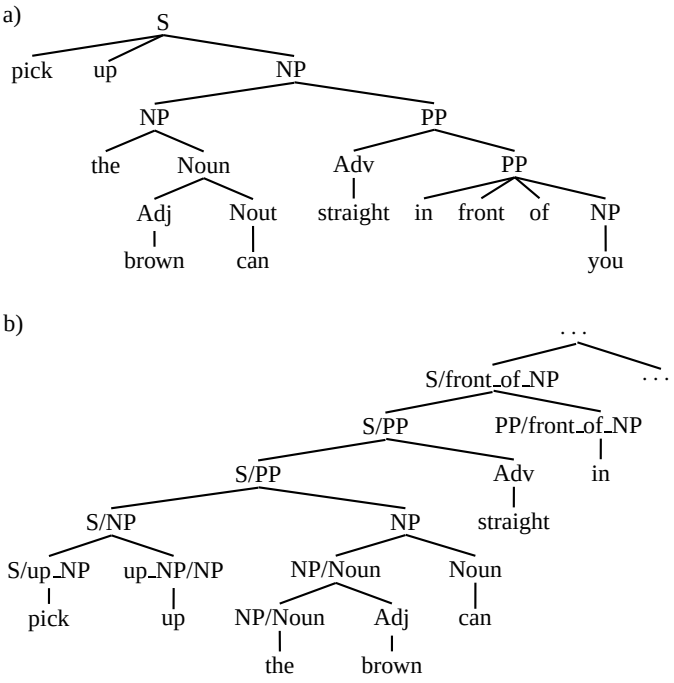

Figure 2: (a) ordinary phrase structure tree and (b) right-corner transform of this tree for the sentence 'pick up the brown can straight in front of you.'

(such as a pushdown automaton).[1]

This right-corner DBN model is similar to the Hidden Markov Models (HMMs) that keep track of probability distributions over state variables in conventional speech recognizers, but allows a bounded stack of variables over incomplete constituents (phrases and clauses) to be maintained, rather than a single state variable, at every point in time. These incomplete constituents are represented by 'slashed' categories like VP/NP, representing a verb phrase (VP) lacking a noun phrase (NP) to follow it – or in other words, a transitive verb.

In this manner, right-corner derivations can be recognized incrementally while still preserving explicit representations of intermediate constituents at all levels of the integrated model: e.g. representing subphones (corresponding to the onset, middle, and ending sounds of individual phonemes, extracted from Sphinx's existing acoustical models [24]) in the DBN's lowest ($i = 0$) level, partial phonemes in the next ($i = 1$) level (which is isomorphic to a hidden Markov model, also extracted from existing acoustical models), partial words in the following ($i = 2$) level, and partial phrases at subsequent ($i > 2$) levels, until eventually the denotation of a complete sentence can be recognized in the top level, at the end of the utterance.

When they are mapped to the right-corner DBN model (see Figure 3), the transformed trees are decomposed into sets of recognition rules, which are probabilistically weighted based on their frequency in the training corpus. The right-corner DBN (shown in

[1]This observation has been used to justify constraints on internal recursion (e.g. arising from rules of the form 'S → a S b'), but not left or right recursion (e.g. arising from rules of the form 'S → S a b' or 'S → a b S'), by converting context-free grammars into finite state automata [4, 18], but this conversion makes constituent structure unavailable for interpretation at run time.

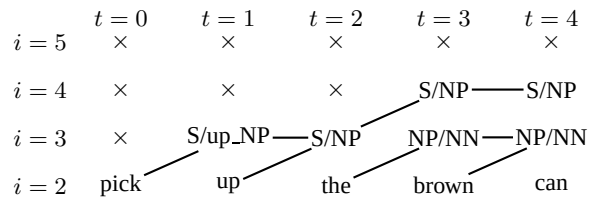

Figure 3: Right-corner derivation of 'pick up the brown can straight in front of you,' mapped to random variable positions in DBN. Taken together, each stack forms a complete analysis of the recognized input at every time frame t.

Figure 4) uses two kinds of rules, each mapping a pair of adjacent incomplete constituents L_t^{i-1} and L_t^i to a higher-level pair of adjacent incomplete constituents L_{t+1}^i and L_{t+1}^{i+1}. One kind of rule combines the two adjacent incomplete constituents (e.g. VP/NP and NP) into another possibly incomplete constituent (e.g. VP) in L_{t+1}^i, leaving L_{t+1}^{i+1} to be filled in by a higher-level rule; and the other kind of rule passes the less recently recognized constituent L_t^i up to the next higher level L_{t+1}^{i+1}, so that it can be combined with some later constituent when one becomes available. The probabilities governing these two kinds of rules (induced over the DBN-mapped transformed grammar) is called a *composition model*.

Phrasal or clausal constituents from phrase-structure-annotated corpora can then be associated with denoted entities (for example, the segmented region of pixels that a noun phrase 'the brown can' refers to in a training example) before they are mapped to a right-corner DBN and decomposed into composition rules. The resulting rules do not have to be specific to the entities in the training environment, however. The specific entities can be abstracted away using 'coindexation patterns' which describe the entities denoted by each resulting constituent purely in terms of coindexations from entities used in the composed constituents. Although some structural information is lost in certain parts of the right-corner transform (e.g. the prepositional phrase in Figure 3 could be a noun phrase or a verb phrase modifier), these coindexation patterns ensure that the dependency information from the original phrase structure tree will be preserved.[2]

Entities must still be initially introduced at some point, however. This is done using a *lexicalization model* which defines the probability with which a word (a symbol at some particular 'lexical' level of the DBN) can be replaced with a syntactic category and a set of entity features, allowing distributions over syntactic categories and denoted entities to be easily calculated for references to entities in any new environment.

4.1 Lexicalization model

The lexicalization model is where the sensor information interacts with the right-corner DBN language model. It controls the probability with which a word W (for example, the word 'covers,' recognized by its component phonemes), may be replaced with an incomplete constituent category P (for example VP/NP) and an associated entity or vector of entities E in the current environment (in this case, the entity that is covered and the entity that is covering it).

[2]It should be pointed out however that the bounded stack of the DBN representation limits the number of available PP attachment configurations to those that involve only a bounded amount of internal recursion (see [14, 19] for arguments that such limitations are appropriate for modeling human language).

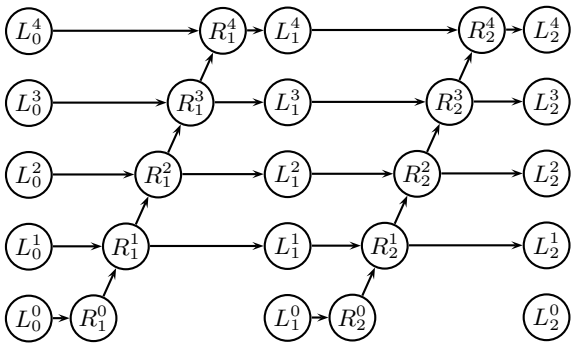

Figure 4: DBN implementation of finite-stack recognizer. Random variables $L_t^0, L_t^1, L_t^2, L_t^3, L_t^4$, at each time frame t assign probability distributions to instances of (possibly incomplete) constituent labels. Random variables $R_t^0, R_t^1, R_t^2, R_t^3, R_t^4$ at each time frame t assign distributions to rules for combining labels, propagating labels to higher levels, or retaining labels at their current levels.

The lexicalization model takes as input a vector of features representing properties of entities in the environment, and returns a distribution over these entities. In the current implementation, the possible features include the (wheeled robot) agent's left and right wheel speeds, the slope and intercept of a line representing the direction of a pointing gesture by the speaker (as seen from an overhead camera), and X, Y-coordinate locations and H, S, V (hue, saturation, value) scores for average colors of entities in the world (segmented color-blobs in images from the same overhead camera). Since these different sensor modalities are not always available simultaneously, this model takes in feature "packets" which are tagged with the type of input represented. For example, a new reading for wheel speed will be of the form "w 4.3 4.8."

The desired output of the lexicalization model is:

$$\mathsf{P}(E, P \mid W) = \mathsf{P}(E \mid P, W) \cdot \mathsf{P}(P \mid W), \tag{4}$$

where E is a vector of entities, P is a pre-terminal symbol as described in Section 4 (e.g. NP/NN), and W is a terminal symbol, i.e. a word. $\mathsf{P}(E \mid P, W)$ and $\mathsf{P}(P \mid W)$ are obtained from the same training data used for building the syntactic models, which consists of a corpus annotated with syntactic constituents and denotations. For a given word in the training set, its word model, $\mathsf{P}(E \mid P, W)$, is built by collecting the features of all entities that it referred to in the training set. A gaussian distribution over the feature set is obtained using Maximum Likelihood Estimation (MLE), and is then normalized to the set of entities (each entity is assigned a probability proportional to the probability of its features). $\mathsf{P}(P \mid W)$ is obtained using frequency counts in the syntactic corpus. When the lexicalization model receives a feature packet, it uses the word model $\mathsf{P}(E \mid P, W)$ and $\mathsf{P}(P \mid W)$ to compute a distribution over entities and pre-terminals for every modeled word.

5. EVALUATION

The evaluation of the system's capabilities was performed using two different experiments. The first experiment evaluated the system's ability to accurately yield correct denotations of complex directives and the second experiment evaluated the benefit of using an environment- sensitive communicative agent architecture.

5.1 Experiment 1

The denotational language model was evaluated on collected directives to a voice-directed mobile manipulator arm in front of a shelf stacked with everyday household objects (cereal boxes, soft drink cans, etc.), which was photographed using a 3-D laser scanning camera.[3] The resulting 3-D point cloud was polygonized into a triangle mesh and segmented into entities e_i corresponding to convex regions of this mesh, each with continuous features $\vec{f}_{e_i}$ specifying the entity's size (exposed surface area), shape (ratio of longest to second longest perpendicular dimensions), spatial location (3-D coordinates of centroid), and color (average hue, saturation, and intensity over all pixels in the segment). Word meanings were modeled for adjectives and prepositions using multivariate gaussians in this feature space (defined on color, size, and shape features for adjectives, and on differences in centroid coordinates for prepositions), which were developed partially by hand as a domain-independent language resource. Verbs and common nouns were considered domain-specific and were trained automatically on a version of the collected corpus of arm directives that was annotated with phrase structure (labeled brackets) and constituent denotations (in the associated training environment). The compositional model was trained on (right-corner transforms of) the denotation-annotated phrase structure trees in this same annotated corpus. All training and testing using this corpus was done using the leave-one-out method of cross-validation.

The accuracy of the sentence-level denotations obtained from the integrated denotational language model was tested against that of denotations obtained by parsing and interpreting the single sentence output of a trigram HMM-based language model trained on transcriptions of the same collected corpus, using a parser and interpreter trained on the annotated version of the same corpus (again using leave-one-out cross-validation). Due to the large amount of noise in this rich environment, the single-best pipelined language model yielded 0/165 sentences with correct denotations; whereas the integrated denotational model yielded 54 parses, 10 of which had correct denotations ($p<.1$ due to chance) – a statistically significant improvement ($p<.01$ using a two-tailed t-test). These results were fairly evenly distributed across task environments.

5.2 Experiment 2

The benefit of this environment-sensitive communicative agent architecture was evaluated using a relatively self-contained subset of recognizable directives relating to wheel movement: '*start/stop moving*,' '*start/stop turning left/right*,' and the relevant sensors for the denotational recognizer: left and right wheel tachometers. 75 input utterances were collected by asking two subjects to direct a voice-controlled mobile robot using the above commands.

The 'lexicalization model' part of the denotational language model (as described in Section 4) was trained on a small set of moving and turning scenarios staged by a trainer, consisting of three scenarios for each directive. These scenarios were intended to correspond to the preconditions of 'sample events' that might be provided for each type of directive by an (experienced) user teaching the system different ways of changing trajectory. The system was then supplied with a pre-existing 'composition model' (as described in Section 4) trained on directives that were transcribed and annotated with phrase and clause constituents, and with the objects (in some independent training scenario) denoted by each constituent.

An agent with the integrated environment-sensitive (denotational) recognizer described in the previous sections was compared with a baseline agent whose directive recognizer did not incorporate this kind of environment information. This baseline recognizer used a conventional trigram Hidden Markov Model (HMM)-based language model trained on the sample set of directives described above. Of the 75 collected utterances, the integrated denotational model recognized 71 correctly, whereas the baseline HMM-based model recognized only 58 correctly. This represents a 70% reduction in recognition error due to the environment-sensitive / denotational architecture. This is a statistically significant improvement with $p \leq .01$ using a two-tailed t-test.

The linear-time recognizer ran in approximately 10 to 20 times real time (so on average, a one-second utterance takes 10 to 20 seconds to process) on a 2.4GHz Pentium 4 desktop computer. This processing speed is on par with that of other experimental systems used in speech recognition evaluations, and can be optimized to run more efficiently through various techniques [6].

[3]Subjects were asked to direct the manipulator arm to pick up several objects from the shelf. The objects were visually designated (by pointing), in order to avoid biasing subjects toward any linguistic description. As a result, some of the collected directives contain very long, complex definite descriptions. The manipulator arm was a non-functional prop during this data collection.

6. CONCLUSION AND FUTURE WORK

This paper has described an implemented robotic agent architecture in which the environment, as sensed by the agent, is used to guide the recognition of spoken and gestural directives given by a human user. This architecture has been observed to reduce recognition error by up to 70% over a conventional HMM-based spoken language interface in controlled tests. Moreover, this architecture allows hypothesized denotations to be dynamically extracted during recognition, while users are still speaking, presenting the exciting possibility of allowing an interfaced agent to provide incremental feedback in other modalities, e.g. using gaze or pointing gestures to indicate understanding (or misunderstanding) of indented meanings. In future work, this possibility will be examined in greater detail.

Also, following [23], the denotational model described in this paper models references using tractable distributions over candidate entities or relations (tuples of candidate entities) of bounded arity. Others [8] have correctly argued that this representation is inadequate for intensional references such as goals or destinations which do not correspond to existing entities. This approach is therefore being extended to employ sampled spatial coordinates or attribute values in a particle filter as potential referents for descriptions of spatial regions or hypothetical entities. This extension is independently motivated by the efficiency of this technique for approximate probability estimation in complex multiply-connected time-series models such as those described above [16].

For further information and downloads of right-corner DBN tools, visit `http://www.cs.umn.edu/research/nlp/`.

7. ACKNOWLEDGMENTS

This research is supported by grants from the Digital Technology Center Initiative Program and the Grant-In-Aid Program at the University of Minnesota Twin Cities, and by National Science Foundation CAREER award 0447685.

8. REFERENCES

[1] A. V. Aho and J. D. Ullman. *The Theory of Parsing, Translation and Compiling; Volume. I: Parsing*. Prentice-Hall, Englewood Cliffs, New Jersey, 1972.

[2] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. M. Blei, and M. I. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.

[3] D. P. Benjamin, D. Lonsdale, and D. Lyons. Integrating perception, language, and problem solving in a cognitive agent for a mobile robot. In *Proceddings of the Third Internation Joint Converence on Autonomous Agents and Multiagent Systems (AAMAS'04)*, pages 1310–1311, New York, NY, July 2004.

[4] A. W. Black. Finite state machines from feature grammars. In *Proceedings of the International Workshop on Parsing Technologies*, pages 277–285, Pittsburgh, 1989.

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

[6] J. Davenport, L. Nguyen, S. Matsoukas, R. Schwartz, and J. Makhoul. The 1998 BBN Byblos 10x real time system. In *1999 DARPA Broadcast News Workshop*, pages 262–264, Herndon, VA, Feb. 1999.

[7] T. Dean and K. Kanazawa. A model for reasoning about persitence and causation. *Computational Intelligence*, 5(3):142–150, 1989.

[8] D. DeVault and M. Stone. Interpreting vague utterances in context. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING '04*, pages 1247–1253, 2004.

[9] M. O. Ernst and H. H. Bulthoff. Mergining the sense into a robust percept. *TRENDS in Cognitive Science*, 8(4):162–169, Apr. 2004.

[10] P. Gorniak and D. Roy. Grounded semantic composition for visual scenes. *Journal of Artificial Intelligence Research*, 21:429–470, 2004.

[11] N. Haddock. Computational models of incremental semantic interpretation. *Language and Congnitive Processes*, 4:337–368, 1989.

[12] M. Johnston, S. Bangalore, G. Vasireddy, A. Stent, P. Ehlen, M. A. Walker, S. Whittaker, and P. Maloor. Match: an architecture for multimodal dialogue systems. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics (ACL'02)*, pages 376–383, 2002.

[13] P. Lamere, P. Kwok, W. Walker, E. Gouvea, R. Singh, B. Raj, and P. Wolf. Design of the CMU Sphinx-4 decoder. In *8th European Conference on Speech Communication and Technology (EUROSPEECH 2003)*, Geneva, Switzerland, Sept. 2003.

[14] G. Miller and N. Chomsky. Finitary models of language users. In R. Luce, R. Bush, and E. Galanter, editors, *Handbook of Mathematical Psychology*, volume 2, pages 419–491. John Wiley, 1963.

[15] D. Milward. Incremental interpretation of categorial grammar. In *Proceedings of the 7th Conference of the European Chapter of the ACL*, pages 119–126, Dublin, Ireland, 1995.

[16] K. P. Murphy and M. A. Paskin. Linear time inference in hierarchical hmms. In *Proceedings of Neural Information Processing Systems*, pages 833–840, 2001.

[17] H. G. Okumo, Y. Nakagawa, and H. Kitano. Integrating auditory and visual perception for robotic soccer players. In *Systems, Man, and Cybernetics (IEEE SMC '99) Conference Proceedings*, pages 744–749, Tokyo, Japan, Oct. 1999.

[18] F. C. N. Pereira and R. N. Wright. Finite-state approximation of phrase structure grammars. In *Meeting of the Association for Computational Linguistics*, pages 246–255, 1991.

[19] S. Pulman. Grammars, parsers and memory limitations. *Language and Cognitive Processes*, 1(3):197–225, 1986.

[20] S. J. Rosenkrantz and P. M. Lewis, II. Deterministic left corner parser. In *IEEE Conference Record of the 11th Annual Symposium on Switching and Automata*, pages 139–152, 1970.

[21] D. Roy. Learning words and syntax for a visual description task. *Computer Speech and Language*, 16(3):353–385, 2002.

[22] D. Roy, P. Gorniak, N. Mukherjee, and J. Juster. A trainable spoken language understanding system for visual object selection. In *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP'02)*, pages 593–596, 2002.

[23] W. Schuler. Using model-theoretic semantic interpretation to guide statistical parsing and word recognition in a spoken language interface. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL '03)*, pages 529–536, Sapporo, Japan, 2003.

[24] K. Seymore, S. Chen, S.-J. Doh, E. Eskenazi, E. Gouvea, B. Raj, M. Ravishankar, R. Rosenfeld, M. Siegler, R. Stern, and E. Thayer. The 1997 CMU Sphinx 3 English broadcast news transcription system. In *Proceedings of the 1998 DARPA Speech Recognition Workshop*, pages 55–59, 1998.

[25] J. M. Siskind. Visual event classification via force dynamics. In *AAAI/IAAI*, pages 149–155, 2000.

[26] 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.

[27] G. Wasson, D. Kortendamp, and E. Huber. Integrating active perception with an autonomous agents. In *Proceedings of the Second International Conference on Autonomous Agents*, pages 325–331, Minneapolis, MN, may 1998.