A TRAINABLE SPOKEN LANGUAGE UNDERSTANDING SYSTEM FOR VISUAL OBJECT SELECTION

Deb Roy, Peter Gorniak, Niloy Mukherjee, and Josh Juster

The Media Laboratory
Massachusetts Institute of Technology
20 Ames Street, Cambridge, MA 02142

ABSTRACT

We present a trainable, visually-grounded, spoken language understanding system. The system acquires a grammar and vocabulary from a "show-and-tell" procedure in which visual scenes are paired with verbal descriptions. The system is embodied in a table-top mounted active vision platform. During training, a set of objects is placed in front of the vision system. Using a laser pointer, the system points to objects in random sequence, prompting a human teacher to provide spoken descriptions of the selected objects. The descriptions are transcribed and used to automatically acquire a visually-grounded vocabulary and grammar. Once trained, a person can interact with the system by verbally describing objects placed in front of the system. The system recognizes and robustly parses the speech and points, in real-time, to the object which best fits the visual semantics of the spoken description.

1. INTRODUCTION

A central problem in the design of speech and text understanding systems is the representation of the meaning of words and sequences of words. We are investigating applications in which spoken language is used to refer to objects in physical environments. These applications highlight the importance of designing semantic representations that bridge the linguistic world of symbolic structures and the continuous, non-symbolic world of visual observations. Emerging application domains that require visually-grounded language understanding include verbal communication with robots and video retrieval by natural language (cf. [1, 2, 3]).

We describe a visually-grounded spoken language understanding system named Newt. Newt processes spoken referring expressions such as "The green apple to the left of the cup" and locates the appropriate object in a visual scene. Newt is embodied in an active vision system mounted on a two degree-of-freedom pan-tilt base (Figure 1). The vision system includes two color video cameras. The system reported here, however, uses only monocular vision. A small laser mounted between the cameras is used to point to objects on a table top in response to spoken utterances.

This investigation builds on previous work in which we developed a trainable speech *generation* system called DESCRIBER [4]. For DESCRIBER, training input consists of a set of synthetically generated scenes of rectangles of varying shape, color, size, and location paired with spoken descriptions of target objects within the scenes. DESCRIBER learns a three-level visually-grounded language model based on this input. The trained system generates syntactically correct, semantically accurate, and contextually appropriate referring expressing of objects in synthetic scenes. A typical utterance produced by DESCRIBER is "the bright red square to the right of the large rectangle".

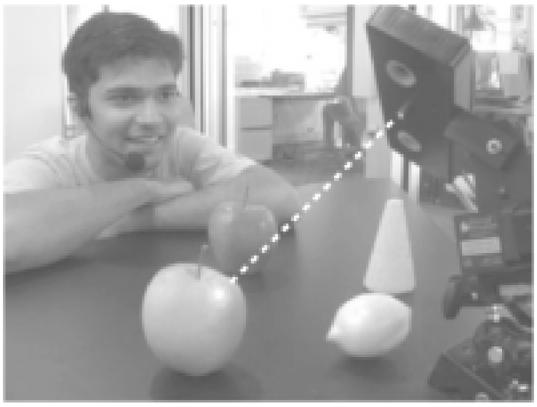

Fig. 1. In response to a spoken object description, Newt (right part of photo) points its laser at the best matching object (path of laser indicated with a dashed line).

The goal of the work reported in this paper was to 'invert' the processes in DESCRIBER, resulting in a speech understanding system for a related but more difficult task. Instead of generating referring phrases of selected objects, we are now interested in finding objects which match the meaning of novel spoken referring phrases. Additionally, the synthetic visual input used in DESCRIBER has been replaced by a real-time color visual system in Newt.

This paper proceeds by first describing Newt's visual and speech processing sub-systems, and then its language acquisition and understanding components.

2. VISUAL SYSTEM

Newt's visual system tracks solid-colored objects placed on a table top in real-time. The system extracts object properties and inter-object spatial relationships which are passed to the language processing system.

2.1. Object Tracking

We model the color distribution of objects using mixtures of Gaussian distributions. Although all objects are constrained to be single-colored, shadow effects of three-dimensional objects necessitate the use of a mixture of Gaussian distributions.

For each object used in the experiments, a color model is created by collecting training images of each object and manually specifying the region within each image that corresponds to the object. The Expectation Maximization (EM) algorithm is used to estimate both the mixture weights and the underlying Gaussian parameters for each object. K-means clustering is used to provide initial estimates of the parameters.

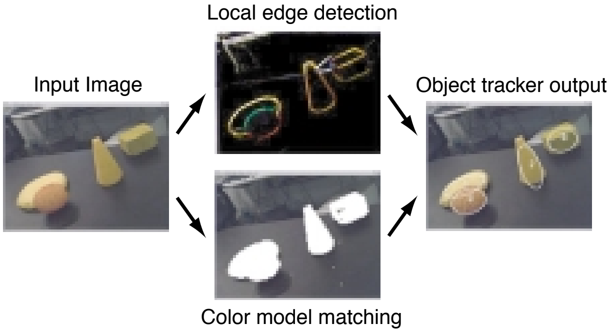

Fig. 2. Object detection combines local edge finding with color-based figure-ground separation.

As shown in Figure 2, input images are split into two parallel processing paths. The first process performs local edge detection in each of the RGB color planes. The outputs from the edge detectors of all three planes are summed to provide an overall estimate of local edges. This image is thresholded, resulting in a binary image in which pixels located at edges are set to 0. We refer to this image as the edge mask. In the second process, each 5x5 patch of pixels is classified as belonging either to one of the known set of objects or to the background. This is done by evaluating each color model (described above) for each pixel patch and thresholding the resulting values. Patches which have high matches with any color model are set to 1 and the remainder are classified as 0. The lower middle image in Figure 2 shows the result of this stage. We refer to this as the object mask.

A final step merges the information from the two processes by performing a pixel-wise multiplication of the edge and object masks. A contour finding algorithm [5] identifies connected regions in the resulting binary image. The integration of edge masks and color based object masks deals effectively with partial occlusions of different colored objects. In Figure 2, four objects are identified and assigned unique indices.

2.2. Object Properties

Once object contours are determined, object properties are extracted and passed to the language acquisition / understanding system. The features extracted include object color, shape, center of area coordinates and bounding box parameters.

The color of an object is represented by the mean of 10x10 pixels centered on the object. The shape of an object is represented by a visual distance metric first proposed in [6]. For each pair-wise pixel on the edge of an object, (i, j), the Euclidean distance d_{ij} is computed and normalized by the mean distance between all pairs of edge points. A one-dimensional histogram H is accumulated for all such pair-wise edge pixel combinations. The inter-pixel distances are divided into 8 bins, i.e., H is composed of 8 bins. The distance between two object shapes is estimated by computing the χ^2 divergence between the objects' shape histograms.

An additional set of four shape related features is computed based on the bounding box of each object. These parameters are: height, width, height-to-width ratio, and area.

2.3. Inter-object Spatial Relations

To enable the system to ground the semantics of spatial terms such as "above" and "to the left of", a set of spatial relations similar to [7] is measured between each pair of objects. The first feature is the angle (relative to the horizon) of the line connecting the centers of area of an object pair. The second feature is the shortest distance between the edges of the objects. The third spatial feature measures the angle of the line which connects the two most proximal points of the objects.

In summary, the vision system provides real-time tracking of multiple objects at approximately 15 frames per second. For each frame, the system determines the number of objects in the scene, a set of object properties and spatial relations.

3. SPEECH RECOGNITION

We have significantly extended our continuous speech recognition system [8] to support processing of interactive spoken language. The recognizer performs real-time medium vocabulary (up to 1000 word) recognition. We chose to develop our own recognizer in anticipation of non-standard decoder features which will be necessary to support rich integration with visual processing. The current system uses a 24-band Mel cepstra representation of acoustics. Subword units consist of triphone modeled by continuous density, three state, Hidden Markov Models. A back-off trigram language model is trained from a mixture of domain specific and domain general speech transcriptions. Speech decoding is accomplished using a time-synchronous Viterbi beam search. A tree-based lexicon network represented by a finite state transducer constitutes the search space in our implementation. The finite state transducer incorporates language model parameters dynamically.

In the standard MAP decoding approach, the recognizer outputs a string of words corresponding to the most likely state sequence. Relying on a single word recognition hypothesis, however, leads to brittle behaviour since recognition errors are inevitable. To gain robustness in our system, acoustic ambiguities are compactly represented using a reduced word lattice structure based on the methods reported in [9]. The word lattice incorporates posterior probabilities based on acoustic and language model parameters. The lattice possesses the topology of a directed acyclic graph that is represented as a weighted Non-deterministic Finite State Acceptor (NFSA). The NFSA is determinized and minimized to yield a compact posterior word graph representing the set of candidate hypotheses. Such a word graph represents edge weighted word level confusions in a compact sequential format. Ambiguous representations are preferred since they retain multiple hypotheses which are better resolved in later stages of processing. Currently, Newt only uses the best hypothesis, but we plan in the near future to consider multiple parses within the word lattice, constrained by semantic possibilities encoded by the visual system.

4. LANGUAGE ACQUISITION FROM SHOW-AND-TELL

Newt learns visually grounded language by a ‘show-and-tell’ procedure. During training sessions, Newt randomly points (using its laser pointer) to one of the objects in its view and waits for the trainer to speak. Newt then forwards the features extracted by the visual system and the utterance detected by the speech recognizer as a training example to its language acquisition system, and proceeds to point to another object on the table. The language acquisition system is distributed amongst a large number of separate processes, each responsible for a different aspect of learning to robustly parse visually-grounded language. The system includes

w-procs - word processes, that each attach to a single unique word and estimate context-free visual semantics for this word,

c-procs - cluster processes, that measure similarities between the w-procs and encapsulate the output of similar w-procs to make them appear to originate from a single process,

g-procs - grammar processes, that combine the outputs of several other processes covering adjacent parts of an utterance using a set of possible combination functions, and an

a-proc - an action process, that listens to all other process activations occurring for a given utterance and rewards correct outputs or acts according to the outputs, depending on whether the utterance appears to be a training example (i.e. is paired with an indicated object) or a command.

The central idea is to gain robustness in parsing by using a distributed set of processes, each of which parses only an island of words within the utterance. The processes self-assemble in hierarchical structures to explain complete utterances and bind them to visual referents. We describe the acquisition of this distributed grammar in the following sections.

4.1. Visually-Grounded Word Learning

Training examples consist of visual features of a target object and its spatial relation to other objects paired with transcriptions of spoken descriptions provided by a human trainer. As training examples arrive, a spawning process ensures that a process is created for each unique word encountered. These word processes (*w-procs*) are responsible for modeling the visual grounding of a word in a context-free manner. For example, the w-proc responsible for ‘blue’ tries to answer the question: How did the objects I saw when I heard ‘blue’ differ from the other objects I have encountered? To do so, each w-proc estimates a multivariate Gaussian over all the features offered by the visual system, conditioned on the occurrence of its assigned word. When the word associated with the w-proc occurs, the process both gives the features of the object that Newt is pointing to as an example to the Gaussian, and publishes the current Gaussian to other running processes. Whenever activated this way, the w-proc also measures how strongly its word w is grounded by computing the Kullback-Leibler distance $KL(w)$ between the word-conditioned Gaussian and a Gaussian background model that is compiled from all objects ever encountered. If its grounding strength is ever below a certain threshold, the process dies. This happens for words that are not grounded in basic visual features in Newt’s world, like ‘the’.

Along with each w-proc a clustering process (c-proc) is spawned and attached to the w-proc. Periodically this c-proc polls all other c-procs for their grounded similarity to itself. This grounded similarity is computed by first forming the semantic word profile

$$s(w) = \begin{bmatrix} KL(w) - KL_{\overline{f_1}(w)} \\ KL(w) - KL_{\overline{f_2}(w)} \\ \vdots \\ KL(w) - KL_{\overline{f_n}(w)} \end{bmatrix}$$

where $KL_{\overline{f_i}}$ is the Kullback-Leibler distance computed after removing feature i from the Gaussian. This profile measures how much each feature contributes to the overall grounding strength of the word. The grounded distance is then calculated as

$$d(s(w_i), s(w_j)) = (s(w_i)/|s(w_i)|)^T (s(w_j)/|s(w_j)|)$$

which measures the degree of overlap between the feature of two words. If the similarity is greater than a threshold, the two c-procs combine into one, relinking all w-procs associated with either to feed into the new c-proc. For a c-proc with multiple associated word groundings, all combinations of similarities between words are calculated and averaged to give a consensus similarity. In effect, c-procs funnel the activations of multiple similar w-procs to make them look identical in origin to higher level processes.

4.2. Learning Grounded Grammar Fragments

As w-procs begin to be activated by words and c-procs channel their outputs, they take responsibility for parts of the utterance heard. For example, the ‘blue’ w-proc signals that it can interpret, in a context-free sense, the occurrence of the word ‘blue’ in a specific location in the current utterance. Whenever two processes cover utterance fragments either next to each other (where ungrounded words are ignored or taken as arguments, as discussed below), the spawning process ensures that grammar processes (g-proc) exist to attempt to combine the outputs of these processes. Each g-proc is linked to two other processes’ output and performs a specific function to attempt to combine them. For example, a g-proc might perform logarithmic pooling on the Gaussians it receives. For Newt, such a process accounts for the concept associated with the phrase ‘yellow cone’, made up of the concepts attached to ‘yellow’ and ‘cone’. During training, other g-procs identify the objects best indicated in the current visual scene by the two Gaussians they receive, and compute a new Gaussian on the spatial features measured between these objects. G-procs are also spawned for grounded concepts that occur with ungrounded words in between them or surrounding them. In this way, ‘rectangle above the’ is grounded in spatial features, where ‘rectangle’ describes every object ever encountered and thus turns up as ungrounded. Notice that g-procs can stack to arbitrary depths, parsing phrases like ‘the large horizontal blue rectangle below the thin red square’.

4.3. Reinforcement Feedback

An active process (a-proc) listens to the output of all processes and tries to account for fragments of an utterance. It computes the probabilities for each object in the scene based on each process’ output, and sends a reward to the process that assigns the highest probability to the object actually indicated and that covers the most words in the utterance. The reward is proportional to the normalized difference between the probability of the correct object and the probability of the closest runner-up. Together with this reward the a-proc sends back the object indicated during training. Each

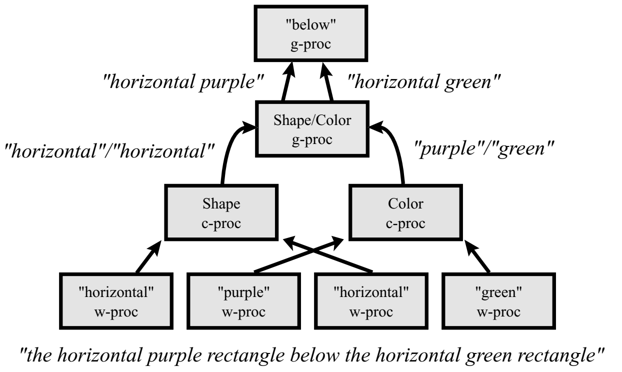

Fig. 3. A sample parse by the language understanding processes.

process that receives a reward in turn sends this reward to the processes that it used to cover a part of the current utterance. If it used different parts of the utterance to designate different objects (as in the case of spatially grounded words like ‘above’) it sends back the appropriate object with the reward message. G-agents maintain an energy measure that they subtract from each time they send a message and add to when they get rewarded. Processes with energy below zero die, trimming useless functions like measuring spatial features to explain ‘yellow cone’. C-agents currently do not use the reward message, but could do so to adjust their clustering threshold and re-cluster to make themselves more useful. W-agents, finally, can be caused to take a different object than the one indicated by the reward message, making sure that the ‘green’ w-proc learns from the right object in ‘the square above the green rectangle’.

5. LANGUAGE UNDERSTANDING

The language acquisition process described in the preceding sections is an on-line process, meaning that the system easily switches from learning to understanding at any point and does the best it can with the examples encountered so far. This switch merely implies that if no object is currently indicated, the w-procs do not take any examples, and the a-proc does not send a reward message. Rather, the a-proc sends the most likely object indicated by the process that covers the most words in the sentence to Newt’s motor controller, which in turn points at the object. Figure 3 shows a sample utterance parsed by the relevant processes, leaving out the a-proc which gathers all other processes’ results.

As a preliminary evaluation, we collected a small dataset of 303 utterances from two trainers in two sessions per trainer. Each utterance describes one object in a scene of four objects chosen from a collection of about 10 objects total, including objects with like shapes but different colors and vice versa. When trained on three of the sessions and evaluated on the fourth for all four sessions in turn, Newt achieves 82% accuracy in picking out the correct object, compared to a random baseline of 25%. Due to the small size of the dataset, we allowed Newt to use an example in the fourth session as a training example after Newt had selected an object for the example’s utterance. Doing this increases performance by almost 10%, indicating that the dataset size is too small to achieve full performance. However, even this preliminary study shows that Newt does learn the correct visual groundings for words and their combinations.

6. CONCLUSIONS AND FUTURE DIRECTIONS

We have presented a complete speech understanding system which maps spoken referring expressions to visually-observed objects. The semantics of referring expressions are grounded in visual primitives provided by a real-time color vision system which tracks multiple objects and extracts their properties and inter-object relations. The system understands color, shape, and spatial language. This work is part of our long term effort to develop a complete semantic representation of spoken language which is based in large part on sensory-motor grounding.

We are expanding on this work in two significant ways. First, we are investigating semantic representations which integrate motor control primitives to encode meanings of words such as “give”, “heavy” and “soft”. Second, we are developing a comprehensive semantic framework which also grounds non-sensory-motor words such as “not”, “because”, and “happen”.

7. REFERENCES

[1] D. Perzanowski, A. Schultz, W. Adams, K. Wauchope, E. Marsh, and M. Bugajska, “Interbot: A multi-modal interface to mobile robots,” in *Proceedings of Language Technologies 2001*, Carnegie Mellon University, 2001.

[2] Kobus Barnard and David Forsyth, “Learning the semantics of words and pictures,” .

[3] G. Iyengar, H. Nock, C. Neti, and M. Franz, “Semantic indexing of multimedia using audio, text and visual cues,” in *IEEE International Conference on Multimedia and Expo (In review)*, 2002.

[4] Deb Roy, “Learning visually-grounded words and syntax for a scene description task,” *Computer Speech and Language*, In review.

[5] T. Westman, D. Harwood, T. Laitinen, and M. Pietikinen, “Color segmentation by hierarchical connected components analysis with image enhancement by symmetric neighborhood filters,” in *Proceedings of the 10th International Conference on Pattern Recognition*, 1990, pp. 796–802.

[6] Deb Roy, Bernt Schiele, and Alex Pentland, “Learning audio-visual associations from sensory input,” in *Proceedings of the International Conference of Computer Vision Workshop on the Integration of Speech and Image Understanding*, Corfu, Greece, 1999.

[7] Terry Regier, *The human semantic potential*, MIT Press, Cambridge, MA, 1996.

[8] Benjamin Yoder, “Spontaneous speech recognition using hidden markov models,” M.S. thesis, Massachusetts Institute of Technology, Cambridge, MA, 2001.

[9] L. Mangu, E. Brill, and A. Stolcke, “Finding consensus among words: Lattice-based word error minimization,” in *Proceedings of EUROSPEECH’99*, Budapest, 1999, vol. 1, pp. 495–498.