WORD LEARNING IN A MULTIMODAL ENVIRONMENT

Deb Roy and Alex Pentland

MIT Media Lab
20 Ames Street, Rm. E15-388
Cambridge, MA 01239
(dkroy,sandy)@media.mit.edu

ABSTRACT

We are creating human machine interfaces which let people communicate with machines using natural modalities including speech and gesture. A problem with current multimodal interfaces is that users are forced to learn the set of words and gestures which the interface understands. We report on a trainable interface which lets the user teach the system words of their choice through natural multimodal interactions.

1. PROBLEM

Most current human-machine interfaces which use natural modalities such as speech and gesture force the user to learn which words and gestures the system understands before the system can be used (see [9], [10], or [4]; a notable exception is [2]). For example, an interface designer who wishes to use speech input must choose the vocabulary which the system will understand. If the user strays from this vocabulary, the system will not respond correctly. The semantics of the words must also be defined by the interface designer but may also not match the expectations of the user [1].

In practice it is extremely difficult to predict what vocabulary a person will use in even the most restricted domains [1]. As Zipf's law would predict, people's choice of words varies widely making it nearly impossible for the interface designer to determine which words an individual user will choose. To compound the problem of predicting vocabulary selection, the semantics of words also vary across users. In some of their experiments, Furnas et. al. found that users will sometimes use the same word to refer to different concepts even within highly limited domains [1]. These findings suggest that the vocabulary and associated semantics used in an interface should not be hard wired by the interface designer.

2. TRAINABLE INTERFACES

Our approach to this problem is to build *trainable interfaces* which let the user teach the interface which words and gestures she wants to use and what the words and gestures mean. In this paper we focus on the problem of building a trainable speech recognizer which lets the user define both the acoustic models and semantics of words they wish to use. We note that our approach to trainable interfaces can also be used for gestures and other non-speech modalities.

We have built a system which learns words from natural interactions with the user. Users teach the system words by pointing to objects and naming them. The system learns acoustic models of words, and infers the semantics of the words by observing the context in which they were heard. For our initial experiments we are using a simple blocks world. The user can refer to blocks using speech and diectic gesture and teach the system words referring to shapes and colors.

In order to learn the semantics of words a logical inference problem must be solved. For example the user might point to the same object and say both "red" and "ball". Over time the system must learn that the word "red" refers to the color attribute of objects and not shape. In addition, the system must deal with noisy acoustic input, so the logical inference must be solved in a statistical framework which can account for noise. A statistical framework is also useful since it can better cope with errors in training data.

3. EMBODIMENT OF THE INTERFACE

We have embodied the interface as an animated character called Toco the Toucan shown in Figure 1. He can move his head to look at any location in 3-D space. His eyes can blink and squint, his beak can open, close and smile, and his eyebrows can be raised, lowered, and tilted. Toco is currently situated in a world populated with blocks with different colors and shapes.

Toco's face and body language may be used as an output display to convey the system's internal state. For example his direction of gaze gives the person immediate feedback of where Toco thinks the user is pointing. Subtle facial cues such as widening of eyes and raised eyebrows are used to signal when Toco is alert and attending to the user. Confidence levels for speech or gesture recognition can also be displayed by showing confusion in Toco's face, or a knowing nod for a clearly understood command.

For output speech generation, we are using a commercial concatenative phonetic speech synthesizer. The synthesizer is driven using phoneme strings which have been learned from listening to users' speech as described later in this paper.

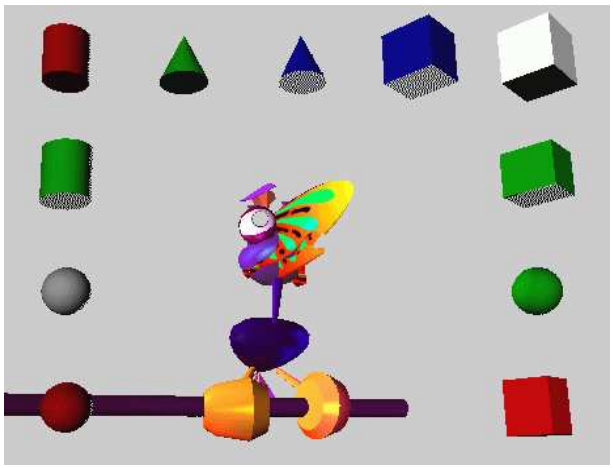

Figure 1: A screen shot of Toco the Toucan looking at a white cube in the virtual environment

4. A MULTIMODAL SENSORY ENVIRONMENT

We have created an environment to facilitate development of multimodal adaptive interfaces based on the smart desk environment [5]. In its current configuration, the user sits at a desk facing a 70" color projection screen which displays Toco and virtual objects. Toco can sense three types of input: the user's gestures, the user's speech, and information about the objects which currently exist in Toco's virtual environment.

4.1. Gesture Tracking

A vision-based gesture tracking system uses two color video cameras to sense the person's hand gestures. One camera is mounted directly overhead and the second provides an orthogonal view from the right side. A Gaussian mixture model of skin color is used to locate and track the user's hand at 30 Hz. The 3-D hand position is estimated by combining estimates from both cameras. For details of the gesture tracking system see [8].

4.2. Speech Analysis using Real-time Phoneme Recognition

Audio is sampled at 16-bit 16 kHz from a head mounted microphone and processed using the Relative Spectral (RASTA) algorithm [3]. The RASTA coefficients are computed on 20ms windows of audio (recomputed every 10ms) and fed into a recurrent neural network (RNN) similar to the system described in [6] to produce a 40-dimensional phoneme probability estimate (39 phonemes and silence) at a rate of 100Hz. The RNN has been trained using back propagation in time on the TIMIT database. A finite state machine is used to detect and segment *speech events* (defined to be spoken utterances surrounded by silence) using the silence estimate from the RNN [8].

The RNN outputs are treated as emission probabilities within a Hidden Markov Model (HMM) framework. Duration models and bigram phoneme transition probabilities for a all-phoneme loop HMM have been computed from the TIMIT training data set. The system currently recognizes phonemes with 68% accuracy on the standard speaker independent TIMIT recognition task. Given a speech event, Viterbi search can be used within the HMM framework to find the most likely phoneme sequence.

We now define a distance metric for comparing a speech event to a reference phoneme string. This distance metric is used for clustering speech events (see Section 5.1). The reference phoneme string may be thought of as a HMM. We can compute a confidence measure that an event was generated by the HMM following methods developed for keyword spotting confidence measures [7] as follows.

First we compute the log probability of an event e using a forced Viterbi alignment with phoneme transitions determined by the reference phoneme string. We denote this as $log(p(reference \mid e))$.

Next we compute the log probability of the event e using a Viterbi search with a phoneme loop model with phoneme bigram transition probabilities estimated from the original TIMIT data. We denote this as $log(p(phonemeloop \mid e))$.

Finally, we can define the normalized distance between the event and the reference string to be:

$$d(ref, e) = log(p(ref \mid e)) - log(p(phonemeloop \mid e)) \quad (1)$$

We have found that this measure works well for both keyword spotting (Toco only responds to users after hearing his name) and for clustering acoustic data (see Section 5.1).

4.3. Synthetic Sensing of Virtual Objects

The third type of input in the multimodal environment lets Toco directly "sense" attributes of virtual objects which are displayed in Toco's graphical world (for example the blocks in Figure 1). Each object is represented by a set of *attribute vectors* which encode characteristics of the object. For example the white cube in the top right corner of Figure 1 is represented as:

```
Object {
   r,g,b = 1.0 , 1.0 , 1.0
   shape = 0, 0, 1, 0
}
```

We use the following notation to represent an element of an attribute vector set:

$$a_j^i(k), \quad i = 1, 2 \ldots n, \quad j = 1, 2 \ldots m_i \quad (2)$$

where a_j^i is the j^{th} element of the i^{th} attribute vector representing the k^{th} object, there are n vectors in an attribute set, and the i^{th} vector has m_i elements. Thus for example the attribute vector set for the white cube described above has $n = 2$, $m_1 = 3$, $m_2 = 4$ and $a_3^2 = 1$. The shape vector discretely encodes shapes in a four-bit binary vector [1] (cone, sphere, cube, cylinder). The purpose for representing objects as attribute vectors is to facilitate learning of

[1] We are currently changing the representation to encode continuous valued attributes so that more complex attribute spaces may be modeled.

word meanings in terms of these attribute primitives. If all perceptually salient attributes of the objects are encoded in the attributes, Toco will be able to discover the meaning of words grounded in these perceptual primitives.

5. LEARNING AND RESPONDING TO SPOKEN WORDS

5.1. Learning Words

Words must be learned at two levels: their acoustic models, and their association with object attribute vectors. Gesture input from the user provides contextual information for learning attribute associations.

Toco's memory consists of a set of *phoneme string clusters*. Each cluster is comprised of a set of one or more phoneme strings, and an *association weight vector*:

$$w_j^i(l), \quad i = 1, 2 \ldots n, \quad j = 1, 2 \ldots m_i \tag{3}$$

where $w_j^i(l)$ is the j^{th} element of the i^{th} weight vector of the l^{th} word cluster, and n and m_i are as defined in Equation 2.

We set the weight vectors of cluster l to the mutual information between the observation of word cluster l and each attribute vector:

$$w_j^i(l) = log \left\{ \frac{p(a_j^i \mid V_l)}{p(a_j^i)} \right\} \tag{4}$$

where V_l signifies that a phoneme string from cluster l was heard. The probabilities of both conditioned and unconditioned attribute vector values are updated using the attributes of the objects which the user points to while speaking training words. Since the elements of the attributes are binary variables, we are able to use simple smoothed relative frequencies to estimate the probabilities in Equation 4 [2].

When Toco first starts running, he has no clusters in memory. When the user first points to an object and utters a word, Toco will create a cluster and initialize it with the phoneme string extracted from the user's speech. The association weight vector for this cluster is then set using Equation 4. Subsequent training examples (i.e. where the user is pointing to an object as she says a word) are incorporated into Toco's associative memory by the following steps.

First we need to define the distance from a speech event to a word cluster. We denote the g^{th} phoneme string of cluster V_l as s_g^l. The distance from cluster V_l to event e is defined as the distance between the event and the closest phoneme string within the cluster:

$$d_{cluster}(V_l, e) = min_g \; d(s_g^l, e) \tag{5}$$

where $d()$ is defined in Equation 1.

Using Equation 5 the index of the cluster closest to the speech event is found:

$$l_{best} = arg \; min_l \; d(V_l, e) \tag{6}$$

and the distance from the event e to the closest cluster is:

$$dist = d_{cluster}(V_{l_{best}}, e) \tag{7}$$

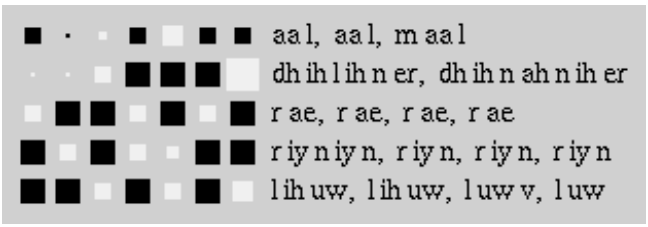

Figure 2: Examples of words learned from the blocks world task. The squares show association weight values. The size of the square indicates its magnitude; white color indicates a positive value and black indicates a negative value. The left three weight columns correspond to color (R-G-B) and the remaining four columns correspond to shape (cube-ball-cone-cylinder). Each row of weights represents a phoneme string cluster with the associated phoneme strings listed on the right. From the top, the words shown are "ball", "cylinder", "red", "green" and "blue". There are currently some systematic segmentation problems which cause clipping of word beginnings and endings. We plan to modify the segmentation algorithm parameters to correct this.

At this point the algorithm compares *dist* to a predefined split/merge threshold. If *dist* is greater than the threshold, a new cluster is formed and initialized with the phoneme string extracted from the event e. If *dist* is less than the threshold, the phoneme string extracted from the event e is added to the cluster $V_{l_{best}}$, effectively merging the acoustic model of the new event with the existing acoustic models of the cluster. After the clusters are updated (by either a merge or split), the weights of the phoneme string clusters are updated according to Equation 4.

Figure 2 shows some of the phoneme string clusters which were learned using this method over the course of a two minute interaction with a user.

5.2. Responding to Words

Toco can respond to words based on the phoneme string clusters which have been formed from previous interactions with the user. When the user says a word without pointing to the object, Toco finds the closest cluster to the speech event using Equation 6. We refer to this as the *activated phoneme string cluster*. The distance to this activated cluster is calculated using Equation 7. This distance is then compared to a fixed response threshold which determines whether Toco will respond to the event. If the distance is greater than the threshold, Toco treats the event as unknown and takes no action.

If the distance is less than the threshold, Toco finds the object in view which has highest association with the activated cluster (see below) and responds by looking towards that object and vocalizing.

5.2.1. Selecting an Object During Word Response

The object is selected by computing the association strength between each object and the activated cluster and selecting the object with the highest association. The association between an object k and cluster V_l is computed as:

$$y_k^l = max_i \frac{\sum_{j=1}^{m_i} a_j^i(k) w_j^i(l)}{m_i}, \quad i = 1 \ldots n \qquad (8)$$

The *max* operator in Equation 8 implements a competition among attribute vectors within the vector set. The dot product of association weights for the cluster and the associated attribute vector is computed for each vector in the set and normalized by m_i, the dimensionality of the i^{th} attribute vector. The association of the cluster to the object is set to the highest normalized vector association.

5.2.2. Generating a Spoken Response

To produce a vocalization the system inverts the process encoded in Equation 8 and finds the phoneme string cluster most strongly associated with the selected object, using the attribute vector which produces the highest dimension-normalized score in the forward application of Equation 8.

A spoken response is generated by choosing a representative phoneme string from the selected cluster and sending it to the phoneme synthesizer. This method of spoken response provides a feedback mechanism for Toco's semantic associations. Inconsistencies in Toco's semantic network are exposed when he tries to echo the user's speech but says an incorrect word. When such an error occurs, the user will immediately know that Toco has not yet learned the proper meaning of the word. When robust acoustic and semantic models have been learned, Toco consistently "parrots" the user's speech. In the blocks world task empirical tests show that approximately 15 to 20 training words are required to teach 4 colors, and 30 to 40 training words are required to teach 8 colors and shapes.

5.3. An Application: Vocabulary Translation

An application of the current system is in translating words between different languages. We have added a mode to the system in which Toco recognizes words using one set of acoustic and semantic weights, and generates output using a second set. If Toco is taught words in two different languages, he can then respond to words spoken in one language with synthetic speech in a second language. The mapping between words is defined by the semantic associations of the words in each language which are grounded in the common object attributes. A related application of this idea is as a communication aid for disabled users with dysarthric (unintelligible) speech. If the user can produce consistent vocalizations, the system can be taught to translate them into clear synthetic speech.

6. SUMMARY AND FUTURE WORK

We have presented results of our initial experiments on word learning in a multimodal environment. Our system demonstrates an interface which learns words and their domain-limited semantics through natural multimodal interactions with people. Toco can learn acoustic words and their meanings by continuously updating association weight vectors which estimate the mutual information between acoustic words and attribute vectors which represent perceptually salient aspects of virtual objects in Toco's world. Toco is able to learn semantic associations (between words and attribute vectors) using gestural input from the user. Gesture input enables the user to naturally specify which object to attend to during word learning.

We are in the process of defining a new task which involves learning spoken directions for spatial navigation. Spatial relations will be encoded in the attribute vectors of virtual objects facilitating the learning of words referring to spatial concepts (for example "left", "right", "near", "far" etc.).

In addition to defining a new task domain, we are planning several technical extensions of the system including the addition of a direct manipulation interface for demonstrating actions so that Toco can learn verbs, and simple grammar learning capabilities which can be used for both speech recognition and generation.

7. ACKNOWLEDGMENTS

Tony Jebara, Mike Hlavac, and Brian Clarkson worked on the vision, graphics and speech subsystems respectively. Thanks also to Allen Gorin and Bruce Blumberg for helpful comments.

8. REFERENCES

[1] G.W. Furnas, T.K. Landauer, L.M. Gomez, and S.T. Dumais. The vocabulary problem in human-system communications. *Communications of the Association for Compuring Machinery*, 30:964–972, 1987.

[2] Allen L. Gorin. On automated language acquisition. *Journal of the Acoustic Society of America*, 97(6):3441–3461, 1995.

[3] H. Hermansky and N. Morgan. Rasta processing of speech. *IEEE Transactions on Speech and Audio Processing*, October 1994.

[4] Sharon Oviatt. Multimodal interfaces for dynamic interactive maps. In *Proceedings of the Conference on Human Factors in Computing Systems*, pages 95–102, New York, 1996. ACM Press.

[5] Alex Pentland. Smart desks, desks, and clothes. In *Proceedings of ICASSP*, pages 171–174, Munich, Germany, April 1997. IEEE Computer Society Press.

[6] Tony Robinson. An application of recurrent nets to phone probability estimation. *IEEE Trans. Neural Networks*, 5(3), 1994.

[7] Richard Rose. *Word Spotting from Continuous Speech Utterances*, chapter 13, pages 303–329. Kluwer Academic, 1996.

[8] Deb Roy and Alex Pentland. Multimodal adaptive interfaces. Technical Report 438, MIT Media Lab Vision and Modeling Group, 1997.

[9] Kristinn R. Thorisson. *Communicative Humanoids: A Computational Model of Psychological Dialogue Skills*. PhD thesis, MIT Department of Media Arts and Sciences, 1996.

[10] Alex Waibel, Minh Tue Vo, Paul Duchnowski, and Stephan Manke. Multimodal interfaces. *Artificial Intelligence Review*, 10(3-4):299–319, 1995.