AUTONOMOUS LEARNING OF ACTION-WORD SEMANTICS IN A HUMANOID ROBOT

BY

LOGAN NIEHAUS

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2011

Urbana, Illinois

Adviser:

Professor Stephen E. Levinson

ABSTRACT

For creation of an artificial agent that is capable of using language naturally, models that only manipulate symbols or classify speech are ineffective. The semantic information which language conveys must be grounded in the agent's complete sensorimotor experience. Typically, patterns from visual, auditory, and proprioceptive data streams which share the same conceptual cause are fused together in an associative memory at the core of the language model. Coupling of motor and auditory modalities, which is crucial for a large part of semantic understanding, presents a particularly difficult challenge. Words and actions both need models capable of capturing spatial and temporal structure, and training algorithms that can learn in a self-organizing, incremental fashion. Presented is a method for online learning of word and action lexicons based on the hidden Markov model. The model is then evaluated through action-word learning experiments implemented on a humanoid robot.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

We begin by defining the term *language*, specifically as it relates to humans, thusly: language is the system of interaction between persons which relates some information about the state of the world. The fundamental long-term goal of this work is the design of an artificial agent which is able to use language in the way that humans do. Besides the enormous potential for benefit that comes with machines that are able to interface naturally with humans, the problem is integral to our understanding of the nature of intelligence. It is hoped that answers in the domain of artificial intelligence will evolve in parallel with those from cognitive science and will eventually provide some small part of an explanation as to how the mind works.

1.1 Historical Motivations

At many points throughout history, there have been attempts to bring the study of language and cognition under the umbrella of mathematics. Many of these relied on the intuition that rational thought could be expressed in terms of formal logic and calculation. While the theoretical limits of formal logic were eventually exposed, the idea that any calculation (and potentially thought) could be performed by mechanical computation seemed for the first time a real possibility. Alan Turing's 1950 paper [1] on the potential of using a computer to emulate the mind was a foundational moment for the field of artificial intelligence. In it he outlines an experiment for gauging the intelligence of a machine based on its ability to converse realistically with a human.

One of the initial possibilities Turing names for a research program aimed toward passing the test is "...that it is best to provide the machine with the best sense organs that money can buy, and then teach it to understand and speak English" [1]. This advice seemed to be almost instantly cast aside, and early research proceeded to tackle AI problems under the familiar paradigms of formal logic and symbolic manipulation. After three decades of moderate success applied to mostly limited task domains, "good old-fashioned AI" seemed to have reached its limits. Purely symbolic systems were inflexible, relied heavily on detailed knowledge databases provided by experts, and were unable to address many fundamental issues relating to using language: namely, where do the symbols get their meaning?

Some of the troubles of this program can be understood through comparison to the field of *cybernetics* [2], which was put forth by Norbert Wiener around the same time as Turing presented his ideas. Cybernetics is a broad field of study which encompasses such systems concepts as control, feedback, communication, information, learning, and their application to both animal and machine. However unlike in symbolic approaches, here humans are understood in terms of how the fundamental structures of which they are composed process information. Cognition (and therefore language) is a function of the phyiscal manifestation of the brain, shaped by its developmental processes, and can not be understood separately from it. This premise has two direct consequences for design of an artificially intelligent agent: the first is that it requires some sort of embodiment. The second is that the developmental processes of the mind must be considered (i.e., learning). It is on this principle of embodiment and learning which the following research rests.

1.2 A Mathematical Approach

Perhaps the best method for introducing the core issues of language acquisition is to start with an intuitive thought experiment. For the mathematically inclined, a machine learning approach to the language learning problem can be formulated in terms of parameter estimation for a given statistical model. Let us assume the goal is to learn a single word, in terms of its speech signal, $A(t)$. Typically, a statistical model, $\mathcal{M}_a$, is created, and its parameters are estimated based on the training sample by any number of mathematical strategies (e.g., maximum likelihood). The maximum likelihood approach could be formulated as

$$\mathcal{M}_a^* = \arg\max_{\mathcal{M}_a} P\left(A(t)|\mathcal{M}_a\right). \tag{1.1}$$

The result is a model $\mathcal{M}_a^*$ that somehow encodes the important features of the given word and can be applied later for any number of purposes, such as recognition. While such a method may learn a speech signal quite well, it still does not convey any of the meaning of the word. The semantic value of a word (its "meaning"), is tied to what information about the world that word represents. This information comes to humans not only through the speech signal but also through every modality of the sensorimotor system. Without this information, a computer can not learn a natural language in any meaningful sense.

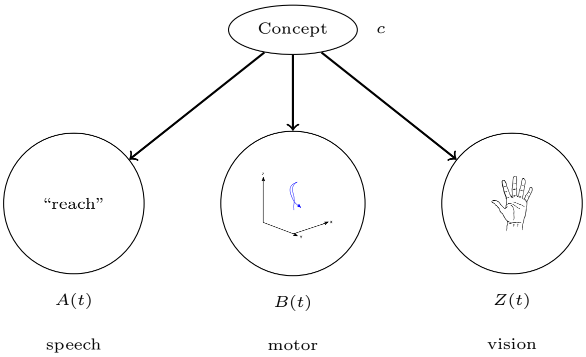


Figure 1.1: Concept learning.

A new worldview, in which a concept is grounded in many senses, is needed. Consider first Figure 1.1. This simplified view shows how a concept may be presented. Information about the action "reach" is stored in individual models of its sensory effects (the word "reach," watching the action being performed, performing it yourself), as well as in a combined model which integrates these effects. The *concept* learning problem can be reposed as

$$\mathcal{M}_T^* = \arg\max_{\mathcal{M}_T} P\left(A(t), B(t), Z(t), c|\mathcal{M}_T\right). \tag{1.2}$$

Here, learning a word and its meaning involves creating some model which can be trained by maximizing the joint probability of grouped streams of multimodal sensory inputs, i.e., creation of an associative memory. If we want our experiments to mirror real language acquisition scenarios, then the signals $A(t)$, $B(t)$, and $Z(t)$ should be expected to be continuous streams containing a number of varied concepts (now a concept "process" $\bar{C}$), which are often repeated many times throughout learning. The task is now twofold: creating a sensory system which experiences those inputs in a realistic way, and creating a model which is able to effectively capture the structure of the inputs in a way that is useful for language.

Developmental robotics is one obvious path to such a solution, as it allows the creation of computational models which are implemented on an embodied platform. In the past decade, a number of elegant solutions and architectures have focused on robots that learn the meaning of words through audiovisual associations [3],[4]. The human sensory system is quite rich, however, and many words lose a great deal of their meaning without reference to spatial and motor reasoning (particularly prepositions, verbs, and a number of adjectives). However, work on semantic grounding in the motor system has been sparse, and many experiments presented often employ trivialized models of either linguistic or motor function.

The first step in improving existing models for action-word learning is to clearly define the action-word learning scenario. Such an experiment will proceed as follows: a human tutor is present with the robot. The human will share his/her linguistic knowledge of action words by first having the robot produce an action (or in some cases, the action will be produced on the robot with human guidance). During the performance of the action, the tutor will name the action aloud to the robot. This procedure will be performed many times, with many different words being presented multiple times. The goal of such an experiment is to mirror a typical interaction between an adult tutor and a child. Here, the adult will name aloud objects or actions which share the joint attention of both participants. Such interactions are fundamental for the acquisition of linguistic knowledge: in order for the child to understand what the linguistic labels refer to, there must be some shared sensory input between them and the tutor.

Going forward, meaningful integration of motor function into a unified multimodal framework will be key to developing a realistic use of natural language by an artificial intelligence. The goal of this work is to put forward such a framework. Special focus will be given to the challenge of acquisition of a symbol-level representation of speech and motor input. It is also important that the framework satisfy a set of conditions, based on certain characteristics which are considered hallmarks of language acquisition in humans. Models used should be unsupervised and self-organizing. No innate knowledge can be given to the system; and all models may learn only online and incrementally, through sensory experience and interaction.

The rest of the work will be presented in this paper as follows. Section II contains a review of biological motivations/justifications for an embodied approach and a brief discussion on some current approaches to the problem, as well as an overview of technical work whose methods are directly applicable to improved solutions. Section III covers the proposed solution for the learning task. Section IV presents a set of small experiments on real data, which aims to show the application of such an architecture to many different types on input. A small set of results for a combined action-word learning scenario is also presented. Finally Section V will explore the various shortcomings of the models and ways in which real behavioral concepts can be applied to make these simple experiments more capable of mirroring actual language acqusition processes.

CHAPTER 2

BACKGROUND

As alluded to in the introduction, the chosen method for understanding language is a computational view: the brain as an information processor. What are the computational mechanisms by which the process of language acquisition can be described? The first steps of such a journey might start at the simplest levels, i.e., the cognitive building blocks on which language rests. The means by which sensory information is acquired, the way in which it is stored, and how it links to language are all such pieces. The body of work that contributes to developing computation models for these comes from both biology and mathematics. Studies in neuroscience and psychology provide general guidelines for the development of cognitive behaviors, and mathematics provides the tools with which to emulate these behaviors.

2.1 Studies of the Mind

As mentioned several times in the introduction, some approximation of the sensorimotor system is necessary for authentic use of natural language as we understand it. Specifically, issues relating to the symbol grounding problem [5] require physical embodiment. While there is no universally accepted definition, robots are the class of systems which meet the requirements of integrated computation and embodiment. But which parts of the sensorimotor system does the robot need to be equipped with? Can a satisfying approximation be acheived with only cameras and microphones? Or is some sort of motor system necessary for a full linguistic capacity? Numerous philosophical arguments can be levied for or against such a proposition, but empirical evidence may yield answers which are somewhat less equivocal. Recent gains in understanding of neuroscience and developmental psychology have served to highlight the close interdependence of action and language in the brain.

Functional MRI studies have demonstrated the simulataneous activation of the motor cortex upon hearing an action word, in areas which correspond somatotopically to the location of the word [6]. Even conservative readings of these results would seem to point to the conclusion that at least part of the semantic information present in action words is represented in terms of its motor embodiment. Such research comes on the heels of studies done on the F5 cortical region of macaques, considered a homologue of Broca's area in humans. It was discovered [7] that certain neurons in F5 fired when the monkey both performed an action and observed a human performing the same action (e.g., grasping). Because of this behavior, such neurons were dubbed "mirror neurons". Some have used this as an action basis for perceptual understanding [8] (i.e., motor knowledge of a grasp is directly recruited and necessary for decoding the utterance "grasp").

However, such views are controversial, and often run counter to empirical evidence [9]. Instead, what the results may imply is a process by which sensorimotor information is linked as a separate conceptual object that then provides "cascading" activation to other sensory modalities. Such a proposition is important, as it serves to bridge a long-standing divide between two hypotheses about how our internal representations of the world are constructed. The first is the idea that internal concepts are abstract objects which can exist and be manipulated independently of sensory information – often called the symbolic or disembodied hypothesis. At the opposite end is the belief that our conceptual understanding is composed only of our various sensory representations, and comprehension is sensorimotor simulation – the sub-symbolic or embodied hypothesis. The approach is taken here, as is taken in [9], that while language is clearly grounded in the continuous sensory domain, at some level a symbolic representation of the world is important, and that these two ideas are not incomptable. From this a difficult but necessary goal becomes quite clear: we must find a way to represent our continuous sensory world in terms of categories, grouped by both their latent and salient structure.

This conclusion allows for the application of a large class of well-studied machine learning techniques which can organize diverse sensory data into a discrete (symbolic) lexicon. Further guiding the choice of which particular algorithms and models to use are a small set of developmental and organizational principles. Foremost among these are the concepts of incremental/online learning and self-organization. Online learning refers to the criterion that the robot be continuously updating its understanding of the world; sensory information is processed and utilized as it is presented. The principle of self-organization is the goal that any model emlpoyed be given as little structure as necessary (i.e., no prior knowledge of phonetic representations or linguistic categories). Ideally, linguistic function emerges from a small set of basic computations.

Unfortunately, these requirements are still quite broad, and it is difficult to develop a robust, general algorithm that is able to tractibly navigate the amount of sensorimotor data acquired by the agent. Evidence is beginning to show that humans themselves have a toolbox of information processing "tricks" which they use to guide their learning processes. Such techniques are based on detecting intermodal synchrony [10], event contingency [11], and joint attention [12]. Algorithmic use of these heuristics can dramatically simplify the machine learning strategies required both to create symbolic representations of sensory knowledge and to tie them together in an associative memory.

2.2 Developmental Robotics

Because of previous failures in computer natural language processing systems, focus over the past decade has shifted heavily into the area of developmental robotics. The robotics work so far has only begun to address integration of motor and linguistic function; but it has already provided many reliable methods for symbol association, as well as methods for basic knowledge representation.

The work in [13] focused on creating audiovisual associative memory in order to learn a small set of concepts for the toys in the robot's pen. Symbolic associations were made using a "cascading" architecture learning cooccurance of modal classes. The classes of objects based on visual inputs were quite easily clustered; but words, however, proved more difficult. A single phonetic classifier was used to produce inputs for a word model. The word model representation consisted of a histogram of these phones over a given segment of speech, as well as energy and utterance length information. Unfortunately, the word lexicon was not capable of online training and used a built-in representation of speech. This issue highlights the difficulty in modeling a word's speech signal, which is capturing its nonstationary statistics. This model bypassed the problem by simply collapsing its temporal structure into a purely spatial one.

Roy [3] presented a model of "Cross-channel Early Lexical Learning" for audiovisial language learning which properly addresses the issues above. He uses a hidden Markov model (HMM) for word representation and learns a lexicon of words by applying a prior phonotactic model to the phone classification stream. Object classes are learned by basic clustering techniques, such as measuring the feature distance between visual representations. During a live experiment, a tutor provides linguistic labels of a jointly attended object to the robot. A semantic model of word-object pairs is created by the maximizing mutual information over a set of lexical candidates. As in [13] though, there is no way of capturing temporal sensory structure without relying on pre-built models.

Work in humanoid robotics which begins to tackle the issue of integrating action and language has focused primarily on the modeling of motor inputs, while generally ignoring the challenge of speech lexicon acquisition [14],[15],[16]. In the framework presented by Marocco [15], speech inputs are given by pre-defined symbols and have no representation beyond this. Additionally, the neural networks used for learning are trained in "batch" mode. A main goal of the research presented here is to expand upon these initial attempts at linking motor knowledge and language, by augmenting the existing statistical techniques for semantic association with new methods for representing sensory data that includes temporal structure.

2.3 Statistical Methods for Representation

The main shortcomings to be found with many of the current architectures applied for learning in developmental robotics lie with the general way of representing sensory information. Here, representation is defined as the way in which sensory knowledge and information are encoded and applied to various aspects of cognition. In the examples above, general sensory representations are often nonexistent [15], or nonrealistic [13]. It is instead desired that methods for developing representations adhere to the basic developmental principles of self-organization and online learning.

Automatic speech recognition (ASR) has already laid considerable groundwork for sensory representation, though this is not a goal that it usually aspires to. Statistical models, namely the hidden Markov model (HMM), are nearly ubiquitous in ASR [17],[18]. Their ability to capture the spatial and temporal structure of the speech signal and convert it to a symbolic representation provides a powerful tool in the representation of sensory data which is intrinsically time-varying. While modern ASR systems' reliance on expertly trained data seems to make them unfeasible from a developmental standpoint, a few early, often overlooked results from the field point to potential for such use. Cave and Neuwirth [19] first demonstrated the potential of the HMM for unsupervised learning on English text. Results showed that the model parameters converged to an internal representation in which each symbol was linked to a orthographic class (consonants, vowels, whitespace). Poritz [20] later showed that a model trained on speech converged to a phonetic representation (vowels, nasals, plosives, fricatives, etc.).

With these results, it was showed that the HMM could capture underlying structure in the data, with almost no prior assumptions (aside from those inherent to the model) of what such a structure might be. Additionally, it did this by looking at only sample data and not through extrinsic guidance at the model level. This lack of need for modality-specific structure suggests that it may also be useful as a general representational tool. Studies of the human cortex (the part of the brain implicated in most of the cognitive abilities discussed here) have already shown a uniformity in physical structure across different sensory/conceptual processing regions [21], and some evidence suggests that there may be a set of general computational and functional principles which underly this uniformity [22],[23]. While the HMM is not a model of such functionality, the decision to use it as a general structure for sensory representations across many modalities is not without merit.

Indeed, application of the HMM to action representation is well established in robotics and other areas. Of particular interest for this work are approaches for learning lexicons of action primitives modeled by HMMs. Kulic et al. [24] propose a method to incrementally build a lexicon of full-body actions, as well as an organizational structure for grouping and generalization, by hierarchically clustering HMM encodings of the observational history. Calinon and Billard [25] propose a similar architecture, in which the HMM exemplars are continuously updated on the presentation of new data but without a need to store the entire training history for a class. Such an approach will serve as a starting point in tackling the issues of action and speech representation which have not been adequately addressed by the robotics experiments above.

CHAPTER 3

MODEL OVERVIEW

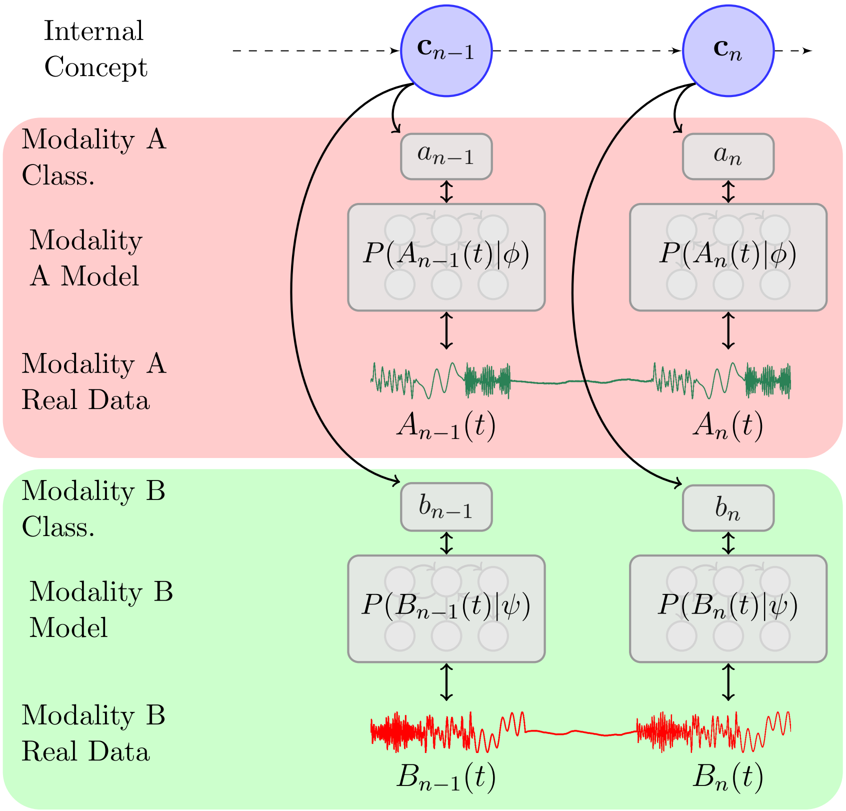


Figure 3.1: Graphical model of problem structure.

Based on the proposed action-word learning scenario, the goal is now to formulate an abstract view of what the experiment may look like in terms of the data it presents. Consider the diagram in Figure 3.1, a conceptual illustration of the world during the learning task. There are some number of hidden "causes" which emit some finite length observation sequence over various sensory modalities (e.g., a word and an action). These separate modalities are assumed to be independent of one another. It is also assumed that the observation signals corresponding to each concept are segmentable (e.g., the audio recording of "reach, grab, punch," can be segmented into individual recordings of "reach," "grab," and "punch").

The assumptions made about the model of the world (Figure 3.1) are necessary to break down equation (1.2) into components for which the parameter estimation problem is tractable. The first such assumption is the independence of each modality's observations, conditioned on the concept and the total model $\mathcal{M}_T$. If we consider $\mathcal{M}_T = (\mathcal{M}_a, \mathcal{M}_b, \mathcal{M}_c)$ to be the tuple of the concept model along with its individual modality models, equation (1.2) can be rewritten as

$$\begin{aligned}\mathcal{M}_T^* = \arg\max_{\mathcal{M}_T} & P(A(t)|\bar{C}, \mathcal{M}_c, \mathcal{M}_a) \\ & P(B(t)|\bar{C}, \mathcal{M}_c, \mathcal{M}_b) P(\bar{C}|\mathcal{M}_c).\end{aligned} \tag{3.1}$$

By applying the assumption that $A(t)$ and $B(t)$ are sparse and segmentable signals, the process $\{A(t), B(t)\}$ can be divided into a set of the "interesting" parts of the signal, $\{A_n(t), B_n(t)\}_{n=1}^N$. The final crucial assumption is that each sequence is generated according to some modality-specific lexical template (i.e., a word in a vocabulary or an action in a set of primitives). That means that the signals $A_n(t)$ and $B_n(t)$ are drawn with statistics modeled by some element of the set of statistical models for the modality, called the modal "lexicon." We can then reference $A_n(t)$ and $B_n(t)$ with a symbol pair, $\{a_n, b_n\}$, corresponding to the index of model from which they were most likely drawn: $a_n \in \mathcal{Q} = \{q_1, q_2, \ldots q_K\}$ and $b_n \in \mathcal{S} = \{s_1, s_2, \ldots s_L\}$. Underlying this transformation is the motivation that it is only desired to learn symbolic-level associations between actions and words.

This step is important, as it allows us to separate the larger task into the concept-symbol association problem and the problem of creating word and action lexicons, $\mathcal{Q}$ and $\mathcal{S}$. While the individual observations $\{a_n, b_n\}$ are still considered to depend on the concept c_n, the lexical models they represent should be trained to learn the structure of their separate modalities. For a single modality, the lexical creation problem then becomes an online clustering task, in which we wish to group a series of (spatially) continuous observations $\{A_n(t), B_n(t)\}_{n=1}^N$ into some smaller set of classes. When $A_n(t)$ and $B_n(t)$ have nonstationary statistics, as is the case for speech and action, this becomes particularly challenging. Addressing the general lexical creation problem and applying it to language learning is the primary contribution of this work.

3.1 Proposed Model

The paradigm of classification, thresholding, and either modification or addition of clusters is the underlying mechanism of this framework. As mentioned in the model overview, the goal is to take a signal $A_n(t)$, and mark it as belonging to some class of similar signals, referenced with symbol q_k. Each class is abstracted using some statistical model for that class. However, $A_n(t)$ is a signal with nonstationary statistics, and a model needs to be used which adequately captures this. Such a model is the hidden Markov model (HMM).

At a high level, clustering seeks a low-dimensional representation of a data set through grouping of similar samples. Cluster membership for a sample is usually determined by distance to each groups' exemplar, according to some metric. Here, group exemplars are coded in HMMs, using probability to measure membership. The clustering process itself is based on a competitive learning approach, similar to the one used in [26]. Competitive learning methods are well suited to problems for which online training is needed. Such an approach works as follows: a set of HMMs used to represent the cluster centers evaluate the distance between a new input sequence and the cluster by calculation of the likelihood function. The model to which the new sequence is "closest" will be incrementally modified to fit the new data, while all other HMMs will take no action. Sequences which are not close enough to any cluster (given some threshold) are considered novel, and a new cluster center is initialized using that sequence.

A block diagram of the data flow for the lexical acquisition task is shown by Figure 3.2. A stream of sensory data (potentially preprocessed as feature vectors) is first saliency-gated for activity. This gating function may be a simple signal-energy-based threshold or a more complex behavior-based heuristic. After this, "interesting" segments are sent to the online clusterer, which then produces a symbolic output.

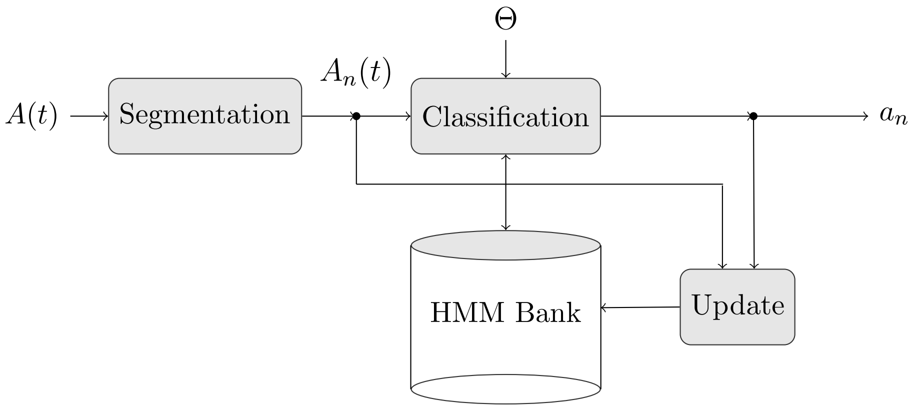


Figure 3.2: Block diagram of data flow for the proposed architecture.

3.1.1 The Hidden Markov Model

(The following section is a brief overview of the terms and definitions associated with the HMM as they pertain to its use in this thesis. Those unfamiliar with the HMM are advised to first read the tutorial by Rabiner [17]). The HMM is a doubly stochastic process, defined by a set of state-observation pairs X_t, Y_t, where X_t is a discrete-time, unobservable process with Markov dynamics, and Y_t is an observable process, with output distribution dependent only the value of the current hidden state X_t. The exact model considered here is a set of HMMs, which each have their own hidden state and output distribution parameterizations.

For each individual HMM, the parameterization can be divided into those which pertain to the internal model and those which pertain to the observation statistics. The internal process X_t is drawn from some finite state space $\mathcal{I} = \{1, \ldots r\}$. The state of the process at time t depends only on the state of the process at time $t-1$, with this transition being characterized by the $r \times r$ matrix A. The entries of A are given by

$$A = [a_{ij}] = P\left(X_t = j | X_{t-1} = i\right). \tag{3.2}$$

At each time step, an observation Y_t will be generated from a distribution whose parameters depend on the current unobservable state at that time, X_t. Usually these distributions are modeled as Gaussian PDFs or discrete probability mass functions. We will first consider single multivariate Gaussian observation distributions. Here, observations Y_t are continuous vectors over $\mathbb{R}^d$. This distribution is of the form

$$f_Y(y) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\tfrac{1}{2}(y-\mu)'\Sigma^{-1}(y-\mu)\right), \tag{3.3}$$

$$\mu = \left[\mathrm{E}\left[y_1\right], \mathrm{E}\left[y_2\right], \ldots, \mathrm{E}\left[y_d\right]\right], \tag{3.4}$$

$$\Sigma = [\mathrm{Cov}[y_i, y_j]]_{i=1,2,\ldots,d;j=1,2,\ldots,d}. \tag{3.5}$$

There are three basic applications commonly discussed with relation to HMMs. The first is finding the likelihood of an observation sequence given a set of model parameters. The second is finding the most likely internal state sequence given an observation sequence. The final is the training task, finding the set of model parameters which maximizes the likelihood of an observation. For this application only the first and third solutions are needed for calculation of cluster membership and update of cluster prototypes, respectively. The second problem is important for the traditional single-level HMM, and is discussed in Section 3.2.

3.1.2 Formal Description of Lexicon Creation

Let K and r be the cardinality of set of individual HMM classifiers for a modality, $\mathcal{Q}$, and internal state spaces of each HMM, X, indexed by $k \in \mathcal{K}$ and $i, j \in \mathcal{I}$ respectively. Furthermore, let T be the length of a given observation sequence $y(t) = \{y_1, \ldots y_T\}$. Each HMM is parameterized by a vector $\phi_k \in \mathbb{R}^p$, corresponding to its index in the bank q_k, for which p is the total number of parameters for the model. These parameters could belong to a number of different families of distributions; but for the purposes of this model, discussion will be limited to use of the multivariate Gaussian as the observable distribution. The result is the following description of a given modality's model:

$$\mathcal{M}_a = (\phi_1, \phi_2, \ldots \phi_k, \ldots \phi_K), \tag{3.6}$$

$$\phi_k = (A, \mu_1, \mu_2 \ldots \mu_r, \Sigma_1, \Sigma_2, \ldots \Sigma_r). \tag{3.7}$$

To aid in the derivation of solutions for the classification and training problems, the forward prediction filter and the class conditional density vectors are now defined. The forward prediction filter is the distribution over the hidden states at a point in time given all past observations and the parameter set. The expression for the filter can be written as

$$u_t(\phi_k) = [u_{t_1}(\phi_k), u_{t_2}(\phi_k), \ldots u_{t_r}(\phi_k)]' . \tag{3.8}$$

in which each element of this column vector is given by

$$u_{t_i}(\phi_k) = P\left(X_t = i | y_{t-1}, \ldots y_1, \phi_k\right). \tag{3.9}$$

This equation is similar to what is generally known from Baum [27] as the forward probability. Similarly to Baum's forward algorithm, the prediction filter can be calculated upon each observation recursively using

$$u_{t+1}(\phi_k) = \frac{A'(\phi_k) F(y_t|\phi_k) u_t(\phi_k)}{f'(y_t|\phi_k) u_t(\phi_k)}. \tag{3.10}$$

The quantity $F(y_t|\phi_k)$ in equation 3.10 is a diagonal matrix containing the observation probabilities conditioned on the output distribution parameters for each state. This matrix can be expressed by

$$F(y_t|\phi_k) = \operatorname{diag}\left[f_1(y_t|\phi_k), \ldots f_r(y_t|\phi_k)\right], \tag{3.11}$$

for which each element is defined by

$$f_i(y_t|\phi_k) = P(y_t|\mu_i(\phi_k), \Sigma_i(\phi_k)). \tag{3.12}$$

With these two auxiliary elements defined, it is now possible to discuss the application of the algorithm. The algorithm can generally be divided into two steps, the first a classification step, and the second a training step (optional). The classification step calculates the membership of a sequence for each HMM in the model; and if desired, the model with the greatest membership will update its parameter set to better "fit" the classified input.

The classification problem can formally be described as picking the index k of the parameter set which has the highest "ownership" among all sets in $\mathcal{Q}$.

$$\begin{aligned}\hat{k} &= \arg\max_{k\in\mathcal{K}} \mathcal{P}_k \\ &= \arg\max_{k\in\mathcal{K}} \frac{1}{T}\log\left[P\left(y_1, y_2, \ldots y_T|\phi_k\right)\right]. \qquad (3.13)\end{aligned}$$

The ownership $\mathcal{P}_k$ is the length-normalized log likelihood of an output sequence for each HMM's parameterization. To derive $\mathcal{P}_k$, it is helpful to begin by deriving the likelihood function $P(y_1, \ldots, y_T)$ in terms of the forward prediction filter u_t and the emission probability vector $f(y_t)$,

$$\begin{aligned}P(y_1, \ldots y_T|\phi) &= P(y_1|\phi)P(y_2|y_1,\phi)P(y_3|y_2,y_1,\phi)\ \ldots \\ &= \prod_{t=1}^{T} P(y_t|y_{t-1},\ldots y_1,\phi) \\ &= \sum_{x_1=1}^{r} P(X_1,\phi)P(y_1|x_1,\phi) \\ &\quad \prod_{t=2}^{T}\left(\sum_{x_t=1}^{r} P(y_t|x_t,y_{t-1},\ldots,\phi)P(x_t|y_{t-1},\ldots,\phi)\right) \\ &= f'(y_1|\phi)u_1(\phi)\prod_{t=2}^{T} f'(y_t|\phi)u_t(\phi). \qquad (3.14)\end{aligned}$$

As a final step, the initial state $u_1(\phi)$ is set to some probability vector $\pi(\phi)$. Taking the log and normalizing by the observation length yields

$$\mathcal{P}_k = \frac{1}{T}\sum_{t=1}^{T} log\left[f'(y_t|\phi_k)u_t(\phi_k)\right]. \qquad (3.15)$$

Training an HMM is the task of fitting a model to a given set of observations or, in other words, estimating the parameters of the model which maximize the likelihood of a model producing the given observations. The Maximum Likelihood Estimator formulation in terms of the observed data Y and the space of HMM parameterizations Φ is given by

$$\phi^* = \arg\max_{\phi\in\Phi} P(y_1, \ldots y_T|\phi). \qquad (3.16)$$

The most popular method of approaching equation 3.16 is the Baum-Welch algorithm [27]. Baum's algorithm is itself a special case of the more general Expectation Maximization algorithm [28], which allows for the esimation of parameters in situations for which there are both observed and unobserved random variables, such as for the HMM. For such latent variable models, direct calculation of 3.16 is intractable; the EM algorithm instead calculates the expectation of both Y and X (together often called the "complete data"), given a guess of the parameter set, and then maximizes this quantity. The expectation step is given by

$$Q(\phi|\phi^{(k)}) = E\left[\log p(X,Y)|y,\phi^{(n)}\right], \tag{3.17}$$

for which

$$\log p(X,Y|\phi) = \log \pi(\phi) + \sum_{t=1}^{T-1} \log a_{x_{t+1},x_t} + \sum_{t=1}^{T} \log f(y_t|x_t,\phi). \tag{3.18}$$

The second step consists of finding a new estimate $\phi^{(n)}$ which maximizes Q, which is usually referred to as the *auxiliary function*. This new parameter set is then used to recalculate equation 3.17, and the process is repeated. Update equations for the maximization step are given in [27]. An attractive property of these equations, and EM algorithms in general, is that the the likelihood function $P(Y|\phi^{(n)})$ is nondecreasing with successive iterations (increasing n). Furthermore, any limit point ϕ^* of the sequence $\phi^{(n)}$ is a critical point of the likelihood function. However, a side effect of this property is that $P(Y|\phi^*)$ is not necessarily a global maximum and can occur at any maximum or inflection point.

Another even more important caveat of Baum's algorithm is that training of an HMM requires all of the data to be presented at once. Such a property is not ideal for a system in which it is desired to continually update the parameters of an HMM to fit a series of examples. Doing so would require storage of past training data in order to function. Instead, a different method for updating the parameters is implemented. In particular, the Recursive Maximum Likelihood Estimation (RMLE) algorithm is used [29],[30]. The RMLE algorithm is a stochastic gradient descent technique, and an alternative approach for solving equation 3.16. Stochastic gradient algorithms are generally of the form

$$\psi_{t+1} = \psi_t - \epsilon \nabla C(\psi_t), \tag{3.19}$$

for which $C(\psi_t)$ is the cost function given the current parameter at time t. In the maximum likelihood estimation case, the cost function is the negative log likelihood function. Intuitively, this approach means that at each iteration, the parameter is moved along the manifold in the direction of increasing likelihood, with step size scaled by ϵ. For the RMLE algorithm specifically, the cost function at each step can be calculated using the forward prediction filter. Therefore, the RMLE algorithm takes the form

$$\phi_{t+1} = \Pi_G(\phi_t + \epsilon_t S(y_t, u_t, w_t|\phi_t)), \tag{3.20}$$

in which S is called the *incremental score vector.* The incremental score vector can be calculated by taking the derivitive of the incremental log likelihood function (the term of equation 3.15 inside the summation) with respect to each parameter, in accordance with equation 3.19. The result is

$$\begin{aligned} S^{(l)}\left(y_t, u_t, w_t|\phi\right) &= \frac{\partial}{\partial \phi^{(l)}} \cdot \log f'(y_t|\phi) u_t(\phi) \\ &= \frac{\frac{\partial}{\partial \phi^{(l)}}(f'(y_t|\phi)u_t(\phi))}{f'(y_t|\phi)u_t(\phi)} \\ &= \frac{f'(y_t|\phi)w_t^{(l)}}{f'(y_t|\phi)u_t(\phi)} + \frac{\partial f'(y_t|\phi)/\partial \phi^{(l)} u_t(\phi)}{f'(y_t|\phi)u_t(\phi)}. \end{aligned} \tag{3.21}$$

A complete derivation for the term $w_t^{(l)} = \partial u_t(\phi)/\partial \phi^{(l)}$, as well as a proof that equations 3.20 and 3.21 provide for a consistent estimator of the true parameter set ϕ^*, can be found in [30]. The operator Π_G is a projection back onto the allowable parameter manifold, which serves to enforce probability constraints. Unlike the Baum algorithm, individual steps of RMLE algorithm are not guaranteed to increase likelihood. However, it is possible for the RMLE algorithm to continuously receive new training data, without requiring previous exemplar storage.

An outline of the complete clustering algorithm is given in the following pseudocode (Algorithm 1). The lexicon begins empty, with $K = 0$. A segmented, real-valued, discrete-time sequence of observations $Y_n(t)$ is presented. For practical implementations, individual observations y_t will likely be transformations of sensor data into a latent feature space. The algorithm will then try to classify $Y_n(t)$ based on its likelihood (eq. 3.15) given each model. If the winning model produces a likelihood greater than the novelty threshhold Θ, then the RMLE incremental update is used for each observation (3.21). If no model meets the threshold, a new model is created with parameters trained again by the RMLE algorithm. Obviously, the setting of Θ is of great practical importance and has many implications for algorithm performance. A more detailed disucssion of these is saved for Chapter 5.

Algorithm 1 Lexicon Creation Algorithm

```
$K \leftarrow 0$
while $Y_n(t)$ do
  for $k = 1$ to $K$ do
    $\mathcal{P}^* = \max(\mathcal{P}^*, P(Y_n(t)|\phi_k))$
    $\hat{k} = \arg\max \mathcal{P}^*$
  end for
  if $\mathcal{P}^* < \Theta$ then
    $K++$
    $q_{K+1} = \text{train}(Y_n(t), q_{K+1})$
  else
    $q_{\hat{k}} = \text{train}(Y_n(t), q_{\hat{k}})$
  end if
end while
```

3.2 Associative Memory Model

The division of the action-word learning task into two subproblems – the lexicon learning problem and the semantic grounding problem – results in the ability to rewrite equation (3.1) using symbol strings as the observations. The auditory process $A(t)$ now becomes $\bar{A} = [a_1, a_2, \dots a_N]$, the motor process $B(t)$ becomes $\bar{B} = [b_1, b_2, \dots b_N]$, and the "conceptual" process becomes $\bar{C} = [c_1, c_2, \dots c_N]$. Each point of the concept sequence c_n can take on integer values from 1 to a fixed M, the number of concepts the system can learn. Additionally, let it be assumed that the action-word pair $\{a_n, b_n\}$ shares the same conceptual origin, and that this relationship can be determined by use of some heuristic (see Section 5.1). The problem statement is now posed as

$$\begin{aligned}\mathcal{M}_C^* &= \arg\max_{\mathcal{M}_C} P(\bar{A}, \bar{B}, \bar{C}|\mathcal{M}_C) \\ &= \arg\max_{\mathcal{M}_C} P(\bar{A}|\bar{C}, \mathcal{M}_C)P(\bar{B}|\bar{C}, \mathcal{M}_C)P(\bar{C}|\mathcal{M}_C).\end{aligned} \tag{3.22}$$

If it were so desired to implement an optimization algorithm for equation (3.22), knowledge of the concept sequence $\bar{C}$ would be necessary. However concepts are merely a representational tool, and their actual state is inaccesible to us as we think and speak. Therefore $\bar{C}$ is an unobserved process; and if it is further considered to be a Markov process, the model $\mathcal{M}_C$ is an HMM with symbolic observations, and its parameters can be estimated online with the RMLE algorithm (the "cascaded hidden Markov model" method developed by [13]). Keeping this structure in mind, maximization now takes place only on the observed processes, yielding

$$\mathcal{M}_C^* = \arg\max_{\mathcal{M}_C} P(\bar{A}|\mathcal{M}_C)P(\bar{B}|\mathcal{M}_C), \tag{3.23}$$

with $\mathcal{M}_C = (A_c, \mathcal{O}_a, \mathcal{O}_b)$. The parameter A_c is the transition matrix of the hidden process, and $\mathcal{O}_a, \mathcal{O}_b$ are the discrete observation matrices of each modality, which contain the conditional probabilities of an observation symbol for each internal state:

$$[\mathcal{O}_a]_{m,k} = P(a_i = k|c_i = m). \tag{3.24}$$

The indicies k and m are references to the elements of the set of lexical items (clusters) for that modality and internal state space respectively. The values of these matrices can be parsed for evaluation of the ability of the algorithm to capture the conceptual structure of the world in the way which we might expect. For example, one would expect that for a simple action word such as *reach*, the concept which has the highest probability of producing a observation symbol corresponding to the word "reach" would also have a high probability of producing a symbol corresponding to any gestures in the lexicon to which a human would also ascribe the label *reach*. Such analysis is key for understanding the results of the experiments in Chapter 4.

Another important tool for understanding the functionality of the HMM is the *Viterbi algorithm* [31], a version of the dynamic programming algorithm which can be applied to find the value of the hidden state, c_n. The Viterbi algorithm efficiently solves the optimization problem

$$\bar{C}^* = \arg\max_{\bar{C}} P(\bar{A}, \bar{B}, \bar{C}|\mathcal{M}_c) \tag{3.25}$$

by exploiting the structure of the hidden Markov model. Unfortunately, like the Baum-Welch algorithm, the Viterbi algorithm calculates the sequence for data presented in a batch-processing scenario. One possible method for approximating equation (3.25) is by finding the most likely state at each step using the current observation and forward prediction filter:

$$\tilde{c_n} = \arg\max_{m=1,\dots M} f_m(y_t|\phi)u_{t_m}(\phi). \tag{3.26}$$

This method is less optimal than the Viterbi algorithm in terms of performance, but it has been shown in [13] that the gains are modest for the given application. However, with either of these methods, it is possible to view in real-time the system's "guess" of the underlying concept, and compare this with the concept the human tutor intended to convey.

An overall architecture for the concept learning problem, which expands on Figure 3.2 is presented in Figure 3.3. It consists of the original lexical creation blocks, the concept learner discussed above, as well as a synchrony detector which bundles simultaneously occuring multimodal inputs as observation pairs. This architecture was implemented for the online action-word learning task set at the beginning of this document.

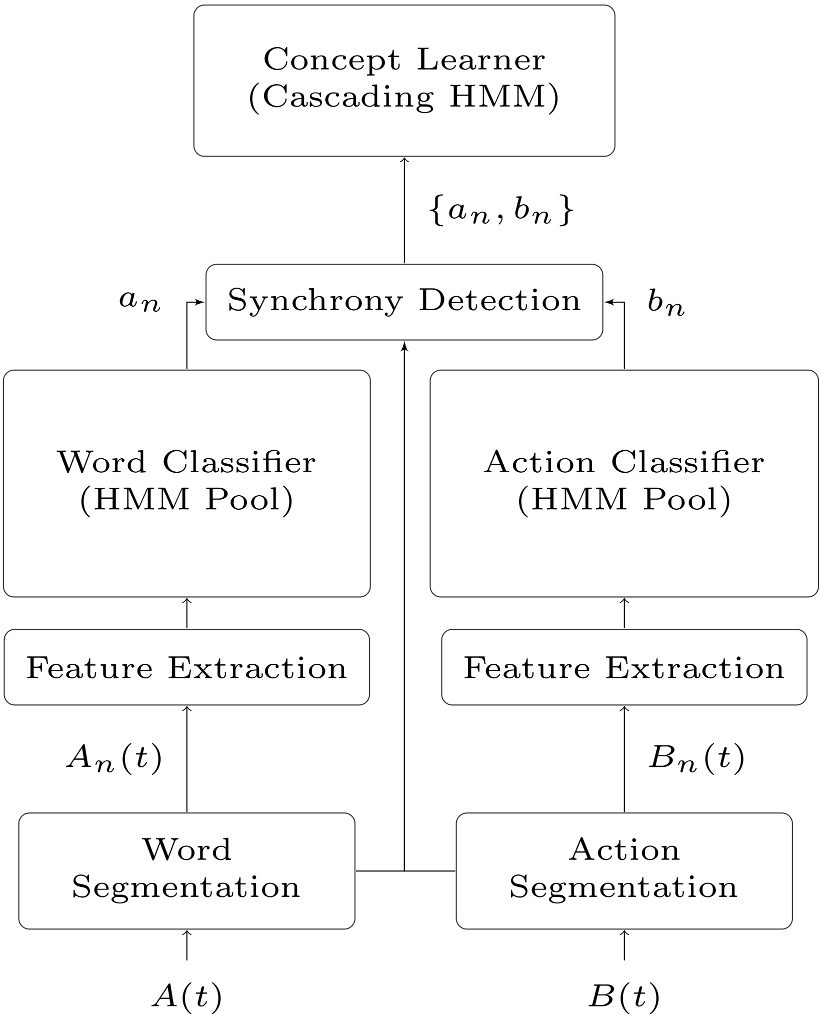


Figure 3.3: Combined architecture for speech and action grounding.

CHAPTER 4

EXPERIMENTAL RESULTS

The models outlined above were first tested using two real data sets: the first from a stream of fluid speech and the second from set of simple gestures. These tests aimed to show the viability of the sensory "lexicon"-building architecture of Figure 3.2. Following this, a third experiment based on the human-robot action-word learning scenario was performed using the multimodal architecture of Figure 3.3.

4.1 Experiment I: Word Recognition

In the first experiment, speech was used as the training data for the sensory lexicon-learning algorithm. Segmentation was performed at the word level through application of a signal energy metric. Feature extraction consisted of a spectral transform, as well as a phonetic classifier. The goal of this experiment was to set up the lexicon learner, presented a solid stream of speech, to acquire a set of words with minimal confusion.

First, a stream of speech was taken from a single male speaker, at 22050Hz and 16Bit/s. The speech consisted of a random assortment of twelve different action-word utterances. This stream was then sent to preprocessing in frames 512 samples long ($\sim$ 25ms). For each frame, the discrete Fourier transform (DFT) was taken and a bandpass window tuned to the range of typical speech was applied. The energy at each frame was used as the voice activity signal. Before thresholding, this activity signal was filtered by a tenth-order FIR filter with cutoff frequency at 1/10th of the Nyquist rate. This filtering operation was performed to smooth out jumps in the voice activity level due to brief pauses from phonemes such as stops. This filtered activity signal was then thresholded to produce the "interesting" sequences which would be presented to the clusterer/classifer.

After segmentation, the signal then underwent other feature extraction. Frames 512 samples wide ($\sim$ 25ms) were used to compute 30 Mel-frequency cepstral coefficients (MFCCs) [32]. The first 15 of these 30 MFCCs were used as the training features. The first stage of the sequence learner was a phone classifier, which consisted of a single HMM with 14 internal states and Gaussian observation distributions. This phonetic classifier was first primed for the sequence-learning task by training with the RMLE algorithm on a minute of normal speech generated by the speaker. After this training period, model parameters were fixed, and it was used to classify the phonetic category (cf. Poritz [20]) of novel speech based on the maximum likelihood classification of equation (3.26). Such an intermediate step does not violate any principles of online learning or self-organization and is indeed regarded as a biologically plausible mechanism for phoneme and word acquisition (Brandl provides a clear review of evidence justifying these claims in [33]).

The input from the classifier to the lexicon learner for each speech segment is a stream of discrete symbols. The HMMs in the lexicon themselves had hidden dimensionality of 7, and discrete observation distributions of the form presented in equation (3.24). For the initial design, the threshold for deciding if a sequence was considered novel or not was fixed to a value slightly above the observed mean of $\mathcal{P}_k$ for correctly classified sequences. For training on novel sequences, the parameter update was performed for 60 iterations on sequence, with a learning rate of $\epsilon = 0.001$. Update sequences each trained an additional 20 times.

The samples were presented to the combined system as continuous stream of speech which contained twelve different action words repeated eight times each, in a randomized order. For an online, unsupervised learning problem, the results are best displayed as the confusion matrix listed in Table 4.1.

Referencing the diagonal entries of the confusion matrix as the ground-truth classes for each word, we calculate an overall word-recognition error rate of 7.29%. While it is difficult to make comparisons of this performance metric to other such metrics for typical speech recognition applications, we still consider this rate acceptable for the given task.

As vocabularies become larger and larger, confusion will become very common for this simple word model. While confusion can always be sacrificed for "lexicon stability" (the eventual settling of lexicon size) by changing of the threshold value Θ, it is ultimately ideal that lexicon create precisely as many

Table 4.1: 12-Word Model Confusion Matrix

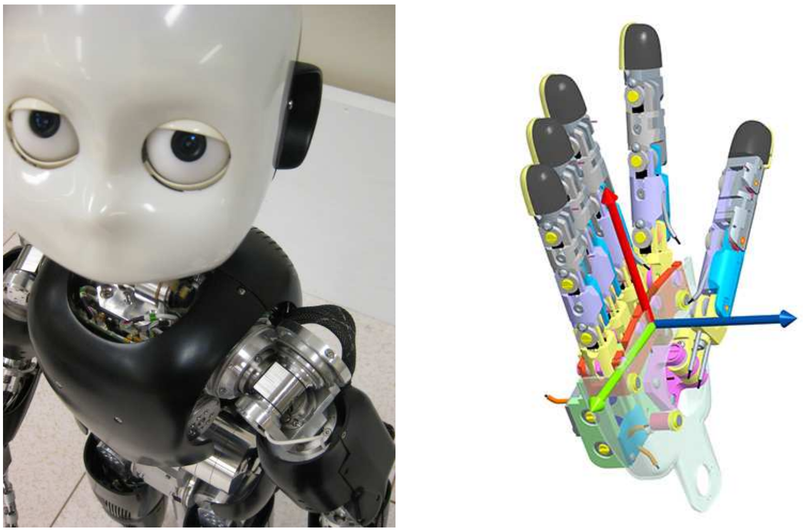
Lex. Ind.	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
reach	8											
slap		8										
grab			8									
wave	1			5						2		
punch	3				5							
hit						8						
turn							8					
salute								8				
greet									8			
attention										8		
sweep										1	7	
drink												8

elements as intended, with no confusion. One way to achieve the ideal balance is to widen the distance between the membership values during correct and incorrect classifications, thereby making the threshold a more effective discriminator. This widening could be done through expansion of the set of acoustic features used, the size of the phonetic classifier, and model orders for the HMMs to improve results. A second method would be to provide more top-down control measures in addition to the threshold, as will be discussed in Chapter 5.

4.2 Experiment II: Action Recognition

The goal of the second experiment was to explore the model's abilities with respect to action recognition. This test resembles very closely the previous one done on speech. A solid stream of joint data was first preprocessed, then segmented and finally the sequences were presented to the lexicon-learning algorithm.

The live stream of data in this case consisted of the set of 7 joint angles $\{\theta_i\}_{i=1}^{7}$, taken from the right arm of the iCub robotic platform [34]. The iCub, pictured in Figure 4.1a, is a fully anthopomorphic robot which serves as the primary experimental platform for testing the work presented here, as well as other language learning models. In total the robot has 53 degrees of



(a) iCub humanoid robot

(b) Location of end-effector reference frame

Figure 4.1: Photograph of the iCub humanoid robot [34], with close-up schematic drawing of right hand [35].

freedom (DOF), including 16 for each arm. For this, only the first 7 DOFs were exercised, corresponding to shoulder pitch, roll, and yaw; elbow angle; and pronosupination, roll, and yaw of the wrist. The actions were generated by first powering down the robot's motors, then using an instructor's hand to guide movement. Four actions were performed during the test: raising the hand in a "greeting" pose, a "sweep" motion back and forth, a "drinking" motion (as if raising a cup to its head), and extending the arm straight down at the side (standing at "attention"). Each of these actions were performed many times, returning to a standard rest position after each action.

The preprocessing step consisted of first using the forward kinematics of the robot [35] to transform the seven-dimensional joint space into a three-dimensional Cartesian or "task" space, based on the position of the end effector, whose reference frame is located on the palm of the right hand (Figure 4.1b). Next, the derivitive was calculated for each direction. The magnitude of this derivitive was then used for motion activity detection, by low-pass filtering the signal and thresholding, as was done in the voice activity detection algorithm. These activity-gated sequences of end effector positions were used as the observations for the classifier. Each HMM in the classifier had a hidden dimensionality of 5. Novel sequences were trained for ten iterations, and updating sequences were trained for five, both with an

Table 4.2: Action Confusion Matrix

Lex. ind.	**0**	**1**	**2**	**3**	**4**
greet	6	0	0	0	0
sweep	0	6	0	0	0
drink	0	0	6	0	0
attention	0	0	0	3	3

epsilon of 0.001. As in Experiment II, the action data was presented as a continuous stream and the novelty threshold was set to a fixed value.

For the motor data, some changes need to be made to ensure that the algorithm works properly. A fundamental part of the algorithm is that HMMs which code cluster centers need to be able to recognize incoming data. However, because samples are presented to the classifier sequentially, it often happens that only a single trajectory has been presented as training data for a given cluster. The result is that the Gaussian observation distributions which tend to fit different parts of the path often have extremely low variances in directions orthogonal to the principal axis. This low variance makes the recognition problem very difficult, as even small amounts of noise or deviation from this prototype trajectory will cause very low likelihoods.

In order to generalize these trajectories, the allowable values of the parameter manifold were constrained. These constraints were enforced by taking the eigendecomposition of the covariance matrix for each output distribution and searching for all eigenvalues less than χ. In this case, χ is the desired minimum variance along any principal component. After each update step, a constrained covariance matrix $\tilde{U}_{t+1}$ is calculated by

$$\tilde{U}_{t+1} = U_{t+1} + VRV^T, \tag{4.1}$$

$$R = diag\left[\lambda_1 - \max(\chi, \lambda_1), \ldots \lambda_r - \max(\chi, \lambda_r)\right]. \tag{4.2}$$

The terms V and λ above refer to the matrix of eigenvectors of U and the individual eigenvalues of U respectively. It is quite possible that such a step may affect some of the convergence properties of the RMLE algorithm, however no such issues were noted during the course of the experiment.

Table 4.2 presents the confusion matrix for the action recognition experiment. For this training set, the algorithm created a lexicon of five elements, with two of these classes corresponding to variations on the action "attention." The error rate for this experiment was calculated as 12.5%, but this value holds less meaning for these results, as there was no actual "confusion" between the two classes of the action labeled "attention." The semantic learning experiment will further demonstrate how this "extra" class is automatically adapted into the learning algorithm.

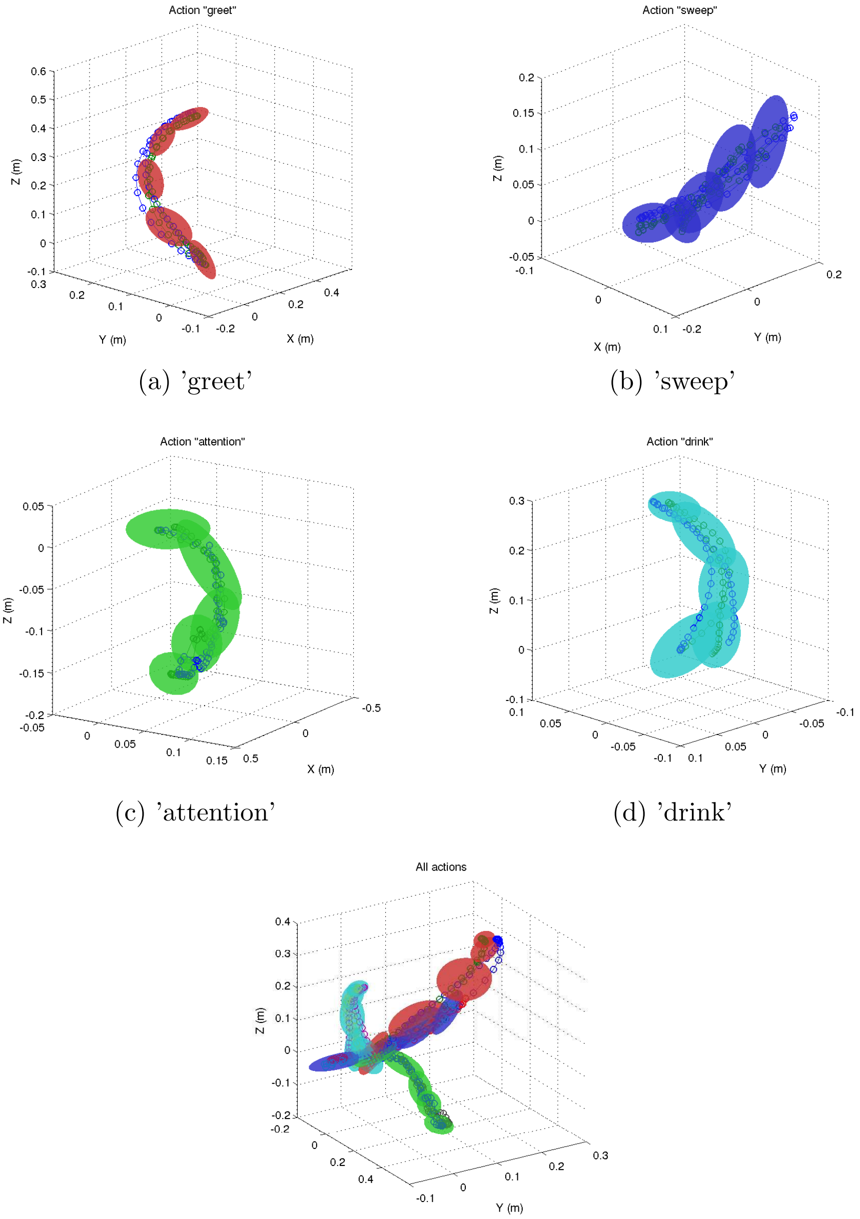
As opposed to the word learning experiment, where it was difficult to visualize the internal workings of the HMM on high-dimensional data, the action learning experiment allows for visual evaluation of the learning algorithms. Figure 4.2 shows sample trajectories for the actions used, as well as a visualization of the HMM output distributions used to model them.

One problem with the current preprocessing chain that these figures highlight is the lack of rotation, scale, and initial/final position invariance. While this representational shortfall is not within the scope of the lexicon creation algorithm, any future system which seeks to build a more complete action lexicon will need to address this issue. Nevertheless, these results demonstrate the ability of the algorithm to cluster action data in a way which effectively encodes the important spatial and temporal features inherent to motion, and allows to symbolic access to these representations.

4.3 Experiment III: Action Semantics

The final experiment was a test of the robot's ability to learn four different action words, in the setting of the human-robot interaction scenario that was put forth at the beginnning of this thesis. The integrated multimodal architecture of Section 3.2 was implemented using the same equipment and techniques of the previous two experiments, now with the addition of the cascading hidden Markov model (CHMM) layer as an associative memory.

The system was initialized with no prior knowledge, and empty lexicons, just as in experiments I and II. The human tutor sequentially produced the four actions from Experiment II: greeting, sweeping, drinking, moving the arm to the side. Corresponding narration was also given during the action performance, with words "greet," "sweep," "drink," and "attention." This proccess was repeated several times in order to generate a data set large



(a) 'greet'

(b) 'sweep'

(c) 'attention'

(d) 'drink'

(e) all actions

Figure 4.2: Plots showing sample trajectories in Cartesian space, as well as a visualization of their HMM encodings. The colored ellipsoids represent the Gaussian output distributions of the HMM.

enough for meaningful convergence. The individual modality models were run with the same segmentation methods and feature extraction algorithms used in experiments I and II.

Symbolic output from each modality was produced by the lexicon creation/clustering algorithm. Cross-modal symbol pairs were created by checking the input stream for synchronous activity (i.e., narration during the action). Specifically, if there was any overlap in the activity detection signals generated for each modality, the corresponding a_n and b_n produced by these periods of activity are passed onto the CHMM as an observation pair.

These pairs were presented to the concept learner, which had a fixed internal state space size of four. The observation parameters were the observation matrices $\mathcal{O}_a$ and $\mathcal{O}_b$, representing the word and action symbols respectively. Since the number of possible observation symbols, and thus the size of the matrices, is not known in advance, each matrix was set to handle up to ten symbols. This number is more than twice the clusters expected to be created by each lexicon, so it is unlikely that any observed symbol will exceed allowable bounds. These "extra" symbols – those which the CHMM can handle but are never created in lexicon – will not be observed by the CHMM, and therefore their observation probabilities will eventually decay to zero. The corresponding empty rows of $\mathcal{O}_a$ and $\mathcal{O}_b$ are not displayed in the results.

Because generating action data on the robot by the means used for Experiment II requires prolonged direct manipulation of the robot, it is ideal for the CHMM to converge to its results with as little training data as possible. To this end, the learning rate ϵ was set higher than usual: $\epsilon = 0.075$ as opposed to $\epsilon = 0.001$ for the modal lexicons. Furthermore, because the order of presentation of the action words is not a consideration for this experiment, the decision was made to remove the Markovian relationship of the hidden process. For the limited data set of concepts presented in random order, enforcing this condition can slow convergence.

The motor modality settled on a model order of 5 (as in Experiment II), while the speech modality used 4. Because of the limited number of words, confusion did not occur for speech, but a pattern of confusion similar to Table 4.2 arose for actions. The symbolic data generated by each lexicon is graphed in Figure 4.4a. These symbols were then coupled into pairs and used as observations for the CHMM.

Figure 4.3 is a plot of the $\mathcal{O}$ matrix values for each input (lighter values

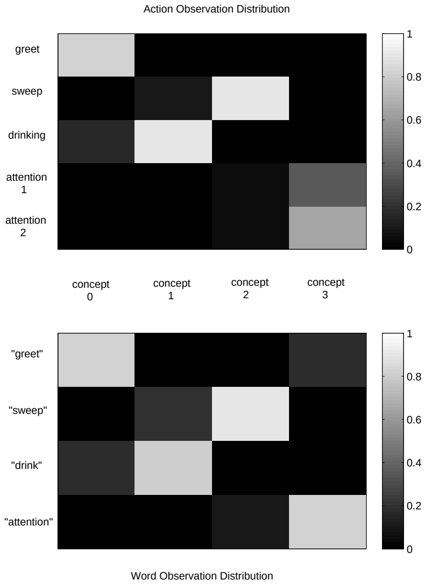
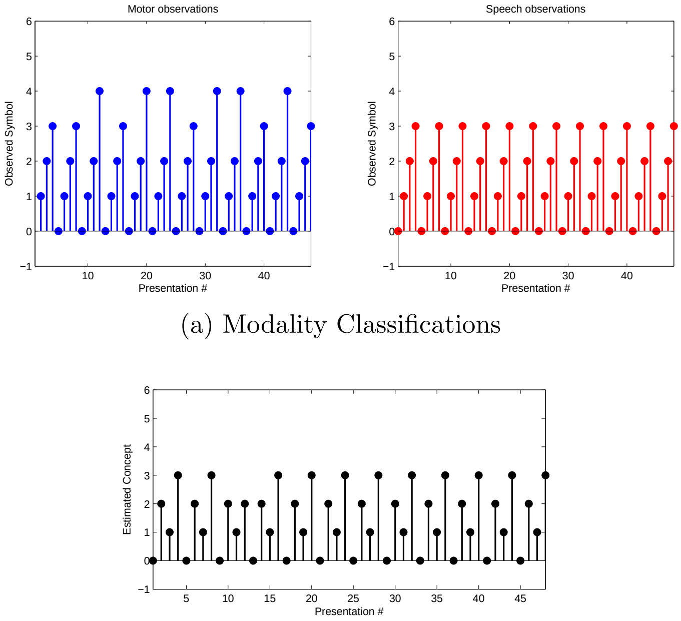


Figure 4.3: Observation probability matrices for the learned concept model. Columns correspond to the internal concept, while rows index the observation symbols.



(a) Modality Classifications

(b) Estimated Concept

Figure 4.4: Classified training data used for concept learning and the corresponding estimated concept state at each observation.

correspond to larger entries). Looking at the values for each state, the linking through concepts of action and speech is readily apparent. For “concepts” 0, 1, and 3, the probability mass falls almost entirely on a single observation for each modality (e.g., “greet”/greet have probability mass near 1 for concept 0). Concept 3 presents some confusion based on the classification of the action “attention” into two separate clusters. However, this is not an issue as co-occurance with the corresponding speech utterance results in these being integrated into a single concept.

Additionally, the value of the internal state of the semantic model, calculated with equation (3.26), can be used to evaluate the model’s “comprehension” of its inputs as time progresses. Figure 4.4b shows the estimated internal state at each observation, with the symbolic classification of each of the modal lexicons above. Ideally, given four words and four matching concepts, a given internal state would correspond exclusively to a single word and only be estimated as the present state when that word was presented. While there is minor confusion in this regard at the beginning of the experiment, after only three to four presentations of each action/word pair the model has no further confusions. An even more impressive observation can be made by comparing this result with the time histories of the model’s parameter estimates over the training session. It can be seen in Figure 4.5 that although the parameters have not yet converged to stable final values by the end of the experiment, concept classification errors are rare.

A third and final observation from this experiment is that the associative memory is able to smooth over some of the issues caused by the novelty threshold. Specifically, problems stemming from an incorrect novelty decision (e.g., many lexical elements are created for what is deemed a single action/word) are mitigated by the HMM binding both actions to a single word, or vise versa. While this is by no means an indication that the novelty question is insignificant, it does allow some latitude in terms of the setting of the threshold.

Overall, the results are nearly ideal; and though not novel with respect to the functioning of the HMM associative memory, they do demonstrate the ability of the lexicon-learning algorithm to interface with it and to expand the range of possible inputs with which to create semantic understanding.

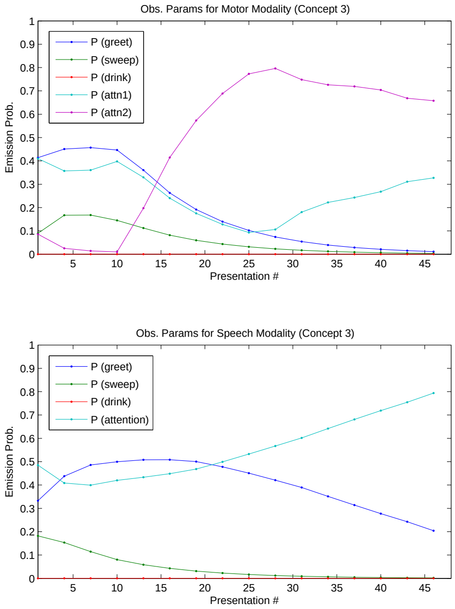


Figure 4.5: Sample time history of the observation distribution parameter estimates for a given concept.

CHAPTER 5

DISCUSSION

5.1 The Hidden Markov Model as a Lexical Model

From the results of the experiments performed in the preceding chapter, the general approach of combining the HMM, a statistical model, with competitive learning techniques, was confirmed in its viability as a means of capturing diverse sensory inputs in a symbolic representation. However, this architecture also showed that while it expanded the range of sensory information that previous methods were able to represent, it also required examination of some of the ad-hoc methods needed in practical implementation.

Two major issues with our lexicon-creation model were identified during the speech and action learning experiments. The first of these was the problem of model order selection: the decision of whether to consider a poorly explained input as “novel” or as a variation on another concept. For the previous experiments, this was handled by a fixed threshold, set at the discretion of the experimenter. The second issue was that of salience, especially present in the motor experiment: what constiutes an “action”, and more importantly what makes an action semantically interesting?

With regards to the model order selection problem (it should be noted that reference to “model selection” refers to the size of the set of HMM classifiers and not r, the model order of the HMMs themselves), it is unlikely that common methods would perform well in this situation, given that the learning of lexical elements in humans is guided by complex processes, incorporating both intrinsic and extrinsic inputs. Child-directed speech, in which a parent is speaking to an infant or toddler and which is generally considered the model setting for these experiments, displays highly idiosyncratic characteristics such as limited vocabulary, heavy redundancy, and stereotypical use of words [36]. Therefore it might be more reasonable to use a heuristic approach which takes advantage of this behavior, such as using redundancy to imply novelty of the input.

The second issue in the current model is that of deciding which inputs are salient or interesting, and how to segment them. In these experiments, segmentation was based entirely upon signal energy and was not related to the feature-level representation. Again, behavior-based heuristics are an interesting and computationally tractable route to a solution. One such method that has been discovered and already explored computationally is that of Acoustic Packaging [37],[38]. Here it was noted that infants used cross-modal salience cues in the form of narration to attend to and package sequences of events. This behavior was found to be useful for application in visual action segmentation.

5.2 Future Work

As stated in the introduction, this work is only a small part of a larger program which aims to understand the underlying processes of language acquisition in humans by exploring computational models of both behavior and biology. The driving philosophy behind this work is that language is an emergent phenomenon: a result of a more general computational process, implemented in the brain, which is able to determine some structure of the world as presented to it through the sensorimotor system. The lexicon-learning algorithm proposed here, as well as the cascading HMM, is put forth as a model of some of the functional building blocks of such a cognitive process.

In the immediate future, we would like to explore new ways in which the pairing of the associative memory and the lexicon-learning algorithm could be used to give the robot expanded capabilities or to refine and improve current capabilities. The first such path is to test the viability of the lexicon learner and the CHMM as not only the top levels of the semantic system but also as features which can be used to abstract even higher-level representations. It is easy to imagine how the CHMM's internal process could serve as input to another lexicon to learn conceptual "events," or an action primitive lexicon might feed into a higher-level lexicon, giving the robot a way to represent compositional motor skills. Another avenue for further research is to employ the semantic information gathered by the CHMM to improve the lexical (specifically word)-recognition error rate. Most current ASR systems utilize only phonetic and word-order information for speech classification, and the system presented here could possibly be used to demonstrate the importance of top-down semantic feedback on our perceptual capabilities.

Such motor abilities are fundamental to the robot's ability to autonomously gather information about its environment. Use of HMMs means that the each lexical element can be run in reverse to generate speech or motor data. Gaussian mixture regression techniques [39] are already being applied to generate sample trajectories of actions that are represented as HMMs. The lexicon-learning algorithm is therefore an important stepping stone for not only future language specific experiments but also future work on improved frameworks for environmental and social interaction.

5.3 Conclusion and Final Comments

Although the data sets tested thus far have been both quite simple or small, the algorithms presented here do hold promise for achieving their goal of providing some of the computational tools by which computers can not only acquire the elements of language itself but also the information about the world in which language grounded. However, just as important is the way in which the algorithm learns. By following a set of basic principles for cognitive development, we have tried to create a minimally structured framework through which linguistic ability is not enforced but rather emerges. We think not only that such an approach is necessary in order to make meaningful progress on this problem but also that it provides great hope that perhaps we will be able to employ simple models to find a path around the great morass of behavioral complexity which we observe in language acquisition.

REFERENCES

[1] A. Turing, "Computing machinery and intelligence," *Mind*, vol. 59, pp. 433–460, 1950.

[2] N. Wiener, *Cybernetics or Control and Communication in the Animal and the Machine.* Cambridge, MA: The M.I.T. Press, 1948.

[3] D. Roy, "Grounded spoken language acquisition: Experiments in word learning," *IEEE Transactions on Multimedia*, vol. 5, no. 2, pp. 197–209, June 2003.

[4] K. Squire and S. Levinson, "HMM-based semantic learning for a mobile robot," *IEEE Trans. on Evolutionary Computation*, vol. 11, pp. 199–212, 2007.

[5] S. Harnad, "The symbol grounding problem," *Physica D*, vol. 42, no. 1-3, pp. 335 – 346, 1990.

[6] F. Pulvermueller, "Brain mechanisms linking language and action," *Nature Reviews Neuroscience*, vol. 6, p. 576 582, 2005.

[7] V. Gallese, L. Fadiga, L. Fogassi, and G. Rizzolatti, "Action recognition in the premotor cortex," *Brain*, vol. 119, no. 2, pp. 593–609, 1996.

[8] G. Rizzolatti and M. Arbib, "Language within our grasp," *Trends in Neuroscience*, vol. 21, pp. 188–194, 1998.

[9] B. Z. Mahon and A. Caramazza, "A critical look at the embodied cognition hypothesis and a new proposal for grounding conceptual content," *Journal of Physiology-Paris*, vol. 102, no. 1-3, pp. 59 – 70, 2008.

[10] L. E. Bahrick, R. Lickliter, and R. Flom, "Intersensory redundancy guides the development of selective attention perception and cognition in infancy," *Current Directions in Psychological Science*, vol. 13, pp. 99–102, 2004.

[11] G. Gergely and J. Watson, "Early social-emotional development: contingency perception and the social biofeedback model," in *Early social cognition: understanding others in the first months of life*, P. Rochat, Ed. Mahwah, NJ: Erlbaum, 1999, ch. 5, pp. 101–136.

[12] M. Tomasello and M. J. Farrar, "Joint attention and early language," *Child Development*, vol. 57, no. 6, pp. 1454–1463, 1986.

[13] K. Squire, "HMM-based semantic learning for a mobile robot," Ph.D. dissertation, University of Illinois at Urbana-Champaign, 2004.

[14] A. Cangelosi and T. Riga, "An embodied model for sensorimotor grounding and grounding transfer: Experiments with epigenetic robots," *Cognitive Science*, vol. 30, no. 4, pp. 673–689, 2006.

[15] D. Marocco, A. Cangelosi, K. Fischer, and T. Belpaeme, "Grounding action words in the sensorimotor interaction with the world: experiments with a simulated iCub humanoid robot," *Frontiers in Neurorobotics*, vol. 4, no. 7, pp. 1–15, May 2010.

[16] Y. Sugita and J. Tani, "Learning semantic combinatoriality from the interaction between linguistic and behavioral processes," *Adaptive Behavior*, vol. 13, no. 1, pp. 33–52, 2005.

[17] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," in *Proceedings of the IEEE*, vol. 77, 1989, pp. 257–285.

[18] S. Levinson, *Mathematical Models for Speech Technology.* New York, NY: John Wiley and Sons Ltd., 2005.

[19] R. Cave and L. Neuwirth, "Hidden Markov models for English," in *Proc. Symp. on the Application of Hidden Markov Models to Text and Speech*, 1980, pp. 16–56.

[20] A. Poritz, "Linear predictive hidden Markov models and the speech signal," in *Proc. of ICASSP 82. IEEE International Conference on Acoustics, Speech and Signal Processing.*, 1982, pp. 1291–1294.

[21] V. B. Mountcastle, "The columnar organization of the neocortex," *Brain*, vol. 120, no. 4, pp. 701–722, 1997.

[22] W. A. Phillips and W. Singer, "In search of common foundations for cortical computation," *Behavioral and Brain Sciences*, vol. 20, no. 04, pp. 657–683, 1997.

[23] M. Sur, P. Garraghty, and A. Roe, "Experimentally induced visual projections into auditory thalamus and cortex," *Science*, vol. 242, no. 4884, pp. 1437–1441, 1988.

[24] D. Kulic, D. Lee, C. Ott, and Y. Nakamura, "Incremental learning of full body motion primitives for humanoid robots," *8th IEEE-RAS International Conference on Humanoid Robots*, pp. 326–332, December 2008.

[25] S. Calinon and A. Billard, "Incremental learning of gestures by imitation in a humanoid robot," in *Proc. of the ACM/IEEE Intl. Conf. on Human-Robot Interaction*, 2007, pp. 255–262.

[26] M. Law and J. Kwok, "Rival penalized competitive learning for model-based sequence clustering," in *Proceedings. 15th International Conference on Pattern Recognition*, vol. 2, 2000, pp. 195 – 198.

[27] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique in the statistical analysis of probablistic functions of Markov chains," *Ann. Math. Statistics*, vol. 41, pp. 164–171, 1970.

[28] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Royal Statistical Soc. Series B*, vol. 1, no. 39, pp. 1–38, 1977.

[29] F. LeGland and L. Mevel, "Recursive estimation of hidden Markov models," in *Proc. of 36th IEEE Conf. Decision Control*, vol. 36, 1997, pp. 3468 – 3473.

[30] V. Krishnamurthy and G. Yin, "Recursive algorithms for estimation of hidden Markov models and autoregressive models with Markov regime," *IEEE Trans. on Information Theory*, vol. 48, pp. 458–476, 2002.

[31] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimal algorithm," *IEEE Trans. on Information Theory*, vol. 13, no. 2, pp. 260 – 269, 1967.

[32] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 28, no. 4, pp. 357 – 366, 1980.

[33] H. Brandl, "A computational model for unsupervised childlike speech acquisition," Ph.D. dissertation, Universitaet Bielefeld, Bielefeld, July 2009.

[34] G. Metta, G. Sandini, D. Vernon, L. Natale, and F. Nori, "The iCub humanoid robot: an open platform for research in embodied cognition," in *PerMIS: Performance Metrics for Intelligent Systems Workshop.*, Washington DC, USA, August 2008, pp. 50–56.

[35] "ICub forward kinematics - wiki for robotcub and friends," 2011. [Online]. Available: http://eris.liralab.it/wiki/ICubForwardKinematics

[36] T. Cameron-Faulkner, E. Lieven, and M. Tomasello, "A construction based analysis of child directed speech," *Cognitive Science*, vol. 27, pp. 843–873, 2003.

[37] K. Hirsh-Pasek and R. M. Golinkoff, *The Origins of Grammar: Evidence from Early Language Comprehension.* Cambridge, MA: The MIT Press, 1996.

[38] L. Schillingmann, B. Wrede, and K. Rohlfing, "Towards a computational model of acoustic packaging," in *Proceedings of International Conference on Development and Learning*, 2009, pp. 1–6.

[39] S. Calinon, F. D'halluin, E. L. Sauser, D. G. Caldwell, and A. G. Billard, "Learning and reproduction of gestures by imitation: An approach based on hidden Markov model and Gaussian mixture regression," *IEEE Robotics and Automation Magazine*, vol. 17, no. 2, pp. 44–54, 2010.