Learning Words from Sights and Sounds: A Computational Model

by

Deb Kumar Roy

B.A.Sc., University of Waterloo (1992)
S.M., Massachusetts Institute of Technology (1995)

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 1999

Author .....
Program in Media Arts and Sciences
August 6, 1999

Certified by .....
Alex P. Pentland
Academic Head, MIT Media Arts and Sciences Program
Toshiba Professor of Media Arts and Sciences
Massachusetts Institute of Technology
Thesis Supervisor

Accepted by .....
Stephen A. Benton
Chairman, Departmental Committee on Graduate Students
Program in Media Arts and Sciences

Learning Words from Sights and Sounds: A Computational Model

by

Deb Kumar Roy

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
on August 6, 1999, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

This thesis addresses three interrelated questions of early lexical acquisition. First, how do infants discover linguistic units which correspond to the words of their language? Second, how do they learn perceptually-grounded semantic categories? And tying these questions together: How do infants learn to associate linguistic units with appropriate semantic categories?

To address these questions, we have developed and implemented a computational model of Cross-channel Early Lexical Learning (CELL). Learning is driven by a search for structure across channels of sensory input in an information theoretic framework. CELL acquires lexical items which model word-meaning associations with high mutual information.

CELL has been implemented using computer speech and vision processing techniques. A lexicon is acquired from microphone and camera input. This is the first implementation of automatic language acquisition which discovers words and their semantics from only raw sensory input without human-assisted preparation of data.

CELL has been evaluated on natural speech recordings of six caregiver-infant interactions centered around play with common objects. The speech recordings were coupled with visual images of the objects taken from multiple perspectives. CELL successfully acquired a lexicon of shape names from each of the six participants. When compared to an acoustic-only baseline model, cross-channel structure proved to increase performance dramatically.

This work has applications in human-computer interaction. It provides a new approach to creating spoken language interfaces which adapt to the vocabulary and semantics of individual users. Early prototypes show promise for building natural, robust, and personalized interfaces.

Thesis Supervisor: Alex P. Pentland
Title: Academic Head, MIT Media Arts and Sciences Program
Toshiba Professor of Media Arts and Sciences
Massachusetts Institute of Technology

Doctoral Committee

Thesis Advisor .....

Alex P. Pentland
Academic Head, MIT Media Arts and Sciences Program
Toshiba Professor of Media Arts and Sciences
Massachusetts Institute of Technology

Thesis Reader .....

Allen Gorin
Distinguished Member of Technical Staff
AT&T Laboratories - Research

Thesis Reader .....

Steven Pinker
Professor of Psychology, Department of Brain and Cognitive Sciences
Director, McDonnell-Pew Center for Cognitive Neuroscience
Massachusetts Institute of Technology

Acknowledgments

I would like to thank my thesis advisor, Sandy Pentland, not only for his role in shaping the intellectual content of this work, but also for his support and friendship over the years. Sandy introduced me to the world of computer vision, and constantly urged me to consider the importance of perception in the process of language acquisition. Al Gorin also deeply influenced my work. His research in automatic language acquisition first drew me to the topic, and his constant encouragement and constructive criticism has been invaluable. His fascination with building language learning machines is infectious, and I am happy to say I've caught the bug! Steve Pinker helped me develop my ideas about cognitive modeling, and methods for evaluating my work as a model of infant learning. Steve's super-human knowledge and understanding of a vast range of literature has enriched my work in many ways.

Rupal Patel first suggested that I evaluate my work with infant-directed data, and then single handedly arranged the entire data collection process. She also read multiple revisions of this document and made countless improvements. Most of all, Rupal has provided encouragement and emotional support with uncommon patience every step of the way.

Many colleagues have contributed to the work reported in this thesis. Bernt Schiele built a part of the vision system described in this thesis, and gave me many new insights into my own work. Bruce Blumberg's work on synthetic characters inspired me to create Toco the Toucan, and his ideas on cognitive modeling have shaped my ideas of how language learning fits into more general models of cognition. Michal

Hlavac's brilliant artistry lead to Toco's graphical embodiment. Brian Clarkson wrote parts of an early version of the speech processing system. Tony Jebara built the vision system for Toco the Toucan which was shown in Siggraph '97, and also wrote the visual connected regions analysis software used in later versions of my thesis. I also gained many mathematical insights from conversations with Tony.

Many others have provided friendship and stimulating conversations on a wide range of topics: Nitin Sawhney, Marina Umaschi Bers, Claudia Urrea, Sumit Basu, David Cavallo, Yuri Ivanov, Bill Tomlinson, Seymour Papert, Jacqueline Karaaslanian, Rich Schwartz, John Makhoul, Ted Gibson, Jay Keyser, Bill Butera, Janet Cahn, Stan Sclaroff, Eric Scheirer, Giuseppe Riccardi, Jerry Wright, Bishnu Atal, Stefan Marti, Ken Russell, Push Singh, Thad Starner, Trevor Darrell, Irfan Essa, Whitman Richards, Kris Thorisson, Andy Wilson, Jim Davis, Chris Wren, Stephen Intille, Pattie Maes, Karen Navarro, Venkatesh Hariharan, Shahidul Alam, and Jeff Herman.

I would also like to thank Asim Roy, Sumit Basu, Nitin Sawhney, Bill Butera, and Yuri Ivanov for reading parts of this document and providing many helpful suggestions and corrections.

Finally, I would like to thank my parents and sister for their constant support, enthusiasm, and encouragement in my work. My parents taught me that nothing is impossible, and to never stop asking questions.

This research was supported by the AT&T MIT Media Lab Fellows Program.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Infants are born into a 'buzzing, booming confusion' of sensations. From this sensory chaos, they construct mental representations to model structure that they find in the world. These representations enable infants to understand and predict their surroundings and ultimately to achieve their goals.

Around their first birthday, infants first begin to use words[1] which refer to salient aspects of their environment including objects, actions, and people. They learn these words by attending to the sights, sounds, and other sensations in their environment. The acquisition process is complex. Infants must successfully segment linguistic input into units which correspond to the words of their language. They must also identify semantic categories which correspond to the meanings of these words. Remarkably, infants are capable of all these processes despite continuous variations of natural phenomena and the noisy input provided by their perceptual systems.

This thesis presents a computational model of early lexical learning. Learning is driven by a search for structure across channels of sensory input. The model is accordingly named Cross-channel Early Lexical Learning (CELL). It is a cognitively plausible on-line model which processes data incrementally.

[1]The term "word" is used throughout this thesis in accordance with Webster's Dictionary: "A speech sound or combination of sounds having meaning and used as a basic unit of language and human communication."

CELL has been implemented as a real-time system driven by microphone and camera input. This is a significant result: CELL is the first implemented model of language acquisition which learns words and their semantics from raw sensory input without any human-assisted preparation of data. This implementation has been evaluated with infant directed data and has successfully demonstrated early lexical acquisition.

This thesis also explores applications of automatic language learning for human-computer interaction. The CELL architecture provides a new approach for developing adaptive spoken interfaces. We have implemented several prototypes which show promise for building natural, robust, and personalized interfaces.

1.1 Problems of Early Lexical Acquisition

Before specifying the problems of lexical learning addressed by CELL, we provide operational definitions of several terms which are used throughout this thesis. For expository purposes, these definitions are presented in simplified form. They are developed more precisely in Chapter 3.

Lexical items are the output of CELL. Each lexical item specifies a *linguistic unit* and a *semantic category*. Linguistic units model the surface form of a word, i.e. the sound of a word when spoken or its visual form when gestured. A linguistic unit consists of a prototype which defines the ideal form of the word and a radius parameter which specifies the allowable variation relative to this prototype. The semantics of linguistic units are grounded in sensory input. A semantic category specifies a range of sensory inputs which can be grouped and associated with a linguistic unit. Semantic categories are defined by a prototype which specifies the ideal or central form of the category and a radius of allowable variation. For example, a semantic category might specify a portion of the color spectrum, where the prototype corresponds to a particular point of the spectrum, and the radius parameter would specify the allowable

deviation from this prototype. Such a semantic category could be used to ground the semantics for a color term such as "red". A lexical item encodes the association between a linguistic unit and its corresponding semantic category.

This thesis addresses three interrelated questions of early lexical acquisition. First, how do infants discover linguistic units which correspond to the words of their language? Second, how do they learn perceptually grounded semantic categories? And tying these questions together: How do infants learn to associate linguistic units with appropriate semantic categories?

Discovering linguistic units of a language is difficult since most infant-directed utterances contain multiple connected words. There are no equivalents of the spaces between printed words when we speak or gesture naturally; there are no pauses or other cues which separate the continuous flow of words. Imagine hearing a foreign language for the first time. Without knowing any of the words of the language, imagine trying to determine the location of word boundaries in an utterance, or for that matter, even the number of words! Infants first attempting to segment linguistic input face a similarly difficult challenge. This problem is often referred to as the speech segmentation or word discovery problem.

In addition to successfully segmenting linguistic units, infants must learn categories which correspond to the semantics of words. In this thesis we consider only semantic categories which can be grounded in sensory input. For example, categories of shape, color, texture, and motion may serve as semantic categories.

The third problem of interest is how infants learn to associate linguistic units with appropriate semantic categories. Input consists of linguistic utterances paired with nonlinguistic contexts. Each utterance contains instances of one or more linguistic units. Each context suggests multiple possible semantic categories[2]. Given a pool of utterance-context pairs, infants must infer word-to-semantic mappings (lexical items)

[2]This is true in even the simplest of situations. A caregiver might present an apple and say, "Look, it's red!". The utterance contains multiple words, and the context includes instances of several possible semantic categories including object shape, color, size, position, etc.

which best fit the data.

1.2 Scope

All input in CELL is grounded in sensors. Thus, both linguistic units and semantic categories must be defined directly in terms of sensory input. This delimits the scope of CELL. The semantics of many words such as *because* and *love* cannot be grounded directly in the physical world. However, a large portion of words which are learned at an early stage by infants can be grounded in the physical senses (See Section 3.2.1).

Infants exhibit a lag between receptive and productive abilities: they understand words before they start using them [9]. CELL models only the early stages of lexical learning which occur prior to the production of first words. CELL is based on the assumption that in this early non-feedback stage, the learner attends to the world and constructs models which reflect regularities across multiple channels of input. Lexical acquisition is viewed as a process in which these regularities are found and represented internally by the language learner.

This thesis is not concerned with the acquisition of syntax. This work is, however, related to questions of syntax since syntax cannot be acquired until at least some words are known to the language learner [89]. With an initial lexicon in place, the learner may observe structural regularities in sequences of known words leading to syntax acquisition.

1.3 Overview of the CELL Model

This section describes the main concepts underlying CELL. CELL extracts representations of utterances and their co-occurring context from a set of sensors (Figure 1-1). An utterance contains instances of one or more linguistic units. The number of linguistic units and the location of inter-unit boundaries is unknown. Context consists of other sensory input which co-occurs with the linguistic input. The context

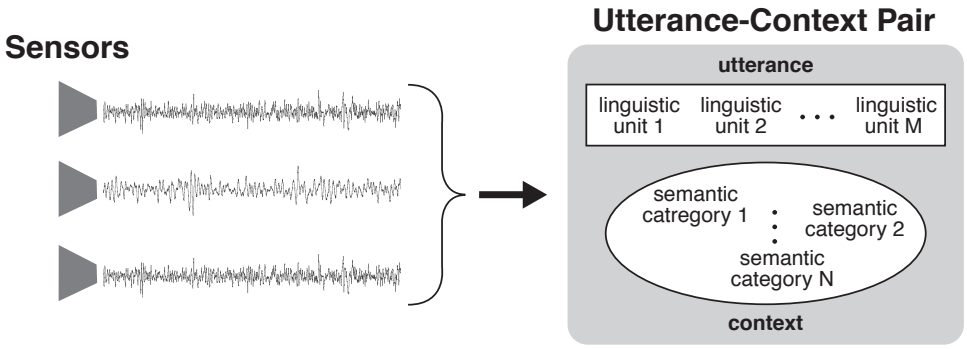


Figure 1-1: CELL extracts representations of utterances and co-occurring context from sensors.

contains instances of multiple semantic categories. A key assumption is that any linguistic unit in an utterance may refer to any semantic category inferred from the co-occurring context.

As an example, consider a learner with two sensors: visual and auditory. Utterances might consist of phonetic representations of spoken sequences recorded by the auditory sensor. The context might consist of representations of visually observable objects and their motions. Potential semantic categories would include categories of shape, color, size, and path of motion. Semantic categories could also consist of conjunctions of attributes, e.g., red objects which are round, or small objects which move along parabolic trajectories.

As utterance-context pairs are encountered, they are "deconstructed" (Figure 1-2). Utterances are "unpacked" into a set of hypothesized linguistic unit prototypes. Contexts are unpacked into a set of hypothesized semantic category prototypes. Any linguistic unit prototype may potentially be paired with any semantic category prototype which is derived from the same utterance-context pair. As shown in Figure 1-2, the deconstruction results in a fully interconnected network of hypothesized prototype pairs.

In a second stage (Figure 1-3), the prototype pairs are filtered and clustered to generate a set of lexical items. Filtering is based on a model of short term memory and attention to recurrent events in close temporal proximity. The clustering process

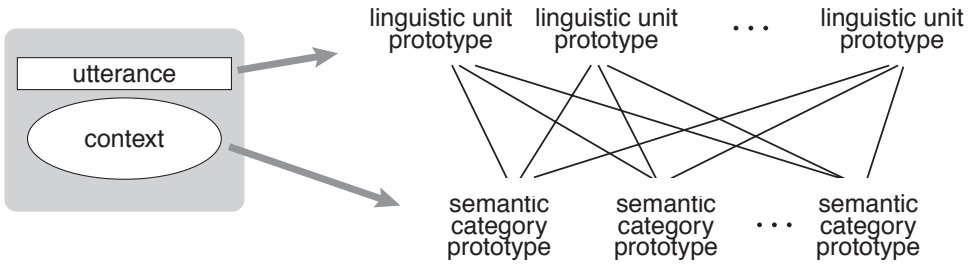


Figure 1-2: Deconstructing input into {linguistic unit, semantic category} prototype hypotheses.

is driven by a search to maximize mutual information between instances of linguistic units and co-occurring semantic categories.

The word learning process in CELL relies on combining evidence from multiple observations. This need for pooled observations raises issues of memory capabilities of the learner. It is unlikely that large amounts of memory are dedicated to storing unanalyzed sensory input[3]. To ensure cognitive plausibility, CELL combines input from multiple situations with only limited dependence on rote memory.

1.4 Interaction between Linguistic and Semantic Learning

A common assumption underlying most previously proposed models of language acquisition is that linguistic unit discovery, semantic category formation, and lexical item formation occur in stages. For the case of learning from spoken input, most models assume that speech segmentation and acoustic unit discovery is driven only by acoustic analysis. There is no clear evidence to support strict modularity and staged sequencing of these tasks. Infants have perceptual and learning capabilities which allow them to leverage contextual information when analyzing linguistic input, and vice versa. This thesis explores the possibility that linguistic and semantic input

[3]However, the ability for limited rote memory is known to exist [87].

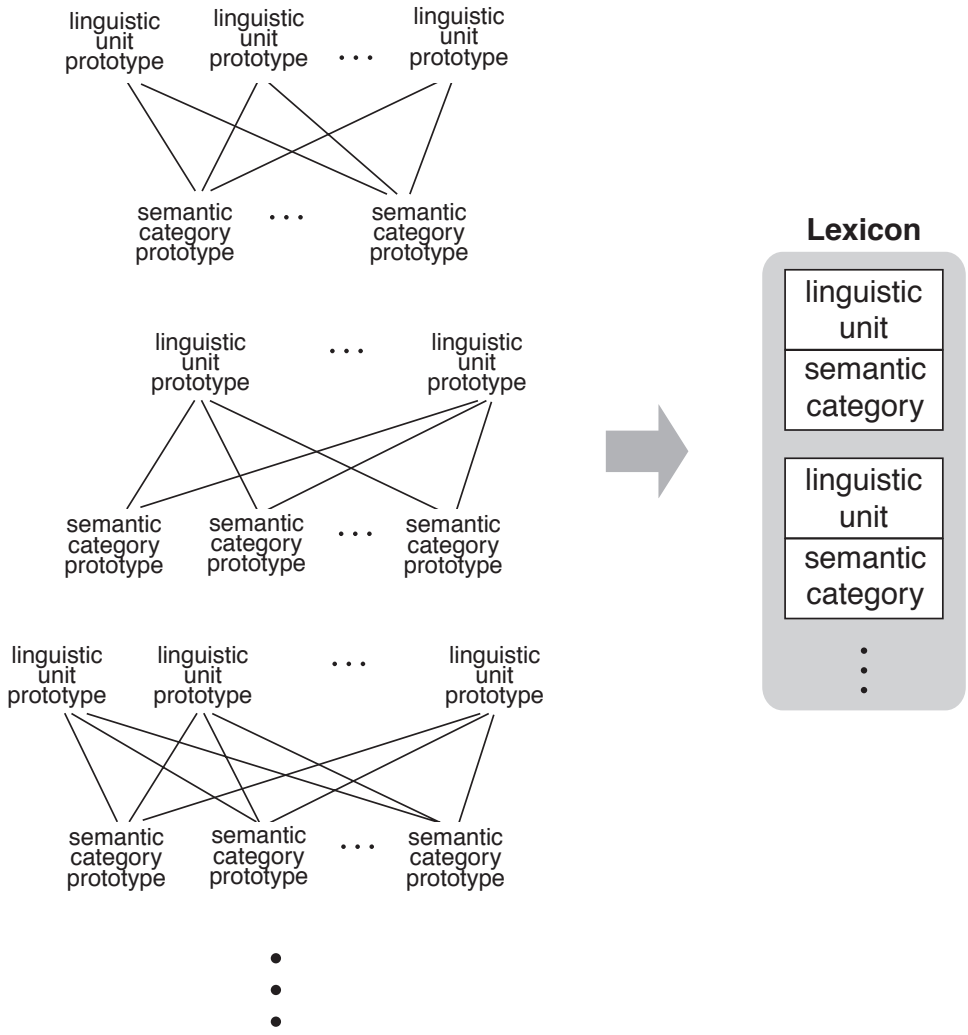


Figure 1-3: Building a lexicon from word-to-meaning hypotheses.

is used in a mutually constraining process to accelerate learning.

1.5 Sensory Grounded Input

All input in CELL is derived from raw sensory signals. In the current implementation, linguistic input comes from a microphone, and semantics are grounded in contextual information obtained from a color video camera. The sensory grounded nature of CELL differs significantly from other models of language acquisition which typically provide human-generated representations of speech and semantics to the system. In these models, speech is often represented by text or phonetic transcriptions. As a result, each time a word appears in the input, the model receives a consistent sequence of tokenized input (e.g., [51, 3, 23, 22]). Semantics are usually encoded by a predefined set of symbols or structured sets of symbols (e.g., [114, 1, 115, 31]). The problems presented in Section 1.1 would be simplified greatly if CELL had access to such consistent representations.

We have chosen to avoid such simplifications for three reasons. First, models which operate with raw sensory input work under constraints which are closer to the natural constraints under which infants learn. Infants only have access to their world through their perceptual systems. There is no teacher or trainer who provides consistent and noiseless data for the infant. Similarly, there should be no equivalent teacher or trainer to help a computational model. Thus, unlike any previous model of language learning, CELL grounds linguistic and semantic input in sensors.

Consider the difference between raw audio and phonetic transcriptions of speech. In raw speech, pronunciations vary dramatically due to numerous factors including phonetic context, syllable stress, the speaker's emotional state, the speaker's age, and gender. On the other hand, a trained transcriptionist will abstract away all these factors and produce a clean phonetic transcription. From a language modeling perspective, such transcriptions may seem equivalent to using raw audio with some

"noise" removed. We argue that the differences go much deeper. Transcriptionists will leverage their knowledge of language to overcome ambiguities in the acoustic signal. These sorts of effects based on pre-existing knowledge are bound to trickle into any model which relies on human-prepared input data. In addition, raw speech also contains prosodic information which provides cues for segmentation and determining points of emphasis. Such information is lost when only a phonetic transcript is used.

Second, we are interested in addressing the symbol grounding problem [50]. Symbols ultimately derive meaning from their relationships with the physical world. Machines which manipulate ungrounded symbols are ultimately limited by their disassociation from reality. Instead of treating language learning as only a symbolic processing problem, we are interested in grounding the semantics of lexical items in the physical world. This is achieved by grounding semantic categories in sensory input.

Third, a sensory-grounded computational model leads to powerful applications for human-computer interaction. The implementation of CELL presented in this thesis has led to the creation of adaptive spoken interfaces which learn the communication patterns of individual users.

1.6 Adaptive Spoken Interfaces

An important motivation for implementing CELL was to build better human-computer interfaces [101]. A fundamental problem which plagues current speech interfaces is their inherent rigidity and scripted feel. Unless the user knows what words and phrases may be spoken, when they may be spoken, and how they should be pronounced, the interface will fail. The problem lies in the fact that there are wide variations in how people express their intentions through words. People vary in how they pronounce words, what words they choose to utter, and the semantics they associate with particular words and phrases. It is impossible for an interface designer to anticipate and

preprogram all possibilities.

CELL provides a basis for creating spoken language interfaces which learn individual users' speech patterns, word choices, and associated semantics. Since CELL is grounded in sensory input, it is easily applicable to situations with natural, noisy input. This thesis presents prototypes of adaptive interfaces based on CELL.

1.7 Road Map

The remainder of the thesis is organized as follows:

- Chapter 2: Background on relevant aspects of infant development, previous related work on models of lexical development, and a review of computational modeling techniques.

- Chapter 3: Description of CELL at an implementation-independent level of abstraction. This chapter describes a general model for acquiring lexical items from multiple streams of sensory input.

- Chapter 4: An implementation of CELL grounded in camera and microphone input. This implementation uses computer speech and vision processing techniques to implement CELL for use with natural, noisy sensory input for the task of learning words which refer to colors and shapes.

- Chapter 5: An evaluation of the implementation using infant-directed data. Natural speech directed towards prelinguistic infants was recorded from six caregivers as they played with their infants. CELL successfully acquired a lexicon from this speech and co-occurring visual input.

- Chapter 6: Application of CELL for developing adaptive spoken interfaces for human-machine interaction. We present prototype interfaces which interactively adapt to individual users.

- Chapter 7: Contributions of this thesis, future directions and concluding remarks.

Chapter 2

Background

This chapter presents a review of several topics relevant to this thesis. Due to the broad range of topics, we highlight only selected works. We begin by reviewing some aspects of infant development which relate to language. Models and theories of lexical learning are presented in the following section. The third section discusses several computational modeling techniques which form a basis for implementing sensory grounded models of lexical learning.

2.1 Language Development

2.1.1 Speech Perception

Within the first few days of life, infants are able to differentiate between the sounds of their native language and other languages, possibly by attending to prosodic properties of speech [76]. A preference for listening to speech over other auditory input also arises in the first few months or even days of life [26]. These initial biases for attending to speech coupled with a presumed bias for mapping sensory forms to possible meanings paves the way for language acquisition.

Infants seem to be born with auditory processing abilities which enable them to distinguish all phonemes of all languages. For example, at the age of one month,

infants exhibit categorical perception of speech along phonemic boundaries [36].They begin to ignore phonetic differences which are not phonemic [64, 123]. It appears that at least some phonetic structure of the native language is acquired before lexical acquisition begins. The frequency distribution of various phonemes might drive this initial acquisition of phonetic structure. As early words are acquired, infants may use semantic constraints to learn further phonetic distinctions [59].

2.1.2 Visual Perception

Similar to the auditory system, the human visual system is designed to extract predefined salient aspects of the environment. Consider the case of color perception. The retina consists of rods and cones which operate in dim and bright light, respectively. Cones come in three kinds, each maximally sensitive to a different part of the color spectrum. It is likely that these color sensitivity biases lead to patterns in how the world's languages categorize color [35]. Although different languages and cultures vary in number of color terms used, there are strong correspondences between the foci, or central exemplars of color categories across languages [14].

Biological structures in the visual system also provide analyzers for spatial and temporal aspects of input. For example, the human visual system is sensitive to specific spatial orientations of edges [65], providing a basis for shape perception. Other independent analyzers exist for extracting information along many dimensions including spatial frequency, spatial position, temporal frequency, temporal phase, temporal position, and direction of motion [72].

Although an infant's visual system develops considerably in the first year of life, basic abilities which may facilitate language learning are present at a prelinguistic stage [75]. Experimental findings show that infants perceive color in roughly the same manner as adults [18]. Infants as young as one month exhibit a bias for attending to the overall shape of an object over other details [77]. Infants also attend to salient motions in the visual scene [25].

Object permanence refers to the ability to mentally represent an object which is not perceptually observable. The capability to form such mental representations is assumed to be a cognitive prerequisite to naming [16]. Baillargeon has provided evidence that infants may be able to represent hidden objects as early as 3.5 months of age [6].

Many of the perceptual abilities required to represent and categorize objects, people, colors, sizes, and actions are present in the infant at a very young age, before they produce their first words.

2.1.3 Sensitivity to Correlations

Infants are sensitive to statistical correlations or co-occurrences between multiple perceptually observable attributes. In the visual domain, Younger and Cohen found that 10-month-old infants were able to perceive differences in correlations between features of two-dimensional animal drawings [126]. Sensitivity to correlations in acoustic input has been demonstrated by Saffran, Aslin, and Newport [107]. In this study, infants were able to detect patterns of speech based on only conditional probabilities of sound sequences.

2.1.4 First Words

Children begin producing their first words sometime after the age of 9 months [10]. The most common type of first words are nominals which refer to objects, followed by words referring to actions [13, 11, 41]. Huttenlocher and Smiley [57] conducted a study to determine patterns of object naming in ten infants raised in the Mid-western region of the United States. They studied language production patterns in the second year of life. They found that the vast majority of early object names referred to mobile objects which are easily held by infants. Examples of common words across subjects included *ball*, *shoe*, *hat*, and *apple*.

Infants exhibit a lag between comprehension and productive abilities: they rec-

ognize words before they start using them. Experimental evidence suggests a lag of at least five months between comprehensive and productive abilities for early vocabularies [9, page 363]. Infants possess some ability to detect words in fluent speech contexts by the age of 7.5 months [60]. Studies have also shown that infants can recognize their own names at the age of only 4.5 months [73].

The lag between recognition and production suggests that the earliest stages of lexical learning are accomplished without feedback on the learner's productions. When infants eventually attempt to communicate verbally, they will receive feedback which helps to shape further lexical development. Feedback does not have to be explicit. If a child asks for an object and receives it, attaining the desired object provides implicit positive feedback. Failure to achieve their goals serves as negative feedback.

The meanings of words acquired by young children often differ from adult meanings. Children often under-extend the meaning of a word, applying it to a smaller subset of phenomena than in adult usage. Over time the child learns to extend (or decontextualize) the meaning so that it better matches adult meaning [63]. In other cases a word might become over-extended and again this is corrected both by feedback from adults, and as other words are learned and "take over" portions of semantic space.

2.1.5 Infant Directed Speech

Infant directed speech (IDS) differs from normal adult directed speech along several significant dimensions [117, 118]:

Structurally simplified Sentence structure is simple compared to adult speech, containing fewer complex phrases with subordinate clauses.

Shortened utterance length On average, utterances contain fewer words.

Redundancy Words and phrases are often repeated in close temporal proximity.

Reference to immediate context The topic of speech is tightly bound to the infant's immediate context, i.e., IDS refers to the "here and now".

Exaggerated prosodic contours Average pitch is raised, and prosodic contrasts are exaggerated in IDS compared to adult directed speech.

Characteristics of IDS are common across numerous cultures. Understanding these characteristics may be helpful for building computational models of language learning.

2.2 Models and Theories of Lexical Acquisition

2.2.1 Principles of Lexical Acquisition

This section summarizes several guiding principles of lexical acquisition. Infants seem to be innately driven by the principle of reference: words refer to objects, actions, and attributes of the environment [45]. Observational learning may be used to deduce word meanings from cross-situational experiences. Joint attention plays an important role in learning terms of reference. Infants are more likely to connect words with their referents when engaged in joint attention with their caregivers [7]. Learning from immediate contexts poses potential problems for verb learning. Verbs are less likely to co-occur with the associated actions, especially for non-observable verbs such as *think* [43].

The philosopher, Quine, posed a well-known problem in observational learning: an infinite number of possible meanings can be inferred from a finite set of utterance-context pairs [91]. Imagine a person in a foreign land points to a rabbit running and exclaims "gavagai!". How is the observer to infer the referent of "gavagai"? Possibilities include not only rabbit, but also furry, undetached rabbit parts, and abstract referents such as rabbithood. A likely answer to this problem is that all infants have certain biases which constrain the set of possible meanings of words.

Markman has proposed a set of constraints which children follow to avoid Quine's paradox [74]. The whole object assumption proposes that children will assume a novel label refers to a whole object rather than its parts, substance or other properties. The taxonomic assumption proposes that children expect labels to refer to basic-level categories. This assumption lets them rule out thematic relations such as the particular spatial context in which the object is observed. The mutual exclusion assumption proposes that children prefer to assign only one label to a concept. Although this assumption obviously can fail (e.g., *dog* and *poodle*), Markman suggests that it is a good strategy for bootstrapping the inference process. This assumption may force the learner to consider attributes, substances, and parts of objects as possible referents which were initially avoided due to the whole object assumption. Problems related to different levels of taxonomic reference (*car* vs. *toy*) are resolved only later in the learning process.

2.2.2 Speech Segmentation

In this section, we review several models and theories of speech segmentation and word discovery. Let us begin with the hypothesis that infant directed speech contains many words spoken in isolation. The child could first memorize these word forms and then use them to bootstrap the segmentation process. They must locate these known words in longer utterances and find word boundaries which are defined by their end points. Studies of infant directed speech do not support such a theory. Although IDS is syntactically simple [88], caregivers rarely use isolated words [3]. A more likely strategy for speech segmentation may be to remember sound sequences at the beginning and end of pause delimited utterances since these will also correspond to the beginnings and ends of words.

Cutler suggests that segmentation strategies are based on the prosodic rhythm of spoken language [28]. She proposes that infants use stress units, syllable units, and mora (subsyllabic) units to locate word segments in English, French and Japanese

respectively. An unresolved aspect of this theory is how the infant is to decide which prosodic cue to initially use.

In a related theory, Cutler and Mehler suggest that infants are born with a bias for periodicity in auditory signals. This would result in vowels, which have relatively steady state properties in the frequency domain, to be highlighted and contrasted from consonants. A periodicity bias allows the infant to detect syllable structure of speech which can act as scaffolding for the word segmentation process.

Another potential cue for segmentation is to use the distributional statistics of phonemes. The predictability of a phoneme given its prior context is high within words but low at word boundaries. Infants could learn the distributional characteristics of the language and then attempt to segment utterances at points of low predictability. Saffran, Aslin, and Newport provided support for this theory by demonstrating that 8-month-old infants are able to find word boundaries in an artificial language based on only distributional cues [107].

Harrington, Watson and Cooper wrote a computer simulation of word boundary detection based on phoneme triplets found within words and between words [51]. They tabulated all within-word phoneme trigrams which occurred in a database of the 12,000 most commonly occurring words of English. The simulation detected word boundaries by looking for instances of trigrams which did not occur in the 12,000 word database. The system was able to detect 37% of word boundaries with a false alarm rate of 11%. The performance could probably be improved by using trigram probabilities rather than discrete occurrence tables.

Brent explored the problem of speech segmentation by using a combination of phonotactic constraints and distributional regularities in a minimum description length (MDL) framework [23]. Phonotactic constraints dictate the allowable phoneme sequences of a language. For example, although /cat/ is an acceptable syllable in English, /pcat/ is not. Distributional regularities encode the frequency statistics of phoneme sequences. Brent's word segmentation algorithm attempts to maximize the

cumulative probability of postulated words based on distributional statistics while constraining the search with phonotactics. The MDL constraint was used to insure a compact lexicon. The model was implemented and tested on a corpus of hand-transcribed speech and resulted in segmentation into word-like units. Although the system relies on consistent transcribed input, the underlying model could be applied to raw audio as well, although performance would drop significantly.

De Marcken proposed a model of speech segmentation which learned a hierarchically structured lexicon using MDL constraints in a Bayesian framework [31]. In contrast to Brent's model, this model has richer a priori constraints due to the hierarchical nature of the model. Brent reports the successful acquisition of word-like units from unsegmented text. The model was also tested with raw acoustic input but its performance degraded drastically. When input was switched from transcribed data to raw speech, de Marcken only provides sample output and remarked that "Except for isolated sentences, the segmentation of speech data are not particularly impressive." [30].

To summarize, there are several types of cues which infants are able to detect and use to segment speech. Although studies show that infants can notice each of these cues, it remains to be established which strategies or combinations of strategies are actually used by infants for speech segmentation.

2.2.3 Word Learning

Jusczyk has proposed the WRAPSA (Word Recognition and Phonetic Structure Acquisition) model to explain how infants develop perceptual abilities necessary to recognize words in fluent speech [59]. In the model, phonetic distinctions are initially language-independent and become tuned to the native language with experience. Jusczyk proposes that phoneme discrimination shifts are initially driven by frequency patterns of the language. The infant's auditory analyzer learns to attend more to frequently occurring sounds, which logically are the phonemes of the native language.

As the infant starts to learn first words, the semantics help drive more precise learning of the sound pattern of the target language. Language independent phoneme discrimination abilities are gradually pruned in the process. Word segmentation is aided by prosodic analysis to arrive at syllabic structure which is used to chunk input speech. Word recognition is achieved by matching stored acoustic templates. Acoustic templates are represented in terms of syllabic structure and acoustic features which become richer as the lexicon size grows.

Sankar and Gorin created a computer simulation of a blocks world in which a person can interactively type sentences which are associated with objects of various colors and shapes [108]. Input representations are abstract: segmented text serves as linguistic input, and discrete valued vectors encode object shape and color. Using an interactive graphical interface, objects can be brought to the system's attention and labeled with text strings. The system learns to identify information-bearing words and associate them with their color or shape semantics. This model used a network with internal structure designed to reflect the target semantic domain. In particular, the model had sub-networks specialized for learning shape and color terms. By partitioning the semantic space, the system learned faster.

Feldman et al., have initiated a research project to study language acquisition with computational models [38]. The goal of the project was to build a system which can learn the appropriate fragments of any natural language from sentence-picture pairs. Sentences are represented as segmented text, and the pictures are synthetic blocks-world style images. This work led to interesting results in learning labels for spatial relations between objects [96], and verb names [5]. In recent efforts, the group has incorporated structural knowledge about the human body into their connectionist learning networks. Verbs, spatial relations, and metaphor are grounded in terms of their model of embodiment. Similar to Sankar and Gorin's system, this model incorporates structural constraints of the world to constrain and ground the semantics of language.

Gorin developed a language acquisition system which learned from raw acoustic speech and semantic annotations [46]. In initial experiments the system accepted interactive feedback which was used to associate isolated spoken words with one of three semantic classes [47]. A second system learned to answer 20 almanac questions about the 50 American states [79]. In later versions, input consisted of connected utterances paired with one of 20 semantic class tags. The system is able to learn which words were useful for a classification task using a measure of semantic relevance based on mutual information. In Gorin's framework, the *salience*, or semantic importance of a word is estimated using the weighted mutual information between the occurrence of the word, and the the occurrence of classification tags. Gorin's system did not attempt to learn semantic classes, and also did not address the problem of speech segmentation.

Siskind has developed a computational model which acquires the semantic and syntactic roles of words as well as phrase structure. Input consists of segmented text sentences paired with semantic annotations of the contextual situation [115, 116]. The learning mechanism employed a cross-situational strategy to assign semantic primitives to words which best account for all data in memory.

2.2.4 Bootstrapping Syntax

Although it is beyond the scope of this thesis to address the acquisition of syntax, we briefly discuss the question of syntactic category acquisition. Categories, such as verbs and nouns, form the building blocks for learning the rules of grammar including grammatical relations, cases, and phrase structure configurations. Without syntactic categories a learner will be unable to acquire the rules of the language. One-to-one mappings between syntactic and semantic classes do not seem to exist. For example, although persons, places, and things count as nouns, so do *heat*, *year*, and *flight*. It is difficult to find semantic groundings shared by all of these terms which give them the status of nounhood. Bootstrapping theories offer solutions to these

problems by providing strategies for deriving syntactic categories from perceptual input. We consider two theories, semantic bootstrapping [89], [48] and prosodic [80, 44] or phonological [81] bootstrapping.

Semantic bootstrapping proposes that the language learner uses semantic categories to seed syntactic categories. Perceptually accessible categories such as objects and actions would seed syntactic classes for nouns and verbs. Once these initial categories have been established, novel input utterances are used in combination with constraints from Universal Grammar to deduce phrase structure. In turn, the acquired phrase structure can be applied to input utterances with novel words to expand the syntactic classes through distributional analysis. This theory assumes that the learner has already acquired words and their semantics without the use of any syntax.

Prosodic or phonological bootstrapping encompasses a variety of theories which use information in the phonetic, phonotactic, and prosodic aspects of acoustic input to guide the process of syntax learning. For example, Gleitman has suggested that the prosodic structure of English correlates with syntactic structure [80, 44], enabling a language learner to acquire syntax from prosody. Research indicates that a combination of phonological cues may lead to the acquisition of some syntactic knowledge (see the collection in [81] for various approaches to this theory). A potential objection is that many of the prosody-based inference schemes are language-dependent and would fail if applied without modification to other languages [89].

2.3 Computational Modeling Techniques

2.3.1 Speech Recognition

This section provides a brief survey of common techniques found in many speech recognition systems today. These techniques may be employed to implement computational models which process raw acoustic input.

Spectral representations have become widely accepted as a suitable representation

of speech. Mel-scaled cepstral representation has proven to be especially useful for speech recognition applications [93]. The mel scale is a log-linear scale which approximates the frequency tuning curves of the ear. The cepstrum is a Fourier transform representation of the log magnitude spectrum. To deal with environmental noise, several models based more closely on human audition have also been proposed including RASTA-PLP [53] which was used to implement CELL and is described in Section 4.2.1 (see also [42]).

Automatic speech recognition is based on creating models of reference speech sounds which serve as templates to match against novel auditory input. Multiple models compete as likely matches for the input speech. Novel input is recognized by comparing the input to each template and selecting the best match.

The simplest method of creating a speech model is to use a spectral template derived from a single exemplar of the word or phrase. To compare input which is different in duration the template must be stretched to match in length. Dynamic time warping (DTW) can be used to non-linearly stretch the template in order to properly align with input sequences. The use of DTW has largely been replaced by the use of Hidden Markov Models.

Hidden Markov Models (HMMs) are a powerful method for modeling speech as a sequence of observations generated by a hidden finite-state structure [92]. The Estimation-Maximization algorithm can be used to train parameters of the HMM from labeled speech. Viterbi decoding is used to recover the hidden state sequence from an observed sequence (i.e., novel speech). Multiple HMMs, each representing a different unit of speech (e.g., a phoneme or a word) can compete to explain an observation. Recognition is achieved by selecting the HMM which has the highest likelihood of having produced the observation. The HMM provides the same non-linear time warping capabilities of DTW, but in addition the HMM is able to encode information about multiple utterances in a single probabilistic model. It does so by encoding statistical models of variations in the signal based on a training corpus.

Although similar capabilities may be achieved by maintaining multiple templates and using DTW, the HMM framework provides a principled statistical framework for doing so.

Neural networks (NN) have also proven to be useful in speech recognition tasks [105], [121], [97]. Neural networks with feedback weights are called recurrent neural networks and are useful for short time scale recognition tasks such as phoneme recognition. In practice, neural networks perform at approximately the same level as HMMs for phoneme level recognition tasks. In higher level tasks, the HMM framework is better suited since statistical language models and human knowledge of domains are difficult to incorporate into NN-based systems.

Large vocabulary speech recognition systems require language models or grammars to constrain word hypotheses during lexical search. Two approaches are commonly used for specifying language models. The first consists of defining context-free grammars (CFG) which define legal word transitions based on word class definitions and possibly recursive rewrite rules. A second approach is to define a statistical n-gram model which defines probabilities of words conditioned on n previous words. Context-free grammars are easy to design for small domains but typically lead to highly constrained systems. To successfully use the systems, the user must be aware of the allowable sequences of words that may be spoken. For larger domains, hand written grammars become unwieldy and difficult to design. In comparison, statistical grammars lead to more robust recognition but they require massive amounts of training data. Most effort in training statistical grammars goes into finding ingenious ways to extract reliable probabilities from small amounts of data.

Subword units are often used in large vocabulary systems since sufficient data to train individual word models is typically impractical. Context dependent units (e.g., generalized triphones) may be used to account for co-articulation effects of continuous speech. New words can be modeled by concatenating HMM models of the appropriate constituent word models [2]. Variance statistics of the subword models

are automatically incorporated into the resulting model, providing a better model of possible alternate pronunciations of the word than a single template combined with DTW would provide.

2.3.2 Visual Object Recognition

Haralick and Shapiro divide two-dimensional object representations into 5 broad classes [49]. An object can be represented using its segmentation mask, a binary image where only the pixels corresponding to the object are set to 1. Scale, rotation, translation and skew invariant measurements of the mask can be used as a compact representation of an object which can be compared using Euclidean or other distance metrics. Hu's moment invariants are a classic example of this method [56]. In a second representation class, only the boundary of the object rather than the full mask image is stored. Third, salient local features such as holes, corners and edges of the object are extracted. The features are stored in a representation which retains spatial relationships and allows objects to be compared based on alignment of features. A fourth class, suitable for stroke based objects (such as written and printed characters), consists of skeleton representations which attempt to encode the underlying "stick man" structure of objects. Finally, a representation-by-parts approach attempts to decompose complex objects into component parts.

The main trade off between representations lies between detail of representation versus robustness of matching novel instances of a class of objects. For example, a global representation based on segmentation masks combined with moment invariants is a relatively robust method of comparing objects. The largest likely source of error would be from poor foreground/background segmentation. Many objects, however, which may be discriminated using some more detailed representations such as boundary cues may fail to be separated using moment invariants alone. On the other hand, feature based methods also suffer from problems of noise which may cause failure of higher level matching processes that rely on specific feature correspondences.

Three dimensional object recognition adds the complexity of dealing with multiple viewpoints of an object. Two basic strategies exist: explicitly model the object's 3-D form, or treat multiple two-dimensional views of an object as its 3-D representation. The latter is referred to as view-based representations [120, 82, 94, 109]. The CELL implementation uses a view-based approach based on histograms [109].

2.3.3 Machine Learning

In this section we briefly review three areas of machine learning which are relevant to the thesis: clustering, Hidden Markov Models, and neural networks.

Clustering techniques are used to group data such that members of each group are close to one another according to some predefined distance metric [34]. Clustering is typically thought of as an unsupervised problem. Given a set of samples without labels, the goal is to divide the samples into useful groups. Most clustering techniques are iterative and attempt to minimize cost function which penalizes grouping distant samples together. A commonly used clustering method is the K-means algorithm [71]. Given a set of samples, the algorithm begins by randomly assigning each sample to one of K sets. It then computes the mean of each set, and re-assigns each sample to the set with the closest mean. The calculations of means and assignment of samples is iterated until there is no change in sample groupings. The ISODATA algorithm provides a set of heuristics to automatically determine the number of clusters, K, in a set of data [8].

Hidden Markov Models model dynamic processes using stochastic finite-state machines [92]. The most difficult aspect of employing HMMs is setting their parameters to maximize the likelihood of an HMM for a set of training data. The Baum-Welch algorithm, a special case of the EM algorithm, provides a solution to the training problem based on an iterative algorithm which converges on a local solution of the parameter search space. A second problem is to recover the most likely state sequence given an observation sequence. Conceptually, all paths through the HMM must be

enumerated and the path with the highest likelihood is chosen. In practice this search, which is exponential with respect to observation length, can be reduced to a linear search based on the Viterbi dynamic programming algorithm.

Artificial neural networks, also called connectionist or parallel distributed processing models, are dense interconnection of simple computation units [15]. A subset of the units are designated for input, and another subset for output. When an activation pattern is applied to the input units, the pattern is propagated through the network. Each unit computes its output activation as some (typically non-linear) function on the sum of the weighted input activations. Once the propagation is complete, the response of the network to the input can be read off the output units. The back-propagation algorithm provides an iterative solution to the problem of learning connection weights for multilayer networks with non-linear units [105]. The algorithm learns the transition weights of a network for a given set of training data. The training data consists of pairs of input activation vectors and desired output activation vectors. An interesting extension for modeling dynamic processes such as speech is to add time delay units and feedback weights which let activation levels from any part of the network from a previous time step become input for the current time step [37, 58].

2.4 Discussion

This chapter has highlighted a broad range of topics in infant development, theories and models of language acquisition, and computational modeling techniques. From these, we can summarize four key ideas relevant to this thesis.

Experimental evidence suggests that prelinguistic infants are able to perceive and discriminate visual forms and speech sounds. They are also sensitive to correlations and probabilistic associations in their environment. The built-in mechanisms available to CELL prior to learning are modeled on these assumptions. CELL includes

"innate" knowledge for extracting useful representations from sensory input, and for performing probabilistic analyses on these representations.

Most previous models of language learning including those reviewed in this chapter assume that at least some input is represented in a clean symbolic form. In contrast, CELL is grounded in raw sensory input. None of the models to date ground both linguistic and semantic input in sensors.

In general, models which explain speech segmentation are based on acoustic analysis alone. Models of higher level learning assume that linguistic input is already segmented into words. An underlying assumption of these models is that learning to segment the speech signal (i.e., word discovery from continuous input) precedes acquisition of word meaning and syntax. In contrast, CELL proposes that acoustic and semantic analysis occur together in a combined learning framework.

Computational tools developed in the fields of computer vision, speech processing, and machine learning can be applied to implement models of sensory grounded language acquisition. Such models may more closely approximate the actual tasks performed by infants in realistic noisy environments. CELL has been successfully implemented using these techniques and has been evaluated on raw infant-directed speech and camera images.

Chapter 3

Cross-Channel Early Lexical Learning

The Cross-channel Early Lexical Learning (CELL) model has been developed to understand how infants acquire early words from multiple streams of sensory input. Lexical learning is driven by a search for structure across input channels, hence the term *cross-channel*. CELL constructs models of this structure by creating *lexical items*. A lexical item contains a specification of a linguistic unit (for spoken language this would correspond to the acoustic model of a word), and a specification of a corresponding perceptually grounded semantic category to which the linguistic unit refers. CELL is a model of *early* stages of the acquisition process. Later stages which involve feedback from caregivers as the language learner attempts to produce words is not modeled. The model is computational which means its processes and memory structures are specified with sufficient precision to enable implementation on a computer and evaluation with test data.

This chapter presents an implementation-independent description of the model. The following three chapters describe the implementation, evaluation, and applications of the model. The separation of the model from implementation emphasizes the general nature of CELL, which may be applied to a variety of domains beyond

those implemented in this thesis. We do believe, however, that implementation and data-driven evaluation is essential to assess the viability of this and any other model.

This chapter begins by revisiting three problems we wish to address in this thesis. The bulk of the chapter then presents the core model in terms of processes and memory structures which acquire lexical items from sensory input. The last part of the chapter presents several extensions of CELL. These extensions are not central to the thesis but provide a basis for integrating CELL with other theories of language learning and cognition.

3.1 Problems Addressed by CELL

Chapter 1 introduced three problems of early lexical learning. We restate these problems to delimit the scope of issues addressed by CELL.

Problem 1: Linguistic Unit Discovery The first problem is to discover linguistic units which correspond to words of the target language. This is a challenging problem since most infant-directed language consists of multiword utterances [3]. Furthermore, even though we assume that underlying linguistic units are discrete in nature, these units will be rendered differently each time they are produced due to variations in the speakers' identity, age, gender, emotional state and other factors. To complicate matters, linguistic units are only observable through noisy sensory signals which are effected by background noise. The learner must cope with this noisy evidence and infer the underlying linguistic units.

Problem 2: Semantic Categorization A second problem is to learn semantic categories which serve as referents of linguistic units. No knowledge about innate semantic categories is assumed by the model; instead, categories appropriate for the target language must be learned from positive examples. The learner must once again confront the problems of variations of natural phenomena and

noise. For example, two shapes of red rarely look *exactly* the same. The learner must form a category for the color red based on a finite number of exemplars.

Problem 3: Linguistic-to-Semantic Mappings The learner must associate linguistic units with appropriate semantic categories. These associations must be inferred from collections of experiences since single examples will have multiple interpretations.

These three problems are treated as different facets of one underlying problem: to discover structure across linguistic and contextual sensory input.

3.2 Lexical Acquisition

This section begins with an overview of the CELL architecture, and then proceeds with details of the model.

3.2.1 Overview of the Model

An overview of the model is illustrated in Figure 3-1. Sensors provide input for the model. In an implementation these might include microphones, cameras, touch sensors and other physical sensory devices. Feature analyzers extract multiple channels of input from the sensors. These analyzers represent innate biases which determine aspects of sensor signals that will be represented in the model. There is no one-to-one mapping between channels and sensors. Multiple channels may be derived from a single sensor. Conversely, a single channel may combine evidence from multiple sensors. The concept of a channel is illustrated below with several examples, and is more precisely defined in Section 3.2.4.

A subset of the input channels are designated to carry linguistic information (e.g., spoken utterances) and are called the *linguistic channels*. Section 3.2.2 discusses how channels might be separated into linguistic and contextual groups. Examples of

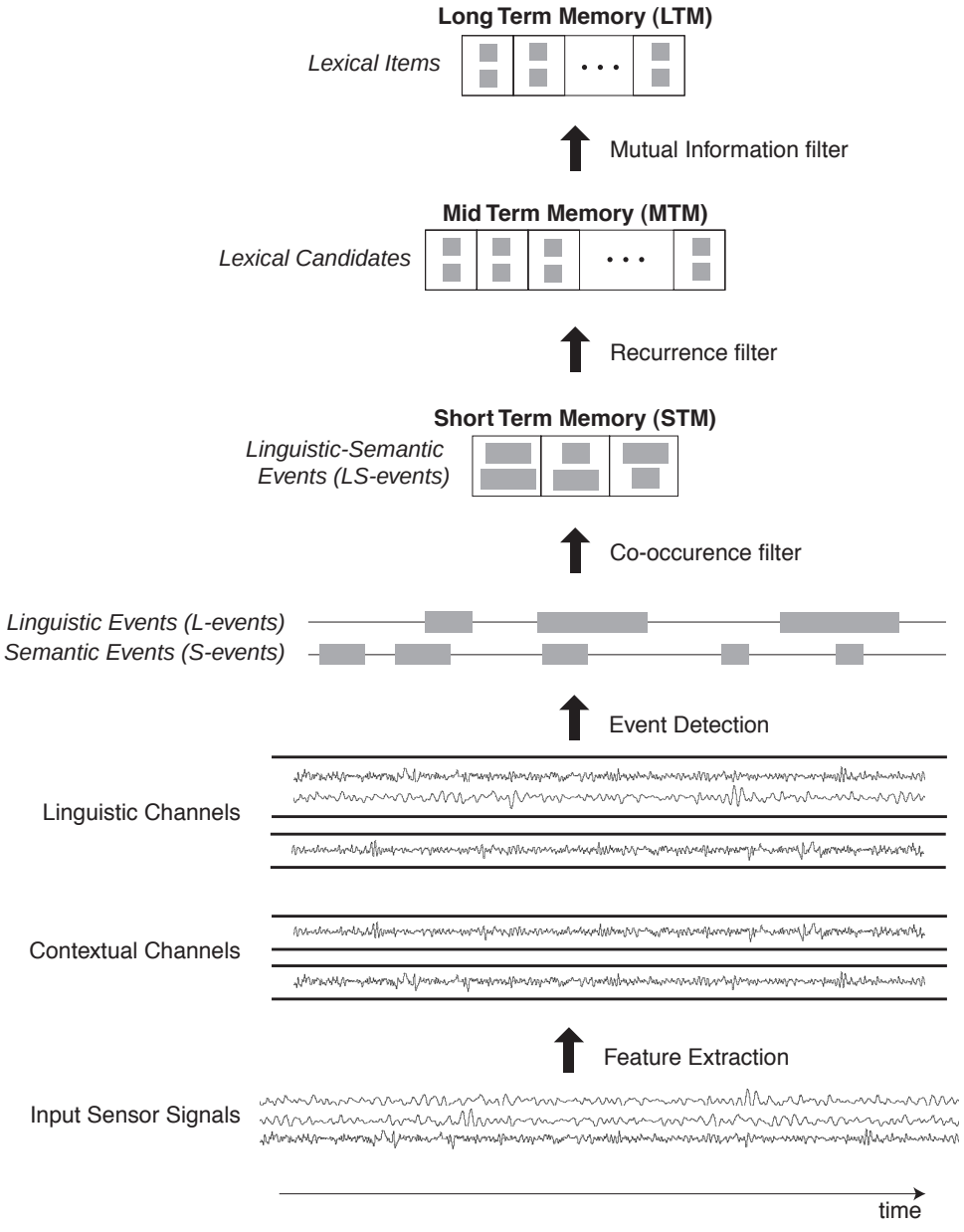


Figure 3-1: Overview of data flow through the CELL model.

linguistic channels include representations of:

- Speech sounds in terms of phonemes or other sub-word units
- Pitch contours of speech (i.e., the fundamental frequency, F0)
- Visually observed lip movements which aid speech understanding
- Hand gestures which complement spoken language, and form the primary communication channel for learners with hearing impairments

A linguistic unit may be defined in terms of one or more linguistic channels. Consider an example in terms of the above four channels. In English, *red* might be defined in terms of a sequence of speech sounds and a corresponding sequence of lip motions. Fundamental frequency and hand gestures might not be relevant for the unit *red*. In a tonal language such as Thai, however, the fundamental frequency contour would also need to be specified. In American Sign Language (ASL), the first two channels would be disregarded, while lip motions and hand gestures would be critical.

All remaining input channels are referred to as *contextual channels*. They encode information about the environment in which the language learner is situated. We assume that contextual channels carry semantic information about the co-occurring linguistic stream. Examples of contextual channels include representations of:

- Shapes of objects
- Colors of objects
- Size of objects
- Spatial relations between objects
- Motion of objects

- Identity of human faces

An infant's early conception of the semantics of *ball* might be grounded in visual stimuli of round objects. A contextual channel which represents the shape of objects would serve as a basis for learning such visual semantic categories. Similarly, each of the listed channels would lead to different classes of semantic categories. The semantics of *red* and *big* could be grounded in object color and size channels, respectively. The semantics of the words *above* and *beside* could be formed using the spatial relations channel. Verbs such as *push* and *open* might at least partly be grounded in the motion channel. A channel encoding the identity of faces may be used to learn the early semantics of people's names. In addition, the meaning of a word may be defined in terms of combinations of contextual channels.

CELL does not specify the channels of input. The lists above are provided as examples of the sorts of input that CELL may operate on. Chapter 4 presents a system which implements three of these channels: phonemic representation of speech, and object shape and color. It is interesting to note that computer models and implementations exist for all the representations listed above (for example, for the linguistic channels see [93, 99, 119, 124, 20] and for the contextual channels [96, 17, 29, 120]). To search for structure across input channels, CELL performs the following operations:

- Hypothesize prototypes of linguistic units which correspond to words
- Hypothesize prototypes of semantic categories
- Maximize mutual information between linguistic units and semantic categories based on hypothesized prototypes
- Create lexical items based on prototypes which result in high mutual information

The first and second operations generate large numbers of hypotheses of possible linguistic units prototypes and their corresponding semantic categories prototypes. The third step considers each prototype pair as the basis for forming a lexical item. The fourth step selects the best hypotheses and generates lexical items on their basis. Using these four operations, CELL simultaneously addresses the three key problems of lexical learning stated earlier.

The CELL architecture was designed to achieve cognitive plausibility and to facilitate real-time implementation on standard computer platforms. As Figure 3-1 indicates, CELL has a layered memory structure which serves to funnel data from input to long term memory. This architecture reduces dependency on rote or verbatim memory while focusing processing on salient parts of the input stream.

3.2.2 Assigning Linguistic vs. Contextual Channels

CELL assumes that input features are grouped into channels innately. For example, all features which collectively specify the shape of an object should be grouped separately from all features which specify the color of an object. Knowledge of the grouping of features into channels is innate and is inherently encoded in the design of the feature analyzers.

A difficult question is: How does the learner decide which channels are linguistic, and which are contextual? We do not try to answer this question in CELL; instead, CELL assumes the separation of channels is known. In this section, however, we discuss two possibilities for how a learner could come to know this decision.

One possibility is that the learner begins by treating all channels as equal and is willing to form lexical items (or, perhaps, some precursor to lexical items) which model structure across *any* set of channels. Over time, the learner notices which channels are common across most lexical items, and assign these common denominators to be the linguistic channels. From this point on, the learner would focus attention on finding lexical items which involve these channels.

Another possibility which may work in cooperation with the above hypothesis is that infants have innate biases which lead them to focus on intentional behaviors generated by their caregivers [90]. Innate biases to attend to cues such as exaggerated prosody and eye contact may help the infant focus on speech and gesture which carry linguistic information in all natural human languages. All other channels would be assumed to carry context. These hyptotheses are speculative and require further study.

3.2.3 Innate Learning Biases

A critical innate learning bias is the sensory apparatus available to a learner. The sensors impose strong limits on what is learnable. In addition, a critical element of any learning system is the choice of representation extracted from the sensors. For example, the human visual system has innate mechanisms for detecting edges at specific angles of orientation [65]. These biases facilitate learning categories of shapes which are invariant to other aspects of the stimulus such as color and size. Frogs, which do not have similar spatial edge receptors, are unlikely to represent and recognize shapes in any manner similar to humans [69].

In general, if we can anticipate aspects of the raw input signal which will likely be useful for the learning task at hand, we can accelerate learning by making these representations innate. Infants are all born with similar sensory and motor systems. The structure and known functions of these biological mechanisms provide insights for representations which might be built into CELL. Representations are not discussed further in this chapter since their choice depends on the learning task in question. Chapter 4 returns to the issue and describes the choice of representations for a specific implementation.

Similarity measures which compare representations are another important learning bias. To form categories, a measure of similarity between samples must be available. The nature of the measure critically affects how past experiences are categorized, and

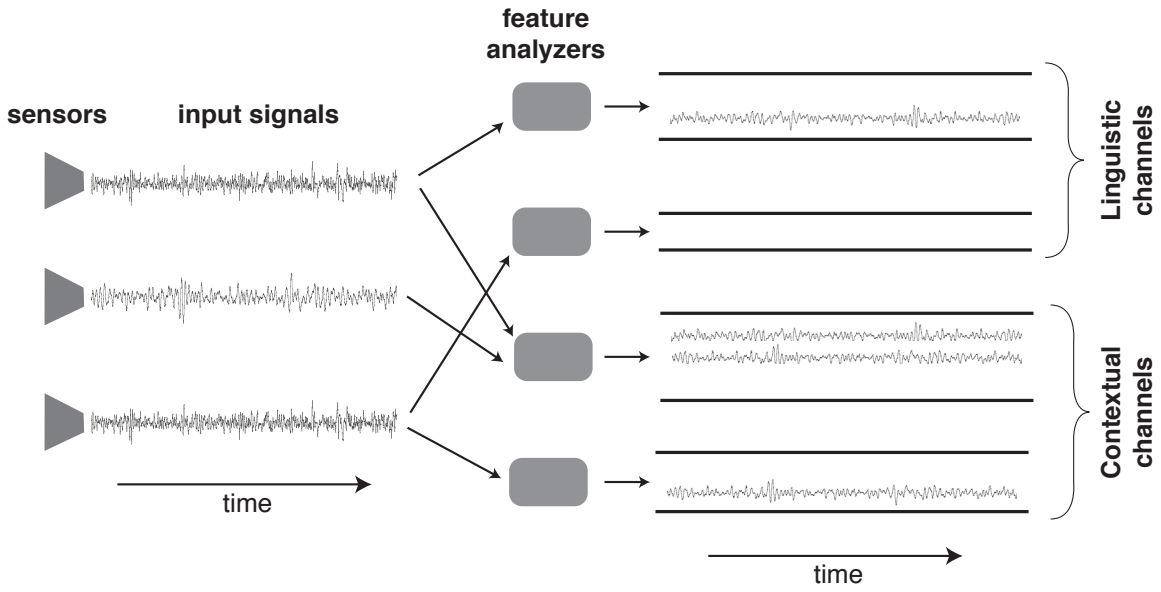


Figure 3-2: Channels of features are extracted from sensory input signals.

how new input is related to existing representations.

3.2.4 From Sensors to Channels

We now present details of each aspect of CELL. Sensors generate input signals which encode information about the learner's environment. Figure 3-2 shows a set of sensors (left) that generate time-varying signals conveying real-time information about the world.

A set of feature analyzers extract salient information from the sensory signals. As Figure 3-2 indicates, the number of feature analyzers does not have to equal the number of sensors. Furthermore, feature analyzers may receive input from one or more input signals, and the same input signal may feed into multiple feature analyzers. The output of each analyzer is a set of time-varying features which are grouped into channels. Channels represent different aspects of the environment.

A subset of input channels convey linguistic information to the learner and are referred to as linguistic channels. The remaining input channels are referred to as contextual channels. They carry information about the environment which encodes

the underlying semantics of linguistic input.

Definition 1 *Linguistic features are time-varying values extracted from sensory input which encode linguistic input. The surface form of words are represented by these features.*

Contextual features are time-varying values extracted from sensory input which encode non-linguistic input. The semantics of words are derived from contextual features.

Linguistic channels are groups of linguistic features.

Contextual channels are groups of contextual features.

Examples of linguistic and contextual channels were provided in Section 3.2.1, and possible methods for distinguishing linguistic from contextual channels was discussed in Section 3.2.2.

3.2.5 From Channels to Discrete Events

Continuous streams of features from input channels are segmented into discrete chunks called *events* (Figure 3-3). These events are referred to as *L-events* when generated from the linguistic channels:

Definition 2 *An L-event is a sequence of linguistic features delimited by pauses which correspond to an utterance.*

L-events capture activity in all linguistic channels for the period of time corresponding to an utterance. For spoken input, an L-event corresponds to a spoken utterance delimited by silence. A speech/silence detector may be used to implement this event detector. In natural situations, L-events will usually contain multiple concatenated words (See section [3]).

Another set of segmentation processes operate on the contextual channels to generate *S-events* which record non-linguistic information:

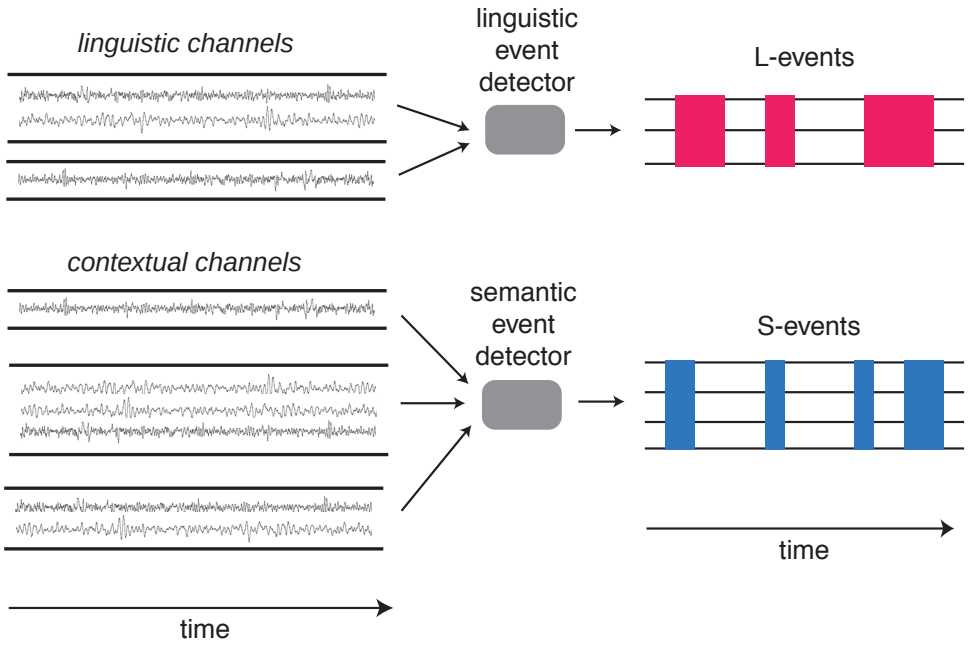


Figure 3-3: Event detectors chunk continuous data from input channels into L-events and S-events.

Definition 3 *An S-event is a sequence of contextual features.*

An S-event captures activity in all contextual channels corresponding to a discrete period of time. S-events are generated in response to salient events in the environment. The definition of a salient event depends on the nature of representations in the contextual channels. For example, if one of the contextual channels encodes color, an S-event might be generated whenever a highly color-contrasted scene or object is encountered by the learner. As a second example, a channel carrying motion might be used to generate events whenever a significant visual action is witnessed. S-events record all contextual channels regardless of which channel actually triggers the event.

3.2.6 Unpacking Events

Events can be divided into discrete time segments, and along channels. Figure 3-4 illustrates the process of event segmentation along the time axis. The event segmenter identifies natural segment boundaries in an L-event or S-event. CELL must discover

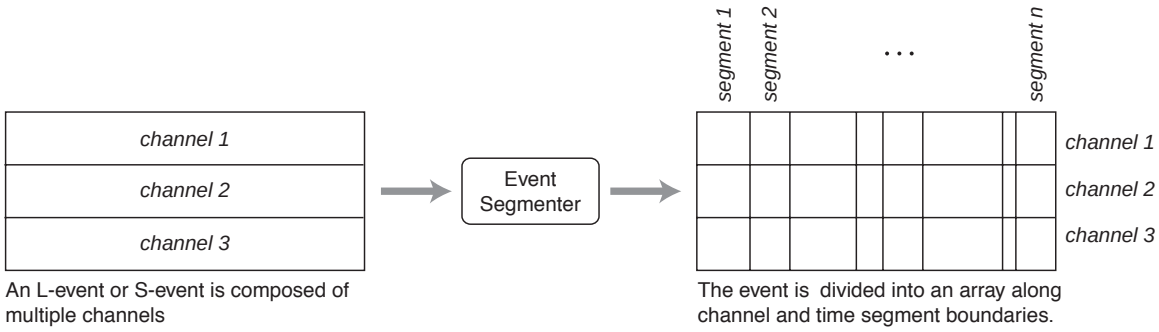


Figure 3-4: An event segmenter locates potential segmentation boundaries along the time dimension. The combination of channel groupings and segment boundaries leads to a grid decomposition of an event.

units from these unanalyzed blocks of data. Event segmentation identifies all possible linguistic unit boundaries. As an example, consider an L-event derived from three linguistic channels: phonemes, F0, and visual lip movements. Each channel may be segmented in time based on the contents of any of the linguistic channels. Each major transition in F0 contour, phoneme transition, or salient lip movement could result in a segment boundary. Each of these boundaries forms a hypothesis of the location of a linguistic unit boundary.

An S-event may similarly be divided into time segments. Consider a channel which represents motion. Each significant change in velocity, or each point of contact between objects, could lead to a segmentation boundary. These boundaries would form potential boundaries of a complete visible action.

Channel and time segment boundaries define the finest granularity at which an event can be analyzed. Segments are non-overlapping; the complete set of segments generated from an event can be concatenated to create the original event. Based on this channel-segment array, we can define the concept of a subevent.

Definition 4 *An L-subevent is a subsequence of an L-event composed of one or more linguistic channels.*

An S-subevent is a subsequence of an S-event composed of one or more contextual channels.

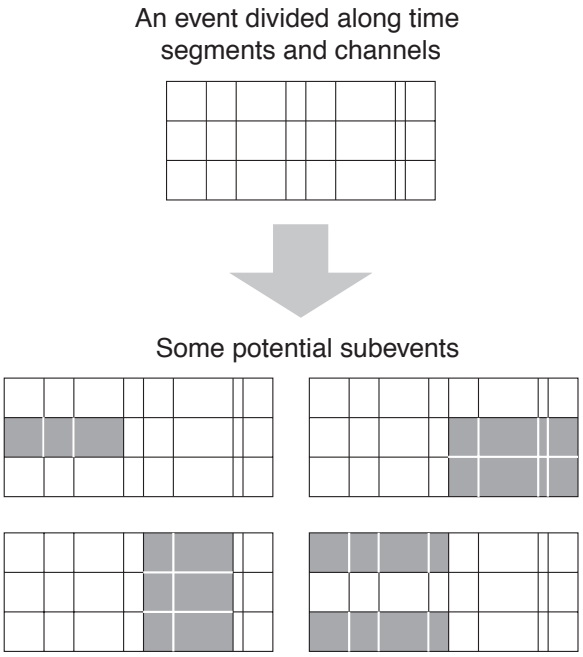


Figure 3-5: Subevents may be extracted from an event using the time-channel decomposition generated by the event segmenter. Each of the darkened regions represents examples of possible subevents extracted from the event at the top.

Subevent end-points coincide with event segmentation boundaries.

Subevents consist of any possible contiguous combination of segments, composed of any possible combination of channels.

Any L-subevent is potentially an instance of a linguistic unit of the target language. The set of all possible L-subevents derived from an L-event represents the complete set of hypotheses of linguistic units embedded within that L-event. Similarly, an S-event can be partitioned into a set of all possible S-subevents which may correspond to the semantics of words in the target language. Figure 3-5 illustrates this concept. At the top of the figure, an event is shown with three channels and eight time segments. Below, are four of many possible subevents derived from this event.

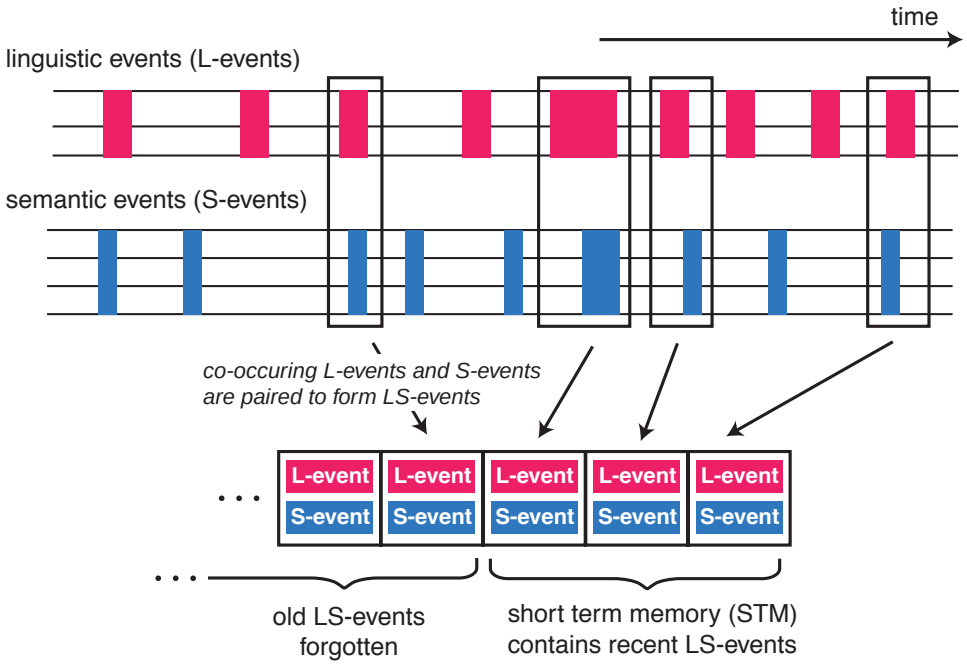


Figure 3-6: A co-occurrence filter is used to select linguistic events (L-events) and contextual events (S-events) which occur together. Co-occurring events are paired into LS-events and placed in short term memory (STM).

3.2.7 Co-occurrence Filtering

Infant directed speech usually refers to the immediate context [88]. Words and their meanings are often perceived together. In terms of CELL, we assume that L-events contain linguistic units whose reference may be found in co-occurring S-events. A natural bias to facilitate early lexical learning is to only attend to L-events and S-events which occur together. To facilitate this bias, S-events and L-events feed into a *co-occurrence filter* which detects events that overlap in time (Figure 3-6). When co-occurring events are detected, the filter generates an LS-event:

Definition 5 *An LS-event consists of an L-event paired with an S-event which overlap in time.*

The LS-event is the first structure in the model in which representations derived from linguistic and contextual signals are coupled.

LS-events are placed in a *short term memory (STM)* which maintains a complete record of recent LS-events. The STM is a first-in-first-out (FIFO) buffer. When a new LS-event is added, the oldest element of the STM is discarded. The size of the buffer depends on the implementation, but in general it is expected to be relatively small.

There are two related motivations for buffering incoming data through the STM. The first consideration comes from a cognitive modeling constraint. Humans have limited verbatim memory of sensory events, typically in the range of 7 ± 2 events [78]. The STM in CELL models this memory capacity. A second motivation for the STM is to conserve computational resources. CELL processes contents of the STM using an exhaustive search for recurring subevents. The search space grows factorially with the size of the STM, motivating a low upper bound on buffer size.

3.2.8 Recurrence Filtering

Infant directed speech is highly redundant [117]. A typical sequence might be, "Oh, look at the ball! The red ball! Does it go fast? Big ball!". Whole or partial phrase repetitions are common. Thus, we can expect that salient words will often recur in close temporal proximity. This observation motivates the next stage of processing. The recurrence filter searches for matching L-subevents and S-subevents across multiple LS-events in STM. The search is invoked each time a new LS-event arrives in STM.

To match subevents, a method of comparison must be established. For this purpose, a pair of distance measures, $d_L()$ and $d_S()$ are assumed to be known. To compare L-subevents, we use $d_L()$. Similarly $d_S()$ is used to compare S-subevents. These measures return a scalar value of dissimilarity which approximate perceptual distances that are computed by humans. The distance metrics operate on representations generated by the feature analyzers and are thus implementation dependent.

Both L-subevents and S-subevents may be grounded in any subset of linguistic

and contextual channels, respectively. The distance metrics only compare subevents which are grounded in the same set of channels. Subevents grounded in different channels cannot be matched, by definition. In the remainder of this chapter, however, we assume that all subevents are grounded in the same set of channels. This reduces cumbersome notation which would otherwise need to be carried at each step to indicate that distance metrics are only applied to subevents with matching channels. All processes which are presented in the remainder of this chapter may be applied to each group of subevents with matched channels without loss of generality.

The search for recurrence is exhaustive within the STM. Figure 3-7 provides pseudocode of the search procedure. The distance between each pair of L-subevents and S-subevents is compared to their respective thresholds t_L and t_S which determine when a pair of subevents match. These thresholds are set liberally so that many recurrency matches are produced at the expense of many false alarms. The next level of CELL is designed to filter out unwanted data.

When a match is found between two or more L-subevents and corresponding S-subevents, a representative L-subevent and S-subevent are extracted and placed in the next level of memory. The L-subevent and S-subevent pair forms a hypothesis of a possible linguistic unit and its corresponding meaning in the target language.

Figure 3-8 summarizes the process of recurrence filtering. Lexical candidates generated by the recurrence filter are stored in the *mid-term memory (MTM)*. The MTM is a buffer of lexical candidates being considered by the learner. Similar to the STM, the MTM is also a FIFO buffer of limited but significantly larger size.

3.2.9 Linguistic Units and Semantic Categories

CELL uses a model of linguistic units and semantic categories based on prototypes. Consider a linguistic unit in the target language. Each time an instance of this unit is produced it will be realized with natural variations, presumably centered around some idealized form. A simple model of the unit is to posit a central exemplar, or

```
/* Consider all pairs of LS-events in short term memory */
for each pair of LS-events in STM, LS_i and LS_j {

  /* Compare each pair of L-subevents in LS_i and LS_j */
  for each L-subevent in LS_i, L_i {
    for each L-subevent in LS_j, L_j{
      if d_L(L_i, L_j) < t_L then set L_match = TRUE
    }
  }

  /* Compare each pair of S-subevents in LS_i and LS_j */
  for each S-subevent in LS_i, S_i {
    for each S-subevent in LS_j, S_j{
      if d_S(S_i, S_j) < t_S then set S_match = TRUE
    }
  }

  /* check for matches of L-subevents and co-occuring S-subevents */
  if L_match = TRUE and S_match = TRUE
    then recurrent match found
  }
}
```

Figure 3-7: Pseudocode listing of the recurrence filter. An exhaustive search for recurrent L-subevents and co-occurring S-subevents is performed over the short term memory.

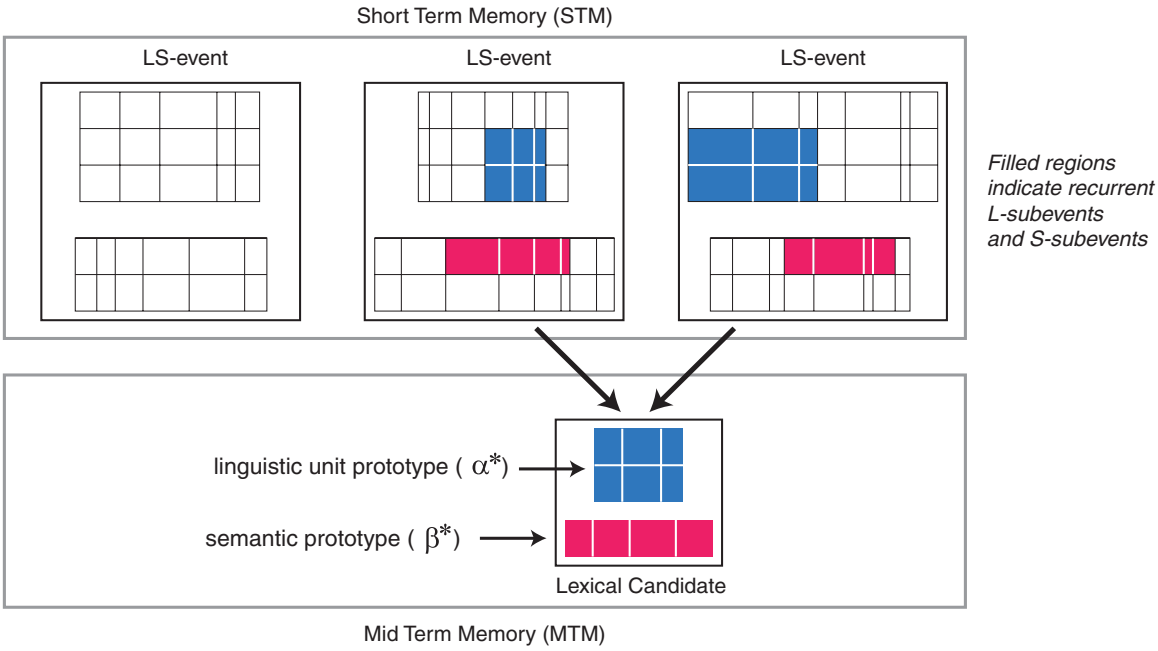


Figure 3-8: Lexical candidates are generated when recurring L-subevents and S-subevents are found in the STM.

prototype, and a error tolerance to allow for variation. Similarly, a semantic category may be modeled by a prototype and allowable error tolerance where the prototype specifies an ideal realization of the semantic category.

The connection between subevents, linguistic units, and semantic categories can now be made: L-subevents and S-subevents serve as prototypes of linguistic units and semantic categories, respectively. These notions are formalized by the following definitions:

Definition 6 *An L-unit, α, is a model of a linguistic unit in the target language.*

An L-prototype, α^*, is the prototype of the L-unit, α. An L-subevent may serve as an L-prototype.

An L-radius, δ^α, specifies the allowable deviation from an L-prototype.

An S-category, β, is a model of a semantic category.

An S-prototype, β^*, is the prototype of an S-category, β. An S-subevent may serve as an S-prototype.

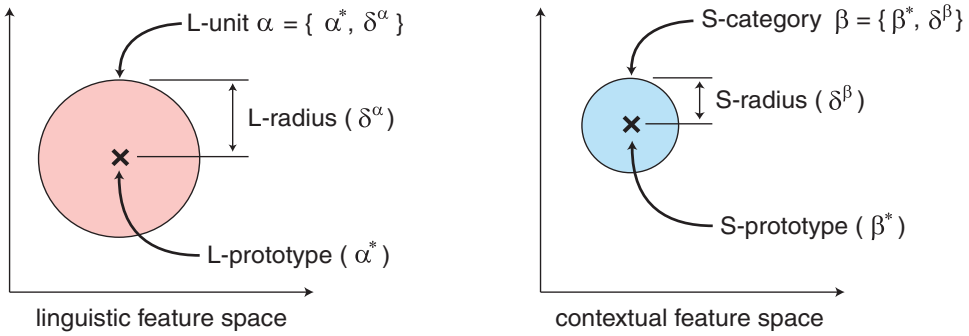


Figure 3-9: A lexical item consists of a linguistic unit (L-unit) and an associated semantic category (S-category).

An S-radius, δ^β, specifies the allowable deviation from an S-prototype.

A lexical item is the union of an L-unit and an S-category.

A prototype and radius define a range of inputs which may be grouped together for the goal of forming lexical items. A novel subevent is said to *match* an L-unit or S-category if it is within the radius of the prototype:

Definition 7 *An L-subevent, l, matches an L-unit, α, if $d_L(\alpha^*, l) \leq \delta^\alpha$.*

An S-subevent, s, matches an S-category β if $d_S(\beta^*, s) \leq \delta^\beta$.

These definitions are illustrated in Figure 3-9. Each plot represents an abstract space in which linguistic units and semantic categories may be defined. Any point in the two-dimensional space corresponds to an instance of a linguistic unit or semantic category. The distance between two points in each space is determined by the distance metrics $d_L()$ and $d_S()$. The prototypes define the center of each model, and the radius determines the scope of the model. A subevent matches an S-category or L-unit if it falls within the model's radius.

The stages of CELL presented thus far can be viewed as a series of attentional filters. A continuous stream of sensory input is processed by feature analyzers, event detectors, a co-occurrence filter and a recurrence filter to produce L-prototypes and corresponding S-prototypes which may lead to the formation of a lexical item.

Definition 8 *A lexical candidate is a hypothesized exemplar of a lexical item. It is composed of an L-prototype and a corresponding S-prototype.*

3.2.10 Creating Lexical Items

The recurrence filter identifies lexical candidates based on repeated observations of similar subevents in close temporal proximity. Resulting lexical candidates are placed in the MTM. The next stage of processing combines evidence from multiple lexical candidates to create hypotheses of lexical items. To create lexical items, CELL first calculates an optimal L-radius and S-radius for each {L-prototype, S-prototype} candidate in MTM. A measure of goodness based on mutual information is generated for each hypothesized lexical item in the process of radii optimization. The process described in this section is invoked each time a new lexical candidate is added to MTM[1].

Let us assume that the MTM contains N lexical candidates $m_1, m_2, \ldots, m_N$. For each candidate m_i we can denote a set of associated terms by adding an index i to the notation introduced earlier:

α_i^*	L-prototype in m_i
δ_i^α	L-radius associated with α_i^*
β_i^*	S-prototype in m_i
δ_i^β	S-radius associated with β_i^*
$\alpha_i = \{\alpha_i^*, \delta_i^\alpha\}$	L-unit derived from m_i
$\beta_i = \{\beta_i^*, \delta_i^\beta\}$	S-category derived from m_i
$\{\alpha_i, \beta_i\}$	Lexical item derived from m_i

Let us choose a candidate from MTM at random and designate it as a reference point called m_{ref}. This reference candidate can be evaluated as the basis for forming a lexical candidate. In the model, each lexical candidate in MTM is treated as the

[1]Processing begins once the MTM is filled.

reference and evaluated in turn. Based on this reference candidate, we wish to derive an L-unit α_{ref} and a S-category β_{ref}. The method of calculation of the radii δ^{α}_{ref} and δ^{β}_{ref} are described later in this section.

Each lexical candidate m_i may be thought of as an experiment which determines the outcome of two random variables when compared to m_{ref}:

$$L = \begin{cases} 0 & if\ d_L(\alpha^*_{ref}, \alpha^*_i) > \delta^{\alpha}_{ref} \\ 1 & if\ d_L(\alpha^*_{ref}, \alpha^*_i) \leq \delta^{\alpha}_{ref} \end{cases} \tag{3.1}$$

$$S = \begin{cases} 0 & if\ d_S(\beta^*_{ref}, \beta^*_i) > \delta^{\beta}_{ref} \\ 1 & if\ d_S(\beta^*_{ref}, \beta^*_i) \leq \delta^{\beta}_{ref} \end{cases} \tag{3.2}$$

These binary variables L and S indicate whether a lexical candidate m_i matches the L-unit α_{ref} and the S-category β_{ref}, respectively. The distance metrics $d_L()$ and $d_S()$ depend on the implementation of the model and are assumed to be innate (in Chapter 4 we provide examples of these metrics for audio and visual input).

Mutual information is used to evaluate the degree of cross-channel structure captured by pairing an L-unit and S-category. Mutual information is a measure of the reduction in uncertainty of one variable due to knowledge about a second variable [27]. We assume that the mutual information between L and S will be high if $\{\alpha_{ref},\beta_{ref}\}$ corresponds to an actual lexical item of the target language. This assumption is based on the fact that infant-directed speech often refers to the immediate context [117]; knowledge about the existence of a word will greatly reduce uncertainty about the presence of the word's referent, and vice versa.

To simplify notation, we denote the event $L = i$ with l_i and $S = j$ with s_j. The mutual information between L and S is defined as:

$$I(L;S) = \sum_i \sum_j P(l_i, s_j) \log \left[\frac{P(l_i, s_j)}{P(l_i)P(s_j)} \right] \tag{3.3}$$

The summations are over 0 and 1 for both i and j. Mutual information is a

symmetric measure since $I(L;S) = I(S;L)$.

The probabilities in Equation 3.3 are estimated using relative frequencies:

$$P(L=i) = \frac{|l_i|}{N} \tag{3.4}$$

$$P(S=j) = \frac{|s_j|}{N} \tag{3.5}$$

$$P(l_i, s_j) = \frac{|l_i, s_j|}{N} \tag{3.6}$$

N is the number of lexical candidates in MTM. The vertical bars denote the count operator. For example $|l_i|$ is the number of MTM items for which $L = i$, and $|l_i, s_j|$ is the number of candidates for which $L = i$ and $S = j$. Probability estimates derived from relative frequencies will be noisy when counts are small, requiring some form of smoothing.

3.2.11 Maximizing Cross-Channel Mutual Information

We now turn to the question of how to calculate the values of the radii δ^{α}_{ref} and δ^{β}_{ref}. The definition of L and S in Equation 3.1 and 3.2 depend on the radii δ^{α}_{ref} and δ^{β}_{ref}. In turn, $I(L;S)$ depends on L and S and thus is a function of the radii as well. We can write that $I(L;S) = f(\delta^{\alpha}_{ref}, \delta^{\beta}_{ref})$ for some function $f()$. To determine the values of the radii, CELL performs a search over all values of $\delta^{\alpha}_{ref}, \delta^{\beta}_{ref}$ to find a combination of radii which maximizes the mutual information between the linguistic unit and semantic category:

$$I_{\max}(L;S) = \max_{\delta^{\alpha}_{ref}, \delta^{\beta}_{ref}} I(L;S) \tag{3.7}$$

This maximization in Equation 3.7 is illustrated by a simple example in Figure

3-10. Let's assume that the MTM contains 27 lexical candidates. One of them is used to define an L-prototype and S-prototype (labeled $\alpha^*$and β^* in the figure) and the remaining candidates are labeled *a*-*z*. Each graph in the left column plots the L-prototypes in MTM. The S-prototypes are plotted on the right. The top left graph shows a circle centered on α^* to indicate a L-unit defined by α^* and a small L-radius, δ^α. The subset {a,i,f,g,q} match this L-unit. Similarly, the circle in the top right plot indicates a S-category defined by β^* and its δ^β. The subset {f,i,j,h,l,m,q,r,s,t,x,z} match this S-category. To aid the reader, the elements common to the L-unit and S-category, ({f,q,i}, are shown in bold italics. In the next two rows, the L-radius is expanded to show its effect on mutual information (the S-radius is held constant). In this example, mutual information is highest for the mid-sized L-radius.

The search must compute $I(L;S)$ for each combination of L-radius and S-radius and choose the combination which maximizes the mutual information. In the example in Figure 3-10, only the L-radius was varied (with large steps). A complete search would also co-vary the S-radius to find a global maximum. If the MTM has N candidates, the search must compute the mutual information of $O(N^3)$ configurations (for each of N candidates, starting from a size of zero, each radii may be stepped through $N-1$ values to include an additional candidate in each step).

The search process is used to determine optimal radii for each lexical candidate in MTM. For each hypothesized lexical item which results, $I_{max}(L;S)$ is found. $I_{max}(L;S)$ will be high for lexical candidates which successfully capture structure across linguistic and contextual channels. $I_{max}(L;S)$ will be low for candidates which don't generalize well to the contents of the MTM.

Mutual Information Filter

The next stage of CELL is referred to as the *mutual information filter* (Figure 3-11). This filter selects lexical candidates for which $I_{max}(L;S) > T_{MI}$ where T_{MI} is referred to as the mutual information threshold. The selected lexical candidates are coupled

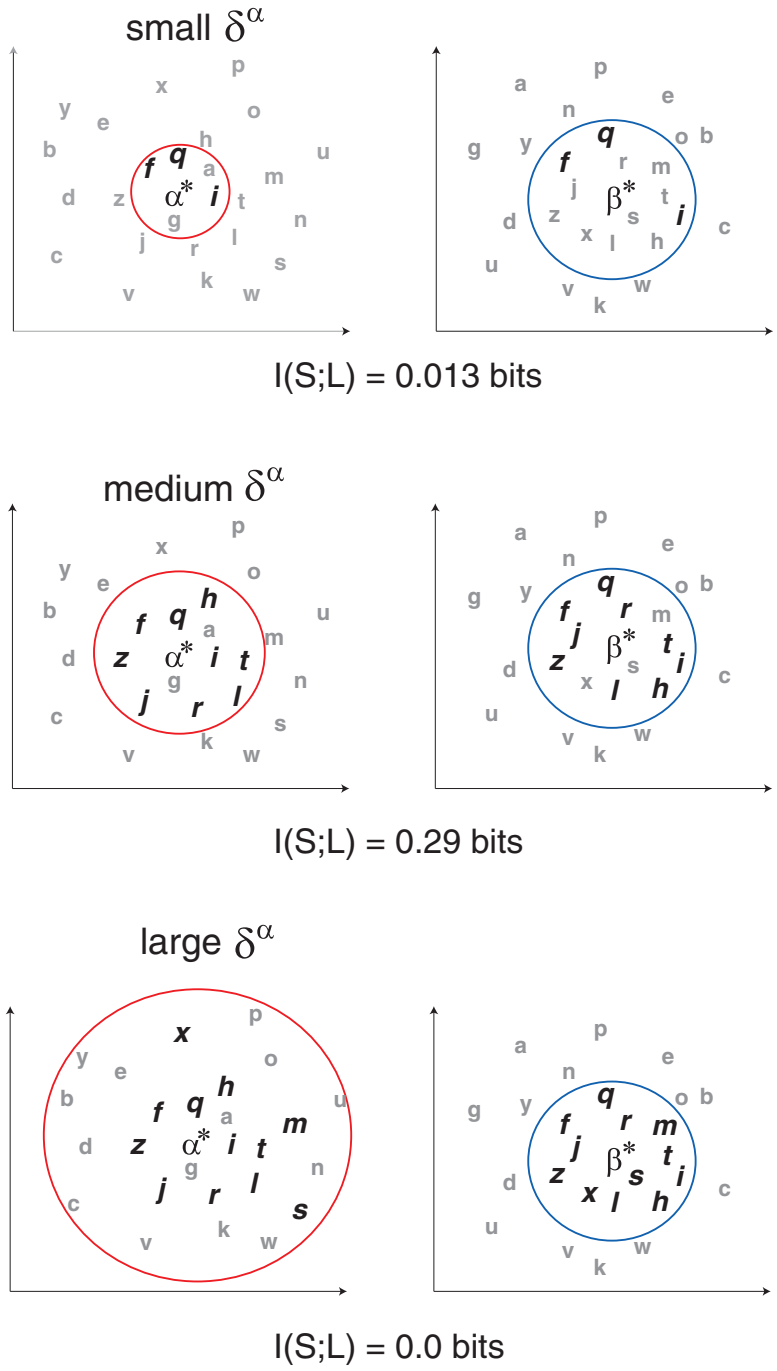


Figure 3-10: Illustration of the search for the optimal radii δ^α and δ^β to maximize mutual information, $I(L;S)$. For a mid-sized L-radius, $I(L;S)$ is greatest.

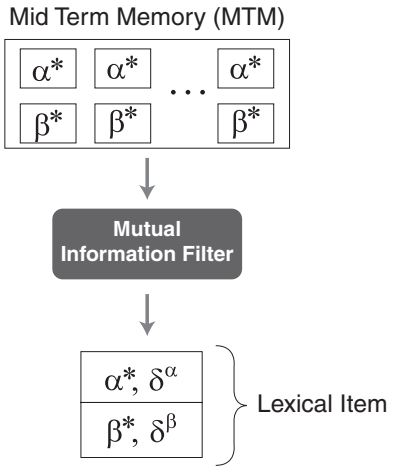


Figure 3-11: Lexical items are selected using a mutual information filter. For an item to be selected, the mutual information between the associated L-unit and S-category must exceed a selection criterion.

with their optimal radii to form lexical items. Lexical items are stored in long term memory (LTM). In contrast to STM and MTM which are FIFO buffers, LTM is a long term repository of information.

Once a lexical item has been created, the candidates which contributed to its formation are removed from MTM by a garbage collection process. This includes the MTM item which serves as the prototype of the lexical item (m_{ref}), and all lexical candidates which match both the L-unit and S-category defined by m_{ref}.

An important property of the lexical acquisition process in CELL is the ability to combine cross-channel evidence. The process we have described effectively combines different similarity metrics via the mutual information search procedure. By using mutual information to look for structure across channels, the model avoids the difficult problem of directly combining different similarity metrics [104].

Extensions of CELL for learning the value of T_{MI} and for removing poor LTM items based on environmental feedback are discussed in Section 3.3.

3.2.12 Summary

CELL is driven by a continuous flow of sensory input. Sensory signals are processed by feature analysers to extract salient aspects of the input. Short and mid term memory serve to buffer partially analysed data which are likely to lead to the formation of lexical items. The flow of data through CELL is summarized below as a review of the acquisition process.

- Input originates from a set of sensors.
- Feature analysis extracts representations of salient aspects of the environment which are grouped into linguistic and contextual channels.
- Event detection packages continuous feature streams into natural chunks of data.
- Event segmenters unpack events, identifying the finest granularity boundaries of analysis within events.
- The co-occurrence filter detects L-events and S-events which occur together in time. Resulting LS-events are placed in STM, a short term FIFO memory buffer. The STM provides limited rote sensory memory of recent salient sensory events.
- The recurrence filter finds {L-subevent, S-subevent} pairs which occur multiple times within the STM. Representations of recurrent subevent pairs are placed in MTM, a larger FIFO buffer. These subevent pairs are hypotheses of prototypes of potential lexical items. The recurrence filter searches the contents of STM each time a new LS-event is generated by the co-occurrence filter.
- Optimal L-radius and S-radius values are calculated for each lexical candidate in MTM. The maximized mutual information is recorded for each candidate. The MTM is re-analyzed each time the recurrence filter adds a new candidate.

- Hypothesized lexical items with high mutual information are stored in LTM.

3.3 Extensions

This section presents several extensions of CELL. These extensions are not central to the thesis but provide a basis for integrating CELL with other theories of language learning and cognition.

3.3.1 Recognizing Novel Input

Items in LTM may be used to recognize novel input to make cross-channel associations. When a novel input utterance contains a linguistic unit which is stored in the acquired lexicon, the associated semantic category may be retrieved. In figure 3-12, a novel L-event is detected using the feature analysis, event detection, and event segmentation components described earlier. A search procedure then looks for matches between L-subevents in the incoming L-event and previous L-units stored in LTM. This procedure is similar to the search used in the recurrence filter in that it considers all possible subevents in the L-event. Unlike the recurrence filter, each L-prototype in the LTM is compared to L-subevents without attempts to unpack L-prototypes. In situations where more than one lexical item matches the same subevent, the match with the smallest distance is selected. A similar approach is used to index into lexical items in LTM based on an incoming S-event (Figure 3-13). When an instance of a semantic category is observed in the environment, the associated linguistic unit is retrieved.

3.3.2 Top-Down Feedback

The core CELL model processes data in a bottom-up fashion. Top-down feedback may be added to accelerate learning. One type of feedback is illustrated in Figure 3-14. An additional stage has been added to CELL prior to recurrence filtering. Recall that

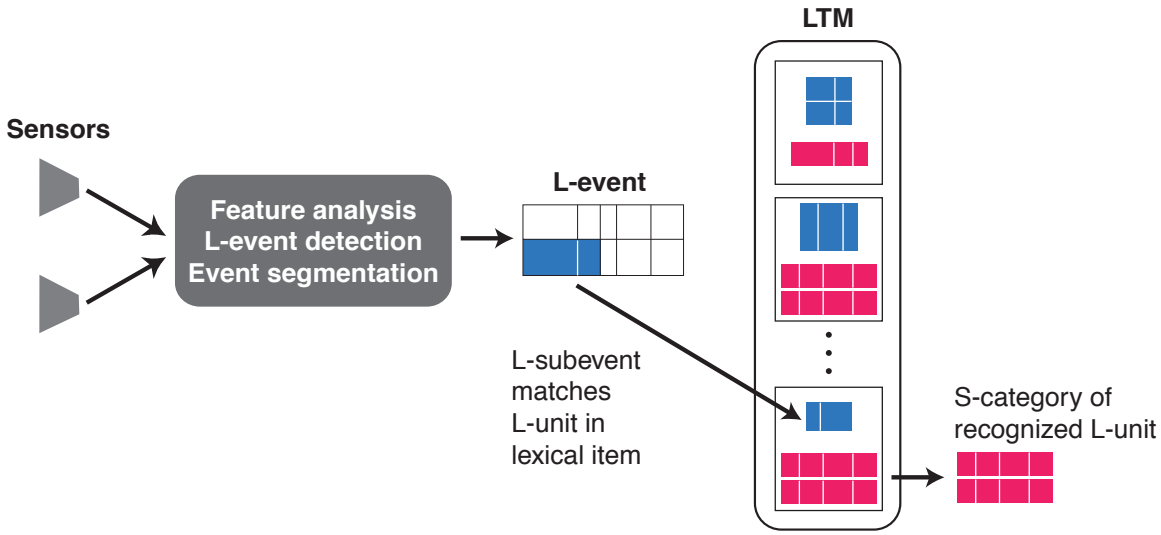


Figure 3-12: Finding a linguistic unit which corresponds to a novel S-event.

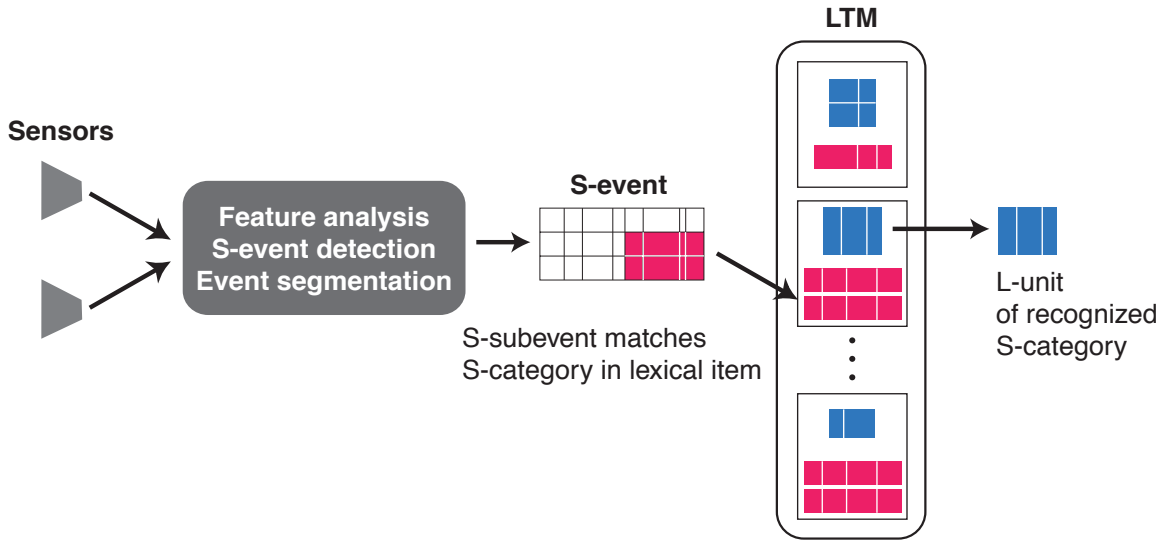


Figure 3-13: Finding a semantic category which corresponds to a novel L-event.

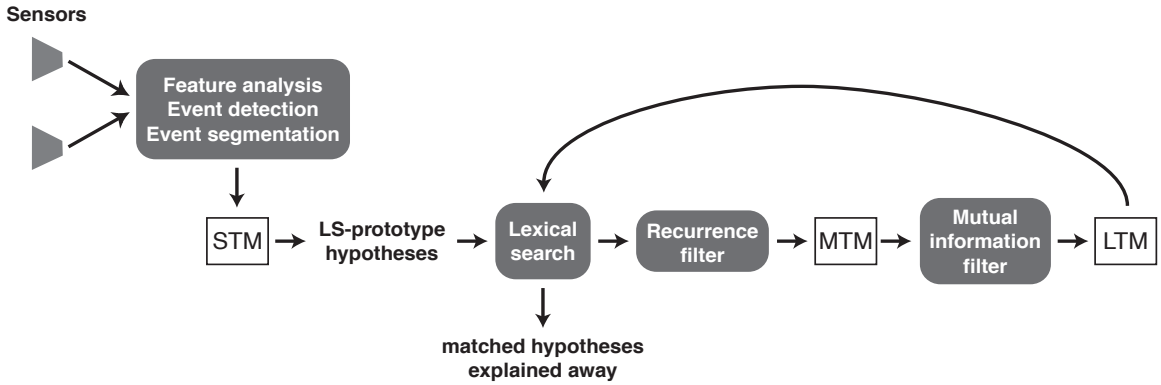


Figure 3-14: The lexical search component looks for matches between LS-hypotheses and existing lexical items in LTM. Matched items are not processed further.

the recurrence filter performs a search for matching {L-subevent, S-subevent} pairs in STM. The new stage first checks if each {L-subevent, S-subevent} pair matches an existing lexical item. To match, both the L-subevent and S-subevent must match the same lexical item's L-unit and S-category. Matches are "explained away" by the contents of the LTM and are not processed further. As the learner's lexicon grows, this stage will increasingly filter input which is consistent with prior knowledge of the language.

3.3.3 Clustering Lexical Items

In Section 3.2.8 we presented a model of a lexical item as a single prototype and constant radius of variation. This model assumes a rather homogeneous distribution of samples around a single prototype. For the current discussion, we may refer to these as elementary lexical items. In general, we might expect more complex distributions which are not well modeled by a single elementary lexical item. Elementary lexical items may be combined to form conglomerate lexical items which model complex distributions.

Consider the following clustering algorithm:

```
for each pair of lexical items in LTM, i and j {
    if d_L(α_i^*, α_j^*) ≤ δ_i^α or d_L(α_i^*, α_j^*) ≤ δ_j^α {
        if d_S(β_i^*, β_j^*) ≤ δ_i^β or d_S(β_i^*, β_j^*) ≤ δ_j^β {
            Cluster items i and j
        }
    }
}
```

In this algorithm, items are clustered to form *conglomerate lexical items*. The distance between a subevent and a conglomerate item is defined as the average distance between the subevent and each prototype in the conglomerate item.

For two items to be clustered, the L-prototype of one item must match the other's L-unit, and the S-prototype of one item must match the other's S-category. An example of this clustering process is illustrated by Figure 3-15. We start with two elementary items of the sort generated by CELL. The middle box superimposes the two items and we see that both L-units and S-categories overlap. The bottom frame shows the contour formed by the merger of models.

An interesting situation arises when only the linguistic or semantic model of two lexical items match (Figures 3-16 and 3-17). These branching structures represent primitive forms of synonyms and homonyms. Additional learning mechanisms could detect such structures and build higher level representations of relations between lexical items.

As relationships between items are established, structured networks of lexical items may emerge. In Figure 3-18, several clusters of items have been formed. This network helps visualize an advantage of cross-modal structure. Consider the large cluster of S-categories in the upper right corner of the figure. Using the semantic distance metric $d_S()$ these categories are close together and thus confusable. Their associations to easily separable linguistic units lets the learner keep the models separate. Similarly, linguistic units which are confusable are pulled apart by their semantic

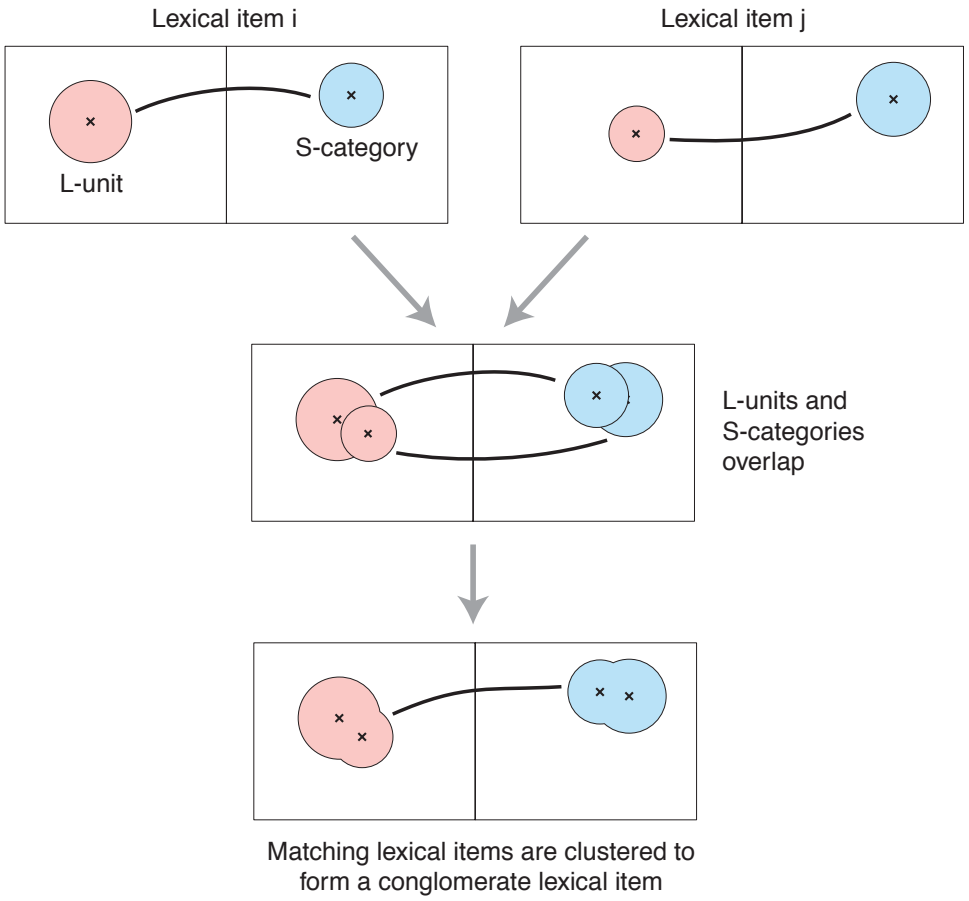


Figure 3-15: Lexical items with matching L-units and S-categories are merged to form a conglomerate lexical item.

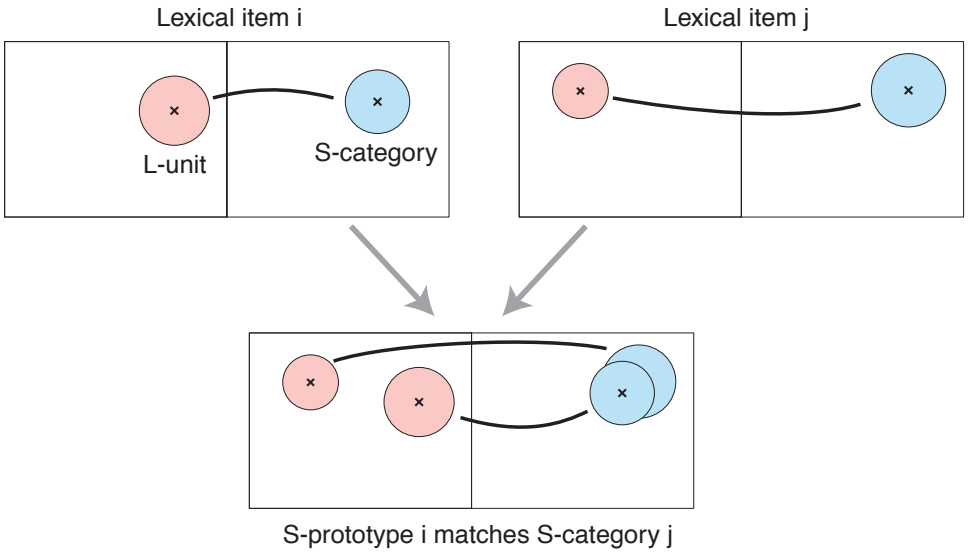


Figure 3-16: Two lexical items have overlapping S-categories but distinct L-units, suggesting the existence of a synonym.

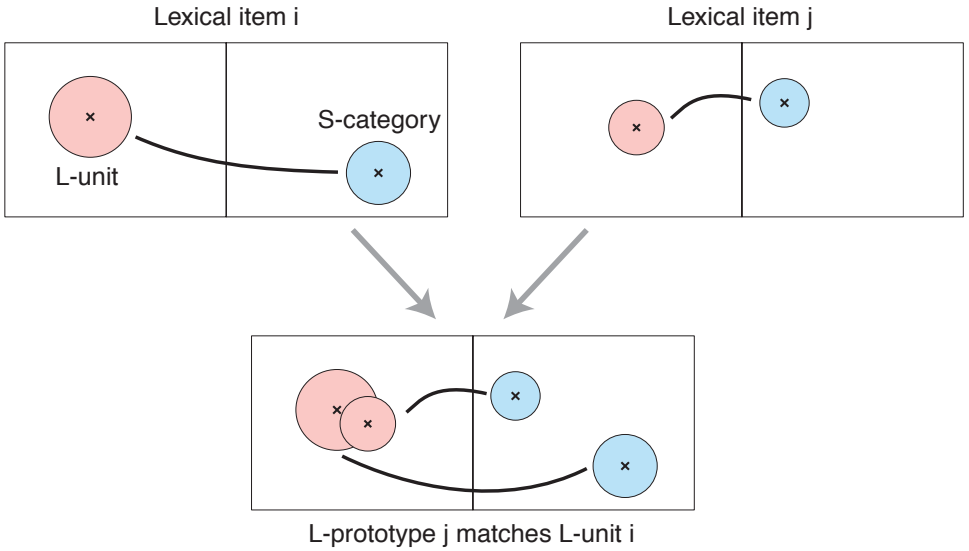


Figure 3-17: Two lexical items have overlapping L-units but distinct S-categories, suggesting the existence of a homonym.

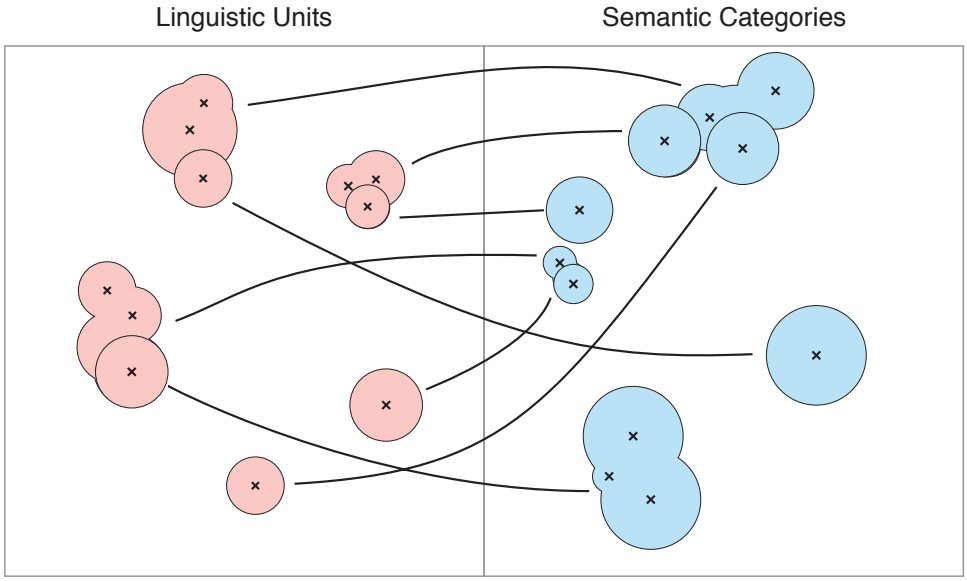


Figure 3-18: A network of associations between linguistic and semantic components of lexical items.

associations. The merging algorithm presented earlier is based on the premise that if there is significant overlap in both linguistic and semantic space, then the two items must be variations of one underlying lexical item of the target language.

3.3.4 Word Classes and Syntax Acquisition

Problems of syntax learning are beyond the scope of this thesis. However, syntax is clearly an important aspect of later stages of language development. We believe that any model of lexical acquisition must provide a basis for the learner to begin acquiring syntax. In this section we sketch the relation between CELL and one theory of syntactic development, semantic bootstrapping.

The semantic bootstrapping hypothesis posits that the language learner uses semantic categories to seed syntactic categories [89, 48]. For example, perceptually accessible categories such as objects and actions would seed the syntactic classes of nouns and verbs. Once these seed categories have been established, input utterances are used to deduce phrase structure (in combination with constraints from a

Universal Grammar). In turn, the phrase structure can be used to interpret input utterances with novel words. Distributional analysis can then be used to expand syntactic classes.

The contextual channels in CELL provide a basis for establishing word classes which seed syntactic categories. For example, consider a configuration of CELL with two visually-grounded context channels: shapes of objects, and the motion of objects. Any word which is grounded in the shape channel would belong to Class A, and any word grounded in the motion channel would belong to Class B. These classes of words could seed syntactic classes corresponding to nouns and verbs in the target language.

3.3.5 Environmental Feedback

In this section we discuss a framework for assigning confidence values to lexical items, and for setting the mutual information filter threshold, T_{MI} (T_{MI} was introduced in Section 3.2.11).

The learner begins with some initial T_{MI}. If the threshold is too low, unreliable lexical items will be added to LTM. If the threshold is too high, valid lexical items will not be extracted from the MTM. Figure 3-19 proposes a framework for adjusting T_{MI} and lexical item confidence levels based on environmental feedback.

A set of goals motivate the learner to take actions in the world. The action selection component chooses actions which optimize goal satisfaction. As shown in the figure, action selection may be influenced by lexical items in LTM. For example, the learner might produce the name of a desired object to enlist the help of a caregiver for obtaining that object.

As an action is executed, the learner monitors feedback from the environment. Feedback is used to adjust confidence levels of individual lexical items and the global parameter T_{MI}. The confidence level of an item increases if using it results in positive feedback (i.e., the learner's goal is achieved). Negative feedback (i.e., goal is not obtained) causes the confidence level to decrease. Continued negative feedback results

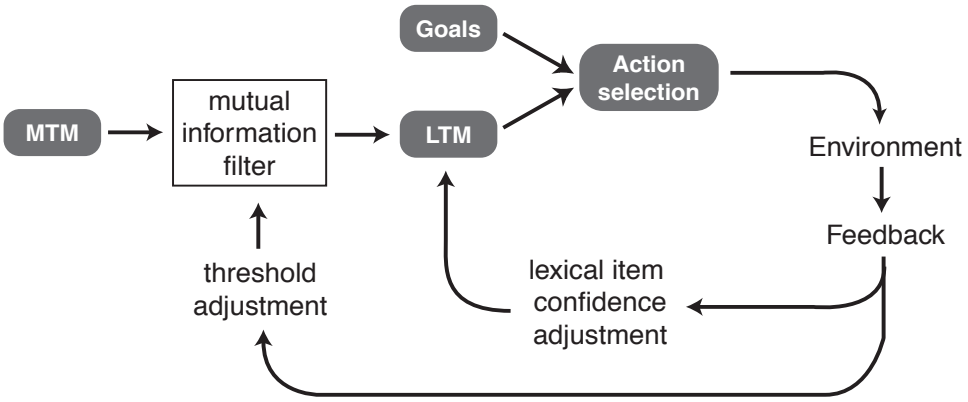


Figure 3-19: Feedback from the environment due to success or failure to obtain a goal is used to adjust confidence levels of lexical items, and to adjust the mutual information threshold.

in the removal of an item from LTM.

On a slower time scale, the learner monitors overall levels of confidence in the LTM. If confidence in most current lexical items grows with feedback, T_{MI} is reduced to let in more items from MTM. If actions based on the LTM lead to significant negative feedback, T_{MI} is increased to insure that future items encode greater cross-modal mutual information.

Chapter 4

An Implementation of CELL with Audio-Visual Input

CELL has been implemented for the domain of shape and color name learning in a spoken English environment. The system implements all components of the core model, and some extensions. It is grounded in microphone and camera input and uses speech recognition and computer vision techniques to process sensory input. This chapter provides implementation details of the system. Chapters 5 and 6 discuss evaluations and applications of the implementation.

We restate the three problems of early lexical acquisition in the context of the current implementation:

Acoustic Unit Discovery The linguistic channel is derived from microphone input. A person (the teacher or caregiver) is expected to speak natural and fluent English. From this continuous multiword input, the system must segment speech and discover acoustic units which correspond to English words.

Color and Shape Categorization The contextual channels carry representations of object shapes and colors derived from camera input. The second problem is to discover color and shape categories which correspond to the semantics of

English shape and color terms.

Speech-to-Visual Association Inference The third problem is to establish associations between acoustic units and corresponding shape and color categories.

Figure 4-1 summarizes the components of the implementation in terms of the CELL architecture. Input is provided by two sensors. A color CCD video camera provides images of objects presented to the system, and a microphone senses acoustic speech signals. Three channels of features are extracted from these sensors. A phoneme analyzer produces time-varying estimates of 39 English phoneme probabilities. These 39 features are grouped to form the linguistic channel[1]. A visual analyzer detects objects in the scene. When an object is present, two contextual channels of features which represent the object's shape and color are extracted. Several innate mechanisms are available to the system prior to learning. These mechanisms are described as we proceed through the chapter, and are summarized in Section 4.9.

Table 4.1 provides a summary of the implementation of each component of CELL.

4.1 Contextual Channels

The system was developed to learn spoken names of shape and color categories. With this goal in mind, a visual processing system has been implemented to:

- Acquire images of a stationary object from multiple perspectives.
- Extract a representation of object color which is invariant to changes in illumination.
- Extract a representation of object shape which is invariant to changes in scale and in-plane rotation.

[1] Although CELL is capable of analyzing multiple linguistic channels, this implementation includes the phonetic channel alone.

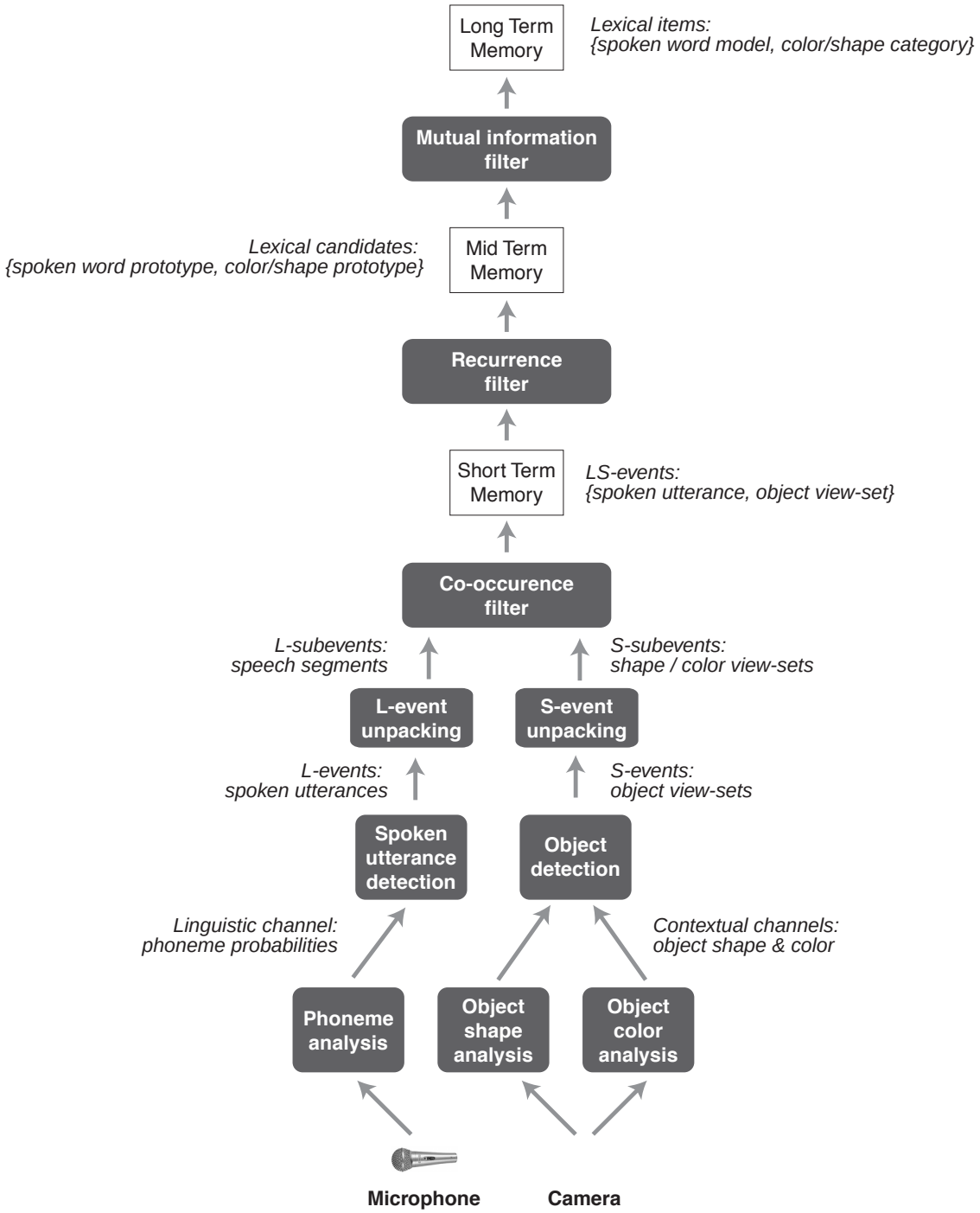


Figure 4-1: Overview of an implementation of CELL for color and shape name learning based on acoustic and visual sensory input.

Table 4.1: Summary of CELL implementation.

Structure	**Implementation**
Linguistic Channel	Phoneme probabilities extracted from microphone input
Contextual Channels	Channel 1: color of any object in view of the camera; Channel 2: rotation and size invariant representation of object shape
L-event	A pause delimited spoken utterance
S-event	Representations of an object's shape and color viewed from multiple perspectives
LS-event	{L-event, S-event}
L-prototype	A speech segment extracted from an L-event
S-prototype	Representations of either an object's shape or color viewed from multiple perspectives
Lexical Candidate	{L-prototype, S-prototype}
L-Radius	Allowable acoustic distance from L-prototype
S-Radius	Allowable distance from an S-prototype using a visual distance metric
L-Unit	{L-prototype, L-radius}
S-Category	{S-prototype, S-radius}
Lexical item	{L-unit, S-category}

To acquire images of an object from multiple perspectives, visual input to the system is provided by a camera mounted on a four degree-of-freedom (DOF) robotic platform as shown in Figure 4-5. The robot is essentially an armature with servo-driven joints which enable it to actively direct the position of the camera. This robot has been used both as an active image capture device (Chapter 5) and, after adding several animation features, as the embodiment of a life-like human-computer interface (Chapter 6).

4.1.1 Image Processing

Three-dimensional objects are represented using a view-based approach [120, 82, 94, 109]. In this approach, an explicit three dimensional model is not recovered. Instead, multiple two-dimensional images of an object are captured from multiple perspectives and grouped to collectively form a model of the object. Figure 4-2 shows the stages of visual processing which are used to extract representations of object shapes and colors. The result is a histogram representation of shape that is invariant to changes in scale and in-plane rotation. The histogram representation of color is invariant to changes in illumination.

Foreground / Background Segmentation

The video signal from the CCD camera is sampled at a resolution of 160x120 pixels at a rate of 10Hz. Each pixel is represented by red, green and blue color components denoted R, G, B. To eliminate variation due to illumination, the chromaticity coordinates [49] or illumination-normalized color of a pixel may be computed by:

$r = \frac{R}{(R+G+B)}$
$g = \frac{G}{(R+G+B)}$
$b = \frac{B}{(R+G+B)}$

After illumination normalization, only two free parameters are necessary to represent the chromaticity of a pixel. In our system, images are characterized by the

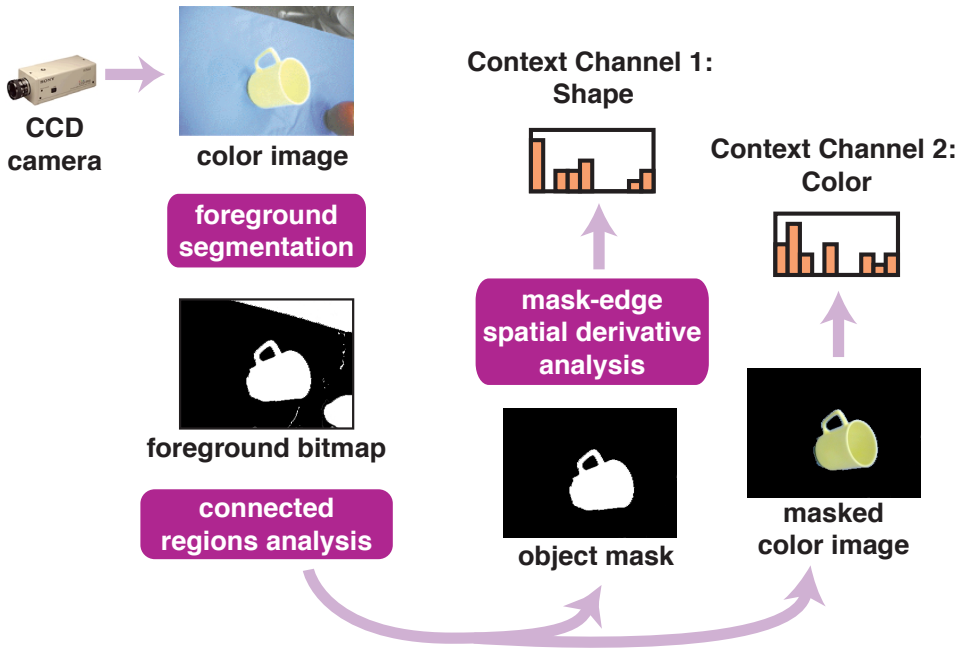


Figure 4-2: Extraction of object shape and color channels from a CCD camera.

normalized red and green components.

Foreground/background separation is simplified by assuming a relatively uniform background color. A Gaussian model of the normalized background color is computed using a set of sample background images. We have found 10 images of the background taken from various perspectives to be sufficient for establishing a baseline. The mean and co-variance of the average background pixel is estimated from these 10 images. For a novel image, the probability that each pixel was generated by the background model was thresholded to classify the pixel as foreground or background. The threshold was set to produce clean object masks.

Object Detection

Connected regions analysis is applied to binary foreground / background images. Connected regions analysis identifies and uniquely labels each set of pixels in a binary image which are connected to each other by a path of pixels with binary value 1 [55, page 28]. A scene which contains a single foreground object will often result in an

image with multiple connected foreground regions due to noise and the presence of multiple objects. For example in Figure 4-2 the foreground map contains three large connected regions and several small ones due to noisy segmentation.

In the robotic setup, non-target objects often come into view along the periphery of the image. The visual system must thus select the connected region which corresponds to the target object. This selection is guided by the assumptions that the object will be of some minimum size, and that it will be located near the center of the image. The object is kept centered by active camera control as described in Section 4.1.2.

In our implementation of connected regions analysis, the binary image is scanned left-to-right, line by line. If a candidate pixel has binary value 1, the value of its four neighbors (to its left, top-left, top-center and top-right) are checked. If any of those pixels are set to 1, the algorithm uses its label for the current pixel; otherwise, the pixel is assigned a new label. After the entire image has been scanned, if neighboring active pixels are found with different labels, the labels are merged into a single class. This final step produces a set of connected regions, each assigned a unique label. The centroid of each connected region is computed and the region whose centroid is closest to the center of the image is selected as the object in the scene. This region is denoted $O = \{(x,y)_1, (x,y)_2, \ldots, (x,y)_M\}$ and consists of M pixel locations.

The above description assumes that the target object is in view. This is not always the case[2]. To decide whether an image contains an object, three criteria must be met. First, the centroid of the selected region must be in the center of the image within a predefined maximum tolerable error. The error tolerance is determined by the accuracy of the robot armature and was set to 38 pixels. Second, the area of the connected region, M, must exceed some minimum threshold to reject noise due to poor background models. This threshold was set empirically to 100 pixels. Third, objects must be completely in view. To reject clipped views, any object mask which

[2]Chapter 6 describes a real-time application of this system in which a person can place and remove objects interactively.

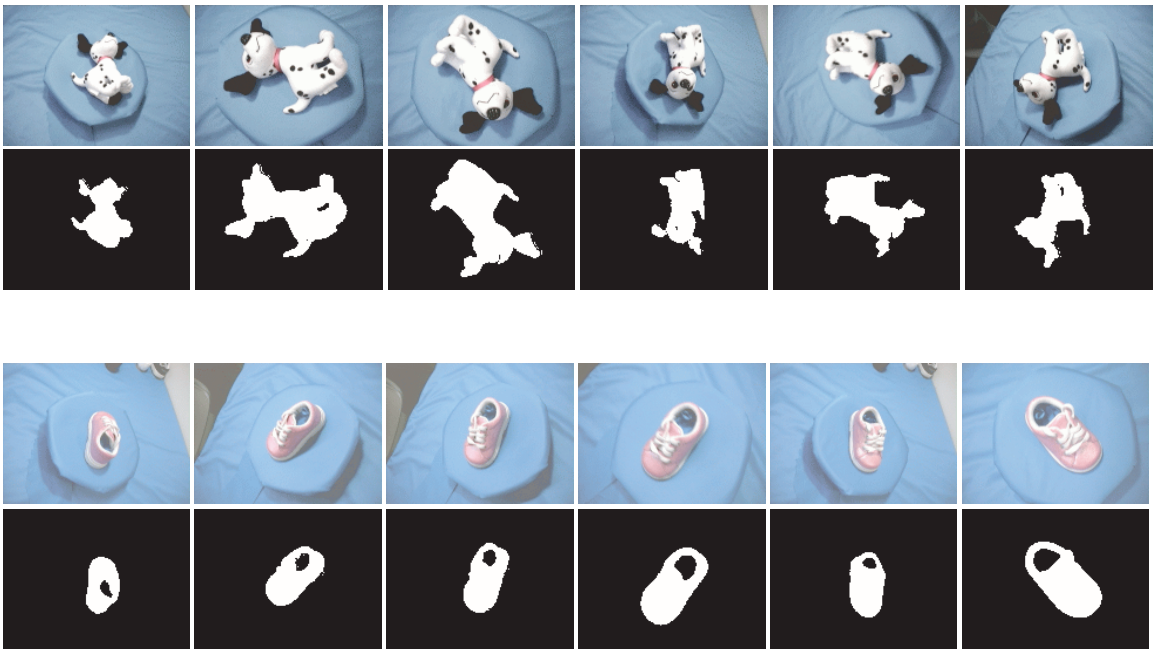


Figure 4-3: Sample images of a toy dog and shoe. Below, the corresponding object mask.

includes pixels at the edge of the image is rejected. Several sample images of objects and their corresponding masks are shown in Figure 4-3. The images of a toy dog and a shoe were taken with the robot-mounted camera. The interior of the shoe was similar in color to the background and is thus not part of the object mask. Below each original image, the foreground image after connected region analysis is shown. Each image is the result of a change of position of the camera and the turntable.

Object Color and Shape Representation

Object color and shape is represented using a histogram approach based on the work of Schiele and Crowley [109]. Schiele and Crowley have demonstrated that histograms of local image features are a robust representation for visual object recognition. In our implementation, a two-dimensional color histogram, H_c, is generated by accumulating (r, g) values for each pixel specified by the region O. The normalized red and green chromaticity values are divided into 8 bins leading to an $8 \times 8 = 64$ bin histogram.

The shape of an object is also represented using a two-dimensional histogram, H_s. To compute this histogram, the system performs the following steps:

1. Locate all pixels which are at the edges of the object. A pixel is defined as an edge point if it is part of the object mask but one or both horizontally adjacent pixels are not part of the mask.

2. Estimate π_i, the angle of the tangent to the mask edge at each edge pixel i.

3. For each pairwise combination of edge pixels i, j:

   - Compute the Euclidean distance, d_{ij}, between the pixels, normalized by the mean distance between all pairs of edge pixels in the object.
   - Compute relative angle between edges $\delta_{ij} = |\pi_i - \pi_j|$

4. Accumulate a two-dimensional histogram of (d_{ij}, δ_{ij}) for all pairwise combinations of edge pixels. Both the inter-pixel distances and the relative angles are divided into 8 bins so that H_s is also composed of $8 \times 8 = 64$ bins.

This representation of shape is invariant to changes in scale since the inter-pixel distances, d_{ij}, are normalized by the size of the object. The representation is also invariant to in-plane rotation of the object since only relative angles are stored in the histogram.

To give insight into these representations, we have generated images of several histograms. Figure 4-4 shows four images, their corresponding object masks, and the resulting color and shape histograms. The symmetrical shape of a ball leads to a near diagonal region of activation in the shape histogram (top right). The shoe has circular regions which lead to diagonal elements in shape as well. The diagonal activation from top-left towards bottom-right is due to the inner cavity of the shoe. Also, notice similarities between the more complex shape histograms of the two dogs.

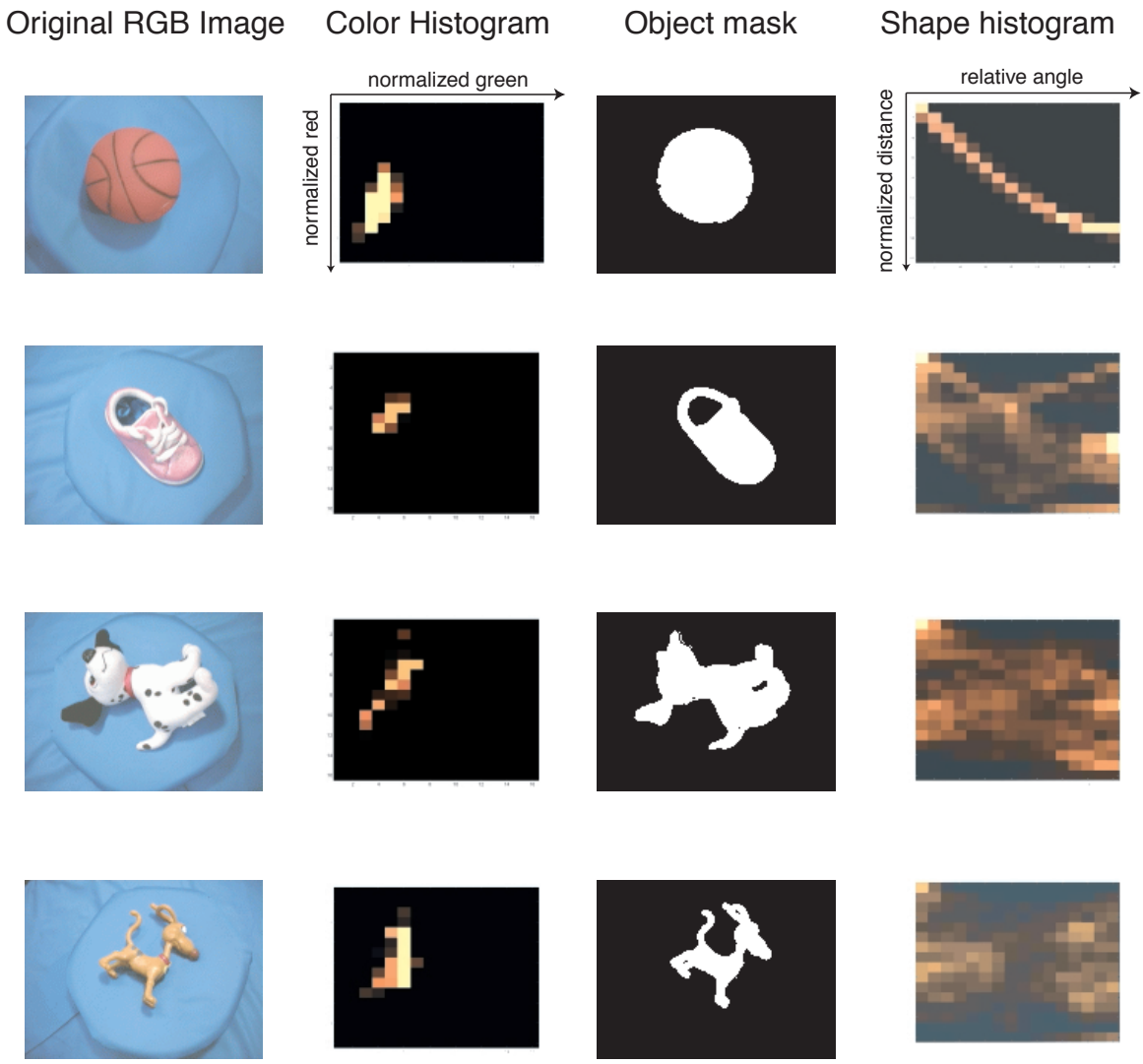


Figure 4-4: Examples of shape and color histograms computed for four images.

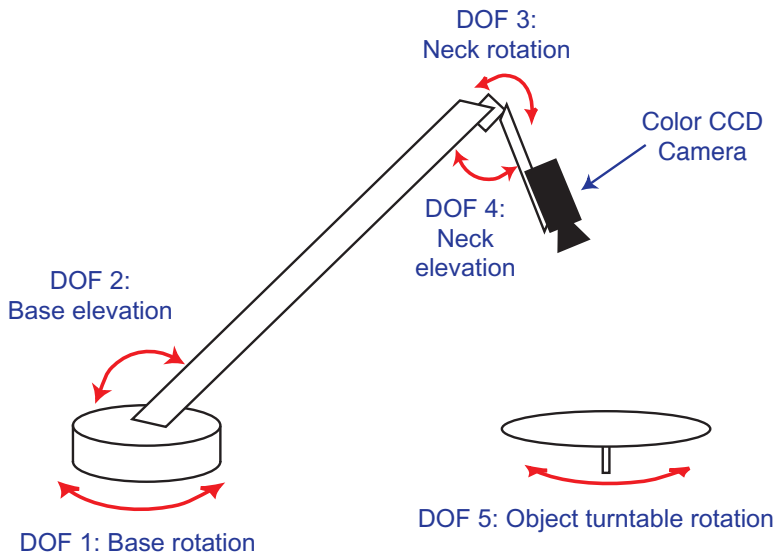


Figure 4-5: The robot has four degrees of freedom: two at the base and two at the neck. A turntable provides a optional fifth degree of freedom for viewing objects from various perspectives.

4.1.2 Active Camera Control

The CCD camera was mounted on a four DOF robotic armature enabling active positioning of the camera. The mechanical platform is shown schematically in Figure 4-5. The robot was built using aluminum and plastic hardware. The base joints were powered with an off-the-shelf motorized camera mount. The neck joints were powered with standard radio control (R/C) servos designed for model aircraft control. A third R/C servo was used to drive the turntable. All motors were connected to the host computer through a pair of serial ports.

The robot was designed to gather multiple views of a stationary object placed on a viewing surface in front of it. When the turntable was used, the robot was used to gather images of an object placed at the center of the turntable. Control of the robot is achieved through a *visuo-motor map*, a look-up table which specifies settings of all four servo motors in order to center the camera on a specific point on the viewing surface. A separate map was created for several different target positions on the

viewing surface. For a given target location, the corresponding table supplies a list of legal joint positions for the robot which will orient the camera to the desired location. Different views of an object were obtained by placing the object on a target location and sampling from the associated visuo-motor map.

A fragment of a sample motor map for the system is shown below. The units for the base joints range from -2000 to 2000, and the neck joints range from -90 to 90 units. Each row of the table specifies the settings of the four joints to direct the camera towards an intended position on the viewing surface.

Base Elevation	Base Rotation	Neck Elevation	Neck Rotation
-350.0	0.0	90.0	-30.0
-425.0	-75.0	85.0	-34.0
-575.0	-375.0	68.5	-36.5
-725.0	-675.0	51.0	-46.0
-800.0	-600.0	51.5	-47.5
-950.0	-300.0	58.0	-51.0

The visuo-motor map is created using a target finding procedure. To initialize the procedure, a small circular object (such as a tennis ball) is placed at the desired target location. The robot is manually set to a position which brings the target object into view. An iterative process then finds a setting of neck elevation and rotation which best centers the target. The foreground/background separation and connected regions analysis methods described above are used to calculate the center of the target. Once the neck has been aligned, the set of four servo positions are found and recorded. To build the complete visuo-motor map, the training procedure systematically steps the base elevation and rotation joints through all mechanically possible values (with steps of approximately 5 degrees). For each step, the optimal neck setting for centering the target are recorded. For many of the base positions, the mechanical construction of the robot makes it impossible to view the target. The training procedure discovers

these limits since object centering fails in these situations. The result is that the final map does not contain entries for those base positions. The entire procedure takes approximately 40 minutes to build a visuo-motor map with 380 entries. A separate map was generated for several target locations. A novel target location was viewed by interpolating between existing maps.

Once a set of maps have been trained, the robot may be used to acquire images of an object from multiple perspectives. Each entry in a map corresponds to a different vantage point. Control of the robot does not require inverse kinematic calculations, and it also does not need any information about absolute positions and angles. Instead, the coordinate system of the robot is vision centered. The robot's goal is to keep objects centered for the camera. All visual processing routines function with multiple random views of an object without reliance on absolute position information.

Over time, the alignment of motors to the visual system may shift due to slight changes in the camera mount and the position of the robot relative the target surface. An on-line mode of operation lets the system quickly tune visuo-motor maps to compensate for such misalignments. An existing map is loaded into the system and the robot "practices" finding a target. Any errors in centering an object are corrected using the original training procedure, and the map is updated accordingly.

For some applications (see Chapter 5) we found an additional degree of freedom for rotating a target object is useful for obtaining additional viewpoints. A small turntable was constructed which can rotate an object 360 degrees. The turntable control is coordinated with robot control to result in synchronized movements.

For each incoming image, the system produces a pair of histograms which represent the color and shape of the object. If an object is not in view, the visual channel encodes this information as a binary flag, and the histograms are not generated.

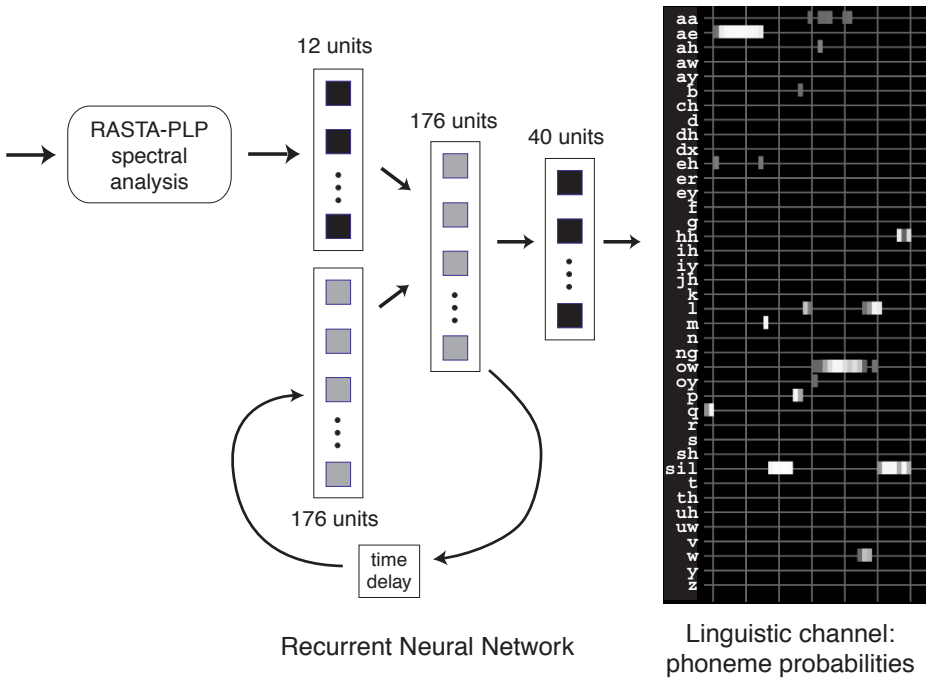


Figure 4-6: Extracting the linguistic channel from microphone input.

4.2 Linguistic Channel

The linguistic channel is grounded in acoustic signals originating from a microphone. Feature analysis generates a time-varying vector of English phoneme probabilities. The implementation thus assumes that the learner has knowledge of the phonetic structure of English prior to lexical acquisition. Figure 4-6 depicts the two stages involved in the linguistic feature analysis. Linguistic features are extracted from a microphone generated acoustic signal. The first stage, Relative Spectra Perceptual Linear Predictive (RASTA-PLP) analysis, extracts a spectral representation of the acoustic signal [53]. A recurrent neural network takes RASTA-PLP coefficients as input and estimates phoneme and speech/silence probabilities.

4.2.1 Acoustic Analysis: RASTA-PLP

The microphone signal is sampled at 16 kHz with 16-bit resolution and converted to the RASTA-PLP representation. RASTA-PLP is a spectral representation of speech.

It is designed to attenuate non-speech components of an acoustic signal. It does so by supressing spectral components of the signal which change faster or slower than speech. To start, the critical-band power spectrum is computed and compressed using a logarithmic transform. The time trajectory of each compressed power band is filtered to suppress non-speech components. The resulting filtered signal is expanded using an exponential transformation and each power band is scaled to simulate laws of loudness perception in humans. Finally, an all-pole model of the resulting spectrum is estimated. In our implementation, 12 all-pole model coefficients are computed on a 20ms (320 sample) window of input. A window step size of 10ms (160 samples) is used, resulting in a set of 12 RASTA-PLP coefficients estimated every 10ms.

4.2.2 Phoneme Analysis: Recurrent Neural Network (RNN)

A recurrent neural network (RNN) has been trained to compute likelihoods of English phonemes based on RASTA-PLP input. When presented a set of spectral coefficients at time t, the RNN produces a 39-dimensional output vector. The first 38 elements of the vector contain likelihood of 38 English phonemes (we use the same phoneme classes as Robinson [97]). The 39^{th} element contains the likelihoods that the input signal is silence. The output activations are all positive values and guaranteed to sum to 1.0[3] and can thus be treated as probabilities. The phonemes encoded in the RNN are listed in Table 4.2.

The RNN is an extension of a standard feed-forward neural network [37, 58]. In a standard feed-forward network, the units of the network compute a non-linear (typically sigmoidal) transfer function on the sum of its input activations. Arcs in a network have associated weights which scale the activations carried by them. Using the back-propagation training algorithm, a three-layer network can be trained to approximate a broad range of input-output transfer functions [106]. To use a trained network, input vectors are applied to the first layer of units. The units' activations

[3]The softmax function is used to normalize output activations.

Table 4.2: Phonemes encoded in the RNN.

Phoneme	**Example**	**Phoneme**	**Example**
aa	caught	l	lay
ae	bat	m	may
ah	but	n	no
aw	about	ow	boat
ay	bite	oy	boy
b	bat	p	pay
ch	chat	q	bat (glottal stop)
d	dog	r	ray
dh	then	s	say
dx	dirty	sh	shoe
eh	bet	sil	(silence)
er	bird	t	to
ey	bait	th	thin
f	fun	uh	book
g	go	uw	boot
hh	hay	v	vision
ih	bit	w	way
iy	beet	y	yacht
jh	joke	z	zoo
k	cat		

are propagated through the network. Output is generated by the final layer of the network.

In a RNN, the output units are a function of present and past inputs. This is accomplished with time delay units which serve as memory for past network activity. The RNN we implemented contains 176 hidden units. These units feedback through a time delay and are concatenated with incoming RASTA-PLP coefficients. The weights connecting the feedback units are learned using an extension of the back propagation training procedure known as back propagation in time [122]. In this method, recurrent weights are unfolded into a static network. The expanded network contains a hidden layer for each time step of a training sequence. Back propagation training with tied weights across layers can then be applied.

The RNN was trained with the TIMIT database of phonetically transcribed American English speech [112]. This database consists of read sentences spoken by 630 speakers from eight dialect regions of the United States. To train the network, each sentence is presented to the back propagation in time training procedure. The target outputs are set using the transcriptions provided in the TIMIT database.

To summarize, the auditory processor receives input from a microphone and produces a vector of phoneme probabilities (and silence) at a rate of 100 estimates per second. Figures 4-7 and 4-8 show examples of output from the RNN run on natural samples of infant-directed speech. Figure 4-7 depicts the RNN output for an utterance produced by a mother to her 10-month old infant while playing with a ball. Symbols for each RNN output are printed along the left and right edges of the plot. The strength of each RNN output determines the brightness of the associated trace as a function of time. Figure 4-8 is an RNN for a longer phrase spoken by the same mother.

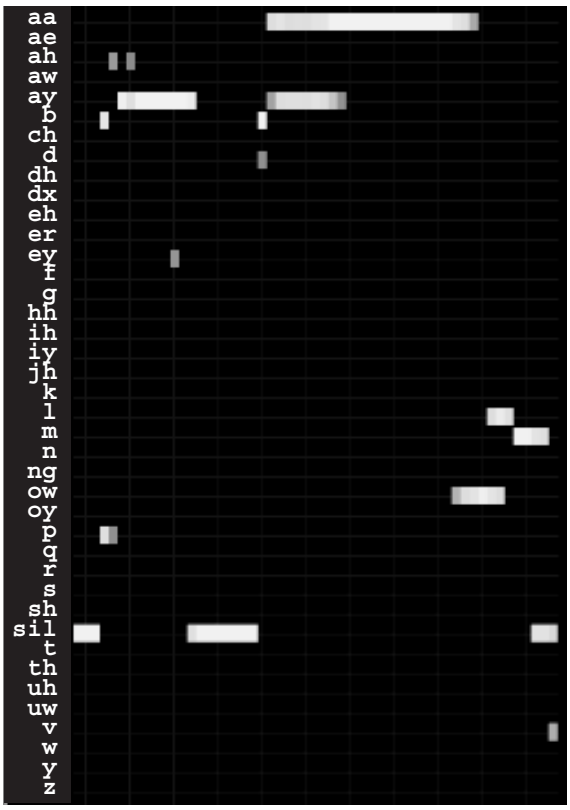


Figure 4-7: RNN output for the utterance "Bye, ball!".

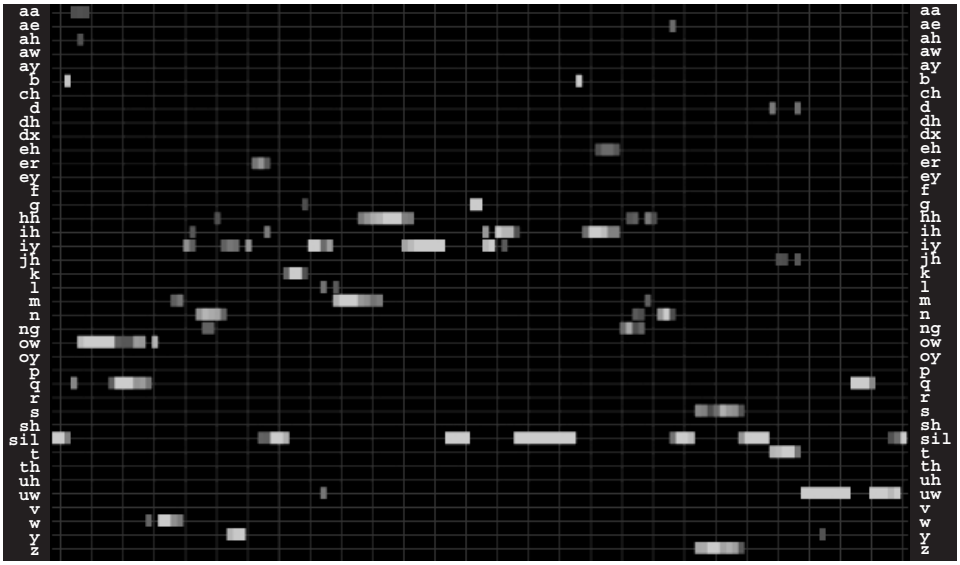


Figure 4-8: RNN output for the utterance "Oh, you can make it bounce too!".

4.3 Event Detection

The visual system and phoneme analyzer provide a constant flow of linguistic and contextual features. A pair of event detectors have been implemented to chunk these streams into L-events and S-events.

4.3.1 S-Events: Object View-sets

An S-event consists of a set of N views of an object, called a *view-set*. When run in an interactive situation (see Chapter 6) the robot continuously searches for the presence of objects on the viewing surface in front of it. When an object is detected, the robot records the image and moves to $N-1$ other locations to gather other views. These view points are selected randomly from a previously learned visuo-motor map. If the object is not present during any of the $N-1$ views, the system discards the collected images and re-initiates the search procedure for a new object. If all N views are successfully captured, shape and color histograms from each view are extracted to generate an S-event. For the evaluations with infant-directed data in Chapter 5, $N = 15$ and the interactive system presented in Chapter 6, $N = 5$.

4.3.2 L-Events: Spoken Utterances

The linguistic feature stream is chunked into *utterances*. Each utterance is composed of an array of phoneme probabilities delimited by silence.

The pseudocode listing in Figure 4-9 specifies the algorithm used to detect utterances from continuous acoustic input. The end-pointing algorithm is designed to detect sustained speech activity. Short bursts of speech are ignored (since they are likely due to environmental noise), and short silences within an utterance are absorbed into a single utterance. The silence estimate from the RNN drives the end-pointing process. The variable SIL is set to 1 if the silence estimate of the RNN is greater than any phoneme probability, and is set to 0 otherwise. Two parameters are used to

control the behaviour of the algorithm. UTTERANCE_START_DELAY determines how many milliseconds of contiguous speech frames must be encountered before the algorithm decides that an utterance has begun. The second timing parameter, UTTERANCE_END_DELAY, determines the duration of silence that must be observed before an end of utterance is detected.

4.4 Unpacking Events

Each L-event is assumed to contain instances of one or more words. Each S-event contains an instance of a shape category, and an instance of a color category. The next two sections describe the finest granularity of analysis on each type of event which is considered by the recurrence filter.

4.4.1 L-subevents: Speech segments

Spoken utterances (L-events) are segmented in time at phoneme boundaries. A phoneme boundary occurs when a speech signal transitions from one phoneme to another. These boundaries serve as hypotheses for potential boundaries which are used by the recurrence filter.

To locate phoneme boundaries, the RNN outputs may be treated as state emission probabilities in a Hidden Markov Model (HMM) framework [19]. When viewed in this way, a sequence of RNN outputs is equivalent to an unpruned phoneme lattice [93, page 43]. A dynamic programming search may be used to obtain probable paths through the lattice.

We have implemented an HMM with 39 states, one for each RNN output. The states are arranged in a parallel configuration (Figure 4-10). Using this HMM structure, a Viterbi search is performed to decode the most likely phoneme sequence in an utterance [93, page 339]. The RNN-HMM hybrid system achieves a phoneme recognition accuracy of 69% on the standard TIMIT speaker independent training

```
state = 1; count_2 = 0; count_3 = 0; count_4 = 0
UTTERANCE_START_DELAY = 50ms; UTTERANCE_END_DELAY = 300ms

for each RNN output vector, l(t) {

    state 1: SILENCE
        if SIL != 1 {
            utteranceStartIndex = t
            state=2 }
        else { state = 1 }

    state 2: POSSIBLE_START_OF_UTTERANCE
        count_2 = count_2 + 1
        if SIL = 1 {
            count_2 = 0
            state = 1 }
        else if {count_2 > UTTERANCE_START_DELAY) {
            state = 3 }

    state 3: UTTERANCE
        if SIL {
            state = 4 }
        else {
            count_3 = count_3 + 1
            state = 3 }

    state 4: POSSIBLE_END_OF_UTTERANCE
        count_4 = count_4 + 1
        if SIL != 1 {
            count_3 = count_3 + count_4
            count_4 = 0
            state = 3 }
        } else if count_4 > UTTERANCE_END_DELAY {
            utteranceEndIndex = t - count_4 - 1
            ProcessUtterance(utteranceStartIndex, utteranceEndIndex)
            count_2 = 0
            count_3 = 0
            count_4 = 0
            state = 1
        }
    }
}
```

Figure 4-9: Pseudocode listing of the utterance end-point detection algorithm.

set. Although this accuracy level is state-of-the-art, an error rate of over 30% in the underlying linguistic representation poses a great challenge for lexical acquisition. In Chapter 5 we show the benefit of leveraging co-occurring contextual information to reduce this acoustic ambiguity.

From a segmentation point of view, 69% phoneme accuracy is quite useful. Although nearly one in three phonemes is incorrectly transcribed, the errors are typically confusions between phonemes within broad classes. For example, a stop consonant might be confused for another stop consonant, but is unlikely to be mistaken for a vowel. As a result, the phoneme transition boundaries generated by the Viterbi algorithm are quite accurate and serve as a useful first step towards locating word boundaries in continuous speech.

The transition probabilities for entering each phoneme state (and silence) are set using bigram phoneme transition probabilities computed using phoneme transcriptions from the TIMIT training data set. State transition probabilities for staying within a state and exiting states were also trained using the TIMIT training set.

After Viterbi decoding of an utterance, the system obtains:

- A phoneme sequence. This is the most likely sequence of phonemes which were concatenated to form the utterance.
- The location of each phoneme boundary.

Each phoneme boundary may serve as a L-subevent start or end point. Any subsequence of an L-event terminated at phoneme boundaries may form an L-subevent. In this implementation, L-subevents are referred to as *speech segments*. A speech segment is a hypothesis of an instance of a spoken word.

4.4.2 S-subevents: Color / Shape view-sets

An S-event is a view-set which consists of N color and shape histograms. Both shape and color are assumed to be static aspects of an object. For this reason, S-events are

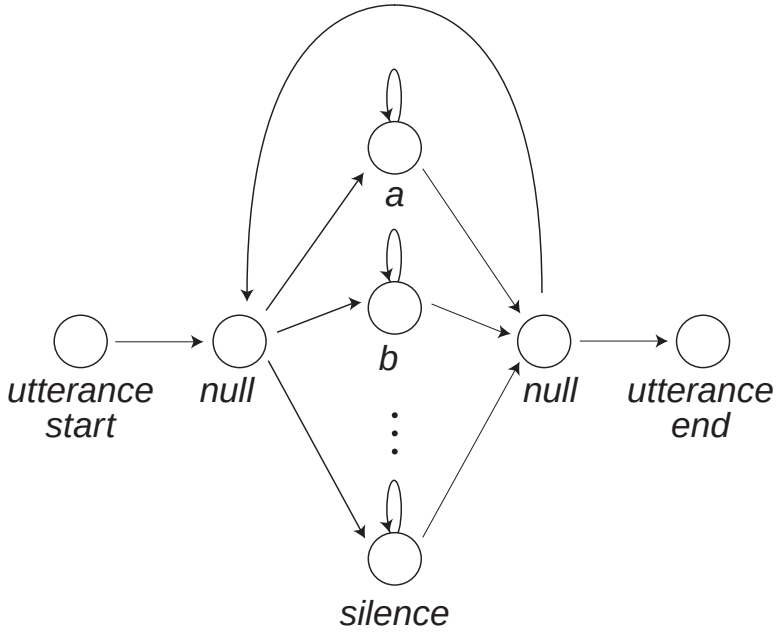


Figure 4-10: A Hidden Markov Model for computing most likely phoneme sequences. HMM state emission probabilities are computed by the RNN.

not segmented in time. An S-event is only divided along channels. A view-set may be partitioned into a color view-set which contains N color histograms, or a shape view-set which contains N shape histograms [4]. A color view-set is a hypothesis of an instance of a color category, and similarly a shape view-set is a hypothesis of an instance of a shape category.

The S-subevents implemented for this thesis are relatively simple since they assume objects are rigid and will remain static during viewing. Many computer vision techniques exist for processing more complicated input including articulated objects and motion. In the future, these techniques could be integrated into the current system.

[4] An extension which we implemented but did not evaluate in this thesis is to treat an entire S-event as an S-subevent. In other words, to let CELL look for semantic categories which are defined as a conjunction of a color and shape categories. For example, the meaning of *apple* might refer to round objects which are red.

4.5 Co-occurrence Filtering

The co-occurrence filter detects when a spoken utterance overlaps in time with a view-set. Its implementation is straightforward. The utterance end-pointing algorithm time stamps the start and end point of each utterance. The visual processing system time stamps the first and last image of each view-set. If the time stamps of the two events overlap, the co-occurring pair of events are bundled into an LS-event. LS-events are stored in a short term memory buffer with a capacity of five LS-events. The short term memory is implemented as a circular buffer which overwrites the oldest contents with new entries.

The co-occurrence filter focuses the system's attention on speech which is heard in the presence of an object (and vice versa). We assume that co-occurrence of events signifies that the utterance may contain one or more words which refer to either the color or shape of the object in view.

4.6 Recurrence Filtering

The recurrence filter searches for recurrent speech segments paired with recurrent color or shape view-sets. To implement this, we must establish a distance metric for comparing speech segments, $d_L()$, and a distance metric for comparing S-subevents, $d_S()$.

4.6.1 Acoustic Distance Metric

To perform recurrency analysis, we must define a distance metric, $d_L()$ which measures the similarity between two L-subevents, in this case speech segments. Recall that a speech segment consists of a sequence of phoneme probabilities generated by the RNN. For each speech segment, we can also obtain the most likely phoneme sequence which generated the RNN output (Section 4.4.1).

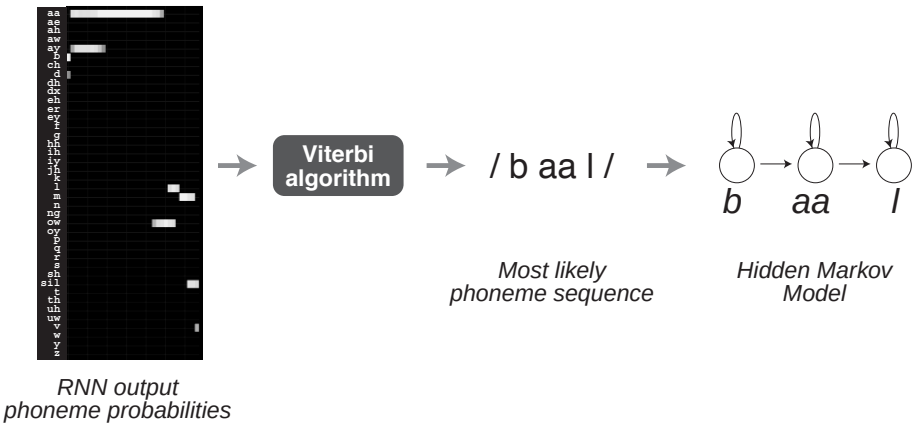


Figure 4-11: The Viterbi algorithm finds the most likely phoneme sequence for a sequence of RNN output. A left-to-right HMM is constructed by assigning a state for each phoneme in the sequence.

One possibility is to treat the phoneme sequence of each L-subevent as a string and use string comparison techniques to derive $d_L()$ [62]. This method has been applied to the problem of finding recurrent speech segments in continuous speech [125]. A limitation of this approach is that it relies on only the single most likely phoneme sequence. A sequence of RNN outputs is in fact equivalent to an unpruned lattice from which multiple phoneme sequences may be derived. To make use of this additional information, we have devised a new method for comparing speech segments.

Let Q be an array of phoneme probabilities generated by the RNN for a speech segment. Using the Viterbi algorithm, the most likely phoneme sequence may be estimated from Q. This sequence may in turn be used to generate an HMM model λ by assigning an HMM state for each phoneme in the sequence and connecting each state in a strict left-to-right configuration. In Figure 4-11 the RNN output for the word *ball* leads to the phoneme sequence /bɑl/. This sequence is used to generate a 3-state left-to-right HMM. State transition probabilities are inherited from a context-independent set of phoneme models trained from the TIMIT training set.

Consider two speech segments, α_i^* and α_j^* [5] and corresponding phoneme probability

[5]Speech segments are L-subevents which in turn serve as L-prototypes in CELL. Hence we use

arrays Q_i and Q_j. From these arrays, we can generate HMMs λ_i and λ_j.

We wish to define a metric for measuring the distance between α_i^* and α_j^*. We do so by testing the hypothesis that λ_i generated α_j^*, and vice versa. The Forward algorithm [93, page 335] can be used to compute $P(Q_i|\lambda_j)$ and $P(Q_j|\lambda_i)$, the likelihood that the HMM derived from speech segment α_i^* generated speech segment α_j^* and vice versa. However, these likelihoods are not an effective measure for our purposes since they represent the joint probability of a phoneme sequence and a given speech segment. An improvement is to use a likelihood ratio test to generate a confidence metric [98, page 318]. In this method, each likelihood estimate is scaled by the likelihood of a default alternate hypothesis, λ^A:

$$L(Q, \lambda, \lambda^A) = \frac{P(Q|\lambda)}{P(Q|\lambda^A)} \tag{4.1}$$

In our metric, the alternative hypothesis is the HMM derived from the speech sequence itself, i.e. $\lambda_i^A = \lambda_j$ and $\lambda_j^A = \lambda_i$. The distance between two speech segments is defined in terms of logarithms of these scaled likelihoods:

$$d_L(\alpha_i^*, \alpha_j^*) = -\frac{1}{2}\left\{\log\left[\frac{P(Q_i|\lambda_j)}{P(Q_i|\lambda_i)}\right] + \left[\frac{P(Q_j|\lambda_i)}{P(Q_j|\lambda_j)}\right]\right\} \tag{4.2}$$

This metric is symmetric, i.e. $d_L(\alpha_i^*, \alpha_j^*) = d_L(\alpha_j^*, \alpha_i^*)$. Logarithms are used to avoid floating point mathematical underflow problems in the implementation. The negative sign converts the likelihood score, a measure of similarity, into a score of dissimilarity (i.e., a distance).

4.6.2 Visual Distance Metric

The color and shape of an object observed in a single image are represented as two-dimensional histograms. An S-subevent consists of N color or shape histograms. To define the distance metric $d_S()$ between two S-subevents, we first define a metric for

L-prototype notation to refer to speech segments.

comparing two individual histograms.

Schiele and Crowley compared several methods for matching histograms [109] and found that the χ^2-test was best for their task of object recognition. The χ^2-test for two histograms H_i and H_j is defined as:

$$\chi^2(H_i, H_j) = \sum_{x,y} \frac{(h_{i:x,y} - h_{j:x,y})^2}{h_{i:x,y} + h_{j:x,y}} \tag{4.3}$$

Where x and y index into the two-dimensional histograms, and $h_{i:x,y}$ is the $(x, y)^{th}$ element of histogram H_i.

To compare two S-subevents, $d_S()$ is defined as the sum of the best M of N matches between individual histograms. Histograms are compared using Equation 4.3. A histogram can only be used once to match another histogram. By choosing only a subset of views for comparing view-sets, the system does not require every view in one view-set to have a matching view in the other view-set. As long as M of the views match, the distance will be small.

For the evaluations in Chapter 5, $N = 15$ and $M = 4$. For the interactive application presented in Chapter 6, $N = 5$ and $M = 3$.

4.6.3 Recurrence Detection

The recurrence filter searches for matching speech segments and S-subevents in the STM. The search is invoked each time a new LS-event is added to STM. A lexical candidate is generated when two or more LS-events contain matching speech segments and shape or color view-sets. To decide whether two subevents match, thresholds must be set for each distance metric. These thresholds are set relatively low so that many lexical candidates are generated at the expense of more false hypotheses. Later stages of processing are designed to remove erroneous candidates. Section 5.7.1 discusses how recurrence thresholds are set in our evaluations. To reduce search time, speech segments are considered only if they contain a vowel and are less than one second in

duration.

The search considers each pairwise combination of speech segment, shape view-set, and color view-set. When multiple LS-events containing matching speech segments and matching colors or shapes are found, a lexical candidate is generated. The candidate contains a representative of each matched set of subevents. The representative is a copy of the "central" member of a set. This central member is defined as the subevent whose cumulative distance to all other matching subevents is minimum. In the case of only two subevents in the set, one of them is chosen at random.

4.7 Maximizing Audio-Visual Mutual Information

Once the MTM has been sufficiently populated, the mutual information between L-units and S-categories may be measured for each lexical candidate. The definition for mutual information in Equation 3.3 may be rewritten by factoring the joint probability:

$$I(L;S) = \sum_i \sum_j P(s_j|l_i)P(l_i) \log \left[\frac{P(s_j|l_i)P(l_i)}{P(l_i)P(s_j)} \right] \tag{4.4}$$

Recall that each candidate in MTM may be thought of as an experiment in which the value of random variables S and L are determined with respect to a reference candidate and set of radii. The terms s_i and l_i are as defined in Equation 3.1. The probabilities in Equation 4.4 are estimated using relative frequencies:

$$\widehat{P}(s_i) = \frac{|s_i|}{N} \tag{4.5}$$

$$\widehat{P}(l_i) = \frac{|l_i|}{N} \tag{4.6}$$

$$\widehat{P}(s_j|l_i) = \frac{|s_j, l_i|}{|l_i|} \tag{4.7}$$

N is the number of lexical candidates in MTM (not including the reference candidate), and vertical bars denote the count operator. To overcome difficulties with small frequency counts, noisy probability estimates are smoothed by linearly interpolating their values with priors [47]:

$$\widehat{P}_2(s_i) = (1 - \alpha_N)\frac{1}{K} + \alpha_N \frac{|s_i|}{N} \tag{4.8}$$

$$\widehat{P}_2(l_j) = (1 - \lambda_N)\frac{1}{K} + \lambda_N \frac{|l_j|}{N} \tag{4.9}$$

$$\widehat{P}_2(s_j|l_i) = (1 - \alpha_N)\widehat{P}(s_j) + \alpha_N \frac{|s_j, l_i|}{|l_i|} \tag{4.10}$$

K is set to the number of candidates in MTM. The interpolation parameters α_N, λ_N, and β_N are set to $N/(m + N)$ for some fixed prior mass m.

For a selected reference candidate, a two-dimensional space of L-radii and S-radii may be searched to locate the point of maximum mutual information. Figure 4-12 presents two examples of mutual information surfaces from an infant-directed speech corpus (see Chapter 5 for details). In each plot, the height of the surface shows mutual information as a function of the L-radius and S-radius. On the left, the L-prototype corresponding to the word "yeah" was paired with the L-subevent of view-set corresponding to a shoe. The resulting surface is relatively low for all values of radii. The leixcal candidate on the right pairs a speech segment of the word "dog" with a view-set of a dog. The result is a strongly peaked surface form. The radii

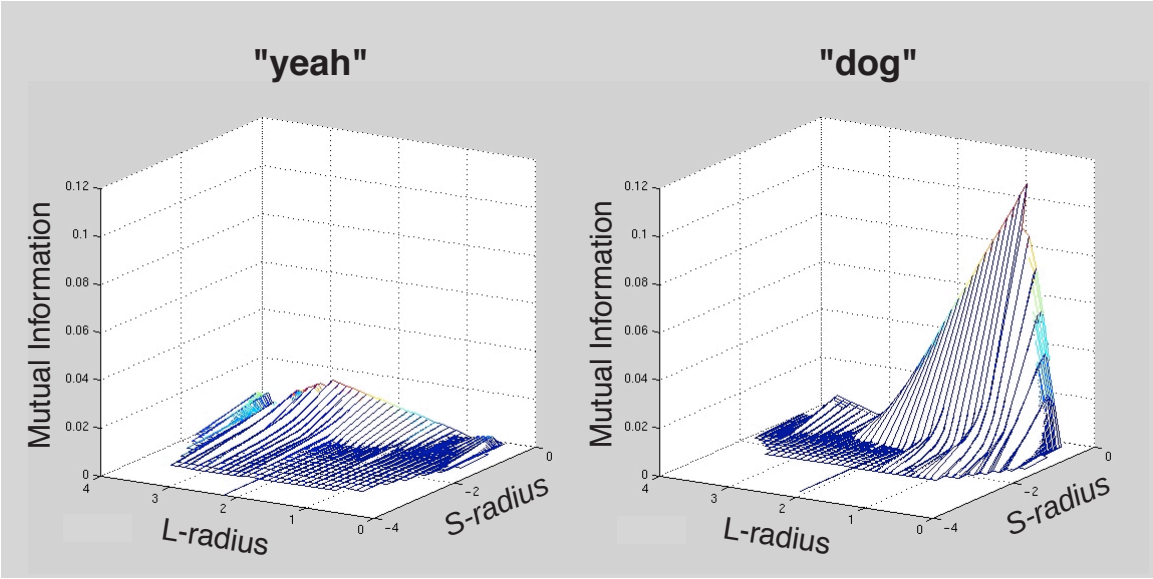


Figure 4-12: Mutual information as a function of L-radius and S-radius for two lexical candidates.

are selected at the point where the surface height, and thus mutual information, is maximized.

Each time a new candidate is added to MTM, all candidates are evaluated. Any candidates which result in high mutual information are promoted to LTM. The radius settings are stored with the candidate to define a new lexical item. When a new item is formed, all candidates in MTM which match both the L-unit and S-category of the newly formed item are removed from MTM.

4.8 Implementation Platform

The system has been implemented on a single SGI O2 workstation (MIPS R10000 CPU) with 128 megabytes of main memory. Speech is recorded using a noise-cancelling head-worn microphone whose output is digitized by the SGI on-board sampler at a rate of 16 KHz with 16-bit resolution. Visual input is provided by a miniature CCD camera mounted on a custom built robotic platform. The camera video signal is also sampled by the SGI's on-board hardware at a resolution of 160

columns by 120 rows of pixels. Each pixel is represented by a 24-bit RGB value.

4.9 Innate Knowledge

Several types of knowledge are built into the system before lexical learning begins. The innate knowledge includes:

- Spoken utterance detection and end-pointing capability.
- Models of English phonemes. These models are trained from the TIMIT database which contains speech samples and phonetic transcriptions. The phoneme models include (1) an acoustic feature to phoneme probability transform (implemented with an RNN), (2) phoneme duration models, and (3) a table of phoneme bigram transition probabilities.
- Visual foreground / background separation and object detection capabilities.
- Representations of object shape and color in terms of histograms.
- Distance metrics for comparing speech segments, color, and shape.

In Chapter 2 we reviewed experimental evidence which shows that infants have similar perceptual abilities prior to lexical acquisition.

4.10 Implementation of Extensions

Two of the extensions presented in Section 3.3 have been implemented. We mention them briefly in this section, and expand on them in Chapter 6.

Recognizing Novel Input: The recurrence filter contains all components needed to analyze novel linguistic and semantic input and search for matches with lexical items in LTM. We have implemented these features and demonstrated them in the context of a real-time adaptive interface.

Co-occurrence of Word Classes: Two word classes may be defined for the audio-visual implementation of CELL: shape terms, and color terms. An analysis module was implemented to search for adjacent words (i.e., words occurring in sequence with no intervening words) from different word classes in the input data. The module estimates the transition probability between these two word classes when in adjacent positions. These word transition probabilities were used for a simple grammar for connected word speech recognition, and to determine word order in a speech generation task.

Chapter 5

Evaluation with Infant-Directed Data

This chapter describes an evaluation of the audio-visual implementation of CELL using natural infant-directed speech and raw visual images. A study involving six caregivers and their prelinguistic infants was conducted to gather a corpus of infant-directed speech. The participants were asked to engage in play centered around seven types of objects commonly named in early infant speech. The speech was then coupled with sets of images of these objects (taken by the robot) and used as input for CELL. In this evaluation only the shape channel was utilized[1]. Chapter 6 discusses an application which uses both shape and color channels.

To compare the added utility of cross-channel learning, we implemented an acoustic-only model which ignores the contextual (visual) channel. The acoustic-only model selects lexical items based on acoustic recurrency in MTM. By doing so, this alternate model approximates the behaviour of a system driven by a minimum description length (MDL) criterion. On three different measures of early lexical learning, CELL out-performed the acoustic-only model across all six participants.

[1]Infants are known to have a "shape bias" and tend to learn names of shapes before colors and other visual classes [67]. Thus this initial investigation evaluates the performance with the shape channel alone.

Table 5.1: Participants of the evaluation study .

Participant	Occupation	Parent Age (years)	Infant Age (months)
CL	Home maker	27	8
CP	Administrator	26	10
TL	At home	17	8
SI	Teacher	27	9
PC	Journalist	44	11
AK	Home maker	23	9

5.1 Participants

Six caregivers and their prelinguistic infants participated in this study. Participants responded to a classified advertisement placed in a local newspaper in Toronto, Canada. By coincidence, all caregivers were female, and five of six infants were male. Table 5.1 summarizes the occupation and age of participants, and the age of their infants.

All participants were native speakers of English [2] and screened to insure that they had no speech or hearing impairments. Each participant confirmed that their infant could not yet produce single words. However, they reported varying levels of limited comprehension of words (e.g., their name, *no*, *dog*, *milk*, *wave*).

5.2 Objects

Participants were asked to interact naturally with their infants while playing with a set of age-appropriate objects. Huttenlocher and Smiley [57] identified a list of object

[2]The experiments were conducted in Toronto and all participants were long-time residents of Eastern Canada.

classes commonly named in early infant speech (Section 2.1.4). We chose seven classes of objects from the top of their list: balls, toy dogs, shoes, keys, toy horses, toy cars, and toy trucks. A total of 42 objects, six objects from each class, were obtained and are shown in Figure 5-1. The objects in each class varied in color, size, texture, and shape.

5.3 Protocol

To ensure controlled experimental conditions, collection of speech samples took place in a sound-treated child-appropriate room. To elicit natural interactions, caregivers and their infants were left alone in the room during sessions. The room was equipped with steerable video cameras and one-way observational windows. Each participant wore a noise-canceling head-worn microphone and wireless transmitter. All speech was recorded on a digital audio recorder for off-line analysis by CELL. Interactions were video taped for annotation purposes only.

Each caregiver participated in six sessions of play with their infants over a two day period (i.e., three sessions per day). Participants signed an informed consent sheet which detailed the experiment. The sheet explained that the goal of the study was to understand how infants learn words from listening to speech and watching their environment. For each of the six sessions, participants were provided with a set of seven objects, one from each of the seven object classes. The order in which object sets were provided was randomized across participants. The objects were placed in a box marked "in-box" at the start of each session. Participants were asked to take out one object at a time, play with it, and then return it to an "out-box".

The mothers were instructed to engage in play centered around the objects, one object at a time[3]. They were *not* told to teach their infants words. They were free to choose the order in which objects were selected for play, and the duration of play

[3]This restriction was made to simplify post-processing and annotation of the resulting speech recordings.

Figure 5-1: Objects used in the infant-directed speech experiments. Six examples of seven different objects: trucks, keys, dogs, shoes, cars, horses and balls.

with each object.

Sample speech recordings from the study were played to two speech-language clinicians who independently agreed that the recordings were natural and representative of infant-directed play. On average, each session lasted for about 12 minutes. Interaction with a specific object ranged between 30 seconds to over 3 minutes.

5.4 Speech Data

A total of 36 sessions of speech recordings were obtained (6 participants, 6 sessions per participant). Utterances were extracted from the recordings using the algorithm presented in Section 4.3.2. These utterances served as L-events for CELL. A sample of automatically extracted utterances for one participant, CL, is shown in Table 5.2.

We assume that infant-directed speech is redundant and that salient words will often be repeated in close temporal proximity [117]. This forms the basis for the recurrence filter in CELL (Section 3.2.8). This assumption was in fact confirmed in all our experimental sessions. Repetition of words occurred throughout all data sets, despite the fact that participants were not specifically instructed to teach their infants, or to talk exclusively about the objects. They were simply asked to play naturally. A temporal "clumping" effect for salient words was evident. For example, the word *ball* would appear several times within the span of half a minute because of the focused and repetitive nature of the interaction. This finding was even more pronounced when caregivers and infants were engaged in joint attention with respect to an object. The STM in CELL may be thought of as a buffer which is large enough to capture temporal clumps. This allows the learner to focus higher level processing efforts on only a short window of recent events in the world.

Table 5.3 summarizes several characteristics of input data sets obtained from all six caregivers. The first column shows the total number of utterances extracted across all six sessions. The next column shows the estimated total number of words

Table 5.2: Transcription of automatically extracted spoken utterances for participant CL. The left column shows the object in play at the time of each utterance. Notice that utterances may contain words which refer to objects not in view (e.g., line two below).

Object	**Utterance**
dog	He's gonna run and hide
dog	He's gonna hide behind my shoe
dog	Look, Savanah
dog	See his eyes?
dog	You like anything with eyes on it, eh?
dog	Just like you he has eyes
dog	Ruf ruf ruf
car	That's what your daddy likes, look!
car	Doors open vroom!
car	The seats go forward, and they go back!
shoe	You're always climbing into the shoes at home
shoe	Savanah! (infant's name)
truck	OK, you want it to drive?
truck	The wheels go around
truck	Your uncle Pat drives a truck like that
dog	He has a red collar
key	Let me see it
key	Do the keys have teeth?
key	You only have two teeth
key	Look through the key hole, look!
key	I see you
key	What are we gonna unlock? you are gonna unlock something?
key	Where you gonna lock it up?
key	What are you gonna do with it?
horse	You always like horses
horse	See this brown horse?
horse	You see him?
horse	See the tail?

accumulated across all these utterances [4]. The total number of words is divided by the total number of utterances to generate the third column which shows that, on average, utterances contained approximately five words. Although one and two word phrases did occur, they were far less common than longer utterances. This highlights the problem of word segmentation from a continuous stream of speech.

To understand the distribution of words, we manually annotated the data sets for keywords and their frequency of occurrence. Keywords which were annotated included: "ball", "car", "shoe", "dog", "doggie", "ruf" (dog's barking sound), "truck", "horse", "horsey", and "key". These annotations were strictly made for to understand the data. The annotations were not used in any way by CELL during evaluation. The total number of keywords in each data set is reported in column four.

The final column in Table 5.3 calculates the percentage of input words that refer directly to the provided object shapes. On average, over 92% of the input speech contains non-keywords. This further illustrates the difficulty of the task of lexical learning with this corpus.

5.5 Visual Data

The robot described in Section 4.1.2 was used to create a database of images for the 42 objects. The motivation was to generate a set of images of each object from a first-person perspective. A set of 209 images were captured of each object from varying perspectives resulting in a database of 8,778 images [5].

From each pool of 209 images, we created view-sets of each object by randomly selecting sets of 15 images. View-sets were compared using a match size of 4 views (see Section 4.6.2)[6].

[4]To calculate this, the speaking rate of each participant was estimated from a sample set of utterances. The cumulative duration of all input utterances was computed and multiplied by the estimated speaking rate to generate the estimated total word count.

[5]Sample images shown in Chapter 4 were taken from this database.

[6]The sampling was designed to minimize re-use of images in different view-sets. No two images

Table 5.3: Summary of input data obtained from parent-infant study.

Participant	Total Utterances	Estimated Total Words	Estimated Words per Utterance	Total Keywords	Estimated Percent Keywords
CL	1141	5168	4.5	425	8.2%
CP	792	2797	3.5	359	12.8%
TL	1275	4152	2.6	356	8.6%
SI	1696	8702	5.1	479	5.5%
PC	1643	8735	5.3	425	4.9%
AK	1056	6955	6.6	429	6.2%
Average	1267	6085	4.6	412	7.7%

To help characterize the image database, we computed pair-wise distances between view-sets. The resulting distances were used to generate a set of histograms (Figure 5-2,5-3,5-4). The horizontal axis marks histogram bins, and the vertical axis indicates bin occupancy. The vertical axis of all histograms have been normalized to aid in visualization. The horizontal axis is held constant to enable comparison between histograms. The histogram bins represent increasing distance between view-sets from left to right.

Figure 5-2 shows a histogram of all distances between different view-sets of the same object. For example, all view-sets of truck C are compared to all other view-sets of truck C. Distances for all 42 objects are accumulated in this histogram. There is little variability in the distances between view-sets of the same object. Distances between view-sets of the same object are relatively small.

Figure 5-3 shows histograms of distances between all view-sets of objects belonging to the same class. For example, all view-sets of truck A were compared to view-sets of

were used more than twice across multiple view-sets. When view-sets were compared, we used the sum of the four best views so the effects of one shared image between view-sets was not significant.

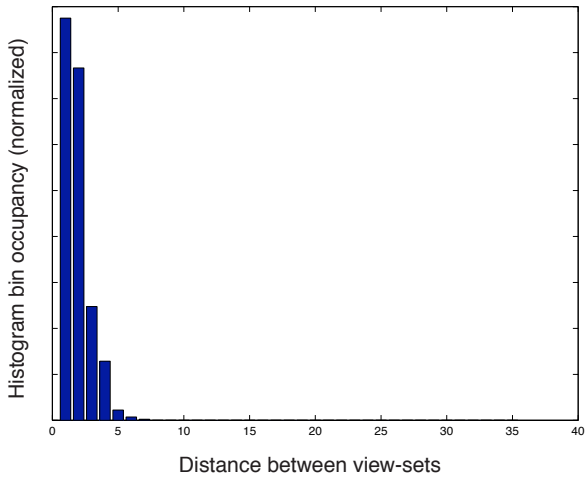


Figure 5-2: Histogram of distances between view-sets of the same object.

truck B, C, D, E and truck F (but not to other view-set of truck A). This histogram shows significantly greater spread than the first indicating large variability of objects within a class. As Figure 5-1 shows, dogs and trucks have large within-class shape variations whereas balls have almost none.

Figure 5-4 shows a histogram of distances between each view-set and all other view-sets across all 42 objects. The bimodal nature of this histogram suggests that the view-sets form clusters based on similarity in shape. The most obvious clusters arise from self-similarity since view-sets of the same object are being folded into this histogram. Objects with similar visual forms such as dogs and horses, and balls and shoes (which both have circular edges), contribute to the first mode. Dissimilar classes such as balls and trucks, or cars and keys result in the second mode.

5.6 Combining Speech and Visual Data to Create LS-events

Caregivers were asked to play with one object at a time. All utterances that were produced by the caregiver between the time when the object was removed from the in-box and placed in the out-box were paired with that object. Video recordings of

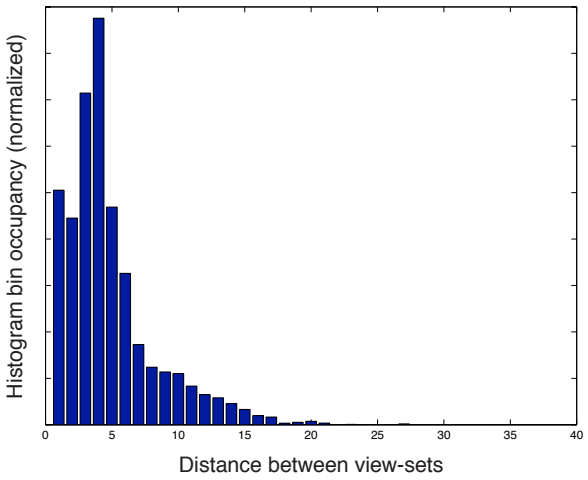


Figure 5-3: Histogram of distances between view-sets of different objects from the same object class.

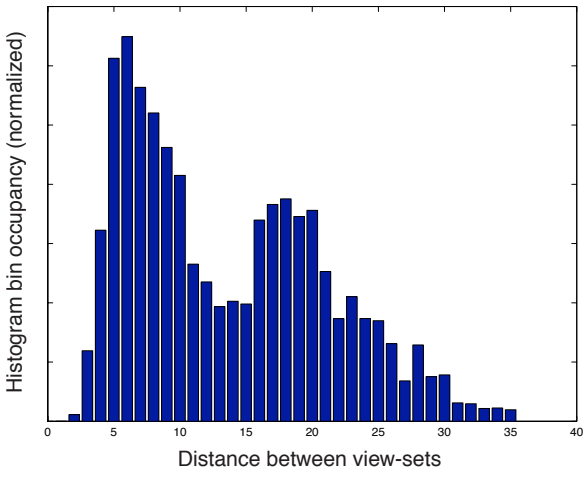


Figure 5-4: Histogram of distances between all pairs of view-sets in the database.

the caregiver-child interactions were used to guide this pairing process. To prepare the data for CELL, we generated an LS-event for each spoken utterance by pairing it with a randomly selected view-set of the corresponding object. By pairing every utterance with a view-set, we effectively perform the function of the co-occurrence filter.

Generating an LS-event for every spoken utterance simplified data preparation. We assumed that all utterances occurred while the infant was looking at the object. In reality, however, infants were not watching the object in some cases. Although this assumption allowed us to use all the recorded speech for evaluation, it may have made the problem of lexical learning even more difficult. Caregivers may have been less likely to refer to an object if they were aware that the infant was not attending to it.

5.7 Processing the Data by CELL

CELL was used to process each participant's data in a separate experiment. The LS-events generated from each participant were presented to the system in the sequence in which they were recorded. The STM size was set to 5 LS-events, and the MTM size was set to 1000 lexical candidates. The result of processing the data is a set of lexical items which are deposited into LTM.

5.7.1 Setting Recurrency Thresholds

To run the recurrency filter, the distance metrics for comparing L-events and S-events must be thresholded. Analysis of the input data suggests natural values for both thresholds. For the visual data, we had to set a threshold to determine matches between shape view-sets. Recall that the distribution of distances between all view-sets in the database have two distinct modes (Figure 5-4). A natural choice for the threshold is the point at which this distribution reaches a minimum between the two

modes. The visual recurrency threshold was set to this value for all six participants. The histogram does not rely on any training labels since all view-sets across unlabelled object classes are used to compute the histogram. The only assumption underlying the calculation of the threshold is that the learner has some visual experience with target object classes prior to lexical acquisition; enough experience to acquire a rough histogram of distances between objects.

An analysis of the distribution of distances between speech segments was used to set the acoustic recurrency threshold. We selected 200 random segments of speech ranging in duration from 100-1000 ms. The segments were taken from one of the participants in the database. The distance between each pair of segments was computed using the metric defined in Section 4.6.1. A histogram of these distances is shown in Figure 5-5. The acoustic threshold was set at the maximum of this distribution. The intuition is that as the radius of allowable error from a prototype grows, the density of samples captured will increase until the mode of this distribution is reached. We set the distance threshold at this point to capture a large number of segments which are within a tight radius of a prototype.

Both methods of setting thresholds rely solely on overall distributions of distances between randomly selected view-sets and speech segments. No training labels or manual assistance is needed to set either threshold. In practice the exact value of either thresholds was not been found to be critical. Small variations from the optimized value lead to similar overall performance.

5.7.2 Recurrency Processing

Recurrence filtering resulted in a series of lexical candidates which were deposited in MTM. Table 5.4 summarizes the data in MTM for each participant. The first column shows the total number of words which passed through STM. This matches the number reported in Table 5.1. The second column shows the number of lexical candidates generated by the recurrency filter, and the third column shows the average

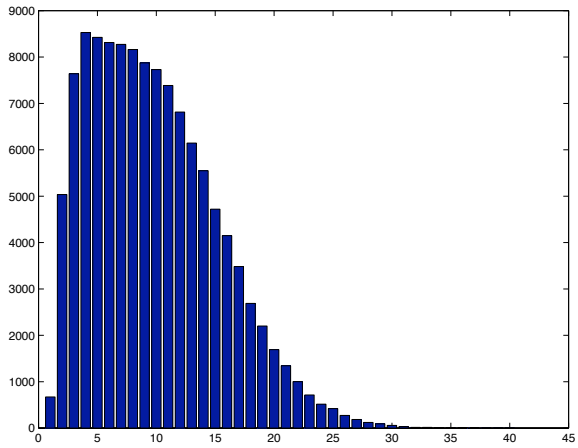


Figure 5-5: Histogram of acoustic distances between randomly selected speech segments less than 1 second in duration.

number of words per speech segment in the MTM.

The average number of words in a lexical candidate L-subevent is approximately three times less than the average number of words in an L-event. The MTM was made large enough to hold all lexical candidates generated by any participant (i.e., greater than 935 candidates). As a result, this implementation does not exercise the FIFO nature of the MTM specified by CELL which is intended to limit MTM capacity for much larger amounts of data than we were able to gather for these studies.

5.7.3 Selecting Lexical Items for LTM

For each lexical candidate in MTM, the mutual information maximization search produces a lexical item hypothesis. In this evaluation, the top 15 items were selected and placed in LTM. We expect that incorporating feedback into the model would enable the system to select an optimal number of lexical items (see Section 3.3.5).

5.8 Baseline Acoustic Only Model

To assess the difference between cross-channel audio-visual learning and mono-channel learning, we implemented an acoustic-only system based on components of CELL.

Table 5.4: Lexical candidates generated by the recurrency filter.

Participant	Total Estimated Input Words	Lexical Candidates	Average Number of Words per Candidate
CL	5168	732	1.5
CP	2797	231	1.7
TL	4152	935	1.1
SI	8702	669	1.7
PC	8735	855	1.6
AK	6955	490	2.1
Average	6085	652	1.6

Although the system was presented with LS-events (i.e., spoken utterances paired with view-sets), only the acoustic portion of the input was used to generate lexical items. The acoustic-only model attempts to identify speech segments which recur most often in the input. The model assumes that some underlying language source concatenates words according to a set of unknown rules. The problem of segmentation persists. The boundaries between words are unavailable to the learner since utterances are spoken fluently and the learner has no prior knowledge of the lexicon. In this model, highly recurrent segments of speech form likely candidates of lexical items in the language. The acoustic-only model implemented here relates to minimum description length approaches in the literature. Lexical items are defined as those which occur most frequently, and thus best explain the input data. In contrast to work by Brent or de Marcken [23, 22, 31], here we are not interested in a best segmentation of the entire input corpus, but rather to identify a set of most likely lexical items of the target language.

The acoustic-only model utilized many of the same components implemented in CELL. The recurrency filter was modified to ignore the visual channel. Recurrent

speech segments were extracted from STM and placed in MTM based on acoustic matching only. The STM and acoustic recurrence threshold were configured identically to the experiments with CELL. A second recurrence filter searched for recurrent segments throughout the entire MTM. Each lexical candidate in MTM was ranked according to how many other MTM candidates it matched acoustically. This threshold was set by hand for optimal performance. We found that a significantly higher threshold (i.e., requiring closer acoustic matches) worked best for this filter. Once the lexical items had been ranked according to acoustic recurrence, the top 15 were selected and placed in LTM. We choose 15 items to allow for direct comparison with the CELL implementation. The S-event paired with each input L-event was carried through the system so that each LTM item had an associated shape model. The underlying assumption is that the meaning of the lexical item is embedded in the context in which the selected speech segment originally occurred. This allowed us to compare if LTM items were associated with the appropriate object.

5.9 Evaluation Measures

Results of the experiments were evaluated using three measures. For each acoustic and visual prototype used to generate a lexical item in our systems, a pointer to the source speech recording and image set were maintained. An interface was built to allow for listening to the original speech recording from which a prototype was extracted. The interface also displayed the images of the corresponding view-set. An evaluator trained in phonetic transcription used this tool to assess the results.

For each LTM item we recorded several types of information:

Measure 1: Segmentation accuracy Do the start and end of each speech prototype correspond to word boundaries in English?

Measure 2: Word Discovery Does the speech segment correspond to a single English word? We accepted words with attached articles and inflections, and we

also allowed initial and final consonant errors. For example the words /dɑg/ (*dog*), /ɑg/ (**dog*, with initial /d/ missing), and /ðədɑg/ (*the dog*), would all be accepted as positive instances of this measure. However /dɑgɪz/ (*dog is*) would be counted as an error.

Measure 3: Semantic Accuracy If the lexical item passes the second measure, does the visual prototype associated with it correspond to the word's meaning? If a lexical item fails on Measure 2, then it automatically fails on Measure 3.

It was possible to apply Measure 3 to the acoustic-only model since the visual prototype was carried through from input to output. In effect, this model assumes that when a speech segment is selected as a prototype for a lexical candidate, the best choice of its meaning is whatever co-occurred with it.

5.10 Results

Table 5.5 lists the contents of LTM for one of the participants using CELL. A phonetic and text transcript of each speech prototype has been manually generated. For the text transcripts, asterisks were placed at the start and/or end of each entry to indicate the presence of a segmentation error. For example "dog*" indicates that either the /g/ was cutoff, or additional phonemes from the next word were erroneously concatenated with the target word. For each lexical item we also list the associated object based on the visual information. The letters A-F are used to distinguish between the six different objects of each object class.

Several phoneme transcripts have the indicator "(ono.)" which indicate onomatopoetic sounds such as "ruf-ruf" for the sound of a dog, or "vroooommmm" for a car. The corresponding text transcript shows the type of sound in parentheses. We found it extremely difficult to establish accurate boundaries for onomatopoetic words in many instances. For this reason, these lexical items were disregarded for all measures

of performance. It is interesting to note that CELL did link objects with their appropriate onomatopoetic sounds. They were considered meaningful and groundable by CELL in terms of the shape channel. This finding is consistent with infant learning; infants often use onomatopoetic sounds to refer to common objects. The only reason these items were not processed further is due to the above stated difficulties in assessing segmental accuracy.

The final three columns show whether each item passes the test of each measure. In some cases a word such as *fire* is associated with a fire truck, or *lace* with a shoe. These are accepted as valid by Measure 3 since they are clearly grounded in specific objects. At the bottom of the table, the measures are accumulated to calculate accuracy along each measure [7].

For comparison, the LTM items acquired by the acoustic-only model are shown in Table 5.6. These results are derived from the same participant's data as Table 5.5. In cases where no discernible words were heard, the text transcript is left blank. CELL out-performs the acoustic-only model across all three measures. This pattern was observed consistently across all six participants[8]. Figures 5-6, 5-7, and 5-8 plot average scores across all six participants for each measure.

Measure 1, segmentation accuracy, poses an extremely difficult challenge when dealing with real acoustic data. The acoustic-only model produces lexical items which correspond perfectly with English words only 1 in 14 times. In contrast, 28% of lexical items produced by CELL were correctly segmented single words. Of these 28%, half of the accepted items are not grounded in the contextual channel (i.e., they fail on Measure 3). For example, the words *choose* and *crawl* were successfully extracted by CELL and associated with car A and ball E respectively. These words do not directly refer to shape categories and thus fail on Measure 3. Yet, there seems to be

[7] Onomatopoetic items do not contribute to the denominator of the sums.

[8] The only exception was that in one instance (participant CL) Measure 1 (segmentation accuracy) increased from 20% to 33% from CELL to the acoustic-only model, respectively.

Table 5.5: Contents of LTM using CELL to process one participant's data (participant=PC).

Rank	Phonetic Transcript	Text Transcript	Shape Category	Segment. Accuracy	Word Disc.	Semantic Accuracy
1	ʃu	shoe	shoe E	1	1	1
2	fair ə	fire*	truck D	0	1	1
3	rək	*truck	truck C	0	1	1
4	dɑg	dog	dog D	1	1	1
5	ɪŋəʃ	in the*	shoe A	0	0	0
6	ki	key	key C	1	1	1
7	ki	key	key E	1	1	1
8	dɑggi	doggie	dog C	1	1	1
9	bɑl	ball	ball C	1	1	1
10	bɑl	ball	ball A	1	1	1
11	kiə	key*	key C	0	1	1
12	ʌʃu	a shoe	shoe B	0	1	1
13	ənðɪsɪz	*and this is	shoe B	0	0	0
14	(ono.)	(engine)	truck A	-	-	-
15	(ono.)	(barking)	dog A	-	-	-
Total				**54%**	**85%**	**85%**

Table 5.6: Contents of LTM using the acoustic-only model to process one participant's data (participant=PC).

Rank	Phonetic Transcript	Text Transcript	Shape Category	Segment. Accuracy	Word Disc.	Semantic Accuracy
1	(ono.)	(engine)	car C	-	-	-
2	dʒudʒudʒu	do do do	shoe A	0	0	0
3	(ono.)	(engine)	truck C	-	-	-
4	(ono.)	(engine)	truck C	-	-	-
5	wʌyugonnʌd	what you gonna do*	shoe A	0	0	0
6	nawhirk	now here okay*	ball B	0	0	0
7	lʌmiyuz	*amuse	car E	0	1	0
8	beybi	baby	horse A	1	1	0
9	ahhiʔ	ah he's*	horse E	0	0	0
10	iah	*be a	ball A	0	0	0
11	wʌyugonnd	what you gonna do*	key A	0	0	0
12	iligʊd	*really good	shoe F	0	0	0
13	iv	-	ball F	0	0	0
14	yulbiə	you'll be a	ball A	0	0	0
15	?ey	*today	dog D	0	1	0
Total				**8%**	**25%**	**0%**

Table 5.7: Summary of results. Each entry shows percentage accuracy for CELL, and in parentheses for the acoustic-only model.

Participant	**Segmentation Accuracy**	**Word Discovery**	**Semantic Accuracy**
PC	54 (8)	85 (25)	84 (0)
SI	25 (0)	75 (10)	42 (10)
CL	20 (33)	87 (60)	80 (20)
TL	17 (7)	50 (35)	25 (14)
CP	17 (0)	50 (8)	42 (8)
AK	33 (0)	92 (45)	67 (27)
Average	28±6 (7±5)	72±8% (31±8%)	57±10% (13±4%)

some structural consistency between the word and the shape which aids the system in producing this segmentation.

For Measure 2, word discovery, almost three out of four lexical items (72%) produced by CELL are single words (with optional articles and inflections) (Figure 5-7). In contrast, using the acoustic-only model, performance drops to 31%. These results demonstrate the benefit of incorporating cross-channel information into the word learning process. The cross-channel structure leads to a 2.3-fold increase in accuracy compared with analyzing structure within the acoustic channel alone. This result has implications for understanding language acquisition in infants. Rather than segment speech as a preparatory step towards acquiring sound-to-meaning mappings, a more efficient strategy may be to combine the segmentation process with the mapping process. The additional structure from the contextual channels may accelerate the overall process of early lexical acquisition.

On Measure 3, semantic accuracy, we see the largest difference in performance between CELL and the acoustic-only model (Figure 5-8). With an accuracy of 57%,

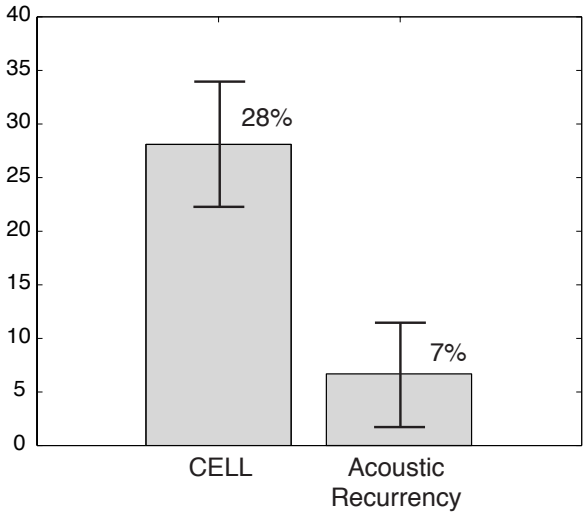


Figure 5-6: Segmentation accuracy (Measure 1) results averaged over all participants. Error bars indicate standard deviation about the mean.

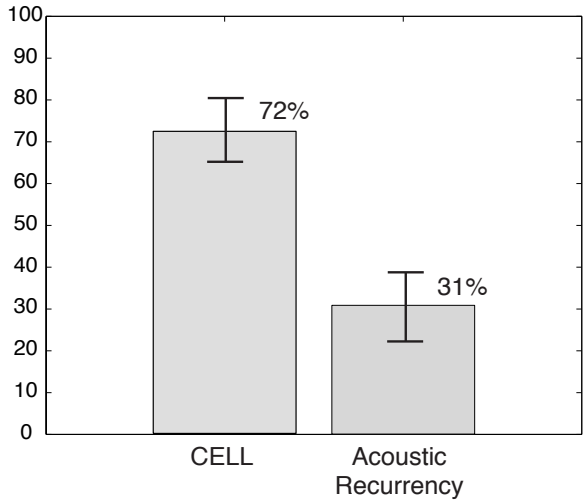


Figure 5-7: Word discovery (Measure 2) results averaged over all participants. Error bars indicate standard deviation of the mean.

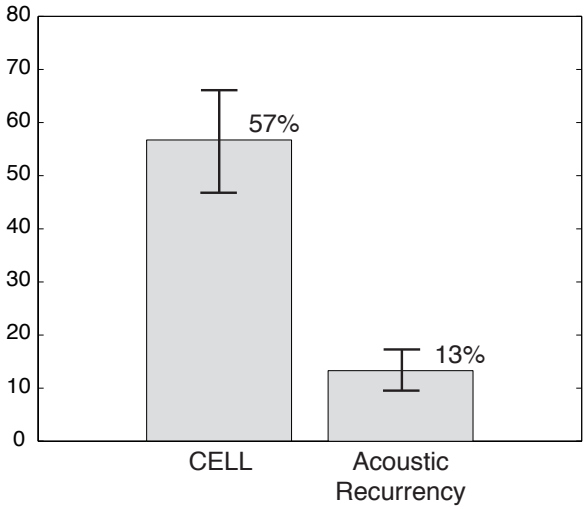


Figure 5-8: Semantic accuracy (Measure 3) results averaged over all participants.

CELL out-performs the acoustic-only model by over a factor of four. In the input speech, less than 8% of words are grounded in shape categories (Table 5.1). In the output, this ratio increases over seven times. The acoustic-only model acquires a lexicon in which 13% of the items are semantically accurate. This is operating at chance level since random guessing will yeild 14% (1 / 7 object classes).

The acoustic-only model is not without merit. It succeeds in learning many words which are not acquired by CELL including "go", "yes", "no", and "baby". These are certainly reasonable words to enter a young infant's vocabulary. This suggests that in addition to cross-channel structure, the learner may also notice within-channel structure to hypothesize words with yet unknown meaning. In a top down process, the learner might then look for the meaning of these words. This is a well explored hypothesis and our findings support it as a mechanism which may operate in parallel with CELL. Similarly, other hypotheses of word segmentation, for example based on prosodic contours [28], may also be employed to improve learning.

5.11 Summary

The results presented in this chapter are significant. They represent the first successful effort to automatically acquire linguistic knowledge from raw infant directed speech and natural raw visual context. The CELL implementation acquires a lexicon of visually grounded words across six different participants without any changes to system parameters. The success of the model on this database serves as an existence proof that the strategies proposed in CELL might be employed by an infant to learn from similar types of input.

Comparisons with an acoustic-only model demonstrate the importance of cross-channel structure in the lexical acquisition process. Semantic accuracy increased over four fold when the visual channel was used. Most surprisingly, cross-channel constraints dramatically increased word segmentation and discovery performance in comparison to the acoustic-only model.

An underlying assumption of modularity is typical in current models of speech segmentation, word discovery, and lexical acquisition. Models of segmentation and word discovery typically operate on the acoustic channel alone. These models assume that sufficient information is available within the structure of a single channel. Similarly, models of lexical learning assume that linguistic units have already been identified by the language learner.

Our results bring the assumption of strict modularity into question. The model demonstrates improved learning by leveraging information across channels at an early stage. Infants are known to possess all the necessary capabilities of sensory processing and correlational analysis necessary to employ the strategies proposed in CELL. Regardless of the details of the model, these results lead us to believe that cross-channel structure is harnessed by infants on the path to language.

Chapter 6

Adaptive Spoken Interfaces

6.1 Introduction

The current state of human-computer interaction (HCI) is strongly biased towards desktop computing based on a windows metaphor and point and click interaction. Although an effective desktop interface for current software, this paradigm is severely impoverished when compared with human-human interaction (HHI). A significant difference between HHI and HCI is that people tend to use rich modalities such as speech and gesture when interacting with one another. These modes of communication come naturally and without conscious effort. The ultimate goal of HCI design is to create interfaces which enable similarly natural and effortless expressivity.

Spoken language is the dominant mode of communication between people and yet it is largely untapped in current human-computer interfaces. This chapter is concerned with the use of speech input for HCI. We begin by presenting some problems with using speech input. Several common approaches for addressing these problems are reviewed. We then describe a new framework for creating adaptive spoken interfaces based on CELL. We present two prototype interactive systems based on CELL. Several application domains are identified in which the adaptive interfaces are effective, and we close with some comments about scalability of the approach.

6.2 Problem

Speech recognition technologies have improved remarkably over the past twenty years leading to commercial software for speech-to-text conversion. These systems work well if the user's speech conforms to predefined normative acoustic models. Performance degrades when the user's speech diverges from expected norms due to accents, speech impairments, or other non-normative acoustic patterns.

Progress has been far slower for speech *understanding* tasks than for speech recognition. In a dictation task, the machine does not have to infer any semantics. The problem is strictly to transcribe, verbatim, acoustics into text without interpretation. In speech understanding systems, the mapping from a user's words to the user's intentions becomes an issue.

A key problem in using speech input is the individual variability of spoken language, both in surface form and semantic content. No two people sound exactly the same, and no two people choose the same words to express the same intents. We summarize these two sources of variability in which we make an explicit distinction between surface form variability and the semantic mapping problem:

Surface form variability The acoustic characteristics of the speech signal may vary widely as a function of the speaker. Some of the many factors include: vocal tract characteristics, accents, speech impairments, age, and gender. Any of these factors may affect the performance of a speech recognition system. A mismatch between the user's speech and the speech used to train the recognizer's acoustic models along any of these dimensions result in a drop in recognition accuracy.

The semantic mapping problem The meaning of a word or phrase, even in identical contexts, may vary from person to person. Experimental evidence suggests this is especially troublesome for command oriented tasks in which input consists of single words or short phrases [39].

The semantic mapping problem was studied by Furnas, Landauer, Gomez, and

Dumais [39]. In their experiments, participants were asked to spontaneously name objects for five application-related domains [1]. In every case, the probability that any two people selected the name for the same object was less than 0.2. The authors concluded that:

> There is no one good access term for most objects. The idea of an "obvious", "self-evident" or "natural" term is a myth! ... Even the best possible name is not very useful...Any keyword system capable of providing a high hit rate for unfamiliar users must let them use words of their own choice for objects [39, page 967].

One way to address these problems might be to design a system with multiple synonyms which cover all possible words chosen by all possible users. There are problems with this approach. First, as the vocabulary grows, recognition accuracy decreases. Second, Furnas et al. conducted experiments in which they created lists of 20 synonyms for each command in their sample task. In a task with 25 commands, the chance that any two people who decided that the same term meant the same command was only 15%. These results suggest that not only is word choice not predictable, but the semantic association of a word is also highly ambiguous, even in fixed contexts.

6.3 Current Approaches

The problem of acoustic variability is well known to the speech research community. The field of speaker adaptation is devoted to this problem. Approaches include both supervised retraining of models, and unsupervised adaptation of models over time (e.g., [111, 40, 33]).

[1] The domains were: (1) verbs used to describe text editing operations in a word processing application, (2) commands for a "message decoder" program, (3) content words used to describe common objects, (4) superordinate terms chosen to describe classified advertisement items, and (5) keywords for cooking recipes.

The semantic mapping problem has received little attention in the speech research community. The problem is often regarded to be a "human factors" problem, not a technical one. Regardless of what sort of problem it is, any speech understanding system designer must nonetheless confront the problem in order to implement a system. The following sections summarize current approaches to overcome the semantic mapping problem.

6.3.1 The Intuitive Design Approach

Perhaps the simplest approach is to assign word-to-meaning or phrase-to-meaning mappings based on the personal intuitions of the interface designer. The problem, as Furnas has demonstrated, is that few users will actually agree with the designer's choice of vocabulary and semantic mappings. Without guidance, the user will attempt to speak naturally and the interface will fail.

6.3.2 Explicitly Structured Interfaces

Many system designers address the problems of the intuitive design approach discussed above by introducing explicit interaction cues to constrain what the user will say. For example, the user might be presented with a voice menu of legal words and their meanings at each junction of a dialog. Such interfaces generally work, but are tedious to use. The user must learn the rules of the interface: what to say, and how and when to say it. Such restrictions result in highly rigid interfaces with a scripted feel. Over time a user may adapt to the interface and become an efficient user, but the potential for natural communication is not realized.

6.3.3 The Brute Force Approach

A third approach is to collect massive amounts of data of people performing a specific task. This data can be used to train statistical word-to-action and phrase-to-action

mappings. This approach has been taken by DARPA[2] sponsored research in two task domains, the Resource Management (RM), and the Airline Travel Information System (ATIS) task. Similar data sets have also been collected in Europe, for example, for the task of rail reservations. Systems built using such training databases have proven to be successful in practice (e.g., [127, 12]).

This approach has two major drawbacks. First, it only works for domains in which input utterances are typically long and contain many words. In such utterances the chance of semantic ambiguity is reduced. For example in the ATIS task, a person is unlikely to issue a one or two word command while reserving a flight (unless answering a directed question). In contrast, many command and control tasks elicit short utterances which are highly ambiguous [39]. For such domains, collecting large amounts of data will not help since the semantic mappings of different users will conflict, making it impossible to train a single average user model.

A second drawback is the high cost in creating a large corpus of sample interactions. The ATIS and RM tasks were monumental efforts which have not been replicated for other domains because of the enormous costs involved. Without a framework for migrating systems to new domains, the brute force approach is too expensive for widespread use.

6.4 Adaptive Interfaces

People overcome surface and semantic variations by adapting to their communication partners [21]. Speech interfaces must contain similar adaptive capabilities if we expect robust and natural spoken interactions with machines. Furnas et al.'s studies motivate the need for spoken language interfaces with adaptive capabilities. The vocabulary of a speech interface should not be preprogrammed. Instead, it should learn to reflect the language usage patterns of each individual user. In a review of speech and text

[2]The United States Defense Advanced Research Projects Agency.

HCI techniques, Hayes and Reddy suggest that it would be highly desirable to build an interface which could adapt to the "idiosyncrasies of individual users" [52, page 240].

To address the problems of acoustic and semantic variability, we may frame spoken HCI as a learning problem. The interface should acquire the acoustic models [3] and sound-to-meaning mappings for each user. Ideally, the interface will efficiently and naturally learn the language patterns of individual users through continuous and natural interactions. The CELL model provides a learning engine for such interfaces since CELL simultaneously addresses problems of acoustic and semantic learning.

A great deal of effort in speech recognition and understanding research has focused on the problems of large vocabularies and speaker independence. Adaptive interfaces shift the emphasis to problems of selecting *appropriate* vocabularies and acquiring accurate speaker *dependent* acoustic and semantic models. This emphasis will lead to personal interfaces which will be in much greater demand than anonymous public services such as information kiosks and telephone network services [83].

An important issue in the design of adaptive interfaces is to ensure a simple and intuitive protocol for teaching the system. The failure of the programmable VCR provides a lesson on the importance of intuitive interfaces [4]. An ideal adaptive interface will combine a powerful learning system with a natural teaching interface. Infants fit this description, and thus a model of infant learning is a natural starting point for creating an adaptive interface.

[3] As mentioned above, techniques of speaker adaptation may be used for unsupervised learning of a person's acoustic characteristics. These methods are effective but are not considered further in this chapter. They may be integrated in the future to yield improved results.

[4] It seems that only young children can actually program VCRs. This suggests that with sufficient human adaptation even a poorly designed interface may be useful. Our goal is to shift some of this adaptation into the machine and remove the burden from the user so that even adults can use them.

6.5 Related Work

On-line machine learning has been applied to create user interfaces which adapt to users' individual preferences and patterns of use [24, 68]. Langley divides adaptive interfaces into two main classes: *Informative* intefaces help filter information based on an adaptive model of the user's interests. *Generative* interfaces generate autonomous actions to aid the user. Examples of informative adaptive interfaces include systems which recommend web pages [86] and music [113]. Generative interfaces have been applied to domains including automated form filling [54], and meeting scheduling [32]. Lieberman has developed a graphical editor which learns new procedures by example [70]. By observing the user as he performs actions, the system learns dependencies between graphical objects and interface operations. These dependencies are generalized to make future interactions more efficient.

6.6 Incorporating CELL into Human-Computer Interfaces

The CELL architecture provides a framework for spoken adaptive human-computer interfaces. We have developed a series of prototypes based on CELL [101, 103, 100, 102, 85][5]. In these interfaces, CELL receives linguistic input from a microphone, and contextual input from one or more devices (Figure 6-1). The contextual channels carry information about the task being performed. CELL searches for acoustic units and contextual categories which have high mutual information.

To understand the concept of adaptive interfaces, consider the task of catalog browsing. Let's assume a point-and-click interface exists for browsing and selecting clothing items by viewing an array of images. Speech input may be used to assist in

[5]These interfaces use early versions of the CELL model. Although all versions include the core functionality of automatic lexical acquisition, they use varying subsets of the CELL components presented in this thesis.

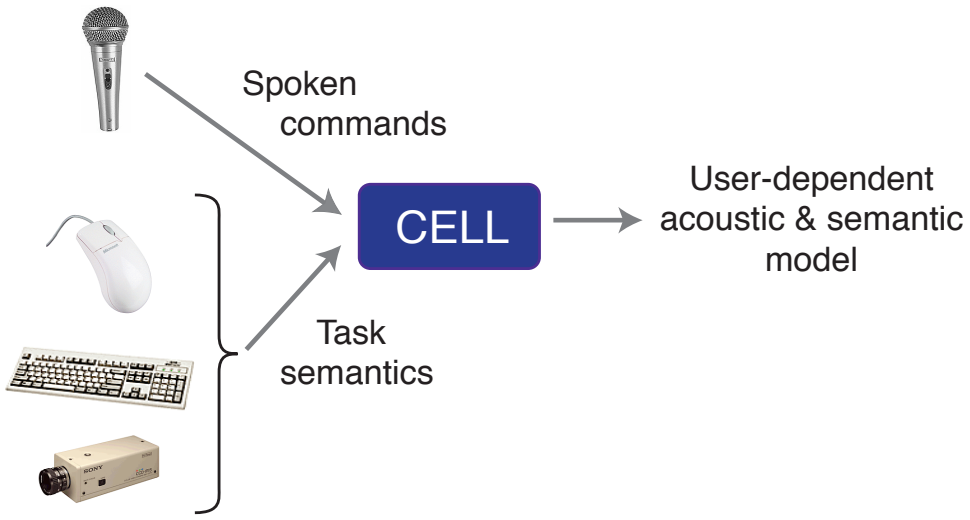


Figure 6-1: Adaptive spoken interfaces based on CELL.

the browsing and selection process. An advantage to using speech is that items not in view may be retrieved. The semantic mapping problem arises in this situation. Users might express their intentions using widely varying terms. For example, one user might call a shirt "ugly" and want to avoid similar shirts in the future. Another user may call the same shirt "funky" and want to purchase it. Even perfect speech recognition will not solve this problem. Simply knowing the text of what a person says does not necessarily reveal their intent.

We can embed CELL in this catalog interface. As the user accesses the catalog with mouse clicks, he may simultaneously express his choices verbally. CELL receives linguistic input from a microphone, and contextual input from visual representations of selected images. The visual representations might include descriptions of shape, color, size, and texture. CELL would learn from on-line interactions. At first, the user would rely entirely on mouse clicks. Over time, CELL would learn a lexicon tuned to that user. As lexical items appear in LTM, the user may use speech in addition to the mouse to access the catalog. The semantic problem would be addressed since the system would learn appropriate associations between acoustic units and visual categories.

Figure 6-2: A player sits in front of a large screen which shows animated graphics. Overhead cameras are used to track the persons hands, and a head-worn microphone senses the player's speech.

We now describe two implemented prototypes of spoken adaptive interfaces.

6.6.1 An Entertainment Application

An early implementation of CELL was incorporated into a multimodal interface for an interactive game [100]. A player sat in front of 60-inch display screen and wore a head-mounted microphone (Figure 6-2). The interface used a combination of vision-based hand tracking and speech input. Two cameras were mounted above the screen and directed towards the player. The images from both cameras were combined to recover the three-dimensional position of the player's hands [4].

The game was entitled "Toco the Toucan" and featured a graphical character named "Toco". The object of the game was to create a mate for Toco. There were six interactive scenes (Figure 6-3). In the first scene the person was asked by a recorded voice prompt to point to locations on a rainbow and speak the name of

the color. Pointing gestures were used to provide contextual input for CELL. The location of the pointing gesture was converted into a RGB value corresponding to the color of the selected portion of the rainbow. The co-occurring speech was paired with this context. After several speech and gesture interactions, a lexicon of color terms was acquired.

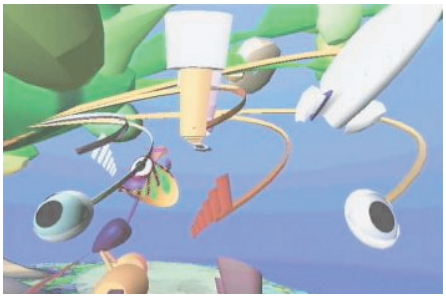
In the second scene, the player selects a part from the "Tree of Life", a fanciful tree from which feathers, beaks, and eye balls could be selected as parts for the mate-to-be. In the third scene, the player used speech to specify the color of the object. The player's speech was compared with each of the lexical items acquired in Scene 1 and the color associated with the best match was applied to the selected part. The story line culminates with the creation of Toco's mate.

The game was demonstrated at Siggraph, an annual large public exhibition [100]. Attendees were invited to participate in the interaction without preparatory instructions for using the interface. Over the course of 5 days, more than 400 hundred people successfully completed the interaction. Although formal usability tests were not conducted, the large number of successful interactions and positive public opinion indicated that the interface was highly successful. As one indication of the game's success, the *Los Angeles Times* selected it as one of the most "cutting edge" exhibits out of the hundreds of displays at the gathering [61].

The success of this system illustrates the potential of adaptive interfaces. The environment was extremely noisy, and there was high variability in how people named colors[6]. By naturally incorporating acoustic and semantic adaptation into the story line, we were able to overcome these problems.

[6]In many instances, the color region was named differently by different players. In addition to common color naming disagreements such as blue-green and purple-violet, players often invented playful adjectives such as *devilish red* and *grasshopper green*.

Scene 1: User points to three colors in the rainbow and names them (lexical acquisition)



Scene 2: User selects a part from the "Tree of Life" by pointing to the part

Scene 3: Part is colored by speech using one of the three lexical items learned in Scene 1

Scene 4: User must select position for new body part using gesture, confirm with speech

Scene 5: A successfully placed part

Scene 6: After two more cycles of Scenes 2-5 the mate is complete and Toco looks on in new-found love

Figure 6-3: Scenes from the "Toco the Toucan" interaction.

6.6.2 A Real-Time Application of CELL with a Robotic Interface

The complete implementation of CELL (Chapter 4) was incorporated into a real-time speech and vision interface embodied in a robotic character (Figure 6-4). Input consists of continuous multiword spoken utterances and images of objects acquired from a CCD camera mounted on the robot. The visual system extracts both color and shape representations of objects which serve as contextual channels for CELL. To teach the system, a person places an object in front of the robot and describes it. Using on-line learning, the system builds a lexicon of color and shape terms grounded in microphone and camera input. Once a lexicon has been acquired, the robot can be engaged in an object labeling task (i.e., lexical generation), and an object selection task (i.e., lexical understanding).

The task of learning color and shape terms was chosen for exploratory purposes, and was not aimed at any particular application domain[7]. Specific application domains are discussed in Section 6.7.

Components of the Robot

The robot is an extension of the active camera system described in Section 4.1.2. The orientation of the camera is determined by a four degree-of-freedom motorized armature. The robotic embodiment facilitates several modes of output including direction of gaze, facial expressions, and spoken output.

Direction of gaze We chose a miniature camera which was embedded in the right eye ball of the robot. The direction of the camera's focus is apparent from the physical orientation of the robot and provides a simple mechanism for establishing joint attention.

[7]Acquisition of shape and color terms could be used for the catalog browsing task discussed earlier.

Figure 6-4: A robotic embodiment of CELL for real-time interaction.

Facial Expressions Several servo-controlled facial features are used to convey information about the internal state of CELL naturally. Movable eyelids which blink after random time intervals produce a life-like illusion. The eyes are kept open when a view-set is being gathered. This is necessary since the camera is mounted beneath the eyelids, and also provides the person with a natural cue that the system is visually attentive. Feathers were mounted on the head which move to provide information of the state of the system. They extend to an attentive pose when the audio processing system detects the start of an utterance. If for some reason the audio processor is not functioning, perhaps due to low microphone levels, the behaviour of the robot conveys this information naturally. The robot's beak can also open and close. Its motions are coordinated with speech output giving the appearance that speech is being generated by the robot.

Spoken Output A phoneme-based speech synthesizer[8] is used to convey internal representations of speech segments. A Viterbi decoder was used to extract the most likely phoneme sequence for a given segment of speech (see Section 4.4.1). This phoneme sequence was resynthesized using the phoneme synthesizer. Naturalness of output is improved by controlling the duration of individual phonemes based on observed durations in the Viterbi decoding.

Acquiring a Lexicon

The robot has three modes of operation: acquisition, generation, and understanding. In the acquisition mode, the robot searches for the presence of objects on the viewing surface. When an object is detected, the system gathers multiple images to build a view-set of the object. If a spoken utterance is detected while the view-set is being gathered, an LS-event is generated and processed by CELL. Lexical items are

[8]We use the TrueTalk speech synthesizer made by Entropic Research Laboratory, Inc. 600 Pennsylvania Ave. SE, Suite 202, Washington, DC 20003.

generated when high mutual information is found between an L-unit (acoustic unit) and either a shape or color category.

To teach the system, the user might place a cup in front of the robot and say, "Here's my coffee cup". To verify that the system has received contextualized spoken input, it "parrots" back the user's speech based on the recognized phoneme sequence (see Section 6.6.2). This provides a natural feedback mechanism for the user to understand the nature of internal representations being created by the system.

The system acquires word order statistics (Section 3.3.4) for the simple case of learning the order of shape and color terms in adjacent positions without intervening words. Lexical items are assigned to either the shape or color class depending on their contextual grounding. The system tracks the distribution of color-shape and shape-color terms for input utterances. In experiments, the system learned that color terms precedes shape terms in English.

Object Description using an Acquired Lexicon

Once lexical items are acquired, the system can generate spoken descriptions of objects. In this mode, the robot searches for objects on the viewing surface. When an object is detected, the system builds a view-set of the object and compares it to each lexical item in LTM. The L-prototype of the best matching item is used to generate a spoken response. The spoken output may describe either shape or color depending on the contextual grounding of the best match.

To use word order statistics, a second generation mode finds the best matching LTM item for the color and shape of the object. The system generates speech for both aspects of the object. The order of concatenation is determined by the acquired word order statistics. When presented with a tennis ball, the robot would say "yellow ball" when it had already learned the words "yellow" and "ball".

Speech Understanding using an Acquired Lexicon

When in the speech understanding mode, the system waits for the user to name objects in terms of shape and color[9]. The input utterance is matched to existing speech models in LTM. A simple grammar allows either single words or word pairs to be recognized. The transition probabilities between word pairs is determined by the acquired word order statistics.

In a second step, the system finds all objects on the viewing surface and compares each to the visual models of the recognized lexical item(s). In a forced choice, it selects the best match and returns the robot's gaze to that object. To provide additional feedback, the selected object is used to index back into LTM and generate a spoken description. This feedback leads to revealing behaviours when an incorrect or incomplete lexicon has been acquired. The nature of the errors provides the user with guidance for subsequent training interactions.

6.7 Application Domains

Adaptive spoken interfaces may be applied to a variety of domains for both human-machine communication and computer-mediated human-human communication. Adaptive spoken interfaces are suitable for tasks with small to mid-sized vocabularies and large expected individual variations in word choice and pronunciation. Many command and control tasks fit these characteristics. A moderate sized vocabulary which is adapted to a user's acoustic and semantic characteristics may facilitate robust voice-activated control. Several application areas are identified below.

Entertainment Current video game environments do not utilize spoken input. Our experiences suggest that spoken input leads to compelling and enjoyable interactions [100]. Synthetic characters which can learn to communicate with human players using speech holds potential for innovative forms of entertainment.

[9]The input utterance is assumed to contain only lexical items in LTM.

Education Spoken adaptive interfaces may be used as tools in constructionist learning [84]. Constructionism emphasizes that learning is an active process. Rather than explicitly teaching a set of rules which govern a concept, we can make systems which enable active discovery of the same ideas. Computational models of language acquisition may be used to develop systems which enable the learner to explore ideas about language. For example, using CELL, they might explore the nature of word-to-meaning mappings, and the concept of word classes and their relation to the physical world. Such a system provides an alternative method for learning about grammar.

Assistive Aids Individuals with communication impairments may use an adaptive interface as a tool for communication. Standard interfaces are often inefficient for this population given that there are large individual difference in speech patterns both within and between speakers. Patel and Roy describe a prototype communication aid for individuals with severe speech impairments based on CELL [85]. The user creates a custom lexicon which maps their vocalizations to symbols on a touch pad. Each symbol is linked to a prerecorded word or phrase such as greetings or requests which can be generated by the system. Over time the system learns to translate a small set of vocalizations to clearly articulated words and phrases. Such a device may be used to communicate with unfamiliar listeners who might not otherwise understand their speech patterns.

Device Control Speech is useful for controlling devices when a person's eyes and hands are busy. For example, when driving a car, it is highly desirable to use speech to control the audio system, telephone, and climate control. Rather than force the driver into using a predefined vocabulary, CELL would enable the driver to easily configure the interface. The existing interface to the devices could be instrumented, providing contextual input for CELL. To configure the interface, the driver would speak while performing task with the non-speech interface. CELL could then acquire a lexicon which connects spoken commands

to actions. Adaptive spoken control of devices may also find application in other personal spaces including the office and the home.

6.8 Scalability

A potential disadvantage of adaptive interfaces is prolonged training time. Depending on the size of vocabulary and complexity of task, lexical acquisition may be impractically slow. Drivers may be willing to teach their cars a vocabulary of 20-30 words, but users may not be willing to teach a catalog browser several thousand words.

An underlying assumption of CELL is that learning begins with an empty LTM. This assumption reflects our interest in modeling the earliest stages of lexical learning. For practical applications, however, the lexicon may be bootstrapped with a default set of lexical items. With interactions, the system can remove items which do not fit the user's patterns and add new items when needed. In effect, our approach provides an adaptive layer to existing speech understanding systems. This layer simultaneously adapts acoustic models and sound-to-meaning mappings.

The default lexicon may be built using conventional means: collect data from a large number of users and generate models which best represent the population. This lexicon will then become personalized over time as an individual interacts with the system.

An exciting alternative is for communities of users connected over a network to share data as a way to produce a default lexicon. Automatic lexical acquisition from communities of networked users may lead to robust systems in a decentralized manner. Such collaborative methods have been shown to be a powerful alternative to centralized control in the context of creating the Linux operating system, a large scale software development [95]. This method is particularly attractive for developing multilingual systems from the ground up.

Chapter 7

Conclusions

7.1 Contributions

In this thesis we presented a model of early lexical acquisition which captures structure between channels of sensory grounded input. Specific contributions of this thesis include:

- A joint solution to three important problems of early lexical learning: (1) Linguistic unit segmentation and discovery, (2) Semantic category formation, and (3) Cross-situational inference of word-to-semantic mappings.

- The first implemented model of early stages of language acquisition which is fully grounded in raw sensory input (CELL). Integrating techniques from speech processing and recognition, computer vision, and machine learning, we have implemented a real-time system which successfully demonstrates on-line lexical learning from natural audio-visual input.

- The first successful evaluation of a model of lexical acquisition using raw infant-directed speech and visual input. The model successfully acquired a lexicon of visually grounded words from the speech of six caregivers. When compared to an acoustic-only model, CELL performed significantly better on measures of

speech segmentation, word discovery, and word-to-semantics mappings.

- A new distance metric for discovering recurrent segments of speech from sets of continuously spoken utterances. A dynamic programming search operates on arrays of phoneme probabilities to locate acoustically similar segments of speech.

- A new framework for adaptive spoken human-computer interfaces has been developed. Interfaces interactively adapt acoustic and semantic models for individual users. The approach holds promise in a variety of application domains including device command and control, entertainment, assistive technology, and multilingual systems.

The CELL model simultaneously discovers linguistic unit boundaries, perceptually grounded semantic categories, and word-to-meaning mappings. This is an important shift from the common assumption that segmentation of the acoustic stream occurs prior to word-to-semantics learning. We have demonstrated that both can happen together, and that in fact, knowledge about one helps accelerate learning about the other.

7.2 Future Directions

Recent advances in perceptual computing enable researchers to apply a new set of tools to age-old questions concerning the nature of language acquisition. This thesis represents a first step in this direction. The work in this thesis has raised many new questions which may be explored in future studies. Some of the most interesting of these directions include:

Robust Sensory Processing The acoustic and visual sensory processors presented in Chapter 4 are stable in environments with low background noise. An important area for future study will be to incorporate robust processing methods into

the CELL architecture. We believe this will require new methods for low level representations of signals, and also higher level models of attention which focus learning on appropriate aspects of the environment.

Expanded Visual Semantic Domains The current implementation of CELL can only learn words which refer to shapes and colors. The semantic domains may be expanded to include reference to spatial relations between objects, people, motions of objects and people.

Additional Input Modalities Infants born blind acquire language in very similar ways to sighted infants [66]. Semantics are grounded not only in vision, but also touch, proprioception, and non-speech auditory signals. Input channels derived from these sensors may be added for richer semantic categories and to broaden the application domains.

Beyond Sensory Grounding Some early words learned by infants do not seem to be grounded in sensory input alone. For example, common words such as *good* and *no* are learned by young infants. To ground such words, we need to model the internal affective and motivational states of the language learner. Certain words are more easily related to their effect on internal state rather than on the large and possibly infinite set of contexts which may cause the same internal state.

Beyond Associations The core CELL model builds associations between words and perceptual categories. Higher level learning mechanisms may be developed to model structure between associations. These higher level structures need not be tied directly to the perceptual system, but can operate on abstractions derived from cross-channel structure.

Linguistic Structure Syntax is a crucial aspect of language. *Dog bites man* and Man bites dog are two very different stories. Syntax provides a set of rules

for mapping words to conceptual structures based on the order in which words occur. The nature of the early lexicon and the definition of word classes is intimately related to processes of syntax learning. Many new insights may be gained by studying syntax acquisition in a sensory grounded framework.

Learning Speech Sounds of a Language Infants learn the phonetic structure of a language before they produce their first words [64]. They learn to make distinctions of only the speech sounds which affect the meaning of words in their language. CELL does not account for this pre-lexical stage of learning. Various methods of unsupervised clustering may be applied to model this problem in a computational framework with raw acoustic data. An interesting possibility is that early lexical learning interacts with the acquisition of a language's phonetic structure. Over time, speech contrasts which have no effect on semantic mappings are discarded.

Multilingual Systems Speech understanding technologies have been developed for under 20 of the 3000 active languages of the world. The cost for developing technology for a new language is prohibitively high. A large amount of accurately transcribed speech, and a lexicon of the language must be available before a new system may be developed [110]. Interfaces which can acquire models of a language automatically present an alternative approach to supporting new languages. Just as an individual user can transfer knowledge of their particular language usage patterns to CELL, a community of networked users could transfer knowledge of a new language to an extended version of CELL.

7.3 Concluding Remarks

An important theme which runs through the work of this thesis is the interplay between human intelligence and artificial intelligence. On one hand, computational models help us understand complex human behaviors. On the other, insights about

human behaviour enable us to build more intelligent machines.

One might ask whether this thesis, broadly classified, is a contribution to the field of cognitive science or artificial intelligence. The answer is both. We have gained new insights into how an infant might leverage structure across multiple channels of sensory input to learn early words. We have also gained insights on how to build adaptive systems which learn the word-to-meaning mappings of individual users from natural interactions.

Computational models are powerful tools for understanding the behaviour of complex systems. Often a model is proposed and hotly debated, but difficult to test. Computational techniques including perceptual computing, pattern recognition and analysis, and machine learning provide a rich set of tools for building and testing complex models with realistic input. A model which behaviorally matches the abilities of an infant does not imply that we have actually uncovered how infants solve the problem. A model which is implemented and successfully evaluated with realistic data does, however, provide an existence proof that it is at least plausible.

Understanding human intelligence can conversely shed light on problems of creating artificially intelligent systems. This is not to say that the *only* way to AI is to understand humans. We have built airplanes which share little in common with birds, and chess playing machines which share as little with their human counterparts. But these examples notwithstanding, humans are an existence proof of many abilities which we cannot observe and study elsewhere. Studying and modeling human intelligence is *a* path to achieving machine intelligence. Moreover, if our goal is to communicate with these systems, it is important to understand human communication and model it in these systems. Otherwise, we may end up with alien intelligences which cannot be related to in any humanly discernible way.

In an era of unbounded growth in information technologies, cognitive science and artificial intelligence are sure to become two sides of the same coin. Exciting new discoveries are waiting to be made at the intersection of these rich areas of study!

Bibliography

[1] J.R. Anderson. Induction of augmented transition networks. *Cognitive Science*, 1:125–157, 1977.

[2] A. Asadi. *Automatic Detection and Modeling of New Words in a Large Vocabulary Continuous Speech Recognition System.* PhD thesis, Northeastern University, 1991.

[3] R.N. Aslin, J.Z. Woodward, N.P. LaMendola, and T.G. Bever. Models of word segmentation in fluent maternal speech to infants. In James L. Morgan and Katherine Demuth, editors, *Signal to Syntax*, chapter 8, pages 117–134. Erlbaum, Mahwah, NJ, 1996.

[4] A. Azarbayejani and A. Pentland. Real-time self-calibrating stereo person tracking using 3-D shape estimation from blob features. In *Proceedings of 13th ICPR*, Vienna, Austria, August 1996. IEEE Computer Society Press.

[5] D. Bailey. *When push comes to shove: A computational model of the role of motor control in the acquisition of action verbs.* PhD thesis, Computer science division, EECS Department, University of California at Berkeley, 1997.

[6] R. Baillargeon. Object permanence in 3.5- and 4.5-month-old infants. *Developmental Psychology*, 23:655–664, 1987.

[7] D.A. Baldwin. Infant contributions to the achievement of joint reference. In P. Bloom, editor, *Language acquisition: Core readings*, pages 129–152. MIT Press, Cambridge, MA, 1991.

[8] G. H. Ball and D. J. Hall. Isodata, a novel method of data analysis and pattern classification. Technical report, Stanford Research Institute Technical Report, Stanford, CA, 1965. (NTIS AD699616).

[9] M. Barrett. Early lexical development. In P. Fletcher and B. MacWhinney, editors, *The Handbook of Child Language*, chapter 13, pages 363–392. Blackwell, Oxford, UK, 1995.

[10] E. Bates, I. Bretherton, and L. Snyder. *From first words to grammar*. Cambridge University Press, Cambridge, UK, 1988.

[11] E. Bates, V. Marchman, D. Thal, L. Fenson, P. Dale, J.S. Reznick, J. Reilly, and J. Hartung. Development and stylistic variation in the composition of early vocabulary. *Journal of Child Language*, 21:85–124, 1994.

[12] M. Bates, R. Bobrow, P. Fung, R. Ingria, F. Kubala, J. Makhoul, L. Nguyen, R. Schwartz, and D. Stallard. The bbn/harc spoken language understanding system. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 111 –114, 1993.

[13] H. Benedict. Early lexical development: Comprehension and production. *Journal of Child Language*, 6:183–200, 1979.

[14] B. Berlin and P. Kay. *Basic Color Terms: Their Universality and Evolution*. University of California Press, Berkeley, CA, 1969.

[15] C.M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, 1995.

[16] L. Bloom and M. Lahey. *Language Development and Language Disorders*. Wiley, New York, 1978.

[17] A. Bobick and A. Wilson. A state-based approach to the representation and recognition of gesture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(12), 1997.

[18] M.H. Bornstein, W. Kessen, and S. Weiskopf. Color vision and hue categorization in young human infants. *Journal of Experimental Psychology: Human Perception and Performance*, 2:115–129, 1965.

[19] H. Bourlard and N. Morgan. *Connectionist Speech Recognition - A Hybrid Approach*. Kluwer Academic Publishers, 1994.

[20] C. Bregler, H. Hild S. Manke, and A. Waibel. Improving connected letter recognition by lipreading. *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 1993.

[21] S.E. Brennan. Lexical entrainment in spontaneous dialog. In *Proceedings of the 1996 International symposium on Spoken Language Dialogue, ISSD-96*, pages 41–44, Philadelphia, PA, 1996.

[22] M.R. Brent. An efficient, probabilistically sound algorithm for segmentation and word discovery. *Machine Learning*, 1999.

[23] M.R. Brent and T. A. Cartwright. Distributional regularity and phonotactic constraints are useful for segmentation. *Cognition*, 61:93–125, 1996.

[24] D. Browne, P. Totterdell, and M. Norman, editors. *Adaptive User Interfaces*. Academic Press, 1990.

[25] I.W.R. Buschnell. Modification of the externality effect in young infants. *Journal of Experimental Child Psychology*, 28:211–229, 1979.

[26] J. Colombo and R. Bundy. A method for the measurement of infant auditory selectivity. *Infant Behavior and Development*, 4:219–233, 1981.

[27] T.M. Cover and J. A. Thomas. *Elements of Information Theory.* Wiley-Interscience, 1991.

[28] A. Cutler. Segmentation problems, rhythmic solutions. In L. Gleitman and B. Landau, editors, *The Acquisition of the Lexicon*, chapter 2, pages 81–104. MIT Press, Cambridge, MA, 1991.

[29] J. Davis and A. Bobick. The representation and recognition of action using temporal templates. *IEEE Conference on Computer Vision and Pattern Recognition*, 1997.

[30] C. de Marcken. The unsupervised acquisition of a lexicon from continuous speech. Technical Report A.I. Memo 1558, MIT Artificial Intelligence Laboratory, 1996.

[31] C. de Marcken. *Unsupervised Language Acquisition.* PhD thesis, MIT Artificial Intelligence Laboratory, 1996.

[32] L. Dent, J. Boticario, J. McDermott, T. Mitchell, and D. Zaborowski. A personal learning apprentice. In *Proceedings of the Tenth National Conference of the AAAI*, San Jose, CA, 1992. AAAI Press.

[33] V.V. Digalakis, D. Rtischev, and L.G. Nuemeyer. Speaker adaptation using constrained estimation of gaussian mixtures. *IEEE Transactions on Speech and Audio Processing*, 3:357–366, 1995.

[34] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis.* Wiley, 1973.

[35] W.H. Durham. *Coevolution: Genes, Cultures, and Human Diversity.* Stanford University Press, 1991.

[36] P. Eimas, E.R. Siqueland, P.W. Jusczyk, and J. Vigorito. Speech perception in early infancy. *Science*, 171:304–305, 1971.

[37] J. Elman. Finding structure in time. *Cognitive Science*, 14:179–211, 1990.

[38] J. Feldman, G. Lakoff, D. Bailey, S. Narayanan, T. Regier, and A. Stolcke. Lzero: The first five years. *Artificial Intelligence Review*, 10:103–129, 1996.

[39] G.W. Furnas, T.K. Landauer, L.M. Gomez, and S.T. Dumais. The vocabulary problem in human-system communications. *Communications of the Association for Computing Machinery*, 30(11):964–972, 1987.

[40] S. Furui. Unsupervised speaker adaptation method based on hierarchical spectral clustering. In *Proceedings of ICASSP*, pages 286–289, 1989.

[41] D. Gentner. Why nouns are learned before verbs: Linguistic relativity versus natural partitioning. In S. Kuczaj, editor, *Language development: Vol. 2. Language, cognition, and culture*. Erlbaum, Hillsdale, NJ, 1983.

[42] O. Ghitza. Auditory nerve representation as a basis for speech processing. In S. Furui and M. Sondhi, editors, *Advances in speech signal processing*, pages 453–485. Marcel Dekker, NY, 1991.

[43] L. Gleitman. The structural source of verb meanings. In Paul Bloom, editor, *Language acquisition: core readings*, pages 174–221. MIT Press, Cambridge, MA, 1994.

[44] L. R. Gleitman and E. Wanner. Language acquisition: the state of the state of the art. In E. Wanner and L. R. Gleitman, editors, *Language acquisition: the state of the art*, pages 3–48. Cambridge University Press, Cambridge, England, 1982.

[45] R.M. Golinkoff, C.B. Mervis, and K. Hirsch-Pasek. Early object labels: The case for a developmental lexical principles framework. *Journal of Child Language*, 21:125–156, 1994.

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

[47] A.L. Gorin, S.E. Levinson, and A. Sankar. An experiment in spoken language acquisition. *IEEE Transactions on Speech and Audio Processing*, 2(1):224–240, January 1994.

[48] J. Grimshaw. Form, function, and the language acquisition device. In C. L. Baker and J. J. McCarthy, editors, *The logical problem of language acquisition*, pages 165–182. MIT Press, 1981.

[49] R.M. Haralick and L.G. Shapiro. *Computer and Robot Vision*. Addison-Wesley, Reading, MA, 1992.

[50] S. Harnad. The symbol grounding problem. *Physica D*, 42:335–346, 1990.

[51] J. Harrington, G. Watson, and M. Cooper. Word boundary detection in broad class and phoneme strings. *Computer Speech and Language*, 3:367–382, 1989.

[52] P.J. Hayes and R. Reddy. Steps towards graceful interaction in spoken and written man-machine communication. *International Journal of Man-Machine Studies*, 19:231–284, 1983.

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

[54] L.A. Hermens and J.C. Schlimmer. A machine-learning apprentice for the completion of repetitive forms. *IEEE Expert*, 9:28–33, 1994.

[55] B.K.P. Horn. *Robot Vision*. MIT Press, 1986.

[56] M.K. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, IT-8:179–187, 1962.

[57] J. Huttenlocher and P. Smiley. Early word meanings: the case of object names. In P. Bloom, editor, *Language acquisition: core readings*, pages 222–247. MIT Press, Cambridge, MA, 1994.

[58] M. I. Jordan. Serial order: A parallel distributed processing approach. Technical report, Institute for Cognitive Science, University of California, San Diego, 1986. Report 8604.

[59] P.W. Jusczyk. From general to language-specific capacities: the wrapsa model of how speech perception develops. *Journal of Phonetics*, 21:3–28, 1993.

[60] P.W. Jusczyk and R.N. Aslin. Infants' detection of the sound patterns of words in fluent speech. *Cognitive Psychology*, 29:1–23, 1995.

[61] Karen Kaplan. Los angeles times, August 11 1997.

[62] J.B. Kruskal and D. Sankoff. An anthology of algorithms and concepts for sequence comparison. In D. Sankoff and J.B. Kruskal, editors, *Time Warps, String Edits, and Macromolecules*. Addison-Wesley, 1983.

[63] S.A. Kuczaj and M.D. Barrett, editors. *The development of word meaning*. Springer-Verlag, New York, 1986.

[64] P.K. Kuhl, K.A. Williams, F. Lacerda, K.N. Stevens, and B. Lindblom. Linguistic experience alters phonetic perception in infants by 6 months of age. *Science*, 255:606–608, 1992.

[65] J.J. Kulikowski, R. Abadi, and P.E. King-Smith. Orientation selectivity of grating and line detectors in human vision. *Vision Research*, 13, 1973.

[66] B. Landau and L. Gleitman. *Language and Experience: Evidence from the Blind Child.* Harvard University Press, Cambridge, MA, 1985.

[67] B. Landau, L.B. Smith, and S. Jones. The importance of shape in early lexical learning. *Cognitive Development*, 3, 1988.

[68] P. Langley. Machine learning for adaptive user interfaces. In *Proceedings of the 21st German Annual Conference on Artificial Intelligence*, Freiburg, Germany, 1997. Springer.

[69] J. Y. Lettvin, H. R. Maturana, W. S. McCulloch, and W. H. Pitts. What the frog's eye tells the frog's brain. *Proceedings of the IRE*, 47(11):1940–1959, 1959.

[70] H. Lieberman. Mondrian: A teachable graphical interface. In A. Cypher, editor, *Watch What I Do: Programming by Demonstration.* MIT Press, Camdridge, MA, 1993.

[71] S. P. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[72] W. Maguire, N. Weisstein, and V. Klymenko. From visual structure to perceptual function. In K.N. Leibovic, editor, *Vision Science.* Springer-Verlag, 1990.

[73] D.R. Mandel, P.W. Jusczyk, and D.B. Pisoni. Infants' recognition of the sound patterns of their own names. *Psychological Science*, 6(5):314–317, 1995.

[74] E.M. Markman. *Categorization and naming in children.* MIT Press, Cambridge, MA, 1989.

[75] J. Mehler and E. Dupoux. *What Infants Know.* Blackwell, 1994.

[76] J. Mehler, P. Jusczyk, G. Lambertz, N. Halsted, J. Bertoncini, and C. Amiel-Tison. A precursor to language acquisition in young infants. *Cognition*, 29:143–178, 1988.

[77] A.E. Milewski. Infant's discrimination of internal and external pattern elements. *Journal of Experimental Child Psychology*, 22:229–246, 1976.

[78] G.A. Miller. The magical number seven plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63:81–97, 1956.

[79] L.G. Miller and A.L. Gorin. Spoken language acquisition in an almanac retrieval task. Technical report, AT&T Bell Laboratories Technical Memorandum, 1993.

[80] J. Morgan and E. L. Newport. The role of constituent structure in the induction of an artificial language. *Journal of verbal learning and verbal behavior*, 20:67–85, 1981.

[81] J.L. Morgan and K. Demuth, editors. *Signal to Syntax*. Erlbaum, Mahwah, NJ, 1996.

[82] H. Murase and S.K. Nayar. Visual learning and recognition of 3d objects from appearance. *International Journal of Computer Vision*, 14:5–24, 1995.

[83] N. Negroponte. Hospital corners. In Brenda Laurel, editor, *The Art of Human-Computer Interface Design*, pages 347–353. Addison Wesley, 1990.

[84] S. Papert. *Mindstorms: Children, Computers and Powerful Ideas*. Basic Books, New York, 1980.

[85] R. Patel and D.K. Roy. Teachable interfaces for individuals with dysarthric speech and severe physical disabilities. In *Integrating Artificial Intelligence and Assistive Technology*, pages 40–47, 1998.

[86] M. Pazzani, J. Maramatsu, and D. Billsus. Syskill and webert: Identifying interesting web sites. In *Proceedings of the Thirteenth National Conference of the AAAI*, Portland, OR, 1996. AAAI Press.

[87] A.M. Peters. *The Units of Language Acquisition.* Cambridge University Press, 1983.

[88] J.M. Pine. The language of primary care givers. In C. Gallaway and B.J. Richards, editors, *Input and Interaction in Language Acquisition*, chapter 1, pages 15–37. Cambridge University Press, Cambridge, UK, 1994.

[89] S. Pinker. *Language learnability and language development.* Harvard University Press, Cambridge, MA, 1984.

[90] S. Pinker. Personal communication, 1999.

[91] W.V.O. Quine. *Word and Object.* MIT Press, Cambridge, MA, 1960.

[92] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, 1989.

[93] L.R. Rabiner and B.H. Juang. *Fundamentals of Speech Recognition.* Prentice Hall, 1993.

[94] R.P.N. Rao and D.H. Ballard. Object indexing using an iconic sparse distributed memory. In *International Conference of Computer Vision*, pages 24–31, 1995.

[95] E.S. Raymond. The cathedral and the bazaar, 1997. http://www.tuxedo.org/ esr/writings/cathedral-bazaar/index.html.

[96] T. Regier. *The human semantic potential.* MIT Press, Cambridge, MA, 1996.

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

[98] R. Rose. Word spotting from continuous speech utterances. In C.H. Lee, F. K. Soong, and K.K. Paliwal, editors, *Automatic Speech and Speaker Recognition*, chapter 13, pages 303–329. Kluwer Academic, 1996.

[99] K. Ross and M. Ostendorf. A dynamical system model for recognizing intonation patterns. In *Proceedings of Eurospeech*, September 1995.

[100] D.K. Roy, M. Hlavac, M. Umaschi, T. Jebara, J. Cassell, and A. Pentland. Toco the toucan: A synthetic character guided by perception, emotion, and story. In *Visual Proceedings of Siggraph*, Los Angeles, CA, August 1997. ACM Siggraph.

[101] D.K. Roy and A. Pentland. Multimodal adaptive interfaces. Technical Report 438, MIT Media Lab Vision and Modeling Group, 1997.

[102] D.K. Roy and A. Pentland. Learning words from natural audio-visual input. *Proceedings of the International Conference on Spoken Language Processing*, 1998.

[103] D.K. Roy and A. Pentland. Word learning in a multimodal environment. In *Proceedings of ICASSP*, Seattle, Washington, May 1998. IEEE Computer Society Press.

[104] D.K. Roy, B. Schiele, and A. Pentland. Learning audio-visual associations from sensory input. In *Proceedings of the International Conference of Computer Vision Workshop on the Integration of Speech and Image Understanding*, 1999. (To appear).

[105] D.E. Rumelhart and J.L. McClelland. On learning the past tenses in english verbs. In J. L. McClelland, D. E. Rumelhart, and The PDP Research Group, editors, *Parallel distributed processing.* Bradford Books/MIT press, Cambridge, MA, 1986.

[106] D.E. Rumelhart and J.L. McClelland. *Parallel Distributed Processing.* Bradford Books / MIT press, 1986.

[107] J. Saffran, R. Aslin, and E. Newport. Statistical learning by 8-month-old infants. *Science*, 274:1926–28, 1996.

[108] A. Sankar and A. Gorin. *Adaptive language acquisition in a multi-sensory device*, pages 324–356. Chapman and Hall, London, 1993.

[109] B. Schiele and J.L. Crowley. Probabilistic object recognition using multidimensional receptive field histograms. In *ICPR'96 Proceedings of the 13th International Conference on Pattern Recognition, Volume B*, pages 50–54, August 1996.

[110] R. Schwartz. Personal communication, 1999.

[111] R. Schwartz, Y.-L. Chow, and F. Kubala. Rapid speaker adaptation using a probabilistic spectral mapping. In *Proceedings of ICASSP*, pages 633–636, 1987.

[112] S. Seneff and V. Zue. Transcription and alignment of the timit database. In *Proceedings of the Second Symposium on Advanced Man-Machine Interface through Spoken Language*, Oahu, Hawaii, November 1988.

[113] U. Shardanand. Social information filtering for music recommendation. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, 1994.

[114] L. Siklóssy. Natural language learning by computer. In H.A. Simon and L. Siklóssy, editors, *Representation and Meaning: Experiments with Information Processing Systems*. Prentice-Hall, Englewood Cliffs, NJ, 1972.

[115] J. Siskind. *Naive Physics, Event Perception, Lexical Semantics, and Language Acquisition*. PhD thesis, Massachusetts Institute of Technology, 1992.

[116] J. Siskind. A computational study of cross-situational techniques for learning word-to-meaning mapping. *Cognition*, 61(1-2):39–91, 1996.

[117] C.E. Snow. Mother's speech to children learning language. *Child Development*, 43:549–565, 1972. speech is redundant in motherese.

[118] C.E. Snow. Mothers' speech research: from input to interaction. In C. E. Snow and C. A. Ferguson, editors, *Talking to children: language input and acquisition.* Cambridge University Press, Cambridge, MA, 1977.

[119] T. Starner and A. Pentland. Visual recognition of american sign language using hidden markov models. In *International Workshop on Automatic Face and Gesture Recognition (IWAFGR)*, Zurich, Switzerland, 1995.

[120] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1), 1991.

[121] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K.J. Lang. Phoneme recognition using time delay neural networks. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37:328–339, 1989.

[122] P. Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78:1150–1160, 1990.

[123] J.F. Werker and C.E. Lalonde. The development of speech perception: Initial capabilities and the emergence of phonemic categories. *Developmental Psychology*, 24:672–683, 1999.

[124] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Real-time tracking of the human body. *IEEE Trans. Pattern Analysis and Machine Vision(???)*, 19(7), 1997.

[125] J.H. Wright, M.J. Carey, and E.S. Parris. Statistical models for topic identification using phoneme substrings. In *Proceedings of ICASSP*, pages 307–310, 1996.

[126] B.A. Younger and L.B. Cohen. Infant perception of correlations among attributes. *Child Development*, 54:858–867, 1983.

[127] V. Zue, J. Glass, M. Philips, and S. Seneff. The mit summit speech recognition system: A progress report. In *Proceedings DARPA Speech and Natural Language Workshop*, pages 179–189, 1989.