A Multimodal Learning Interface for Grounding Spoken Language in Sensory Perceptions

Chen Yu
Department of Computer Science
University of Rochester
Rochester, NY, 14627, USA
yu@cs.rochester.edu

Dana H. Ballard
Department of Computer Science
University of Rochester
Rochester, NY, 14627, USA
dana@cs.rochester.edu

ABSTRACT

Most speech interfaces are based on natural language processing techniques that use pre-defined symbolic representations of word meanings and process only linguistic information. To understand and use language like their human counterparts in multimodal human-computer interaction, computers need to acquire spoken language and map it to other sensory perceptions. This paper presents a multimodal interface that learns to associate spoken language with perceptual features by being situated in users' everyday environments and sharing user-centric multisensory information. The learning interface is trained in unsupervised mode in which users perform everyday tasks while providing natural language descriptions of their behaviors. We collect acoustic signals in concert with multisensory information from non-speech modalities, such as user's perspective video, gaze positions, head directions and hand movements. The system firstly estimates users' focus of attention from eye and head cues. Attention, as represented by gaze fixation, is used for spotting the target object of user interest. Attention switches are calculated and used to segment an action sequence into action units which are then categorized by mixture hidden Markov models. A multimodal learning algorithm is developed to spot words from continuous speech and then associate them with perceptually grounded meanings extracted from visual perception and action. Successful learning has been demonstrated in the experiments of three natural tasks: "unscrewing a jar", "stapling a letter" and "pouring water".

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning—*Language acquisition*; I.2.0 [**Artificial Intelligence**]: General—*Cognitive simulation*; H.5.2 [**Information interfaces and representation**]: User Interfaces—*Theory and methods*

General Terms

Languages, Human Factors

Keywords

Language Acquisition, Machine Learning, Multimodal Integration

ICMI'03, November 5–7, 2003, Vancouver, British Columbia, Canada.

1. INTRODUCTION

The next generation of computers is expected to interact and communicate with users in a cooperative and natural manner when users engage in everyday activities. By being situated in users' environments, intelligent computers should have basic perceptual abilities, such as understanding what people are talking about (speech recognition), what they are looking at (visual object recognition) and what they are doing (action recognition). Furthermore, similar to human counterparts, computers should acquire and then use the knowledge of associations between different perceptual inputs. For instance, spoken words of object names (sensed from auditory perception) are naturally correlated with visual appearances of the corresponding objects obtained from visual perception. Once machines have that knowledge and those abilities, they can demonstrate many human-like behaviors and perform many helpful acts. In the scenario of making a peanut-butter sandwich, for example, when a user asks for a piece of bread verbally, a computer can understand that the spoken utterance of "bread" refers to some flat square piece in the visual scene. Therefore, with an actuator such as a robotic arm, the machine can first locate the position of the bread, then grasp and deliver it to the user. In another context, a computer may detect the user's attention and notice that the attentional object is a peanut butter jar, it can then utter the object name and provide information related to peanut butter by speech, such as a set of recipes or nutritional values. In a third example, a computer may be able to recognize what the user is doing and linguistically describe what it sees. The ability to generate verbal descriptions of user's behaviors can be a precursor to making computers communicate with users naturally. In this way, computers will seamlessly integrate into our everyday lives and work as intelligent observers and human-like assistants.

To progress toward the goal of anthropomorphic interfaces, computers need to not only recognize sound patterns of spoken words but also associate them with their perceptually grounded meanings. Two research fields are closely related to this topic: speech recognition and multimodal human-computer interfaces. Unfortunately, both of them only address some parts of the problem. They do not provide a solution to the whole issue.

Most existing speech recognition systems can not achieve the goal because they rely on purely acoustics-based statistical models, such as hidden Markov models [13] and hybrid connectionist models [9]. These systems have two inherent disadvantages. First, they require a training phase in which large amounts of spoken utterances paired with manually labeled transcriptions are needed to train the model parameters. This training procedure is time-consuming and needs human expertise to label spoken data. Second, these systems transform acoustic signals to symbolic representations (texts) without regard to their grounded meanings. Humans need to interpret the meanings of these symbols based on our own knowledge. For instance, a speech recognition system can map the sound pattern "jar" to the string "jar", but it does not know its meaning.

In multimodal human-computer interface studies, researchers mainly focus on the design of multimodal systems with performance advantages over unimodal ones in the context of different types of human-computer interaction [12]. The technical issue here is multimodal integration – how to integrate signals in different modalities. Generally, multimodal systems using semantic fusion include individual recognizers and a sequential integration process. These individual recognizers can be trained using unimodal data, which can then be integrated directly without re-training. However, most systems do not have *learning* ability in the sense that developers need to encode knowledge into some symbolic representations or probabilistic models during the training phase. Once the systems are trained, they are not able to automatically gain additional knowledge even though they are situated in surrounding physical environments and can obtain multisensory information.

To overcome the above shortcomings, a few recent studies proposed several unsupervised methods for learning words from linguistic and contextual inputs [18, 15]. Among them, the work of Roy [15] is particularly relevant to our work. He proposed a computational model of infant language acquisition, which utilizes temporal correlation of speech and vision to associate spoken utterances with a corresponding object's visual features. The model has been implemented to process a corpus of audio-visual data from infant-caregiver interactions. Our work differs from his in that we focus on building a multimodal learning interface that is able to acquire a lexicon from naturally co-occurring multisensory information in everyday activities.

This paper presents a multimodal learning system that is able to learn perceptually grounded meanings of words from user's everyday activities. The only requirement is that users need to describe their behaviors verbally while performing those day-to-day tasks. Since no manually labeled data is involved in the learning procedure, the range of problems we need to address in this kind of word learning is substantial. To make concrete progress, this paper focuses on how to associate visual representations of objects with their spoken names and map body movements to action verbs. Our work suggests a new trend in developing human-computer interfaces that can automatically learn spoken language by sharing user-centric multisensory information. This advent represents the beginning of a progression toward computational systems capable of human-like sensory perception.

2. A MULTIMODAL LEARNING INTERFACE

We believe that the limits of existing systems lie in the fact that sensory perception and language acquisition of machines are quite different from those of human counterparts. Humans learn language based on our sensorimotor experiences with the physical environment. We learn words by sensing the environment through our perceptual systems and associating spoken language with other sensory perceptions. In light of human language acquisition, we are developing a multimodal learning system that can also learn meaningful semantic representations grounded in the physical environment around us.

To ground language, the computational system needs to have sensorimotor experiences by interacting with the physical world. Our solution is to attach different kinds of sensors to a real person as shown in Figure 1. Those sensors include a head-mounted CCD camera to capture a first-person point of view, a microphone to sense acoustic signals, an eye tracker to track the course of eye movements that indicates the agent's attention, and position sensors attached to the head and hands of the agent to simulate proprioception in the sense of motion. The functions of those sensors are similar to human sensory systems and they allow the intelligent system to collect user-centric multisensory data to simulate the development of human-like perceptual capabilities. In the learning phase, the real agent performs some everyday tasks, such as making a sandwich, pouring some drinks or stapling a letter, while describing his/her actions verbally. We collect acoustic signals in concert with user-centric multisensory information from non-speech modalities, such as user's perspective video, gaze positions, head directions and hand movements. A multimodal learning algorithm is developed that firstly spots words from continuous speech and then builds the grounded semantics by associating object names and action verbs with visual perception and body movements. The central idea is to make use of user's focus of attention to firstly infer his/her referential intentions in speech and then build word-meaning associations. The system consists of several components as follows:

- **Attention detection** finds where and when a user looks based on gaze and head movements.
- **Attentional object spotting and action categorization** extract grounded meanings of words encoded in non-speech contextual information, such as attentional objects and intentional actions.
- **Speech segmentation and word spotting** discover the sound patterns that correspond to words.
- **Multimodal integration** associates spoken words with perceptual features.

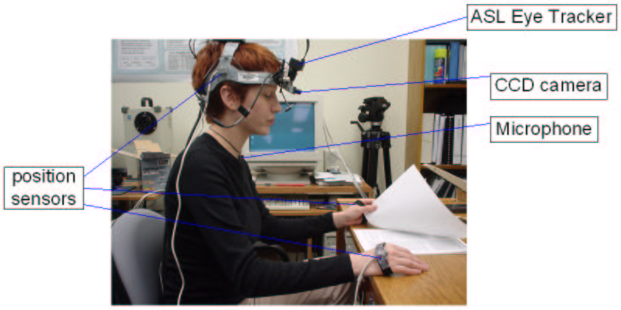

Figure 1. The intelligent system shares multisensory information with a real agent in a first-person sense. This allows the association of coincident signals in different modalities.

2.1 Estimating Focus of Attention

We present a method that utilizes eye gaze and head direction to detect a performer's focus of attention. Attention, as represented by eye fixation, is used for spotting the target object of user interest. Attention switches are calculated and used to segment the action sequence into action units. We developed a velocity-based method to model eye movements using a Hidden Markov Model(HMM) representation that has been widely used in speech recognition with great success [13]. A hidden Markov model consists of a set of N states $S = \{s_1, s_2, s_3, ..., s_N\}$, the transition probability matrix $A = a_{ij}$, where a_{ij} is the transition probability of taking the transition from state s_i to state s_j, prior probabilities for the initial state π_i, and output probabilities of each state $b_i(O(t)) = P\{o(t)|s(t) = s_i\}$. Salvucci et al.[16] firstly proposed a HMM-based fixation identification method that uses probabilistic analysis to determine the most likely identifications for a given protocol. Our approach is different from his in two ways. First, we use training data to estimate the transition probabilities instead of setting pre-determined values. Second, we notice that head movements provide valuable cues to model focus of attention. This is because when users look toward an object, they always orient their heads toward the object of interest so as to make it in the center of their visual fields. As a result of the above analysis, head positions are integrated with eye positions as the observations of HMMs. Figure 2 shows a 4-state HMM used in our system for attention detection. The observations of HMM are 2-dimensional vectors consisting of the magnitudes of the velocities of head rotations in three dimensions and the magnitudes of velocities of eye movements. We model the probability densities of the observations using a two-dimensional Gaussian. The parameters of HMM needed to be estimated comprise the observation and transition probabilities. The estimation problem concerns how to adjust the model λ to maximize $P(O \mid \lambda)$ given an observation sequence O of eye and head motions. We can initialize the model with flat probabilities, then the forward-backward algorithm allows us to train the model parameters using training data (see [13]). Figure 3 illustrates the state transitions of HMM given eye and head motions. Among those four states, dwell and fixation represent attentional states in which users look toward the objects in the visual scene.

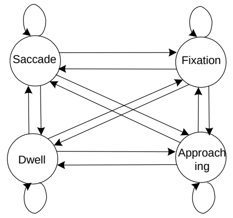

Figure 2. The HMM of eye and head movements. There are four states: saccade, fixation, dwell and approaching. **Saccades** are rapid eye movements that allow the fovea to view a different portion of the visual scene. **Dwells** typically include several fixations and the relatively small amount of time for the saccades between these fixations while head direction is fixated. **Fixations** are relatively stable eye and head positions over some minimum duration (typically 100-200ms). **Approachings** are the fixations of eye-in-head positions while the head is moving as part of upper body.

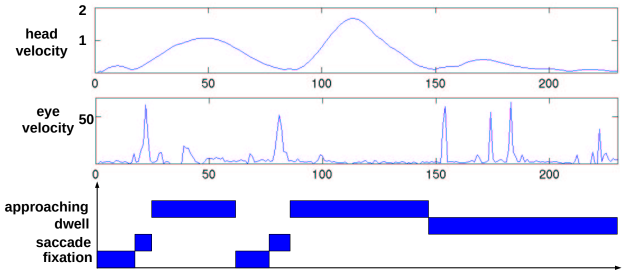

Figure 3. Eye and head movement analysis. The first Row: point-to-point velocities of head position data. **The second row:** eye movement velocities. **The bottom row:** the state transitions of HMM.

2.2 Attentional Object Spotting

Knowing attentional states allows for automatic object spotting by integrating visual information with eye gaze data. For each attentional point in time, the object of user interest is discovered from the snapshot of the scene. Multiple visual features are then extracted from the visual appearance of the object which are used for object categorization.

2.2.1 Object Spotting

Attentional object spotting consists of two steps. First, the snapshots of the scene are segmented into blobs using ratio-cut [21]. The result of image segmentation is illustrated in Figure 5(b) and only blobs larger than a threshold are used. Next, we group those blobs into several semantic objects. Our approach starts with the original image, uses gaze positions as seeds and repeatly merges the most similar regions to form new groups until all the blobs are labeled. Eye gaze in each attentional time is then utilized as a cue to extract the object of user interest from all the detected objects.

We use color as the similarity feature for merging regions. $L * a * b$ color space is adopted to overcome undesirable effects caused by varied lighting conditions and achieve more robust illumination-invariant segmentation. $L * a * b$ color consists of a luminance or lightness component (L*) and two chromatic components: the a* component (from green to red) and the b* component (from blue to yellow). To this effect, we compute in the $L * a * b$ color space the similarity distance between two blobs and employ the histogram intersection method proposed by [20]. If C_A and C_B denote the color histograms of two regions A and B, their histogram intersection is defined as:

$$h(A,B) = \frac{\sum_{i=1}^{n} min(C_A^i, C_B^i)}{\sum_{i=1}^{n}(C_A^i + C_B^i)} \qquad (1)$$

where n is the number of bin in color histogram, and $0 < h(A,B) < 0.5$. Two neighboring regions are merged into a new region if the histogram intersection $h(A,B)$ is between a threshold $t_c (0 < t_c < 0.5)$ and 0.5. While this similarity measure is fairly simple, it is remarkably effective in determining color similarity between regions of multi-colored objects.

The approach of merging blobs is based on a set of regions selected by a user's gaze fixations, termed seed regions. We start with a number of seed regions $S_1, S_2, ..., S_n$, in which n is the number of regions that the user was fixating on. Given those seed regions, the merging process then finds a grouping of the blobs into semantic objects with the constraint that the regions of visual objects are chosen to be as homogeneous as possible. The process evolves inductively from the seed regions. Each step involves the addition of one blob to one of the seed regions and the merging of neighbor regions based on their similarities.

In the implementation, we make use of a sequentially sorted list (SSL) [1] that is a linked list of blobs ordered according to some attribute. In each step of our method, we consider the blob at the beginning of the list. When adding a new blob to the list, we place it according to its value of the ordering attribute so that the list is always sorted based on the attribute. Let $N(A)$ be the set of immediate neighbors of the blob A, which are seed regions. For all the regions $N(A)_1, N(A)_2, ..., N(A)_n$, the seed region that is closest to A is defined as:

$$B = \arg\max_i h(A, N(A)_i); 1 \leq i \leq n \qquad (2)$$

where $h(A, N(A)_i)$ is the similarity distance between region A and $N(A)_i$ based on the selected similarity feature. The ordering attribute of region A is then defined as $h(A,B)$. The merging procedure is illustrated in Figure 4. Figure 5 shows how these steps are combined to get an attentional object.

2.2.2 Object Representation and Categorization

The visual representation of the extracted object contains color, shape and texture features. Based on the works of [10, 17, 20],

```
Algorithm: object segmentation based on gaze fixations
Initialization:
    Compute the color histogram of each region.
    Label seed regions according to the positions of gaze fixations.
    Merge seed regions that are neighbors to each other and are
    close with respect to their similarity.
    Put neighboring regions of seed regions in the SSL.
Merging:
  While the SSL is not empty
    Remove the top region $A$ from SSL.
    Compare the similarity between $A$ and all the regions in
    $N(A)$ and find the closest seed region $B$.
    Merge the regions $A$ and $B$ and compute the color histogram
    of new region $I = A \cup B$.
    Test each neighboring region $A_i$ of $A$:
      If $A_i$ is labeled as a seed region
         Merge the region with $I$ if they are similar.
      Otherwise
        Add the region to the SSL according to its color similarity
        with $I$, $h(A_i, I)$.
```

Figure 4. The algorithm for merging blobs

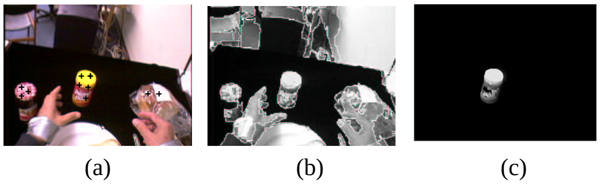
(a) (b) (c)

Figure 5. **Left:** The snapshot image with eye positions (black crosses). **Middle:** The results of low-level image segmentation. **Right:** Combining the eye position data with the segmentation to extract an attended object.

we construct the visual features of objects which are large in number, invariant to different viewpoints, and driven by multiple visual cues. Specifically, 64-dimensional color features are extracted by a color indexing method [20], and 48-dimensional shape features are represented by calculating histograms of local shape properties [17]. The Gabor filters with three scales and five orientations are applied to the segmented image. It is assumed that the local texture regions are spatially homogeneous, and the mean and the standard deviation of the magnitude of the transform coefficients are used to represent an object in a 48-dimensional texture feature vector. The feature representations consisting of a total of 160 dimensions are formed by combining color, shape and texture features, which provide fundamental advantages for fast, inexpensive recognition. Most pattern recognition algorithms, however, do not work efficiently in higher dimensional spaces because of the inherent sparsity of the data. This problem has been traditionally referred to as the dimensionality curse. In our system, we reduced the 160-dimensional feature vectors into the vectors of dimensionality 30 by principle component analysis (PCA) [2], which represents the data in a lower dimensional subspace by pruning away those dimensions with the least variance. Next, since the feature vectors extracted from visual appearances of attentional objects do not occupy a discrete space, we vector quantize them into clusters by applying a hierarchical agglomerative clustering algorithm. Finally, we select a prototype to represent perceptual features of each cluster.

2.3 Action Segmentation and Recognition

Recent results in visual psychophysics [8, 7] indicate that in natural circumstances, the eye, the head, and hands are in continual motion in the context of ongoing behavior. This requires the coordination of these movements in both time and space. In light of this, our hypothesis is that eye and head movements, as an integral part of the motor program of humans, provide important information for action recognition in human activities. We test this hypothesis by developing a method that segments action sequences based on the dynamic properties of eye gaze and head direction, and applies Dynamic Time Warping (DTW) and HMM to cluster temporal sequences of human motion.

2.3.1 Action Segmentation

The segmentation of a continuous action stream into action primitives is the first step toward understanding human behaviors. With the ability to track the course of gaze and head movements, our approach uses gaze and head cues to detect agent-centered attention switches that can then be utilized to segment human action sequences.

In our experiments, we notice that actions can occur in two situations: during eye fixations and during head fixations. For example, in a “picking up” action, the performer focuses on the object first then the motor system moves the hand to approach it. During the procedure of approaching and grasping, the head moves toward the object as the result of the upper body movements, but eye gaze remains stationary on the target object. The second case includes such actions as “pouring water” in which the head fixates on the object involved in the action. During the head fixation, eye-movement recordings show that there can be a number of eye fixations. For example, when the performer is pouring water, he spends 5 fixations on the different parts of the cup and 1 look-ahead fixation to the location where he will place the water pot after pouring. In this situation, the head fixation is a better cue than eye fixation to segment the actions. In either case, there is almost always an identifiable saccade that switches attention from one place to another. Based on the above analysis, the times of action boundary are extracted by finding any of dwells, fixations and approachings that follow a saccade. Hand motions are segmented into several action primitives based on those times. Detailed technical descriptions of action segmentation can be found in [22].

2.3.2 Action Categorization

We collect the raw position (x, y, z) and the rotation (h, p, r) data of each action unit from which feature vectors are extracted for recognition. We want to recognize the types of motion not the accurate trajectory of the hand because the same action performed by different people varies. Even in different instances of a simple action of “picking up” performed by the same person, the hand goes roughly in a different trajectory. This indicates that we can not directly use the raw position data to be the features of the actions. As pointed out by Campbell et al. [5], features designed to be invariant to shift and rotation perform better in the presence of shifted and rotated input. The feature vectors should be chosen so that large changes in the action trajectory produce relatively small excursions in the feature space, while the different types of motion produce relatively large excursions. In the context of our experiment, we calculated three element feature vectors consisting of the hand’s velocity on the table plane ($d\sqrt{x^2+y^2}$), the velocity in the z-axis, and the velocity of rotation in the 3 dimensions ($d\sqrt{h^2+p^2+r^2}$).

Let S denote a hand motion trajectory that is a multivariate time series spanning n time steps such that $S = \{s_t \mid 1 \leq t \leq n\}$. s_t is a vector of values containing one element for the value of each of the component univariate time series at time t. Given a set of

m multivariate time series of hand motion, we want to obtain in an unsupervised manner a partition of these time series into subsets such that each cluster corresponds to a qualitatively different regime. Our clustering approach is based on the combination of HMM (described briefly in Section 2.1) and Dynamic Time Warping [11]. Given two time series S_1 and S_2, DTW finds the warping of the time dimension in S_1, which minimizes the difference between two series.

We model the probability of individual observation (a time series S) as generated by a finite mixture model of K component HMMs [19]:

$$f(S) = \sum_{k=1}^{K} p_k(S|c_k)p(c_k) \quad (3)$$

where $p(c_k)$ is the prior probability of kth HMM and $p_k(S|c_k)$ is the generative probability given the kth HMM with its transition matrix, observation density parameters, and initial state probabilities. $p_k(S|c_k)$ can be computed via the forward part of the forward-backward procedure. Assume that the number of clusters K is known, the algorithm for clustering sequences into K groups can be described in terms of three steps:

- given m time series, construct a complete pairwise distance matrix by invoking DTW $m(m-1)/2$ times. Use the distance matrix to cluster the sequences into K groups by employing a hierarchical agglomerative clustering algorithm [6].
- fit one HMM for each individual group and train the parameters of the HMM. $p(c_k)$ is initialized to M_k/M where M_k is the number of sequences which belong to cluster k.
- iteratively reestimate the parameters of all the k HMMs in the Baum-welch fashion using all of the sequences [13]. The weight that a sequence S has in the reestimation of kth HMM is proportional to the log-likelihood probability of the sequence given that model $\log p_k(S|c_k)$. Thus, sequences with bigger generative probabilities for a HMM have greater influence in reestimating the parameters of that HMM.

The intuition of the procedure is as follows: since the Baum-Welch algorithm is hill-climbing the likelihood surface, the initial conditions critically influence the final results. Therefore, DTW-based clustering is used to get a better estimate of the initial parameters of HMMs so that the Baum-Welch procedure will not converge to a local maximum only. In the reestimation, sequences that are more likely generated by a specific model cause the parameters of that HMM to change in such a way that it further fits for modeling a specific group of sequences.

2.4 Speech Processing

We briefly describe our methods of phoneme recognition and phoneme string comparison in this subsection, which provide a basis for word-meaning association. Further information can be obtained from [3].

2.4.1 Phoneme Recognition

We have implemented an endpoint detection algorithm to segment the speech stream into several spoken utterances. Then the speaker-independent phoneme recognition system developed by Robinson [14] is employed to convert spoken utterances into phoneme sequences. The method is based on Recurrent Neural Networks (RNN) that perform the mapping from a sequence of the acoustic features extracted from raw speech to a sequence of phonemes. The training data of RNN are from the TIMIT database — phonetically transcribed American English speech — which consists of read sentences spoken by 630 speakers from eight dialect regions of the United States. To train the networks, each sentence is presented to the recurrent back-propagation procedure. The target outputs are set using the phoneme transcriptions provided in the TIMIT database. Once trained, a dynamic programming match is made to find the most probable phoneme sequence of a spoken utterance, for example, the boxes labeled “phoneme strings” in Figure 6.

2.4.2 Comparing Phoneme Sequences

The comparison of phoneme sequences has two purposes: one is to find the longest similar substrings of two phonetic sequences and the other is to spot a short string (a pattern) from a long sequence of spoken utterance. In both cases, an algorithm of the alignment of phoneme sequences is a necessary step. Given raw speech input, the specific requirement here is to cope with the acoustic variability of spoken words in different contexts and by various talkers. Due to this variation, the outputs of the phoneme recognizer previously described are noisy phoneme strings that are different from phonetic transcriptions of text. In this context, the goal of phonetic string matching is to identify sequences that might be different actual strings, but have similar pronunciations.

To align phonetic sequences, we first need a metric for measuring distances between phonemes. We represent a phoneme by a 15-dimensional binary vector in which every entry stands for a single articulatory feature called a distinctive feature. Those distinctive features are indispensable attributes of a phoneme that are required to differentiate one phoneme from another in English. We compute the distance between two individual phonemes as the Hamming distance. Based on this metric, a modified dynamic programming algorithm is developed to compare two phoneme strings by measuring their similarity. A similarity scoring scheme assigns large positive scores to pairs of matching segments, large negative scores to pairs of dissimilar segments, and small negative scores to the operations of insertion and deletion to convert one sequence to another. The optimal alignment is the one that maximizes the accumulated score.

2.5 Word-like Unit Spotting

Figure 6 illustrates our approach to spotting word-like units in which the central idea is to utilize non-speech contextual information to facilitate word spotting. The reason we use the term “word-like units” is that some actions are verbally described by verb phrases (e.g. “line up”) but not single action verbs. The inputs are phoneme sequences(u_1, u_2, u_3, u_4) and possible meanings of words (objects and actions) extracted from non-speech perceptual inputs. Those phoneme utterances are categorized into several bins based on their possible associated meanings. For each meaning, we find the corresponding phoneme sequences uttered in temporal proximity, and then categorize them into the same bin labeled by that meaning. For instance, u_1 and u_3 are temporally correlated with the action “stapling”, so they are grouped in the same bin labeled by the action “stapling”. We need to point out here that, since one utterance could be temporally correlated with multiple meanings grounded in different modalities, it is possible that an utterance is selected and classified in different bins. For example, the utterance “stapling a few sheets of paper” is produced when a user performs the action of “stapling” and looks toward the object “paper”. In this case, the utterance is put into two bins: one corresponding to the object “paper” and the other labeled by the action “stapling”.

Next, based on the method described in Subsection 2.4.2, we compute the similar substrings between any two phoneme sequences

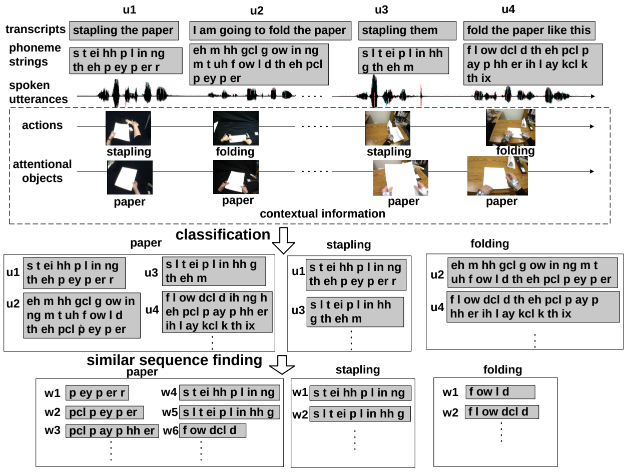
u1
u2
u3
u4
transcripts
stapling the paper
I am going to fold the paper
stapling them
fold the paper like this
phoneme strings
s t ei hh p l in ng th eh p ey p er r
eh m hh gcl g ow in ng m t uh f ow l d th eh pcl p ey p er
s l t ei p l in hh g th eh m
f l ow dcl d th eh pcl p ay p hh er ih l ay kcl k th ix
spoken utterances
actions
stapling
folding
stapling
folding
attentional objects
paper
paper
paper
paper
contextual information
classification
paper
u1 s t ei hh p l in ng th eh p ey p er r
u2 eh m hh gcl g ow in ng m t uh f ow l d th eh pcl p ey p er
u3 s l t ei p l in hh g th eh m
u4 f l ow dcl d ih ng h eh pcl p ay p hh er ih l ay kcl k th ix
stapling
u1 s t ei hh p l in ng th eh p ey p er r
u3 s l t ei p l in hh g th eh m
folding
u2 eh m hh gcl g ow in ng m t uh f ow l d th eh pcl p ey p er
u4 f l ow dcl d th eh pcl p ay p hh er ih l ay kcl k th ix
similar sequence finding
paper
w1 p ey p er r
w2 pcl p ey p er
w3 pcl p ay p hh er
w4 s t ei hh p l in ng
w5 s l t ei p l in hh g
w6 f ow dcl d
stapling
w1 s t ei hh p l in ng
w2 s l t ei p l in hh g
folding
w1 f ow l d
w2 f l ow dcl d

Figure 6. **Word-like unit segmentation**. Spoken utterances are categorized into several bins that correspond to temporally co-occurring actions and attentional objects. Then we compare any pair of spoken utterances in each bin to find the similar subsequences that are treated as word-like units.

in each bin to obtain word-like units. Figure 7 shows an example of extracting word-like units from the utterance u_2 and u_4 that are in the bin of the action "folding". In this way, we spot all the word-like units in each spoken utterance which will be used to associate with their possible grounded meanings to build lexical items. Among those spotted words, some of them have grounded meanings but others are not.

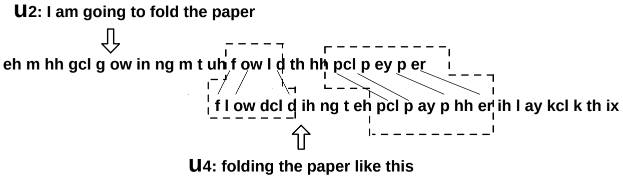
u2: I am going to fold the paper
eh m hh gcl g ow in ng m t uh f ow l d th hh pcl p ey p er
f l ow dcl d ih ng t eh pcl p ay p hh er ih l ay kcl k th ix
u4: folding the paper like this

Figure 7. **An example of word-like unit spotting**. The similar substrings of two sequences are /f ow l d/ (fold), /f l ow dcl d/ (fold), /pcl p ey p er/ (paper) and /pcl p ay p hh er/ (paper).

2.6 Grounding Spoken Words

In the final step, the co-occurrence of multimodal data selects meaningful semantics that associate spoken words with visual representations of objects and body movements (shown in Figure 8). We take a novel view of this problem as being analogous to the word alignment problem in machine translation. For that problem, given texts in two languages (e.g. English and French), computational linguistic techniques can estimate the probability that an English word will be translated into any particular French word and then align the words in an English sentence with the words in its French translation. Similarly, for our problem, if different meanings can be looked as elements of a "meaning language", associating meanings with object names and action verbs can be viewed as the problem of identifying word correspondences between English and "meaning language". In light of this, a technique from machine translation can address this problem. The probability of each word is expressed as a mixture model that consists of the conditional probabilities of each word given its possible meanings. In this way, an Expectation-Maximization (EM) algorithm can find the reliable associations of spoken words and their grounded meanings that will maximize the probabilities.

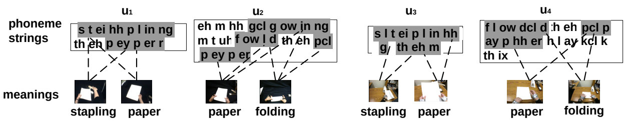
u1
u2
u3
u4
phoneme strings
s t ei hh p l in ng th eh p ey p er r
eh m hh gcl g ow in ng m t uh f ow l d th eh pcl p ey p er
s l t ei p l in hh g th eh m
f l ow dcl d th eh pcl p ay p hh er ih l ay kcl k th ix
meanings
stapling
paper
paper
folding
stapling
paper
paper
folding

Figure 8. **Word learning.** The word-like units in each spoken utterance and co-occurring meanings are temporally associated to build possible lexical items.

The general setting is as follows: suppose we have a word set $X = \{w_1, w_2, ..., w_N\}$ and a meaning set $Y = \{m_1, m_2, ..., m_M\}$, where N is the number of word-like units and M is the number of perceptually grounded meanings. Let S be the number of spoken utterance and all data are in a set $\chi = \{(S_w^{(s)}, S_m^{(s)}), 1 \leq s \leq S\}$, where each spoken utterance $S_w^{(s)}$ consists of r words $w_{u(1)}, w_{u(2)}, ..., w_{u(r)}$, and $u(i)$ can be selected from 1 to N. Similarly, the corresponding contextual information $S_m^{(s)}$ include l possible meanings $m_{v(1)}, m_{v(2)}, ..., m_{v(l)}$ and the value of $v(j)$ is from 1 to M. We assume that every word w_n can be associated with a meaning m_m. Given a data set χ, we want to maximize the likelihood of generating the "meaning" corpus given English descriptions:

$$P(S_m^{(1)}, S_m^{(2)}, ..., S_m^{(S)}|S_w^{(1)}, S_w^{(2)}, ..., S_w^{(S)}) = \prod_{s=1}^{S} P(S_m^{(s)}|S_w^{(s)}) \quad (4)$$

We use the model similar to that of Brown et al. [4]. The joint likelihood of meanings and an alignment given spoken utterances:

$$\begin{aligned} P(S_m^{(s)}|S_w^{(s)}) &= \sum_a P(S_m^{(s)}, a|S_w^{(s)}) \\ &= \frac{\epsilon}{(r+1)^l} \sum_{a_1}^{r} \sum_{a_2}^{r} \cdots \sum_{a_l}^{r} \prod_{j=1}^{l} t(m_{v(j)}|w_{a_{v(j)}}) \\ &= \frac{\epsilon}{(r+1)^l} \prod_{j=1}^{l} \sum_{i=0}^{r} t(m_{v(j)}|w_{u(i)}) \qquad (5) \end{aligned}$$

where the alignment $a_{v(j)}, 1 \leq j \leq l$ can taken any value from 0 to r which indicates which word is aligned with jth meaning. $t(m_{v(j)}|w_{u(i)})$ is the association probability for a word-meaning pair and ϵ is a small constant.

We wish to find the association probabilities so as to maximize $P(S_m^{(s)}|S_w^{(s)})$ subject to the constraints that for each word w_n:

$$\sum_{m=1}^{M} t(m_m|w_n) = 1 \qquad (6)$$

Therefore, we introduce Lagrange multipliers λ_n and seek an unconstrained maximization:

$$L = \sum_{s=1}^{S} \log P(S_m^{(s)}|S_w^{(s)}) + \sum_{n=1}^{N} \lambda_n (\sum_{m=1}^{M} t(m_m|w_n) - 1) \quad (7)$$

We then compute derivatives of the above objective function with respect to the multipliers λ_n and the unknown parameters $t(m_m|w_n)$ and set them to be zeros. As a result, we can express:

$$\lambda_n = \sum_{m=1}^{M} \sum_{s=1}^{S} c(m_m|w_n, S_m^{(s)}, S_w^{(s)}) \qquad (8)$$

$$t(m_m|w_n) = \lambda_n^{-1} \sum_{s=1}^{S} c(m_m|w_n, S_m^{(s)}, S_w^{(s)}) \qquad (9)$$

where

$$c(m_m|w_n, S_m^{(s)}, S_w^{(s)}) = \frac{t(m_m|w_n)}{t(m_m|w_{u(1)}) + ... + t(m_m|w_{u(r)})} \times \sum_{j=1}^{l} \delta(m, v(j)) \sum_{i=1}^{r} \delta(n, u(i)) \qquad (10)$$

The EM-based algorithm sets an initial $t(m_m|w_n)$ to be flat distribution and performs the E-step and the M-step successively until convergence. In E-step, we compute $c(m_m|w_n, S_m^{(s)}, S_w^{(s)})$ by Equation (10). In M-step, we reestimate both the Lagrange multipliers and the association probabilities using Equation (8) and (9).

When the association probabilities converge, we obtain a set of $t(m_m|w_n)$ and need to select correct lexical items from many possible word-meaning associations. Compared with the training corpus in machine translation, our experimental data is sparse and consequently causes some words to have inappropriately high probabilities to associate the meanings. This is because those words occur very infrequently and are in a few specific contexts. We therefore use two constraints for selection. First, only words that occur more than a pre-defined times are considered. Moreover, for each meaning m_m, the system selects all the words with the probability $t(m_m|w_n)$ greater than a pre-defined threshold. In this way, one meaning can be associated with multiple words. This is because people may use different names to refer to the same object and the spoken form of an action verb can be expressed differently. For instance, the phoneme strings of both "staple" and "stapling" correspond to the action of stapling. In this way, the system is developed to learn all the spoken words that have high probabilities in association with a meaning.

3. EXPERIMENTS AND RESULTS

A Polhemus 3D tracker was utilized to acquire 6-DOF hand and head positions at $40Hz$. The performer wore a head-mounted eye tracker from Applied Science Laboratories(ASL). The headband of the ASL held a miniature "scene-camera" to the left of the performer's head that provided the video of the scene from a first-person perspective. The video signals were sampled at the resolution of 320 columns by 240 rows of pixels at the frequency of $15Hz$. The gaze positions on the image plane were reported at the frequency of $60Hz$. Before computing feature vectors for HMMs, all position signals passed through a 6th order Butterworth filter with cut-off frequency of $5Hz$. The acoustic signals were recorded using a headset microphone at a rate of 16 kHz with 16-bit resolution. In this study, we limited user activities to those on a table. Three activities that users performed were: "stapling a letter", "pouring water" and "unscrewing a jar". Figure 9 shows snapshots captured from the head-mounted camera when a subject performed three tasks. Six subjects participated in the experiment. They were asked to perform each task nine times. We collected multisensory data when they performed the task, which were used as training data for our computational system.

The action sequences in the experiments consist of several motion types: "pick up", "line up", "staple", "fold", "place", "unscrew" and "pour". The objects that are referred to by speech are: "cup", "jar", "waterpot" and "paper". For the evaluation purpose, we manually annotated speech data and calculate the frequencies of words. We have collected approximately 960 spoken utterances and on average, a spoken utterance contains 6 words, which illustrates the necessity of word segmentation from connected speech. Among all these words, approximately 12% of them are action verbs and object names that we want to spot and associate with their grounded meanings. These statistics further demonstrate the difficulty of learning lexical items from naturally co-occurring data. These annotations were only used for the evaluation purpose.

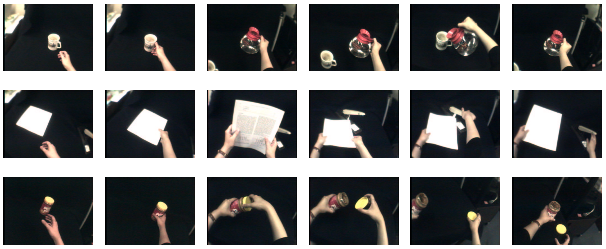

Figure 9. The snapshots of three continuous action sequences in our experiments. **Top row:** pouring water. **Middle row:** stapling a letter. **Bottom row:** unscrewing a jar.

To evaluate experimental results, we define the following four measures: (1) **Semantic accuracy** measures the recognition accuracy of processing non-linguistic information, which consists of recognizing both human actions and visual attentional objects. (2) **Speech segmentation accuracy** measures whether the beginning and the end of phoneme strings of word-like units are correct word boundaries. For example, the string /k ah p/ is a positive instance that corresponds to the word "cup" while the string /k uh p i/ is negative. The phrases with correct boundaries are also treated as position instances because those phrases do not break word boundaries but only combine some words together. (3) **Word learning accuracy (precision)** measures the percentage of successfully segmented words that are correctly associated with their meanings. (4) **Lexical spotting accuracy (recall)** measures the percentage of word-meaning pairs that are spotted by the computational system. This measure provides a quantitive indication about the percentage of grounded lexical items that can be successfully found.

Table 1 shows the results of four measures. The recognition rate of the phoneme recognizer we used is 75% because it does not encode any language model and word model. Based on this result, the overall accuracy of speech segmentation is 69.6%. Naturally, an improved phoneme recognizer based on a language model would improve the overall results, but the intent here is to study the model-independent learning method. The error in word learning is mainly caused by a few words (such as "several" and "here") that frequently occur in some contexts but do not have grounded meanings. The overall accuracy of lexical spotting is 82.6%, which demonstrates that by inferring speakers' referential intents, the stable links between words and meanings could be easily spotted and established. Considering that the system processes natural speech and our method works in unsupervised mode without manually encoding any linguistic information, the accuracies for both speech segmentation and word learning are impressive.

4. CONCLUSION

This paper presents a multimodal learning interface for word acquisition. The system is able to learn the sound patterns of words and their semantics while users perform everyday tasks and provide spoken descriptions of their behaviors. Compared to previous works, the novelty of our approach arises from the following aspects. First, our system shares user-centric multisensory information with a real agent and grounds semantics directly from egocentric experience without manual transcriptions and human involvement. Second, both words and their perceptually grounded meanings are acquired from sensory inputs. Furthermore, grounded meanings are represented by perceptual features but not abstract

Table 1. Results of word acquisition

	sound pattern examples	semantics	speech segmentation	word learning	lexical spotting
pick up	/ p ih kcl k uh p /	96.5%	72.5%	87.5%	72.6%
place	/ p l ey z /	93.9%	66.9%	81.2%	69.2%
line up	/ l ay n ax p/	75.6%	70.3%	86.6%	83.5%
staple	/ s t ey pcl p /	86.9%	70.6%	85.3%	90.6%
fold	/ f ow l d /	86.3%	69.8%	89.2%	87.7%
unscrew	/ ax n s kcl k r uw /	90.6%	73.8%	91.6%	80.6%
pour	/ p aw r /	86.7%	65.3%	91.9%	85.5%
paper	/ pcl p ey p axr /	96.7%	73.9%	86.6%	82.1%
jar	/ j aa r /	91.3%	62.9%	92.1%	76.6%
cup	/ k ah p /	92.9%	68.3%	87.3%	76.9%
waterpot	/ w ax t axr p ux t /	87.5%	71.9%	85.6%	82.3%
overall		90.2%	69.6%	87.9%	82.6%

symbols, which provides a sensorimotor basis for machines and people to communicate with each other through language. From the perspective of machine learning, we argue that the solely statistical learning of co-occurring data is less likely to explain the whole story of language acquisition. The inference of speaker's referential intentions from their body movements provides constraints to avoid the large amount of irrelevant computations and can be directly applied as deictic reference to associate words with perceptually grounded referents in the physical environment. From an engineering perspective, our system demonstrates a new approach to developing human-computer interfaces, in which computers seamlessly integrate in our everyday lives and are able to learn lexical items by sharing user-centric multisensory information.

5. ACKNOWLEDGMENTS

The authors wish to express their thanks to Mary Hayhoe for fruitful discussions. Brian Sullivan was a great help in building the experimental system. This work is supported by the NSF under research grant number EIA-0080124 and by the NIH/PHS under research grant number 2 R01 RR09283.

6. REFERENCES

[1] R. Adams and L. Bischof. Seeded region growing. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 16(6):641–647, June 1994.

[2] C. C. Aggarwal and P. S. Yu. Finding generalized projected clusters in high dimensional spaces. In *SIGMOD*, Dallas, Texas, USA, 2000. ACM.

[3] D. H. Ballard and C. Yu. A multimodal learning interface for word acquisition. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Hong Kong, April 2003.

[4] P. F. Brown, S. Pietra, V. Pietra, and R. L. Mercer. The mathematics of statistical machine translation:parameter estimation. *Computational Linguistics*, 19(2), 1993.

[5] L. Campbell, D. Becker, A. Azarbayejani, A. Bobick, and A. Pentland. Invariant features for 3-d gesture recognition. In *Second International Workshop on Face and Gesture Recognition*, pages 157–162, Killington, VT, Oct. 1996.

[6] J. Hartigan. *Clustering Algorithms*. Wiley, New York, 1975.

[7] M. Hayhoe. Visual routines: A functional account of vision. *Visual Cognition*, 7:43–64, 2000.

[8] M. Land, N. Mennie, and J. Rusted. The roles of vision and eye movements in the control of activities of daily living. *Perception*, 28:1311–1328, 1999.

[9] R. P. Lippmann. Review of neural networks for speech recognition. *Neural computation*, 1(1):1–38, 1989.

[10] B. W. Mel. Seemore: Combining color, shape, and texture histogramming in a neurally inspired approach to visual object recognition. *Neural Computation*, 9:777–804, 1997.

[11] T. Oates, L. Firoiu, and P. R. Cohen. Clustering time series with hidden markov models and dynamic time warping. In *Proceedings of the IJCAI-99 Workshop on Neural, Symbolic and Reinforcement Learning Methods for Sequence Learning*, pages 17–21, 1999.

[12] S. Oviatt. Multimodal interfaces. In J. Jacko and A. Sears, editors, *Handbook of Human-Computer Interaction*. Lawrence Erlbaum, New Jersey, 2002.

[13] L. R. Rabiner and B. Juang. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[14] T. Robinson. An application of recurrent nets to phone probability estimation. *IEEE Transactions on Neural Networks*, 5(2):298–305, 1994.

[15] D. Roy and A. Pentland. Learning words from sights and sounds: A computational model. *Cognitive Science*, 26(1):113–146, 2002.

[16] D. D. Salvucci and J. Anderson. Tracking eye movement protocols with cognitive process models. In *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society*, pages 923–928, LEA: Mahwah, NJ, 1998.

[17] B. Schiele and J. L. Crowley. Recognition without correspondence using multidimensional receptive field histograms. *International Journal of Computer Vision*, 36(1):31–50, 2000.

[18] J. M. Siskind. Grounding language in perception. *artificial Intelligence Review*, 8:371–391, 1995.

[19] P. Smyth. Clustering sequences with hidden markov models. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, page 648. The MIT Press, 1997.

[20] M. J. Swain and D. Ballard. Color indexing. *International Journal of Computer Vision*, 7:11–32, 1 1991.

[21] S. Wang and J. M. Siskind. Image segmentation with ratio cut. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003.

[22] C. Yu and D. H. Ballard. Learning to recognize human action sequences. In *IEEE Proceedings of the 2nd International Conference on Development and Learning*, pages 28–34, Boston, MA, 2002.