INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.
Layered abduction for speech recognition from articulation

Fox, Richard Keith, Ph.D.
The Ohio State University, 1992
LAYERED ABDUCTION FOR SPEECH RECOGNITION FROM ARTICULATION

DISSERTATION

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By

Richard Keith Fox, B.S., M.S.

* * * * *

The Ohio State University

1992

Committee:

B. Chandrasekaran
John Josephson (co-advisor)
Osamu Fujimura
Terry Patten

Approved by

Advisor

Co-advisor

Department of Computer and Information Science
In loving memory of my mother
ACKNOWLEDGEMENTS

There are many people whom I am indebted to. I wish to thank my co-advisors, B. Chandrasekaran (Chandra) and John Josephson (JJ). I wish to thank Chandra for being a wonderful teacher and inspiring me to investigate the mysteries of AI. His lab has been a great workplace and I would not have even considered working on my PhD without his teachings. I have learned more from him than I could have possibly imagined. I am also deeply indebted to JJ, who among other things, has been a friend and a person I could talk to about anything. He has injected his obsession of abduction into me. Without this, I would not have cared to get a phd. Thank you both, for your time, your teachings, your guidance and leadership, and your patience.

I would also like to express my gratitude to Osamu Fujimura, who acted as an advisor on my dissertation committee, and offered me a glimpse into his endless knowledge of linguistics and speech. Osamu has been a wonderful leader for our speech group and has been very sympathetic towards my work and all of the unavoidable delays that have occurred over the last year and a half. Thanks also to Donna Erickson, Reiner Wilhelms and Cari Spring, you have all been very helpful with comments about my work and adding to my knowledge of speech and linguistics. Also thanks to Kevin Lenzo for helping out by running cases during the last 6 months, adding knowledge to ARTREC, and by continuing the work that I have started at
the LAIR. I am indebted to Mary Beckman who supported me for two quarters while working on a previous speech recognition project. I would also like to express my gratitude to other members of that project, Ashok Krishnamurthy, Rob Fox, Larry Feth, Benjamin Ao, and John Barnes for all of the knowledge and help they provided me.

The CIS department has been a wonderful work place. Both the grad students and staff have helped me throughout my 6 years here. I would like to express thanks to everyone in the main office, especially Tom Fletcher and Deanna, also to Jeff Steinbeck and Jane Grissom for all of their help throughout the years. I would also like to express deep thanks to members of my Generals committee, Dik Lee and especially D. Jayasimha for his support and help with my computer architecture minor. Also thanks to Terry Patten for being on my dissertation committee and helping me finalize this document.

I would also like to thank the other LAIR folks for conversation, friendship and exchange of ideas. A big thank you to Pete Angeline, my officemate, for putting up with me over the last couple of years, and for allowing me to play my music in our office and for giving up Neuron for many many weeks during the last 2 quarters. Thanks also to the LAIR secretaries, Sandy, Danny, Radha and Sandra. And thanks to the other AI grad students, Dave Herman, Barbara Becker, John Kolen, Dave Stucki, Sunil Thadani, and Hari Narayarn. Thanks to the friendship of Mike Weintraub, Anne Keuneke, Dean Alleman, Ben Blake, Joe Hollingsworth, Matt Dejongh, Denise Byrnes, and Olivier Fischer, I will always remember you, you have all been great.
friends.

A very big thank you to Diana Smetters for almost single-handedly implementing the Integrated Generic Task Toolset, of which I used extensively in my dissertation work, and Arun Welch for helping me with the Toolset and for teaching me lisp, common lisp, clos, and many other things about computers. Thanks to Bryan Dunlap for all of your help with machine support and various conversation about inane things!

A very deep thanks for the love and support of my family. I wish my mother could have lived to see this day and this document. With all my love, this is for my dad and my sisters and in loving memory of my mother. See, it only took 6 years dad!

Finally, but certainly most importantly, thank you to Julie. For your love and support, your patience and understanding.

I would also like to acknowledge the funding sources which provided me with the research, both directly and indirectly. Thanks to John Westbury and the gang at the Madison Wisconsin Microbeam facility for providing me with pellet data. Their facility is supported by USPHS Grant NS16373 and NIH USPHS Grant DC00820. Additional research has been supported by a gift from NEC Corporation to Osamu Fujimura. My research and other LAIR research has been funded at various stages by DARPA and the Air Force Office of Scientific Research under the grant F49620-89-C-0110, and by the National Science Foundation under grant CBT-8703745.
Vita

February 23, 1964 .................... Born—St. Louis, Missouri

1986 ................................. B.S. Computer Science
University of Missouri, Rolla
Rolla, Missouri

1988 ................................. M.S. Computer Science
The Ohio State University
Columbus, Ohio

1988-1990 .......................... Research Associate
Laboratory for Artificial Intelligence Research
The Ohio State University
Columbus, Ohio

Department of Computer and Information
Science
The Ohio State University
Columbus, Ohio

1988-1989 .......................... Administrative Associate
Department of Computer and Information
Science
The Ohio State University
Columbus, Ohio

Proceedings and Technical Reports


R. Fox, J. Josephson, CV Experiment and Results, LAIR technical report, 1990.


Fields of Study

Major Field: Computer Science

Specializations:
   Artificial Intelligence (B. Chandrasekaran)
   Cognitive Science (J. Josephson)
   Computer Architecture (D. Jayasimha)
## Table of Contents

DEDICATION ......................................................................................................................... ii

ACKNOWLEDGEMENTS ................................................................................................ iii

VITA ........................................................................................................................................... vi

LIST OF FIGURES .............................................................................................................. xii

LIST OF TABLES .................................................................................................................. xiv

CHAPTER PAGE

I Introduction ..................................................................................................................... 1

1.1 Abduction as a Generic Problem Solving Strategy .................................. 5
1.1.1 A Definition of Abduction ................................................................. 7
1.1.2 A Better Definition of Abduction ................................................... 9
1.2 Problems Come in Layers .............................................................................. 12
1.2.1 A Mechanism for Abduction in Layers ........................................ 15
1.2.2 Perception in Layers............................................................................ 18
1.3 Speech Recognition ............................................................................................. 26
1.3.1 The Problems of Automated Speech Recognition ..................... 27
1.3.2 Articulation .............................................................................................. 30
1.3.3 An Example of Articulatory Recognition ..................................... 34
1.4 The Contributions and Scope of this Work .............................................. 35
1.5 A Guide to Reading this Dissertation .................................................... 37

II A Survey of Related Works ..................................................................................... 39

2.1 Previous Work in Speech Recognition ......................................................... 39
2.1.1 Hearsay-II ................................................................. 39
2.1.2 Harpy ................................................................. 40
## ARTREC - a Layered Abduction Speech Recognition System using Articulation as Input

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Introduction</td>
<td>127</td>
</tr>
<tr>
<td>4.2 Articulatory Recognition</td>
<td>127</td>
</tr>
<tr>
<td>4.2.1 Why Articulation?</td>
<td>130</td>
</tr>
<tr>
<td>4.2.2 Articulatory Input - Microbeam Pellet Data</td>
<td>133</td>
</tr>
<tr>
<td>4.3 ARTREC - Knowledge-based System for Articulatory Recognition of Speech</td>
<td>137</td>
</tr>
<tr>
<td>4.3.1 Experimental Data</td>
<td>138</td>
</tr>
<tr>
<td>4.3.2 Lexicon for Pine Street</td>
<td>140</td>
</tr>
<tr>
<td>4.3.3 Detailed Description of the Flow of Processing</td>
<td>141</td>
</tr>
<tr>
<td>4.3.4 Confidences</td>
<td>151</td>
</tr>
<tr>
<td>4.3.5 Types of Articulatory Knowledge</td>
<td>154</td>
</tr>
<tr>
<td>4.4 Conclusion</td>
<td>166</td>
</tr>
</tbody>
</table>

## An Example of ARTREC

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Introduction</td>
<td>167</td>
</tr>
<tr>
<td>5.1.1 Events and Syllable Boundaries</td>
<td>169</td>
</tr>
<tr>
<td>5.1.2 Gesture Hypothesization</td>
<td>169</td>
</tr>
<tr>
<td>5.1.3 Syllable Hypothesization</td>
<td>175</td>
</tr>
<tr>
<td>5.2 Emphasis Detection</td>
<td>179</td>
</tr>
<tr>
<td>5.3 Performance and Complexity</td>
<td>180</td>
</tr>
<tr>
<td>5.4 Conclusion</td>
<td>182</td>
</tr>
</tbody>
</table>

## Experimental Results

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 Introduction</td>
<td>183</td>
</tr>
<tr>
<td>6.2 ARTREC Experiments and Terminology</td>
<td>183</td>
</tr>
<tr>
<td>6.2.1 Experiment 1: Single Abduction</td>
<td>185</td>
</tr>
<tr>
<td>6.2.2 Experiment 2: Bottom-Up Layered Abduction</td>
<td>188</td>
</tr>
<tr>
<td>6.2.3 Experiment 3: Pessimistic Bottom-Up Layered Abduction</td>
<td>189</td>
</tr>
<tr>
<td>6.2.4 Experiment 4: Full Layered Abduction</td>
<td>191</td>
</tr>
<tr>
<td>6.3 Discussion of ARTREC Results</td>
<td>194</td>
</tr>
<tr>
<td>6.4 Prosodic Emphasis</td>
<td>195</td>
</tr>
<tr>
<td>6.5 Conclusion</td>
<td>198</td>
</tr>
</tbody>
</table>
VII Implications of this work ................................................................. 200
  7.1 Introduction .................................................................................. 200
  7.2 Uses of Articulation in Speech Recognition .................................. 200
    7.2.1 How ARTREC can be used to aid Acoustic Speech Recognition 203
    7.2.2 Making Speech Non-Linear ..................................................... 206
    7.2.3 Place of Articulation .............................................................. 206
    7.2.4 Using Prosodic Information ................................................. 208
    7.2.5 Conclusions of Using Articulatory Knowledge ..................... 209
  7.3 Scalibility of ARTREC ................................................................. 209
  7.4 Abduction and the C/D Model ...................................................... 211
  7.5 The Problems with Hidden Markov Models and how Abduction can
    Save the Day .................................................................................. 218
  7.6 Layered Abduction is the Proper Way to Solve Interpretation Tasks 226
  7.7 Conclusion .................................................................................... 230

VIII Future Directions and Conclusion ............................................... 232
  8.1 Introduction .................................................................................. 232
  8.2 The Intention of this Research ..................................................... 232
  8.3 Limitations of ARTREC .............................................................. 234
  8.4 Future Expansions of ARTREC ................................................... 235
  8.5 Expansions to Peirce ................................................................. 238
  8.6 Future Work .................................................................................. 241

BIBLIOGRAPHY .................................................................................... 245
## List of Figures

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>An Example of Abduction</td>
</tr>
<tr>
<td>2</td>
<td>Abduction versus Layered Abduction</td>
</tr>
<tr>
<td>3</td>
<td>Speech Recognition Layers</td>
</tr>
<tr>
<td>4</td>
<td>Hearsay-II Flow of Processing</td>
</tr>
<tr>
<td>5</td>
<td>Sample Phoneme Lattice from Harpy</td>
</tr>
<tr>
<td>6</td>
<td>Hidden Markov Model for phoneme</td>
</tr>
<tr>
<td>7</td>
<td>Byblos Flow of Processing</td>
</tr>
<tr>
<td>8</td>
<td>Sample Belief Network</td>
</tr>
<tr>
<td>9</td>
<td>Sample Network for ECHO</td>
</tr>
<tr>
<td>10</td>
<td>The Subtasks of Abduction</td>
</tr>
<tr>
<td>11</td>
<td>Red Findings to Explain: Antibody to Antigen reactions</td>
</tr>
<tr>
<td>12</td>
<td>Sample Red Composite Explanation with status of other explainers</td>
</tr>
<tr>
<td>13</td>
<td>Essentials</td>
</tr>
<tr>
<td>14</td>
<td>Example of a Hard Decision</td>
</tr>
<tr>
<td>15</td>
<td>Peirce Abducer Task Description</td>
</tr>
<tr>
<td>16</td>
<td>Three Level Machine Flow of Processing</td>
</tr>
<tr>
<td>17</td>
<td>Parallel Peirce</td>
</tr>
<tr>
<td>18</td>
<td>Acoustic Recognition versus Articulatory Recognition</td>
</tr>
<tr>
<td>19</td>
<td>Microbeam Pellet Data from the Pine Street data set</td>
</tr>
<tr>
<td>20</td>
<td>Task Description of ARTREC</td>
</tr>
<tr>
<td>21</td>
<td>ARTREC's flow of processing</td>
</tr>
<tr>
<td>22</td>
<td>Recognition Agent</td>
</tr>
<tr>
<td>23</td>
<td>Abstract Gesture Template</td>
</tr>
<tr>
<td>24</td>
<td>Syllable Boundaries</td>
</tr>
<tr>
<td>25</td>
<td>Pellet data for ARTREC example</td>
</tr>
<tr>
<td>26</td>
<td>Events indicated for ARTREC example</td>
</tr>
<tr>
<td>27</td>
<td>Gestures hypotheses and what they can explain</td>
</tr>
<tr>
<td>28</td>
<td>Noise hypotheses and what they can explain</td>
</tr>
<tr>
<td>29</td>
<td>Best explanation of the events in terms of gestures</td>
</tr>
<tr>
<td>30</td>
<td>Syllable and noise hypotheses</td>
</tr>
<tr>
<td>31</td>
<td>Best explanation of pellet data in terms of syllables</td>
</tr>
<tr>
<td>32</td>
<td>Emphasis Detection</td>
</tr>
<tr>
<td>33</td>
<td>Speech Recognition Task Decomposition Using Articulation</td>
</tr>
<tr>
<td>34</td>
<td>Articulatory Layer in Speech Recognition</td>
</tr>
<tr>
<td>35</td>
<td>The Convertor/Distributor Model of Fujimura</td>
</tr>
<tr>
<td>36</td>
<td>A Layered Abduction Description of the C/D Model Inverted</td>
</tr>
<tr>
<td>37</td>
<td>Functional Model with an Abducer</td>
</tr>
</tbody>
</table>
# List of Tables

<table>
<thead>
<tr>
<th>TABLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Gesture Types</td>
</tr>
<tr>
<td>2</td>
<td>Gesture Template for &quot;final alveolar closure&quot;</td>
</tr>
<tr>
<td>3</td>
<td>Gesture Template for &quot;tongue retroflection&quot;</td>
</tr>
<tr>
<td>4</td>
<td>Syllable Recognition Features</td>
</tr>
<tr>
<td>5</td>
<td>Syllable Rule Out Knowledge</td>
</tr>
<tr>
<td>6</td>
<td>Data Set Description</td>
</tr>
<tr>
<td>7</td>
<td>Experiment 1 Results: Single Abducer Version</td>
</tr>
<tr>
<td>8</td>
<td>Experiment 2 Results: Layered Abduction</td>
</tr>
<tr>
<td>9</td>
<td>Pessimistic versus Normal Abduction Acceptance Criteria</td>
</tr>
<tr>
<td>10</td>
<td>Experiment 3 Results: Pessimistic Bottom-Up Layered Abducer</td>
</tr>
<tr>
<td>11</td>
<td>Experiment 4 Results: Full Layered Abducer</td>
</tr>
<tr>
<td>12</td>
<td>Overall Experimental Results</td>
</tr>
<tr>
<td>13</td>
<td>Overall Experimental Results</td>
</tr>
<tr>
<td>14</td>
<td>Prosodic Emphasis Accuracy</td>
</tr>
</tbody>
</table>
Abduction is *inference to the best explanation*, a form of problem solving whereby an agent explains the appearance (or absence) of data in terms of suitable domain hypotheses. The need for such an explanation is important to reasoning agents that attempt to make sense of the world. Among the activities that an intelligent agent must undertake are communication and data interpretation, information-processing activities responsible for coming to an understanding of the world. These activities can require reasoning over a vast region of knowledge and a large influx of data. In recent research, abduction has proved to be highly useful means of accomplishing a variety of interpretive reasoning tasks in a large number of domains. These tasks include diagnosis (both medical and mechanical), test data interpretation (in many domains), natural language understanding, legal reasoning and theory formation.

Perception seems to fit the description of an abductive task, yet, very little perceptual work has been implemented in an abductive framework. This is a puzzling notion, for it has been posited that perception is a form of inference, a form in which we hypothesize the cause of the inputs [Gregory, 1987, Josephson, 1982, Peirce, 1955, Charniak and McDermott, 1985, Zardozny, 1991, Rock, 1983]. That is, we can state that the goal of perception is to explain the inputs to the percepts. Further, it is
reasonable to assume that we explain these inputs by searching for the best explanation (rather than just any explanation). Therefore, we can describe perception as an abductive task.

Perception is a complex task which draws on disparate sources of knowledge. Perception requires creating coherence out of many channels of input, often noisy input, and several types of knowledge. It has long been proposed that perceptual tasks require mappings between different forms of knowledge, where the knowledge describes concepts and ideas at many levels of abstraction.

For instance, Marr has stated in [Marr, 1982a, Marr, 1981] that visual understanding is a task which takes a visual image (representing as points in a bitmap), maps the image to hypotheses of intensities (of each point of the image), which in turn is mapped to primal sketches (or edges, lines, groupings of lines, etc...) which is mapped to a rough two a half dimensional sketch (orientations, surfaces, distances, overlapping) and finally to a coherent three dimensional image including spatial information. The last step is one of recognition or identification, to classify each element in the three dimensional image. Waltz, in [Waltz, 1975] has stated similar types of mappings across several levels of inference.

Speech Recognition can similarly be cast as a "layered" interpretation task[Erman, 1980, Wolf and Woods, 1980, Josephson, 1988]. The input is the acoustic signal. Auditory features are recognized from the acoustic signal (such as formant frequencies, nasality, voicing, burst and other spectral characteristics) and from these, phonetic characteristics or identifications are made, such as tongue motion, labial motion,
glottal, velar and apical constrictions. Primitive phonetic units can then be ascribed to these phonetic characteristics, such as phonemes or syllables. These units are then aligned together to form morphemes, words and phrases. Finally, syntactic, semantic and pragmatic knowledge can be used to confirm or rule out possible word sequences.

If perception requires layers of inference, mapping from one type of knowledge to another, then a mechanism is needed to implement a strategy for such cascaded inferences. This mechanism can use hypotheses to capture the features thought to exist in the input signal. Hypotheses will exist at many levels of abstraction, in many types of knowledge (e.g. in speech recognition, there will be hypotheses pertaining to auditory features, phonetic units, syntactic units and so forth). The hypotheses formulated and accepted (or believed) at one level can be used as input to a higher level or as a guidance to lower level problem solving. In this way, inferences will occur at increasing levels of abstraction until a final hypothesis at an appropriate level is accepted.

Perception can be thought of as abduction in layers, each layer attempting to use different forms of knowledge, passing conclusions upward and downward to propagate the solution.[Josephson, 1988, Josephson and Josephson, 1993]. Therefore, I propose a mechanism for perceptual problem solving using layered abduction. This mechanism will use different forms of knowledge, different levels of inference, implementing each level as a form of abductive problem solving, and allowing each level to communicate with other levels. Whether perception is truly layered abduction or not is beyond the scope of this work. Instead, I hope to implement a perceptual problem by using our
layered abduction strategy. If successful, it will lend credence towards the hypothesis that perception is a problem of layered abduction.

Layered abduction is a novel idea in the field of Artificial Intelligence. Some previous work has combined abduction with multiple levels of knowledge representation (such as [Pearl, 1987, Reggia, 1985a, Peng, 1986, Thagard, 1989]), however, these past attempts have usually required a single abductive inference which maps findings into hypotheses and these hypotheses into hypotheses of a different form. I describe here, a different form of layered abduction in which multiple abductions can be reached, one conclusion at each problem solving level. Because this approach is novel, I hope to answer many questions pertaining to layered abduction. Among these, I hope to discover what overall behavior a layered abduction system will exhibit, how much explicit control is necessary, and if the individual abductive problem solvers will contribute to solving the overall problem (rather than "fighting each other" or having some form of non-contributive communication). As a problem to implement, I have chosen an area in speech recognition, in particular a problem known as Articulatory Recognition (a variation of the general Speech Recognition problem).

The thesis of this work is to suggest that the speech recognition problem can, at least in part, be solved by a task-level problem solving strategy, namely layered abduction. It will be argued throughout this work that layered abduction is a powerful mechanism for solving most all interpretation-type tasks, and that speech recognition can equally be solved by using such a task. In particular, I will demonstrate this, first by discussing a method for solving layered abduction, and following this up with
an example of a layered abduction system to solve the problem of *articulatory speech recognition*.

1.1 Abduction as a Generic Problem Solving Strategy

Over the last several years, abduction has gained respect and authority in Artificial Intelligence research revolving around interpretation tasks. Abduction is a generic form of inference whereby the agent generates a hypothesis to explain the appearance of some set of findings. The hypothesis should be a good one to be accepted as an abductive conclusion, in fact, it should be the best available hypothesis.


---

1 By "cornerstone", I am referring to a task which is used not only in a number of domains, but one which is used in a number of very different problems as well. Such a task has been called a Generic Tasks before. However, since abductions seem to be widely used, I use the stronger term
are based on abductive inference. Whether we are using an explicit strategy for abduction or an implicit strategy is an interesting question\(^2\) however, the end results of our daily actions are based on coming to interpretive conclusions about the data we are faced with. Thus, we are doing some form of abductive inference.

Figure 1 shows an example of an abductive problem where there are a set of findings and a set of plausible hypotheses to explain those findings. This figure does not show the output of abduction, a conclusion which best explains the data. We can define the task of abduction as obtaining the best explanation independently of describing how to do abduction.

Abduction can be considered a *Generic Task* [Chandrasekaran, 1987, Chandrasekaran, 1988, Josephson et al., 1987]. We can describe abduction as a task by giving its input and output characteristics. Such a description would be: given input of findings to be explained, the output would be a list of highly plausible hypotheses which together explain the input findings better than any other combination of hypotheses. With this task-level description, we can develop a method (or several methods) for solving abductive problems. In chapter 3, I give a detailed discussion of such a method developed over a number of years from research at the Laboratory for Arti-

\(^2\)By an "explicit" abductive strategy, I mean that we try to actively explain the data, making use of explicit hypotheses and explicit explanatory knowledge. The distinction between an implicit and explicit strategy might be important. As will be shown in chapter 3, an explicitly abductive strategy has more power than an implicit strategy. The reason that I make this distinction is that past speech recognition systems can be thought as implicitly abductive. These systems did not explicitly state a goal of explanation, but they still attempted to generate hypotheses which could “cover” the data. However, because explicit explanatory knowledge was not used, these systems lost out on some of the problem solving strategies that I will discuss in chapter 3.
ficial Intelligence Research. One advantage of decomposing a task into a task-level description, and the choosing a method as an implementation is that the task defines the types of problem solving activities required to solve the task which dictates the forms of knowledge needed to solve an abductive problem. I will also discuss the appropriateness of a generic task level solution to perceptual problems in chapter 7.

1.1.1 A Definition of Abduction

We can define an abductive process as follows:

There exists a set of data, D, to be explained. We know of a set of hypotheses,
H, each element of H can explain some part of D. An abduction is to generate h, an element of H, which can best explain D.

In most instances of an abduction, no single h (from H) will explain the entire set of data, and so a conjunction of h’s must be used. This conjunction is called a “composite hypothesis”. Therefore, an abduction is to generate H’, a subset of H, which best explains D. The term “best” means that H’ is more plausible than any other subset of H which can explain D, and H’ has no superfluous parts to it. We can also take “best” to have several other criteria such as the explanation is complete (i.e. H’ explains all of D), is consistent (i.e. no parts of H’ are incompatible), H’ is minimal (i.e. H’ is contains the least number of parts of any full explanation).

Problems arise out of the need to come to an abductive conclusion. The first deals with the need to generate a conclusion “on the fly” (or at run time). Most problems we face that require abductive conclusions are problems in which the input is not previously known. Many instances of abductions are problems which are novel (such as a new diagnostic case or something we have never seen before). Therefore, we cannot simply retrieve an explanation from memory, it must be constructed. Having to construct an explanation requires time to think of hypotheses, to put hypotheses together into a composite.

However, to construct an abductive conclusion means to generate the best explanation. And to obtain the “best” explanation presents a major problem. To ensure that H’ is the best or most plausible hypothesis, it seems that we would need to generate all possible subsets of H, and compare each subset in terms of a set of “bestness”
criteria (e.g. minimality, plausibility, consistency, completeness, parsimony). That is, since our conclusions will typically be made up of some set of lesser hypotheses, we would need to generate all possible combinations of lesser hypotheses and compare them. Without such a comparison, we would not know if our \( H' \) was truly the best.

The necessity to generate all subsets of \( H \) would make abduction an intractable problem. Given a set of \( k \) plausible hypotheses, the set \( H \) would consist of \( 2^k \) different subsets of hypotheses. To create \( 2^k \) composite explanations and compare them is obviously intractable.

Further, in constructing the "best" multipart hypothesis, we must maintain consistency (i.e. make sure that the hypothesis has no mutually exclusive parts). This further complicates the problem. Cancellation effects among hypotheses can complicate matters\(^3\). All of these problems, the need to generate an a conclusion "on the fly", the need for an optimal conclusion, having to maintain consistency, and the problems of cancellation effects make the abductive problem intractable.

We are faced with a dilemma. If abduction is as ubiquitous as we would like to think, how can we get by if we require exponential amounts of time to accomplish the task?

1.1.2 A Better Definition of Abduction

There is a great friction in defining abduction. We seem to use abduction to reason, we generate abductive conclusions "on the fly" (i.e. at run time, or at the time the

\(^3\) A cancellation effect occurs if one hypothesis will cancel out the symptoms or manifestations produced by another hypothesis. This can occur naturally in medical diagnosis for instance.
data is encountered) yet we come to quick conclusions. It takes very little time to parse a sentence or make sense of visual input or to identify a person based on their voice or visual features. In the process of diagnosis, a Doctor can come up with a fast conclusion in seconds. The only instances of abductions which seem to require substantial processing time are those cases in which we must deliberate over the values of individual hypotheses. These instances occur in theory formation and evaluation problems (including legal reasoning) and communication where meanings and words are ambiguous or data is noisy. It also occurs in perceptual problems in which some form of optical illusion or trick might be involved (again, a problem of ambiguity). And in all this, how can we generate an exponential number of possible conclusions? We must redefine abduction to be a more realistic task.

A more practical definition of abduction might be as follows:

Given a set of data D, and a set of hypotheses H, an abduction is to generate a “best” composite explanation H’ to account for as much of D as is confidently and practically possible.

In addition, we can appeal to several clever and efficient strategies in order to compose and critique H’. Some of these strategies are described here.

- We attempt to generate the best “local” explanation for each individual datum, combining our “local” explanations into an overall composite. We will worry about combining the local abductions into a global abduction separately.

- We do not attempt to explain some datum, d, if two or more hypotheses are
equally good explainers. We leave this decision open for the time being. The result of this is three-fold. First, we do not waste time with differentiation between closely plausible hypotheses. Second, the decision of accepting one hypothesis over the other hypothesis may be made for us later on due to other problem solving. Third, by not coming to a conclusion, we do not weaken our confidence in our overall conclusion, we do not introduce the risk of error.

- We use knowledge about hypotheses' interactions in order to drive some of our decision making. Hypothesis interactions are one factor which makes abduction intractable. However, if we believe in a conclusion, we can assume it to be true and use this as leverage for further problem solving. Among the types of leveraging that can be used are ruling out hypotheses which are incompatible with accepted hypotheses; giving more plausibility to hypotheses which are related to or expected by accepted hypotheses; seeking refinements of accepted hypotheses to offer a more detailed explanation.

- We do not attempt to find the most optimal explanation as long as we can feel reasonably sure that our “best but less-than-optimal” decision is “good enough”. This is an important notion in abduction because in real-world abductions, we are certainly not finding an optimal explanation, but one we feel comfortable accepting.

And so, the “best” explanation becomes the explanation which maximizes the explanatory coverage while maintaining as much confidence as possible. However, we
do not force the conclusion to contain complete coverage of the data, and we do not force the conclusion to be the most optimal (in terms of plausibility). By committing to clearly best local abductions, we can attempt to compose a single composite explanation which explains as much of the data as is confidently possible while maintaining a very low possibility of error. If our search of alternative explanations for a datum is thorough and we do not commit to hard decisions (i.e. instances where two or more hypotheses seem equally good) then we can feel secure in our composite explanation.

It is interesting to note that most past researchers have assumed that the solution to an abductive problem should be both optimal and complete. Yet these two criteria cause abduction to become an intractably hard problem. The qualifications of our "best explanation", which I suggested above, have the result of removing the intractability from the problem while still allowing the abduction to be useful. Humans may be wrong even if they have accurate knowledge. This is because, among other reasons, we will not seek an optimal and complete explanation. We will seek a "best" explanation within reason.

1.2 Problems Come in Layers

Most abductive researchers take abduction to be a single mapping of findings to hypotheses. Thagard, Pearl and Reggia have allowed hypotheses to be explained themselves in terms of other hypotheses. However, even in their case, there was a single set of inferences made to explain the original findings. This description seems

---

4The intractability is removed from some types of abductive problems. There are still many forms of intractable abductions. See [Bylander, 1991, Allemang et al., 1987] for a discussion of the various types of intractable abductions.
insufficient to solve many abductive activities.

Diagnosis has been thought of a task requiring a single abduction, explaining the symptoms or peculiar findings in terms of appropriate disease or malfunction hypotheses. Fischer, in [Fischer, 1991], discusses using a Meta-diagnostic level to reason over the diagnostic problem itself. Reggia, in [Reggia, 1983] has used two forms of hypotheses, disease hypotheses and pathological state hypotheses. A complete diagnosis might map the symptoms/findings to pathological states to diseases to etiologies. In generating a causal story of how a disease or malfunction may have caused the resulting findings, several levels of inference might take place. Diagnosis, when complex, may require multiple inferences. This can presumably be accomplished by abduction in layers.

Story comprehension is another problem solving area where researchers have assumed a single abduction to be sufficient to explain the appearance and order of words in terms of concepts and messages (i.e. messages being communicated)[Hobbs et al., 1988, Dasigi, 1988]. However, to understand a message being communicated, one must take into account syntax, semantics, pragmatics and discourse. This requires several levels of inference and can presumably be accomplished by abduction in layers.

Theory formation and evaluation [Thagard, 1989, Fox et al., 1992b] requires interpretation of a large collection of findings into new theories and laws. The findings may be at varying levels of representation, requiring many types of knowledge. Laws may also be at differing levels of representation. Theory formation has, in the past,
required generating hypotheses to explain the findings and observations, and then unifying these hypotheses into a consistent theory. This can be accomplished as two distinct sets of inferences one mapping from findings/observations to hypotheses about the world, and one unifying the hypotheses into a consistent theory. Theory formation and evaluation require leaping across levels of knowledge. Again, this can presumably be solved by abduction by layers.

In perception, the input to the process may be caused by light or sound waves, yet a satisfactory conclusion is one based on object hypotheses or word hypotheses, or one based on concepts. A conclusion which states that the eye is seeing edges and colors is not satisfactory. A conclusion that the we are seeing a truck moving towards us is more sufficient, or that we are hearing a message in the English language from a friend. In perception, there are many forms of knowledge represented at varying levels of detail. For visual interpretation and understanding, Marr, in [Marr, 1982a], has suggested that these levels include detection of edges, shades, and shapes, depth analysis, 2 1/2 dimensional image identification and finally three dimensional object identification. In speech understanding, it has been suggested that the levels of inference pertain to auditory recognition, articulatory recognition, phonetic recognition, syntactic parsing, semantic understanding, pragmatic and discourse reasoning. If the abduction is used as the mechanism to implement a perceptual mapping at one level, layered abduction is used as the overall mechanism.\footnote{The levels described here for visual comprehension and speech recognition are controversial. We are not attempting to prove these are the correct levels, but rather that to solve such problems, multiple levels of inference are needed.}
1.2.1 A Mechanism for Abduction in Layers

To accommodate the complexities described for the above tasks, a more powerful mechanism may be required than abduction. This mechanism should be one which can come to a conclusion (or partial conclusion) at one level of inferencing, and make more abstract inferences based on that conclusion; a mechanism which can "leap" across levels of knowledge representation and unite the different types of hypotheses into a single, coherent idea or story. To accomplish this, the mechanism must have the ability to reason over many different forms of knowledge, yet to come to an abductive conclusion at each level. It must be able to tie together levels of knowledge, and to reason in both an upward motion (data driven) and downward motion (hypothesis driven).

Our approach to layered abduction is a novel idea. It is a variation of abduction where knowledge of hypotheses and data are represented at in many forms and at various levels of abstraction. Abductive conclusions at one level are passed up to the next level and used as findings to be explained. Hypotheses of a more abstract type are then used to explain the hypotheses of a less abstract type found at the lower level. This process continues in a bottom-up manner until hypotheses of a sufficient description are accepted. A higher level may also drive lower level inferences by prompted for more refined hypotheses, or may offer expectations to aid lower level inferences [Josephson, 1988].

Abstractly, the difference between abduction and layered abduction can be seen in figure 2. Here, we see that abduction requires one set of inferences (from findings to
Figure 2: Abduction versus Layered Abduction
hypotheses) while layered abduction requires several such inferences with processing occurring in both top-down and bottom-up directions.

In order to implement layered abduction, one need only build a single mechanism for abduction for each level of the problem solving, and then link together the output of one level to the input of the next level. This requires some preplanning in that one level will expect findings of a suitable form which in turn requires the hypotheses of the previous level to be of that form. Some means of control is needed to operate the layers in a proper order. The simplest form of control is to use a bottom-up strategy where abductive conclusions are passed up from one level to the next level in a serial order. One abductive conclusion is made at the lowest level and passed up to the next level. This level uses the conclusion as findings, and makes an abductive conclusion and passes it up to the next level, and so on until the final level reaches a conclusion.

However, without any top-down processing, the layered abduction mechanism is limited. Top-down processing can be used to determine higher level hypothesis plausibilities, to generate expectations to aid lower level abductions, to doubt puzzling (or ambiguous) data, or to resolve conflicts of paradoxes. These notions are described in more detail in chapter 3.5.3.

To implement a layered abduction system with both bottom-up and top-down processing, one must decide when to pursue bottom-up processing as opposed to attempting some top-down processing. This decision can be made based on available knowledge and the status of the problem solving [Punch, 1989], or made at system construction time. By using bottom-up and top-down processing and a flexible control
strategy, a layered abduction mechanism can generate very plausible partial conclusions at one level, and use these as islands of certainty to leverage further problem solving, both at the same level of inference and at higher and lower levels.

A layered abductive problem solving strategy will combine the explanatory power of abductive strategies with the ability to leap across levels of knowledge representation unifying a conclusion of many types of hypotheses. In this way, a problem solver can offer to explain input findings at the “appropriate” level of abstraction. The “appropriate” level for diagnosis may be a causal story which is unavailable at a lower level of abstraction, or, in story comprehension or theory formation this level might be abstract concepts or theories, also unavailable at lower levels of knowledge. In perception, the “appropriate” level will not be at the level of the input (such as an explanation in terms of lines or acoustic phenomena) but at the level of words, physical objects and “concepts” being communicated.

1.2.2 Perception in Layers

Perceptual tasks are tasks which require rich and diverse knowledge. There is a need to come to an understanding at many levels of representation. It has long been hypothesized that perception is a form of inference [Charniak and McDermott, 1985, Josephson and Josephson, 1993, Josephson, 1988, Peirce, 1955]. Therefore, to accomplish a perceptual task, one must infer across these levels of knowledge representation, and come to an understanding at a much more abstract level.

Speech recognition requires making sense of sounds, of phonetic units, of words, of syntactic and semantic constraints. The inference is not one of simple mapping, but
one of complex mappings and transformation of one type of knowledge into another. Past works in speech recognition have typically used multilayered techniques where data driven and hypothesis driven processes combine. However, in systems such as Byblos, Hearsay and HWIM, the systems have not taken an explicit explanatory approach. The layered abduction strategy discussed here combines the best aspects of the abductive approaches to explanation and the multilayered approaches of complex perceptual systems.

If we can say that abduction is a task used in perception, then the task should be one of layered inference. I will propose here, a means of accomplish the speech recognition task by layered abduction. I will then concentrate on a variation of the speech recognition problem and attempt to solve it by a strategy of layered abduction. We will direct our attention to articulatory recognition, inferring words from data relating to articulation (vocal tract motions) rather than data relating to acoustics. Reasons for using articulation will be discussed later in this chapter.

Speech Recognition as a Layered Task

The speech recognition task, as depicted in figure 3, can be broken into the following subtasks. This description is a proposed hierarchical decomposition of the speech recognition task, and is similar in ways to many other systems such as Hearsay-II and Byblos [Erman, 1980, Chow et al., 1987].

First, to infer auditory characteristics from the acoustic signal. This entails de-
Figure 3: Speech Recognition Layers
terminating the frequencies and magnitude behind a burst of energy, locating formants, determining the slopes of formant transitions between a consonant and vowel-like sound, analyzing the spectral qualities of an energy release, and so on. This type of knowledge will enable the listener to decide if he/she is hearing a phonetic event representing a consonantal sound, vocalic (or vowel-like) sound, a glide, silence or some other sound. This is the first step in classifying the sound in terms of possible phonetic units.

Next, the abstracted auditory information is used to infer articulatory knowledge. Articulation is the process of creating sounds by motions of the vocal tract. The auditory information is the input to a mapping into possible vocal tract motions which can cause such auditory responses. Articulatory hypotheses consist of "closures" and "openings", tongue motions, voicing, and nasality. These hypotheses can be described in terms of qualitative vocal tract motions, and can explain the auditory events found at the previous level.

At the next level, the articulatory hypotheses are combined with auditory hypotheses and used as input to identify possible phonetic units which are the cause of the acoustic signal. Phonetic units are the primitives of speech (for example phonemes, demisyllables or syllables) which we use to generate phrases (words and sentences). The output of the phonetic layer will be a series of temporally ordered phonemes or syllables (or some other form of phonetic unit or feature description).

---

7This is a hypothesized mapping which is by no means certain. The Motor Theory of Speech Perception discusses this mapping although other theories disagree and state that the auditory information is mapped directly to phonetic units. I will discuss the Motor Theory of Speech Perception in the next section.
These strings of phonetic units can then be grouped together into morphemes and morphemic affixes to create words. Knowledge of legal strings of syllables can be used to determine what morphemes and morphemic affixes are possible given the language being recognized. The output of this level will be a string of temporally ordered morphemes.

The next level is the lexical level in which the strings of affixes and morphemes are fitted together to form words. A lexicon will be used to generate plausible word hypotheses based on the input.

Words are then linked together at the highest levels to create phrases and sentences, and more importantly, meanings. The highest levels of the speech recognition problem are mappings of words into the semantic and pragmatic meanings. Discourse and register knowledge can aid in coming to a complete understanding of the message.

Speech Recognition as Layered Abduction

Perceptual reasoning requires hypotheses at different levels of representation, using a variety of disparate knowledge sources. To accommodate this, I propose a task which can accomplish perception in terms of abduction. This new task is called “Layered Abduction”, in which abductions will be performed at different levels of knowledge representation. Hypotheses accepted as explainers at one level will be used as data.

---

Register knowledge is discussed in [Halliday, 1985, Patten, 1988]. Register knowledge refers to the context of what is being discussed, to whom it is being said, and the manner of which it is being said. Register knowledge dictates, among other things, the vocabulary to be used and the tone of the conversation.
to be explained at a higher level. In this way, we can attempt to explain an acoustic signal, not in terms of words, but in terms of auditory hypotheses. These hypotheses then become data and are explained by phonetic units. And so on until we have explained the original signal in terms of words or even meanings.

In speech recognition, we are given an acoustic signal, and we make use of knowledge of discourse and register in order to infer words and meanings. This complicated process occurs very rapidly thanks to highly compiled knowledge all of the levels from the auditory level to the syntactic and semantic levels. To implement this mapping as layered abduction, we need to form levels of knowledge pertaining to each individual mapping. At each level, we require mechanisms that will allow each level to generate an explanation of the input findings in terms of appropriate speech hypotheses (appropriate meaning auditory, articulatory, phonetic, lexical, syntactic, semantic). The actual means of this mechanism will be described in chapter 3 and an example of part of this mechanism will be described in chapter 4.

It might seem unreasonable to place a restriction of "deliberation" on perception. Perception occurs at such a rapid speech, than there should be no need for deliberation. However, I assert that abduction does not need to be a deliberative task. Using an explicit mechanism for forming a composite hypothesis does not necessarily require a deliberative strategy. Rather, much of the knowledge used in abduction can be pre-compiled (as suggested in much perceptual literature such as [Marr, 1982b, Gregory, 1987, Zardozy, 1991]). The abductive task can make use of compiled knowledge as easily as non-compiled knowledge. This issue will be discussed in more detail in
chapter 3.

However, as an introduction to the types of compiled knowledge used in speech recognition, considering the following. Low level feature detection/recognition in speech recognition may be “hard wired” in the auditory mechanism. That is, the ability to propose hypotheses at the auditory level may be one which is compiled as a set of sound-hypothesis “links” so that a sound will instantly cue (or tickle if you will) a hypothesis which can explain that sound. Our perceptual system is very highly tuned and so, hopefully in unambiguous cases, the auditory recognition system will propose only a single hypothesis for each sound or set of sounds so that abduction becomes trivially easy (i.e. choosing the best explanation is very simple because there is only one plausible hypothesis to explain each finding). In addition, top-down knowledge in the form of context will help auditory recognition by allowing for focus of attention and high level expectations. The recognition process at the auditory, phonetic and lexical levels becomes simplified because the highly compiled knowledge is readily available, and further, it will typically yield only plausible hypotheses.

Higher levels of processing in the speech recognition task are less “hard wired” requiring some degree of deliberation (such as inferring meaning from ambiguous words or parsing particularly difficult sentences). Because context is very important to language processing, more run-time consideration must take place to infer correct meanings for words. However, even language processing is incredibly fast which points out that much of language processing is accomplished in a non-deliberative fashion [Patten et al., 1992].
I propose that, while speech recognition (and other perceptual tasks) is abductive, much of the potential problem solving, at least in the lower levels of processing, are highly compiled so that the abductive problem solving is very efficient and quick. A conclusion in terms of a composite hypothesis is still required, but the steps in composing such a hypothesis are greatly simplified by the efficient and highly compiled knowledge used in portions of the abductive subtasks.

Inferring sounds or articulatory gestures (as will be described in chapters 4 and 7) requires knowledge of phonetics. This knowledge, because of training over many years as a child, becomes highly compiled. The task of hypothesis evocation (as described in chapter 3) is highly simplified as sounds directly infer possible hypotheses rather than a lengthy search process for possible hypotheses. The task of hypothesis instantiation is also simplified due to a high degree of knowledge compilation. Further, these hypotheses (gestures, phonetic units) are very simple hypotheses which, when combined by abductive assembly, yield more complex hypotheses. These more complex hypotheses can be reasoned over in more detail at higher levels of problem solving (such as at the syntactic, semantic and pragmatic level) where some deliberation seems to occur.

Different levels in the perceptual process seem to require different amounts of "focus" and deliberation. Lower levels require very little deliberation as the mechanisms are highly compiled. Yet, hypotheses are still necessary to make sense of the input. As these hypotheses are passed up to higher level cognitive processes, more and more attention can be given.
An abductive strategy, as described in chapter 3, will not require great amounts of “processing” time. The task of abductive assembly requires a small amount of processing time once hypotheses have been suggested (by hypothesis evocation and instantiation). Therefore, it seems reasonable to say that abduction, if performed efficiently, is a proper task to solve perceptual problems.

1.3 Speech Recognition

Automated Speech Recognition research has been going on since the 1950s with initial isolated word and digit recognition systems. A systematic serious effort for automated speech recognition started with the ARPA project in 1971. These projects met with some mild success but pointed out many flaws with the state-of-the-art in linguistic knowledge, signal processing and overall strategies for speech recognition. Since that time, many speech recognition systems have come out, but none have solved the entire speech recognition problem.

This problem can be stated as follows:

Recognize multi-speaker, continuous speech, using a large vocabulary and a complex grammar in variable pragmatic environments.

Why has not the field been more successful at creating an automated speech recognition system?
1.3.1 The Problems of Automated Speech Recognition

There are numerous problems with automated speech recognition. No single solution has been appropriate for solving all of these combined problems. The problems of automated speech recognition are as various as the variations in an acoustic speech signal. There are problems that come from differences between speakers (such as dialect, accent, lexical differences), differences within a speaker (such as if that person is speaking quickly or slowly, speaking to a child or a professor, speaking with a cold or in good health), differences within utterances (e.g. a /t/ differs in acoustic features whether it is in the first part of a syllable, following an /s/ or at the end of a syllable). There is obviously enough information in the acoustic signal (along with context or discourse knowledge) for humans to correctly identify the words being spoken. Yet the high accuracy that humans achieve in everyday listening cannot be approached by machines.

Since the information is present in the acoustic signal, there must be a problem with the automated recognizers. The problem must lie with either processing the acoustic signal in a correct manner or interpreting the processed acoustic signal correctly. Processing an acoustic signal seems fairly well understood with techniques developed in the past in Electrical Engineering and Speech Science. Continued work in signal processing should help. Recent new research includes a better modeling of the activities in the human ear which should aid the overall speech recognition problem [Patterson, 1987, Patterson, 1988]. However, signal processing is beyond the scope of this work, and any problems related with signal processing will be ignored
throughout this work. The problems that I wish to address lie in correctly interpreting
the processed speech signal.

What kinds of problems exist in such interpretation? This should be asked as
“what types of knowledge are necessary to correctly interpret the speech signal?”
This is where the key difficulty lies in automated speech recognition. Just what types
of knowledge are necessary, and where can we obtain this knowledge? How should
this knowledge be modeled? How should this knowledge be used?

One major stumbling block with continuous speech recognition is locating the
words from an acoustic signal. By just observing the acoustic signal, it becomes
a very difficult task to determine where one word ends and the next one begins.
Systems which use phonetic “templates” have problems because word boundaries
are necessary to determine the location within the word where a template should
be compared. Without the word boundary, the comparison of a section of acoustic
signal onto a “template” might occur anywhere, at the beginning, middle or end of a
word.[Reddy, 1976]

Another problem that haunts automated speech recognition is the problem of
articulatory dependencies⁹. The speech signal is inherently variant due to the depe­
dencies that occur when one sound follows another sound. When one sound appears
next to another sound, subtle and not-so-subtle changes occur in the acoustic signal.

⁹This term, “articulatory dependencies”, is what I have dubbed to be the effects which arise from
the ways that sounds will be altered depending upon their locations within an utterance. The
dependencies arise because a sound might change dramatically based on the context of the sound
(where context is dictated by the surrounding sounds). In the past, these dependencies have been
given terms such as coarticulation. However, coarticulation implies a linear phonology which is
not an implication that I wish to make here.
For example, an initial /t/ sound in an utterance will have a highly different look (or spectral characteristics) from a /t/ which follows an /s/ sound, or a /t/ which is located at the end of a phrase. [Randolph, 1989] This makes it much harder for a continuous speech recognition system to infer the correct units because it now has to consider a large variation of sounds.

Yet another problem is the lack of definitive linguistic knowledge. As speech science and linguistics become more certain of what to look for in the acoustic signal, speech recognition systems can also improve. And there has been improvements in linguistic knowledge since the early ARPA experiments. However, there is an abundance of unknowns that are still present in the linguistic knowledge. Included in this gap of knowledge are what features are important to recognize certain sounds in different circumstances. An example of this is finding distinctive features for determining place of articulation [Erman, 1980; Fox and Josephson, 1991]. Because of the variability in spectral characteristics exhibited in acoustic speech, linguistic knowledge must state the proper contexts under which certain features are useful or useless. In order to explicitly encode such knowledge, the knowledge must be available.

However, other methods of speech recognition using implicitly encoded knowledge (such as in Hidden Markov Model systems or connectionist systems), lack this knowledge as well, but can bypass the need for explicit features by training recognition models. HMM systems typically use phonemes as primitive phonetic units and all of the models of the system are in terms of individual phonemes. There are problems associated with using such implicitly encoded knowledge in speech recognition.
systems, especially if the primitive units are phonemes and the speech task is one of phoneme concatenation (i.e. appealing to linear phonology where primitive units are concatenated together). Some of the problems of HMMs will be addressed in chapter 7.

And so, there are stumbling blocks to surpass. These involve relationships of how the acoustic signal arises from the pronunciation process, how the variations found in the acoustic signal are caused by the dependencies within the speech process. In order to get around some of these problems, whether they are due to lack of linguistic knowledge or because knowledge about the speech dependencies are lacking, new types of knowledge must be brought into the problem solving. These types of knowledge deal with speech production, namely articulation and prosody.

1.3.2 Articulation

In the past, automated speech recognition has relied on knowledge pertaining to acoustics. It has been hypothesized [Liberman and Mattingly, 1985] that articulatory knowledge is also relevant in interpreting the acoustic signal. This Motor Theory of Speech Perception suggests that, in recognizing words, we are appealing to knowledge of how speech is produced. To understand how speech is produced, we must use knowledge of articulation.

It is known how the vocal tract can give rise to certain acoustic properties (formant frequencies). However, the mapping from the vocal tract "area function" to formants does not reflect the independent, multidimensional control of the articulatory system, and how this gives rise to the speech signal. That is, we can describe a function mapping the overall shape of the vocal trace (for a particular utterance) into speech
signal characteristics, but we cannot describe a function which maps from intentions of the articulatory system to the resulting acoustic signal. Thus, to use articulation to aid acoustic speech recognition, we must discover how to closely relate the two types of knowledge (articulation and auditory knowledge). To do this, we must explore the independencies of articulation, we must come to an understanding of the articulatory system.

Articulation is the process of speech production where the elements of the vocal tract (tongue, lips, larynx, etc...) combine to produce speech. If one could properly model articulation, then one could appeal to this model when considering the causes of the acoustic signal. One could determine what motions were responsible for the sounds, and then by using articulatory knowledge, determine the utterance based on the derived motions. Rather than inferring phonetics from acoustic signal, a speech system could infer articulation from acoustics, and phonetics from articulation. This seems a reasonable hypothesis since the acoustic signal is the result of the articulation.

Among the potential gains of explicit articulatory knowledge are the ability to distinguish articulatory dependencies from other phenomena which might cause confusion in the acoustic signal, the ability to make inferences about prosody, and the ability to clearly notate where one word ends and the next word starts. Articulatory knowledge will help because articulation is more stable than the speech signal created from articulation. For instance, the motion necessary to produce a sound, such as a /p/, is relatively constant no matter what precedes or succeeds the sound whereas the same is not true of the acoustic signal produced from speech. Articulatory re-
search has found stability in articulatory motions independent of articulatory and non-linguistic contexts. These stable items have been termed "icebergs". For more information, see [Fujimura, 1986].

It is believed by some linguists that the proper units of speech are not based on a linear phonology but a non-linear phonology [Browman and Goldstein, 1986, Browman and Goldstein, 1988, Fujimura, 1992, Fujimura et al., 1991]. Phonemes, and even syllables, are based on linear phonology in which these primitive phonetic units are simply concatenated together. To recognize an utterance is to determine, at each position, which phonetic unit best matches the sound. Then, one simply concatenates each phonetic unit together. Coarticulation is a process of "smearing" the phonetic units together to take care of any overlap in the acoustic signal. However, if we model articulation in the speech process, we can use a non-linear form of phonology in which we are more interested in phonetic features (such as bilabial closure) rather than phonetic primitives (/p/).

To use articulation for speech recognition, it is assumed that a system can take the output from the auditory level and hypothesize articulatory causes for the auditory events. For example, a burst of high frequency energy can be caused by a particular type of articulatory event such as constriction along an element of the vocal tract. By analyzing features in the auditory event, a classification of the articulation action can be made, and this will allow a system to infer the place of articulation. Another example is that formant locations can be used to determine the approximate location of the tongue, or relative tongue motions, during the pronunciation of vowels. Other
uses of the acoustic signal can be researched in order to determine more fully, how auditory features can lead to articulatory hypotheses. In turn, articulatory hypotheses can be used to infer phonetic units and other important features.

The Motor Theory of Speech Perception claims that to understand speech, one must understand\(^\text{10}\) how speech is produced. The production of speech requires complex and intricate motions carried out by the vocal tract. To understand speech, according to the Motor Theory, one must have an understanding of how the vocal tract moves in order to create speech. When we learn to speak, we are tying together what we are hearing with what we see of others speaking. We watch their mouths, and attempt to copy those motions. I will argue in this document that articulation plays a large role in our ability to recognize speech.

Articulatory research has revolved around exploring the motions of the vocal tract, of studying acoustic phenomena, and tying the two together. To aid this research, Microbeam Pellet data is now available [Westbury and Fujimura, 1989, Rassner, 1987]. This data (explained later) consists of vocal tract motions, and can be used by researchers to explore how the vocal tract is shaped to create the sounds we hear. In this work, Microbeam Pellet data is used in two ways: first to explore articulation to create appropriate types of knowledge for our speech recognition system, and second, to use as input to the speech recognition system (rather than acoustic data). Because the work being presented involves using articulatory data as input rather than acoustic data, I have termed this task as Articulatory Speech Recognition (or simply

---

\(^{10}\)Understand is used loosely here. The Motor Theory actually dictates that the production mechanism is in some way invoked in order to make inferences about the speech signal.
Articulatory Recognition).

We will also consider a new theory of non-linear speech production, and what roles it may play in our understanding of speech recognition. This theory, the Convertor/Distributor Model [Fujimura and Wilhelms, 1991, Fujimura, 1992, Fujimura et al., 1991] has been a basis for some of our articulatory research. In chapter 7, we will explore the Convertor/Distributor model and discuss its relationship with layered abduction.

1.3.3 An Example of Articulatory Recognition

To demonstrate Articulatory Recognition as an abductive problem, and in order to further the research on layered abduction, a system called ARTREC has been built that uses Microbeam pellet data (motions of the vocal tract) as input, and generates an explanation for the motions in terms of lexical and prosodic information. Using multiple levels of knowledge, ARTREC will infer the cause of the pellet data in terms of a message being communicated. While past speech recognition systems have used syntactic, semantic and other contextual knowledge to support their system, ours will only map the input into lexical items (words) and prosodic information. Without the higher level knowledge, this task becomes more challenging.

ARTREC currently solves a modest problem, articulatory recognition of a small set of words and articulatory gestures. However, I will try to show that high accuracy in continuous multi-speaker speech is possible by appealing to articulatory knowledge. ARTREC makes use of no acoustic input, relying solely on articulatory data. Obviously adding an acoustic signal to the input will improve the recognition
performance significantly, however this would also trivialize the task of recognition due to our limited lexicon size. I will demonstrate the abilities of ARTREC and discuss experimental results and enhancements to this system.

1.4 The Contributions and Scope of this Work

This work focuses on two areas. First, this work introduces a strategy for accomplishing layered abductive tasks. Abduction, or inference to the best explanation, is a generic problem solving method which could be used to solve many varying problems such as diagnosis, story understanding, and perceptual identification problems. Layered abduction is a method whereby abductions are made based on the results of other abductions. I will show that layered abduction is a very plausible means of solving a variety of problems dealing with data interpretation, including perception.

Second, this work introduces the idea of using articulation for speech recognition. This is a novel concept in speech recognition. By appealing to articulation, a speech recognition system can make inferences that were not available to "acoustic signal" speech recognition systems. Examples of such inferences include determining how one sound might affect the pronunciation of another sound and therefore affect the sound itself, how prosody affects utterances, how vocal tract differences affect the sound, and so on.

There has been substantial research in abduction especially over the last few years, however, there has been very little documented about layered abduction. Many questions still need addressing when considering layered abduction. How will abducers interact between layers? Will there be a problem with the abducers settling on
best explanations? What domains are suitable for layered abduction? Are layered abductions even feasible? I hope to answer as many of these questions as possible within the framework of this research.

The core set of work cited in this dissertation is the articulatory recognition system ARTREC. This system is given an English utterance in the form of X-ray Microbeam pellet data without any acoustic signal. ARTREC attempts to identify the words uttered by appealing to knowledge about articulation. As such, ARTREC does not solve any portion of the speech recognition problem. However, ARTREC could be used as a module in a larger speech recognition system. ARTREC uses a strategy of layered abduction, that is, by generating "best explanations" at each level of problem solving. ARTREC will be explained in detail in later chapters. I will also discuss how layered abduction (and ARTREC in particular) can be used to aid the development of a new theory of speech production known as the Convertor/Distributor model.

The problem that this work attempts to solve is a subproblem of speech recognition which I have termed Articulatory Recognition. This work contributes to the fields of Artificial Intelligence and Speech Science. In Artificial Intelligence, this work contributes to the body of research investigating abduction. In particular, the contribution is to show the relevancy and usefulness of layered abduction. In Speech Science, this work contributes to the understanding of articulation, speech organization and prosodic control in realistic speech. In both fields, this work combines past and present issues in automated speech recognition, and offers novel ideas to both fields (that of using layered abduction as a method of speech recognition, and that of
using articulation as explicit knowledge in accomplishing speech recognition).

1.5 A Guide to Reading this Dissertation

This dissertation is laid out in the following manner. Chapter Two is a literature survey of both abduction and speech recognition. The survey on speech recognition is kept short as there have been many examples of speech recognition, but most systems have similarities in both implementations and problems. The survey of speech recognition systems will concentrate on two past systems, Hearsay and Harpy, and two newer systems using Hidden Markov Model approaches, Sphinx and Byblos. The survey of abduction will be more complete and will consider four types of abduction strategies, abduction by heuristics methods, a logic-based approach to abduction using backward chaining, probabilistic reasoning methods, and a connectionism-based approach.

The third chapter will focus on the abduction strategy used in ARTREC. This chapter will describe a particular strategy which was first developed in a system known as Red and later abstracted to be domain independent called Peirce. This strategy will be described in detail and followed by a discussion of how the strategy can be used for layered abduction. Finally, I will consider the various strategies to control layered abduction.

Chapter Four will examine the ARTREC system in detail. I will discuss the task of articulatory recognition, the types of knowledge necessary for articulatory recognition, and the methods with which articulatory recognition are achieved. I will give a detailed view of the flow of processing for ARTREC, and the knowledge built
into ARTREC.

Chapter Five will examine an example of ARTREC on a typical utterance. This chapter will consist of a detailed examination of ARTREC using a pictorial format by showing, at each stage of ARTREC’s processing, what is going on.

Chapter Six will consider experiments and results of the ARTREC system. There were four separate experiments run using ARTREC, these will be discussed and compared. I will discuss in detail what I have learned from the experience of ARTREC.

Chapter Seven will consider the implications of this work. I will try to tie together the use of layered abduction in a variety of problems. I will also attempt to tie together the use of articulation in speech recognition. I will discuss a new theory of speech production known as the Convertor/Distributor model and consider how ARTREC can be used to advance this new theory. Finally, I will turn to Hidden Markov Models, discuss several problems with this approach, and briefly discuss how abduction might be used to get around some of these problems.

The last chapter will describe enhancements to ARTREC. I will attempt to show how ARTREC can be improved. I will also consider variations of the layered abductive strategy. I will conclude with a statement of the benefits of layered abduction and the future of speech recognition.
CHAPTER II

A Survey of Related Works

My work explores new dimensions in both abductive problem solving and speech recognition. This chapter will review some important previous work in both abduction and speech recognition.

2.1 Previous Work in Speech Recognition

Speech Recognition research has had both ups and downs over the last 25 years. The largest single initiative for automated speech recognition started with the ARPA research. Of four large projects, none solved the overall speech recognition problem. In fact, the results were rather disappointing, causing interest in automated speech recognition to diminish. This speech initiative brought forth the systems Hearsay-II, Harpy, HWIM and SDC [Klatt, 1990]. Of these four, I will only comment on Hearsay-II and Harpy, two of the more interesting and successful systems.

Since the mid 1970's, many approaches have been tried. And still, none of the systems have solved the overall speech recognition problem. However, two major stumbling blocks, continuous speech and speaker independence, have been solved to some degree. The latest drive for automated speech recognition has revolved around the use of a particular statistical approach using mathematical models called Hidden
Markov Models (HMMs). One particular advantage of HMMs is the ability to train the system rather than having to engineer the knowledge into the system. Training is accomplished by using speaker data and training algorithms (in a similar way to how one would train a neural network). This frees up the system designer from tedious and detailed manipulations of knowledge structures and allows the system itself to learn how to recognize. Two of the more successful systems, Sphinx and Byblos, will be discussed.

2.1.1 Hearsay-II

From CMU came Hearsay-II, one of the initial attempts at speech recognition. Hearsay attempted to solve the continuous speech problem for a large vocabulary (1000+ words) in a restricted grammar for the task of document retrieval. Hearsay's entire problem-solving strategy was based on using a centralized data organization device known as a Blackboard. Hearsay-II divided the speech recognition problem into a cooperative problem among several parallel subtasks, and using the Blackboard to allow communication between the subtasks.

These subtasks were:

- To infer segments (phonemes) by parametric analysis. Hearsay-II did not attempt to recognize the exact phoneme at each location of the speech utterance, but instead sought each segment’s manner of articulation by examination of auditory features\(^1\). The decision for manner of articulation was made by appealing

\(^1\)Hearsay-II made no attempt to determine the place of articulation. Finding place of articulation by
to four parametric functions.

- To infer segment strings from individual segments. This meant concatenating strings of segments (which were not phonemes, but simply manner of articulation).

- To infer words from segment strings. This task was difficult because many possible words could be formed out of the segment strings. Further, Hearsay-II would not know what set of strings made up a word (there was no easy means of detecting word boundaries).

- To verify words from hypothesized words and analyze word strings syntactically, making predictions of the syntactic categories for the next word or words. Hearsay would use higher level knowledge in order to determine if a word fit the partial solution of other words.

The overall structure of Hearsay-II's processing is shown in figure 4.

Each one of these subtasks was accomplished by a "knowledge source" (KS). Each KS would examine the blackboard, see if there was something for it to do and generate a priority value dependent on several factors. The KS would determine whether appropriate data was available, whether the task it would accomplish was

---

A feature-based mechanism is very problematical. Because of this, Hearsay-II assumed that manner of articulation was enough information to recognize the utterance as long as other knowledge (higher level knowledge) was available. The place of articulation is the location of constriction in the vocal tract that led to the sound. Manner of articulation was the means by which the sound was created (i.e. how the sound was created, not where). Manner types consist of stops (created by complete constrictions in the airflow), fricatives (created by forcing air through a narrow opening rather than complete constriction), nasals (created by channeling the airflow through the nasal cavity) and so on.
Figure 4: Hearsay-II Flow of Processing

high in priority and whether the KS thought it had a high chance of succeeding. A overall priority would be computed for each KS which had data (on the Blackboard) to operate on. The highest scoring KS would then be allowed to execute.

It should be noted that, in principle, these KS's could be implemented in parallel. Each KS would look at the blackboard to see if there was something for it to work
on, and if so, the KS would do its task, placing results out on the blackboard and wait for more partial results. In fact, this would be very beneficial since Hearsay spent a large portion of its time in making control decisions.

Hearsay's strategy was to work on only the best (or highest rated) hypothesis from the partial solutions posted to the Blackboard. That is, each KS would only examine the best hypothesis posted to the Blackboard. Lower level KS's would generate new hypotheses from the acoustic signal and the output of other lower level KS's. Higher level KS's would attempt to verify the best hypotheses that were posted to the Blackboard. Verification was in terms of syntactic and lexical correctness. For example, a word generated from a segment string might not be in the lexicon. Or, a string of words generated from the lexicon might not have been syntactically correct. Verification was able to remove incorrect hypotheses from the Blackboard. If verification removed some hypotheses, then a KS would examine the next best hypothesis.

The overall results of Hearsay-II were somewhat disappointing based on the amount of effort that went into the system. Hearsay-II accomplished the task of speaker dependent continuous speech recognition. It had a large vocabulary (1011 words) using a context-free grammar. The overall word accuracy of Hearsay-II was 91% when using a constrained grammar. A more general grammar resulted in an overall word accuracy of 74%, a far cry from the accuracy needed in a practical automated continuous speech recognition system.

Some of the problems with Hearsay-II are as follows:
• Lack of linguistic knowledge. While Hearsay-II used the state of the art in linguistic knowledge, the knowledge was lacking in many areas including an accurate means of determining place of articulation. Without knowing place of articulation and only knowing manner of articulation and voicing, the choice of a phoneme is actually choosing among several phonemes. For instance, a voiced stop consonant is one of (/p/, /d/, /g/) and a voiceless fricative is one of (/θ/, /s/, /ʃ/). Obviously, a large number of potential words can be generated from the phoneme strings generated by the segmental KS. This leads to many more word hypotheses being generated than is useful. The accuracy of detecting manner of articulation was 90%, which was reasonable but not very encouraging. A 10% error would lead to a large number of incorrectly hypothesized words. Hearsay’s results showed that it correctly recognized a word 77% of the time while it was able to understand the word’s meaning (within the sentence) at a much higher accuracy, 91%. Therefore, Hearsay’s downfall was its recognition ability and not its higher level knowledge.

• Hearsay-II took a segmental point of view, in which speech is thought of as a series of individual phonemes concatenated together. This notion of having segments of phonetic units concatenated together is known as linear phonology. However, linear phonology is an insufficient theory to account for the effects that one sound has on another sound. There are many dependencies that one sound

\(^2\)θ represents the sound “th” phonetically
will have on preceding and succeeding sounds. One solution to this problem is so-called "coarticulation" where new rules can be assigned to show how adjacent sounds will cause changes in the acoustic signal. Hearsay-II attempted to bypass the problem of dependencies by using coarticulation rules. These rules were applied between phonemes in a word and between words themselves. However, these rules were not adequate in detecting some of the many variations that occur in a speech signal.

- The explicit control strategy based on evaluating each KS and determining which KS should be allowed to proceed cost in overhead. At each control decision, Hearsay-II had to spend a large amount of time in deciding what to do next. While the explicitness of this control strategy offered flexibility and programmability, it took away processing time from the already overburdened system.

- Hearsay-II did not work in a strictly left-to-right manner. The left-to-right processing refers to working through the utterance in a temporal order. Hearsay-II instead ignored the temporal ordering and worked on all portions of the utterance at one time. As will be shown in other speech systems, when working on a left-to-right manner, the speech task can become one of searching through possible utterances. Hearsay-II used a "middle-out" approach where islands of certainty were identified anywhere in the utterance and built upon. While this is by no means the wrong approach, there are two seeming flaws to it. There is no cognitive plausibility for such an approach, humans hear with a clear tem-
poral order, recognizing word in the order spoken. Only when word recognition
is delayed because of ambiguity or noise will a human “reparse” an utterance.
Second, recognizing words in temporal order allow the recognizer to generate
expectations of the “next” word. This severely restricts the number of possi­
ble words that one might hypothesize as the word currently being recognized.
Hearsay-II did allow for grammatical expectations, but only after words had
already been hypothesized.

In spite of these flaws, Hearsay-II did prove a couple of very important things.
First of all, continuous speech recognition was possible. It required vast amounts of
knowledge and a cascaded inference strategy. It also required some form of centralized
communication where inferences made at one level could be passed along to another
level. However, for the most part, Hearsay-II did not succeed. Harpy was a more
impressive system in terms of results although Harpy solved the speech recognition
problem in a completely different manner.

2.1.2 Harpy

Harpy was another APRA project, also coming from CMU\(^3\). However, it was a major
departure from the strategy used in Hearsay-II. In fact, just about the only simi­
larities between the systems are that they both attempt to solve continuous speech
recognition using a large vocabulary (1000 words) in the task of document retrieval.

Harpy approached the problem of speech recognition as a search task to navigate

\(^3\)It is the author’s conjecture that CMU is attempting to corner the market on automated speech
recognition. Most of the research in this field comes from CMU, and CMU is attempting to
monopolize this field with very impressive results from Sphinx.
through all of the possible utterances in a left-to-right manner. The assumption is that all of the knowledge necessary for speech recognition can be compiled into a lattice of phonetic, phonological, lexical and syntactic knowledge. This lattice is possible because of the limited number of possible sentences that can be generated. Such a lattice was compiled where each node in the lattice represented a primitive sound (a phoneme) in one of the words from one of the possible sentences. To compile such a lattice took an enormous amount of machine time (13 hours on a PDP-11), however once this was done, a large portion of the speech recognition problem was completed. An example portion of Harpy’s lattice can be seen in figure 5.

Each phoneme was represented as a template of spectral information. This did not take into account other phenomenon such as "coarticulation" or word boundaries. Therefore, additional templates were created wherever Harpy was prone to making errors. A total of 98 phoneme (and other) templates were created.

Harpy’s strategy was based on a dynamical programming algorithm using a beam search technique to find the “most probable” path through the lattice. This path represented the closest match of the spoken utterance. To accomplish this search, the speech utterance was decomposed into a sequence of temporal slices, each of which contained spectral information. At each point in the utterance, the spectral information was compared against a few phoneme templates chosen based on the beam search. The closest match was picked and the algorithm moved on to the next location.

---

4This assumption does hold true in the case of Harpy, which used a 1000 word vocabulary and a very limited grammar.
The results from Harpy were more impressive than those from Hearsay-II. Harpy achieved a word accuracy of 95% in a very restrictive grammar. This grammar was ridiculously simple and useful only in constrained speech tasks such as document retrieval. However, even in this restrictive domain, Harpy limited the form of utterances to very few possible combinations. This was necessary to keep the lattice of possible utterances to a reasonable size. Any increase in vocabulary or grammar complexity would have required recompiling the lattice which, unless constrained in some manner, would have generated an unreasonably large lattice.

The success of Harpy is double-edged. While Harpy showed a means of continuous speech recognition, it was accomplished by making an unrealistic constraint on
grammars. Harpy would have proved to be more successful if the designers could have shown a way to make a more realistic grammar while keeping in the same approach. This was not possible though. More complex grammars brought about much greater errors. A grammar with twice the complexity resulted in a greater than double error rate (the word accuracy dropped to below 85%) indicating that Harpy was not scalable to solve a larger speech recognition problem.

2.1.3 Byblos

Byblos, a more recent speech recognition system, used an overall approach of combining multiple sources of knowledge (similar to Hearsay-II in that phonetic, lexical, and syntactic knowledge was used, but Byblos went further and attempted to use semantic, pragmatic and discourse knowledge as well). However, the way the knowledge would be encoded and used was highly different than Hearsay. The knowledge would be encoded in mathematical models of probability transitions known as Hidden Markov Models (HMMs).

Hidden Markov Models are extensions to Markov models or Markov chains which can be used to encode state transition probabilities [Rabiner, 1988]. The introduction of Hidden units allows encodings of unobservable states as well as observable states. The result is a mathematical model of some set of events with probabilities which allow one to determine the most probable course through the network of states. Figure 6 shows a Hidden Markov Model for a phoneme5

To use HMMs for speech recognition, a phonetic model must be created. This

5This is actually the model used by Sphinx [Lee, 1988].
would be the form of all phonetic units (such as phonemes) used in the speech recognition task. Then, a new model instance is created for every instance of a phonetic unit used in the system. Speech parameters are modeled as probabilistic functions. They are trained by using a Forward-Backward function [L.R. Bahl and Mercer, 1983] and the result is a set of HMMs which can be used as the knowledge for a speech recognition system. This is the advantage of HMMs as training is done automatically without much need of knowledge engineering crucial to systems such as Hearsay-II and Harpy.
Models are created for every phoneme that Byblos must recognize. Training of Byblos' HMMs occurs by initial creation of the template HMM and then subsequent alterations of each HMM by using 15 minutes of annotated speaker training data. Once created, the phoneme HMMs are used in conjunction with a phonetic dictionary (a dictionary of all possible phoneme sequences) to create word HMMs. The word HMMs constitute all possible sequences of probability functions which can form legitimate words. This lexicon (set of word HMMs) is then the search space used by Byblos during the recognition process.

To recognize the spoken utterance, Byblos compares at each segment of the speech signal, the speech data to the best match in the HMM word model. This is similar to Harpy in that a beam search using dynamic programming algorithms is used to drive the recognition process through a lattice of possible utterances. The major differences between Byblos and Harpy are threefold. First, Harpy stored phonemes as spectral templates and did a best match comparison. Byblos uses the probability functions to determine the most probable transition from one node in an HMM to the next node, or from one word HMM to the next word HMM. Second, Byblos does not precompile a lattice as is the case with Harpy but instead, uses the preformed word HMMs and when the end of the current word HMM is reached, a small number of probable succeeding words are generated. Only those very probable words are considered as the next transition. Third, Byblos uses an explicit grammar checker so that, at each stage of the recognition process, unlikely word HMMs can be discarded. This differs greatly from Harpy which expanded the lexicon and grammar into a giant
lattice. Byblos' overall flow of processing can be seen in figure 7.

One major strength of Byblos is the trainability of HMMs. Because the HMMs can be trained on a speaker's voice and mannerisms, the HMMs are suitable for that speaker. By using a technique of speaker adaption, a new speaker need only speak 15 seconds of utterances to adapt the previously created phoneme HMMs to a new set of speaker-specific HMMs.

Byblos achieved high accuracy (98.5% word accuracy for speaker trained HMMs, 97% word accuracy for speaker adapted HMMs) for a 350 word lexicon using a mod-
est context-free grammar. The accuracy dropped to 90% word accuracy without the use of a grammar. Byblos can be thought of as solving the continuous speech, multi-speaker problem using a small lexicon and restrictive grammar while achieving a very respectable word accuracy.

2.1.4 Sphinx

Sphinx, also a CMU speech recognition system, has continued the research on HMMs for speech recognition. Similar to Byblos, Sphinx uses trained HMMs as the phonetic units for speech recognition. However, Sphinx has used a variety of additional techniques to improve on initial accuracies with a lexicon of nearly 1000 words.

Sphinx departs from Byblos by having no explicit grammar. Instead, the grammar is encoded as probabilities (just as the phone and word models use probabilities). Three types of grammars were tried, bi-gram grammar, word-pair grammar and no grammar. The word-pair grammar is the probability of a word following the given word where there is a limited number of succeeding words (approximately 60). Therefore, there is approximately a one in 60 chance for each word. The Bi-gram grammar represents the probability that word A follows word B, where the probabilities are compiled by using the sentences in the training set. Here, there is approximately a one in 20 chance of one word succeeding another. No grammar means that any word could follow any word, no probability was assigned. Here, the probability for each word following the next is a one in 997 (the total number of words that Sphinx knew).

Initial word accuracies for Sphinx were poor, 74.8% for bi-gram grammar, 58.1% for word-pair grammar and 25.8% for no grammar. However, measures were taken to
improve these accuracies. These improvements are:

- Bilinear transforms - transforming the frequencies in the acoustic signal from straightforward hertz scale to the logarithmic scale of barks [Zwicker, 1961, Davis and Mermelstein, 1980]. This more accurately reflects how the human ear processes the acoustic signal.

- Multiple Codebooks - the spectral information used for recognition in Sphinx is broken into a set of vectors. The original set of vectors constituted 256 values. By increasing the amount of spectral information, multiple codebooks became necessary, each one containing some subset of the overall information. This improvement is simply increasing the amount of information used for recognition.

- Increased lexicon - the lexicon and phone HMMs were increased to contain, not only the lexical pronunciation of words, but variations of the word pronunciation which occurred during training. The number of phone HMMs were increased to include variations found during training such as integrating closure-stop pairs into single units, adding flapped versions of /t/ and /d/, reduced nasal /t/’s and other phonetic dependencies which simple phoneme models cannot capture alone.

- Duration - word duration probabilities were introduced to add further knowledge to the system.

After each change was made, the system was rerun on the speaker data using the three types of grammars. Results after all changes showed a definite improvement
with word accuracies being 55.1% for no grammar, 85.7% for word pair grammar, and 91.4% for bi-gram grammar.

Two further changes were made to the system. First, function words (prepositions, the words “and”, “the”, “a”, “in” and other specific words in the English language) were introduced as specialized HMM word models. That is, rather than creating a word model by combining the /æ/ HMM with /n/ HMM and the /d/ HMM, a single word model for “and” was created. Function words are often poorly articulated or overlooked entirely. By created specific function word HMMs, this problem can be handled more easily. A set of 42 function words were trained and introduced to the lexicon replacing previous HMM versions of these words (created by concatenating phoneme HMMs).

Second, triphone models were introduced rather than phoneme HMMs. Triphones take into account the physical interactions of the sound with the preceding and succeeding sounds. Triphones can be thought of as specially trained phoneme models which take into account coarticulation. The advantage of triphones is that dependencies between phoneme sequences can be captured in the triphone models. For instance, the form of a /p/ may differ depending upon the preceding and succeeding sounds. Therefore a triphone for every possible context of /p/ is needed (such as an initial /p/, a final /p/, a /p/ which follows /s/, a /p/ which precedes /t/ and so on). The disadvantage of using triphone models is that the number of phonetic HMMs needed increases drastically. The number of phoneme HMMs in Sphinx was around 50. The number of triphone HMMs after this expansion was around 2400. Obviously,
the complexity of the system was greatly increased.

The results of these two further enhancements was quite striking. The version of Sphinx which used explicit function word models and triphone models had a word accuracy of 71% for no grammar, 94% for word pair grammar and 96% for bi-gram grammar.

Similar to Byblos, Sphinx had to be trained by lengthy speaker training sets. However, once a base set of HMMs was created for a speaker, similar speakers (such as same gender, same dialect) could use speaker adaption of the already existing HMMs in order to generate a reasonable set of new HMMs.

One large criticism can be made of Sphinx in that, the bi-grams are a very biased form of grammar. The bi-grams are generated based on the speaker training data. Therefore, the grammar is biased to the sentences used in the training data. If those sentences were the only ones being used to test Sphinx, then the results do not accurately reflect what Sphinx can do. For example, in using the bi-gram grammar, if no training sentences contained the expression “brown dog”, then if the word “brown” was recognized, Sphinx would not give “dog” much of a chance of being accepted. However, it is not beyond reason to utter the expression “brown dog”. Thus, Sphinx’s high accuracy was based on the bias of the training sentences.

To make Sphinx more realistic, a real grammar is needed, similar to Byblos. This flaw of Sphinx will be discussed in chapter 7 when we discuss implications of layered abduction as it pertains to HMMs.
2.1.5 Conclusion of Past Speech Recognition Systems

This section has examined a few important speech recognition systems. Two were developed during the early and mid 1970's, Hearsay-II and Harpy. While neither of these systems were thought of as particularly successful, they both illustrated the possibilities and limitations of automated speech recognition. Of particular interest were two very diverse means of solving the problem. Harpy was more impressive for two reasons, speed and accuracy. The reason that it was so quick was based on precompiling a lattice of every possible utterance. This in turn led to Harpy being able to consider more possible utterances at one time, which made it more accurate. Harpy also used spectral templates to match against the acoustic signal in order to drive the system to achieve higher accuracy. However, Hearsay-II used a knowledge intensive approach using cascaded inferences and cooperative and opportunistic problem solving. Hearsay-II showed that large knowledge based systems were possible in complex areas. Both systems showed the need for better linguistic knowledge for place of articulation and something better than simple "coarticulation" rules.

Sphinx and Byblos were impressive in their achievements showing for the first time, high accuracy in continuous speech recognition. These two systems used Hidden Markov Models created by training methods using speech data. Once trained, these systems used a method similar to Harpy in that they worked through the search space of possible utterances using a dynamic programming algorithm. However, where Harpy precompiled an entire utterance space, Sphinx and Byblos build theirs at run time, constraining the size by beam search. While Sphinx and Byblos have achieved
high accuracy and solved both the speaker independent problem and the continuous speech problem, they both have some flaws due to the use of HMMs. These will be discussed in detail in chapter 7.

One question arises as to both Harpy and the HMM systems. Are they scalable? This is, of course, a very important question in speech recognition. Harpy's strategy was to precompile all of the possible utterances into a gigantic lattice and use a dynamic programming algorithm to search for the “most likely” utterance. However, with more words or a more complex grammar, this lattice must be recompiled. While this in itself is not a problem, it will create a larger lattice to work with. As shown in experiments, a more complex grammar had the direct result of worse accuracy. A realistic grammar would not even be possible. That is, because a true grammar for English would require the capacity for infinite generation, there would be no possible way to precompile all possible utterances into a grammar. Even with restricted grammars, Harpy was not very expandable. Doubling the lexicon size would result in a much greater lattice (i.e. the lattice would not just double in size). Therefore, Harpy is not very scalable because of the limitations of the approach.

On the other hand, HMM systems are scalable. There is no restriction such as using more complex grammars or more words. However, as will be discussed in chapter 7, HMMs restrict the form of the knowledge. HMM systems can only use probability models. Additional knowledge of a form different from transition probabilities can only be used outside of the system itself. Byblos uses an explicit grammar-checker which can be used to prune away poor word choices in between word models. Sphinx
got by without a real grammar but used probabilities generated from test sentences. In Sphinx's case, a more complicated grammar would mean any grammar at all! In Byblos' case, a more complex grammar would require implementing a more complicated grammar-checker, which is simply a matter of constructing a new module. For these systems to expand their lexicon, the new words would require new word models which are easy to construct once the individual phone models (or triphone models) are trained. Whether the systems would continue to exhibit a high accuracy is questionable, but based on past performance, it is not unreasonable to assume that lexicon sizes could be increased without more that a slight drop in accuracy.

Hearsay-II is also expandable just as the HMM systems are expandable. Rather than requiring new HMMs though, Hearsay-II would require additional knowledge to capture the new words. Additional knowledge would be in the form of phonetic entries into the lexicon (in terms of manner of articulation) and possibly some additional coarticulation and boundary rules. Minor changes would be required to include any new words. The resulting performance would depend on how tricky the new words were (especially if new words closely matched some of the original words), and how many new words were added.

2.2 Previous Works in Abduction

Abductive research in Artificial Intelligence can be traced back to Internist-I, the first system to use an explicit abductive approach. However, abduction is such a common form of inference, other works can be thought of as implicitly abductive. For instance, Hearsay-II's approach was to attempt to infer from the speech signal, what
was said by using a process of hypothesization and composition (similar to what I will
discuss in chapter 3). Hearsay-II did not attempt to make an explanation as such, the
strategy used was an opportunistic hypothesization format where the hypothesized
words would "cover" the acoustic signal. Other past systems can also be thought of
as implicitly abductive. This notion of implicit versus explicit abduction seems to
be important. We can solve a seemingly abductive problem by some non-abductive
(non-explanatory) means. However, if we choose a non-abductive form of problem
solving, do we achieve the same solution? Will the problem solving be as meaningful
or efficient? I will attempt to answer these questions in the next chapter.

Since my point is to show the power of an explicit abductive strategy, I will exam-
ine some researchers who have used such an approach. We will first look at Pople's
Internist-I which used a straightforward hypothesization approach with some domain-
specific heuristics. We will then turn to Jerry Hobbs who used a first order predicate
calculus language and a backward-chaining approach to generate assumptions and
generate a best explanation as the least costly assumptions. We will then consider
two approaches that are large departures from Hobbs and Pople, the probabilistic
approaches of Reggia and Peng, and of Pearl, and finally the connectionist approach
of Thagard.

2.2.1 Pople and Internist

Harry Pople implemented a medical system to diagnose internal diseases. His ap-
proach was to use the power of explanation to come to a diagnosis. His argument
was that diagnosis is actually an example of inference to the best explanation.
Internist [Pople, 1977a, Pople, 1982b] (and later Caduceus) was the implementation of Pople's ideas. Internist used explicitly-represented disease hypotheses in the form of a disease classification tree. The goal of Internist was to come to the best complete and parsimonious explanation for the input data. This was accomplished by a method of hypothesis, scoring and covering (or explaining) based more on a hypothesis-driven method than a data-driven method.

The overall algorithm for Internist consisted of the following steps.

- Score each available hypothesis - this score was computed as a result of the hypothesis' overall explanatory coverage. In detail, the score is a combination of the weights of findings that could be explained by the hypothesis, the weights of the finding which evoked (or suggested) the hypothesis, the weights of the findings which were expected by the hypothesis but not present in the current case, and the weights of the findings present but not explained by the hypothesis. Finding "weights" were a factor known as "import" which stood for the importance of that finding. The import would be used in several ways, including hypothesis scoring.

- Sort all hypotheses from best scores to worst scores. Hypotheses which are scored low are dismissed (at least temporarily).

- A group of competing but high scoring hypotheses are then considered (competing hypotheses are those which can account for some of the same findings).

- If only one hypothesis is being considered, it is considered as true. If several
hypotheses are being considered, then consider the "best" one as true (as long as it is sufficiently better than the next best hypothesis).

- If a hypothesis cannot be concluded as true, then begin to ask questions to differentiate between top competing hypotheses. Analyze the answers and reevaluate the hypothesis scores, beginning this cycle again.

- When a hypothesis is concluded as true, all findings that it can account for are removed. The cycle resumes by creating a new group of competing hypotheses.

- When only findings of import 2 or less are left unexplained, the program stops.

Internist took into account the explanatory power of a hypothesis and the expectations of that hypothesis in creating a diagnostic conclusion. Internist used a hypothesis driven approach as it attempted to find the best hypothesis, which was then used as an explainer (rather than considering a datum and determining what could best explain it). One problem that arose because of this strategy was when hypotheses were not scored appropriately. A hypothesis with a high score had a large chance of being concluded as true even if it couldn't explain any of the findings!

Another problem of Internist was its lack of criticism for the accepted hypotheses. This was brought about because Internist ignored the possibility of alternative explanations once an explainer was found. For instance, if H2 and H3 are alternative ways of explaining something that H1 could explain, and H1 was concluded, then H2 and H3 were ignored. Thus, a potential better explanation was dismissed as soon as any explanation was found. We can summarize this mistake as saying that Internist
would commit too early in the diagnostic process.

In spite of these problems, Internist-1 was a successful attempt at medical diagnosis. What it proved was the advantage of explanatory knowledge. Inference to the best explanation would soon become a means of solving a variety of problems besides medical diagnosis. From the strength of his work, more people have “seen the light” of abduction.

2.2.2 Hobbs

Hobbs et al [Hobbs et al., 1988] describe a method for abduction for the purpose of sentence interpretation using discourse knowledge. They use an algorithm based on backward-chaining across predicates represented in first order predicate calculus. The goal is to determine the best subset of predicates. This abductive conclusion can then be used to explain the sentences used as input. Each predicate has a cost of believing it. The abductive process consists of proving the least costly predicates where the predicates are assigned costs based on two criteria. These criteria are “how specific the predicate is” (the more specific, the higher the cost in believing), and an intuitive sense of “semantic contribution” to the overall proof. The “intuitive sense” is not clearly defined (in fact, it is very ad hoc) but this cost can be adjusted until the abductions provide appropriate behavior.

The abductive strategy revolves around three assumptions. First, it should be possible to assume goal expressions (logical expressions to be proved). That is, the goals can be stated as predicates and well-formed sentences of predicates. Second, assumptions should be expressible at varying levels of specificity (so that there is
a trade off between detail of a hypothesis and the confidence or cost in believing the hypothesis at that level of detail). Predicates can be stated in general terms or specific terms (or somewhere in-between) such that the more general predicates are easier to conclude but have less explanatory power. More specific predicates could explain more but are harder to conclude. Third, there should be a way of exploiting the natural redundancy of natural language found in texts. This is something that has been presupposed by many working in natural language. The belief is that the redundancies can be used to leverage against the ambiguities involved in language.

Hobbs et al have used this form of abduction in a system called TACITUS, for interpretation of malfunction reports (in the domain of mechanical diagnosis). The system must base its decisions of specificity versus correctness depending upon which costs less. Since specificity is a higher cost, the desire to be as specific as possible is curbed. However, if enough evidence points to a specific predicate, then the cost is lower than a more general conclusion. In the end, the abductive conclusion is composed of the predicates which were proven true by the backward-chaining algorithm and were the least costly. In this method of abduction, there is no explicit notion of a predicate explaining a datum. Instead, the predicates taken as a whole are used to explain all of the data (the input sentences).

In order to come to an abductive conclusion, the following types of explanations are required. First, references must be resolved. That is, it must be determined as to what the reference is referring to. Second, determination of implicit relations between compound statements. That is, modifiers must be determined as to what
they modify. Also, syntactic ambiguities must be resolved so that the role of the words is discovered. Also, metonymy must be resolved. Metonymy occurs when a word is used as an association or attribute with something it is related to.

TACITUS is first given some sentence or sentences. It then expresses assumptions in terms of first order predicate calculus sentences. TACITUS's task is then to prove (or disprove) some set of these sentences in order to resolve the references, relations, ambiguities and metonymies. It should be noted that second order predicates are often required to represent some of the assumptions. A first-order simulation is then used in order to resolve these more complex predicates. This can be accomplished by further assuming a constant value for some of the predicates so that they become first-order. However, in making these assumptions, TACITUS must either create all possible assumptions (which would lead to an exponential number of such assumptions) or only make certain assumptions and hope that they are correct. The latter is the case for TACITUS where notions of commonality and typicality (in language) are used to create the assumptions.

In conclusion, Hobbs uses abductive explanation to derive sentence meanings in natural language. His method is one of first-order predicate calculus resolution, and the criteria for the best explanation is the proof (if one can be found) with the least costly assumptions.

2.2.3 Reggia and Peng

for medical diagnosis. A causal network is used in order to produce the best explanation of symptoms in terms of diseases. In order to come to a conclusion, disease hypotheses must be chosen to explain the given symptoms such that a complete cover is found. The criteria for choosing the best set cover is based on principles of minimality (or parsimony), consistency, complete coverage and highest degree of plausibility (highest probability).

Similar to Pople, they use explicit knowledge of causation and explanation. When a finding is to be explained, the possible causes of that finding are considered. However, finding the best explanation is accomplished by using a causal network to determine possible hypotheses, and probabilistic functions using a Bayesian probability calculus to score the hypotheses. This computation requires considering two probabilities, the probability of the disease occurring given the prior probability of that disease, and the probability of causation (the probability that the disease will cause the manifestation to occur).

Generating a best explanation requires only instantiating the manifestations and computing a series of probabilities. The disease hypotheses with the highest probabilities are then considered the best explanation as long as the criteria for bestness are met. Reggia and Peng assume that hypotheses are mutually independent so that covering the data can be accomplished in a piece-meal fashion and that the hypothesis probabilities will not change because of the acceptance of other hypotheses.

Two extensions of this approach over past General Set Covering models are the incorporation of probabilities in order to derive the "best" explanation, and the use
of intermediate states in the model. Past models have only used two sets of states (or nodes), manifestations and diseases. In [Peng, 1986], intermediate nodes are used so that patho-physiological states can also be represented. Thus, to generate the best explanation, one must first determine certain intermediate states in the diagnostic process, and then explain those in terms of diseases.

The approach taken by Reggia and Peng has become popular and used in other areas including language processing [Venugopala and Reggia, 1988, Dasigi, 1988], error classification [Ahuja, 1985] and memory modeling [Reggia, 1985b].

2.2.4 Pearl and Belief Nets

Judea Pearl [Pearl, 1987] has implemented a version of abduction problem solving using a combination of causal networks and Bayesian probabilities to solve problems in the domain of medical diagnosis. In Pearl’s Belief Network scheme, the domain is laid out as a network of hypothesis and manifestation (or data) nodes. Hypotheses which can explain manifestations are directly connected together. Three forms of probabilities are used to connect hypothesis and manifestation nodes, evidential probabilities and causal probabilities (representing the probability of the symptom appearing given the disease and the probability of the disease given the symptom respectively), and prior probabilities for each hypothesis node. Probabilities are assigned to the links between nodes and to hypothesis nodes prior to using the network. It should be noted that deriving accurate probabilities is very problematical. However, Pearl states that accurate probabilities are not necessary for a system to behave in an appropriate manner.
A sample belief network (taken from [Pearl, 1987]) is shown in figure 8.

Figure 8: Sample Belief Network

To use the belief network, certain manifestation nodes are activated (those manifestations which are present in the current case). This causes evidential knowledge to be propagated to the hypotheses nodes. Hypothesis nodes which receive information from manifestation nodes (or other hypothesis nodes) update themselves by computing, via Bayesian probabilities, their beliefs. This computation is performed based on the three forms of probabilities and are computed given the initial activation of manifestations. The belief of a hypothesis node is the probability that the hypothesis node is true based on the values propagated to it and the prior probability stored in that node. Once a hypothesis node computes its own belief, it propagates this
value to other nodes in the network. Beliefs are propagated around the network, moving between hypotheses' nodes and between hypothesis and manifestation nodes until the network settles to some stable state. The hypotheses nodes which have sufficiently high belief are considered true and can be used to explain the activated manifestations. All other hypotheses are considered false.

One restricting problem with the belief network scheme is that most real world situations are complicated in manners which cause a belief network to be "multiply-connected", a situation where the belief network contains cycles. The propagation scheme of passing beliefs will not work in a multiply-connected network. To resolve this problem, Pearl suggests a method in which some hypothesis nodes are preinstantiated as true or false. If the correct nodes are preinstantiated in this manner, all cycles are removed from the network and the network becomes singly-connected. However, to adequately determine if the hypotheses were preinstantiated correctly, the network must be run for each possible combination of truth and false values for the preinstantiated hypotheses. Then, for each run case, an overall "case" probability must be computed. The most probable set of hypotheses (i.e. the best explanation) depends upon which "case" had the highest probability. That case dictates the best explanation as a combination of the preinstantiated "true" nodes plus any other hypothesis nodes which surpassed the activation threshold after the network stabilized. Obviously, this leads to an exponential number of run cases (because we would have to run the network once for each combination of true and false values for the preinstantiated nodes).
Therefore, Pearl's belief networks are useful only in two circumstances, either the domain is restricted such that the network is singly-connected (and this will not occur in problems where more than one explainer is needed for a diagnosis, such as with multiple faults, or in problems with highly interacting hypotheses) or if the minimal number of possible preinstantiated nodes is sufficiently small that the number of runs also remains small in spite of having to preinstantiate some of the nodes in the network.

2.2.5 Thagard and Neural Nets

Paul Thagard[Thagard, 1988] has implemented a version of abduction using neural networks. Thagard's version of abduction is to make a decision between two (or possibly more) theories or two verdicts. Hypotheses are divided into sets supporting one of the possible theories or verdicts. Each set could have potential explainers for a given data item. An example of a portion of an ECHO network is shown in figure 9.

Thagard's network contains explicit hypothesis and data nodes. Explanatory knowledge is given by connecting hypothesis nodes to data nodes where a hypothesis is able to explain that datum. The network is run by propagating activations around the network until it stabilizes. Hypothesis nodes which are still active are considered as part of the composite explanation while hypothesis nodes which are not active are considered ruled out.

Thagard is able to capture a variety of knowledge in his networks (other than explanatory) by adding links between different nodes. Hypothesis interaction knowl-
Sample network showing hypothesis H1 and H2 each explaining E1 and E2. H1 is incompatible with H2. H1 is explained by theory T1. H2 is explained by theory T2.

Figure 9: Sample Network for ECHO

ty (incompatibilities, associations) are given by connecting hypothesis nodes to other hypothesis nodes. Incompatibility connections are made with inhibitory links while association connections are made with excitatory links. Analogy, simplicity and "unification" (where two or more hypotheses must unite to explain a finding) are all possible. Weights for the links are determined based on the amount that a hypothesis can cause a finding, or the degree of association or incompatibility between
Thagard used this strategy, incorporated in a tool called Echo [Thagard, 1989] to implement several different small abduction problems including two on legal reasoning and two on theory formation. Thagard uses Echo in order to make decisions between two bodies of possible explanations. In his examples of theory formation, Thagard implements a theory decision maker to determine which of two theories is more reasonable (in terms of consistency and explanatory power) and in legal decision making to determine if the evidence points towards innocence or guilt.

Limitations of this approach revolve around the way knowledge is encoded. There is no means of "exhaustively searching" for plausible hypotheses in order to explain some finding. Instead, the system builder predesignates the two or three hypotheses that are to be used in the system. Encodings can be thought of as arbitrary and very subjective. However, if the domain and problem break down easily into two (or potentially more) sets of explanatory hypotheses, and one wishes to base a conclusion primarily on consistency and explanatory power, this approach is reasonable. Other criticisms of this approach are the inefficiency of having to train and execute the network (in [Josephson and Fox, 1991], we show that a case implemented in ECHO is inefficient in comparison with the same case implemented in Peirce\textsuperscript{6}), and the unavailability of explanations or other usages of the knowledge encoded into ECHO. Finally, ECHO can only be used currently to decide between two sets of alternatives rather than as a general tool for diagnosis or story understanding.

\textsuperscript{6}Peirce will be explained in detail in the next chapter.
2.2.6 Conclusion on Abduction Research

As can be seen from the above survey, there are many ways of implementing abduction. Each of these strategies has some commonalities: namely, that abduction is the process of generating hypotheses to explain (or cover) a set of data, and the generated hypotheses should be the most plausible at a minimum cost. All of the approaches use explicit hypotheses and the explanatory knowledge is also made explicit. Also, each of these strategies defines the "best" explanation using similar terms such as most parsimonious, simplest, most probable, most consistent and so forth.

The differences in the above approaches lie in how the explanation is generated and how evaluation of hypotheses is done in order to choose the best one. Pople and Reggia weigh the choices of one hypothesis versus another in coming to conclusions about what can explain some datum. For Reggia, local conclusions can be combined with the assumption that hypotheses will not interact. Hobbs uses a variation of first-order predicate calculus theorem proving by backward-chaining in order to find the least costly explanation (where cost is directly proportional to "bestness" or plausibility). Pearl and Thagard use a form of propagating beliefs around a network, letting the network weigh overall and individual choices in hypothesis selections.

Another difference in approaches between Pearl and Thagard and the others is the separation of hypothesization and hypothesis scoring. Reggia, Pople and Hobbs all search for hypotheses separately from scoring and evaluating the necessity of a hypothesis. This is blurred in the works of Thagard and Pearl who consider all available hypotheses in their net.
In the next chapter, I will introduce a particular method for abduction, similar to Pople and Reggia and Peng, which uses explicit hypotheses and explanatory knowledge to incrementally construct a composite hypothesis from lesser hypotheses. I will show the roots of this abductive strategy from past research, and then discuss a domain independent strategy which has evolved from the past research. I will then describe some small prototype systems constructed using this strategy and finally discuss how we can solve layered abductive problems using this strategy.
CHAPTER III

A General Strategy for Layered Abduction

3.1 Introduction

Research over the last 7 or 8 years in the LAIR (Laboratory for Artificial Intelligence Research) has brought about several variations of a powerful abductive problem solving strategy. This strategy is an explicit one, meaning that the phases involved in the abductive problem all require the explicit consideration of hypotheses, and what the hypotheses can explain of the data. This approach is similar to that of Internist and the methods of Reggia and Peng, and differs highly from Pearl and Thagard, who use little or no explicit explanatory control. The degree of explicitness is the degree to which hypothesis knowledge (of various types) is available and used during the course of the abduction. In addition, the explicitness of the strategy depends on whether the system explicitly considers (or weighs) hypotheses against each other in order to come to a conclusion. In Pearl and Thagard's works, there is little notion of "explaining this datum now" but rather, a global effort of generating a consistent best explanation. It will be shown that in our work, there is a very definite notion of examining the causes of each finding in order to come to a conclusion. The LAIR abduction algorithms all have explicit control.
The LAIR work has based abduction on three separate subtasks, hypothesis evocation, hypothesis instantiation and hypothesis composition. Each of these will be explained in detail.

- **Hypothesis Evocation** is the process of finding plausible hypotheses given a set of data to be explained. While abduction does not dictate the means by which hypotheses can be selected or generated from a search space, abduction requires some means of hypothesis generation. One efficient means of searching for hypotheses is by classification. Classification is a task of generating specific hypotheses from general classes of hypotheses. This task can be used as a subtask of abduction in order to generate plausible hypotheses to be used in an abductive explanation. Other means of hypothesis evocation include cueing (i.e. using data to cue specific hypotheses), generation from a functional representation model, heuristic search strategies through some organized hypothesis search-space, and random generation of hypotheses.

- **Hypothesis Instantiation** is the process of determining a hypothesis' relevance for the current case. This involves first scoring the hypothesis to determine plausibility. Such a plausibility score can be found by local matching knowledge using the presence and absence of features of interest to come to some statement of plausibility (whether this statement is a probability, connectionist activation, or qualitative statement of confidence). The second step is to determine what data items the hypothesis can explain in the context of the current case. Hypotheses can account for certain data items in general, and
other items in particular cases. Again, the method of scoring and determining explanatory coverage is not dictated by abduction.

- **Hypothesis Composition** is the process of generating an explanation to account for the data. This process can be an incremental building of a composite hypothesis (such as in Internist, or Red as explained below), or a more holistic composition process (such as in ECHO or Pearl’s Belief Networks). During the composition process, or after the composition process, the composite hypothesis can be criticized for overall plausibility, parsimony, and other evaluating features.

Figure 10 shows this process of hypothesis evocation, instantiation and composition.

The strategy currently used in the LAIR for hypothesis composition is a domain-independent strategy which has evolved from past research in medical test interpretation (from the Red Antibody Identification System) [Smith et al., 1985, Smith, 1985] and medical diagnosis (from the Pathex Liver Specialist [Smith et al., 1988] and others [Punch, 1989]). A domain-independent strategy for hypothesis scoring (part of hypothesis instantiation) also exists. Strategies for hypothesis evocation exist but the choice of which strategy to use depends for the most part on the domain. Cueing hypotheses is one means of generating hypotheses. Hierarchical classification is another means of generating hypotheses. Determining hypothesis coverage is also a task specific to the domain. Work in the LAIR has led to the construction of tools for hypothesis composition, hypothesis scoring, and hypothesis generation via hierarchical
classification. Other tasks involved in abduction can be solved by several methods, none of which are captured as tools (yet).

This chapter will explore the abductive strategy of the Red system, and then consider a general domain-independent problem solving strategy for hypothesis composition (also known as abductive assembly). This chapter will then look at some
problem solvers constructed from this domain-independent strategy. The chapter will conclude with an examination of using the abductive strategy described for the purposes of layered abduction.

### 3.2 Red: for Antibody Identification

Red is a knowledge-based system for Red blood cell antibody identification in the blood-banking domain. While the system has changed over the years, the prototypical version of Red (known in the LAIR as Red 2) is based on a data driven, opportunistic algorithm which first attempts to classify antibody hypotheses and then, using this set of hypotheses, form a composite best explanation.

The data to Red is a set of findings about blood tests pertaining to reactions when mixing test blood serum with donor's blood. Any reactions indicate incompatible antibodies. These reactions are used as findings to be explained. The reactions are indicated as numbers from 0 (no reaction) to 4+ (large reaction). A typical set of reactions can be seen in figure 11.
Hypotheses represent whether types of antibodies exist in the patient’s blood. Antibody hypotheses know what types of reactions and the strengths of reactions which they can account for. This domain is one in which hypothesis coverage is “additive” meaning that two (or more) hypotheses can be combined in order to explain a full reaction.

The system first attempts to find by classification a subset of plausible antibody hypotheses to explain the reactions. Each hypothesis is scored based on local matching with available data. This score comes from patient-specific information, the reactions found in the blood tests, and knowledge of patterns of reactivity associated with antibodies. The result of the classification phase is a set of plausible hypotheses which can be used to explain various portions of the data. This first step is the hypothesis evocation in which a set of hypotheses are found. This first step also is the hypothesis instantiation phase as hypotheses are given an initial plausibility and each hypothesis determines which part of the data it can explain.

The next step in RED is the “abductive assembly” step which is itself a 5-part strategy. This 5-part strategy is the hypothesis composition phase which attempts to find a subset which is both parsimonious and maximally plausible [Josephson et al., 1987].

The first part of the 5-part strategy is accomplished by choosing a reaction for focus of attention, then picking the hypothesis which best explains it (i.e. pick the most highly rated hypothesis which can explain it). This hypothesis is placed into the composite explanation. The effects of believing this hypothesis are taken care of.
This constitutes removing any reactions which this hypothesis can explain (besides the chosen reaction) as well as removing any incompatible hypotheses from the composite. After the first hypothesis is accepted, another reaction is chosen for focus of attention, and the best hypothesis is picked to explain it. This procedure continues until a composite has been constructed which explains all of the reactions. This process can be thought of as generating several "local abductions", one abductive conclusion for each reaction that is considered. The result of all these local abductions is a single composite explanation of (potentially) many antibody hypotheses.

At this point, the composite is an explanation but possibly not the best explanation. The composite is criticized in an attempt to make it better. The composite is checked for parsimony. This entails removing explanatorily superfluous hypotheses (superfluous hypotheses are those which do not contribute to the explanation because other accepted hypotheses are capable of explaining what the superfluous hypotheses explain).

Next, the composite is examined for essential hypotheses. Essential hypotheses are hypotheses which are necessary for explaining some reaction. Without an essential hypothesis, some reaction (or reactions) will not be explainable given the set of hypotheses generated from the classification phase. Essentials are an important notion to our strategy of abduction and will be discussed in more detail in section 3.3.1.

A new composite is now formed. This composite starts with the essential hypotheses. These hypotheses represent the most confident conclusions that can be reached
so far in the problem solving. This new composite initially has only essentials. The findings that the essentials explain are removed. The composite is then filled out, similar to the first part of the strategy by reactions local abductions for all remaining reactions. A reaction is considered and the best hypothesis which explains it is included. This continues until all of the reactions are accounted for.

This new and improved composite is checked for parsimony. Superfluous hypotheses are removed from the composite. The resulting composite is one based on essentials and highly plausible hypotheses. It is also consistent because inconsistent hypotheses are removed from the composite at the time that an incompatible hypothesis is placed into the composite. Figure 12 shows a sample output from the Red-2 system.

Red has existed for many years now in various forms. The original Red system (now defunct) had a simpler abductive strategy. Research has continued on Red in order to explore many strategies for abduction. This research has indicated some interesting ideas that are useful in abduction. In fact, this research has led to the development of better domain-independent strategies for abduction.

3.3 The Necessity for an Explicit Abductive Strategy

The above strategy is one which has much to offer for general abductive problem solving activities. If we remove the domain specific knowledge, we are left with an easy means of generating explanations, a means which can take advantage of many factors used in such decision making processes. By appealing to the explanatory power of hypotheses, we are able to gain more leverage on problem solving than by
Antibody Classification Report

The best available explanation for the reactions in the workup is: antiSlgG antiIgG

The following antibodies have been ruled out: antiD antiC antE antiCw antiM antiFya antiFyb antiJka antiJkb antiKp b antiKp h antiLub antiL

There are no antibodies that have been classified as CONFIRMED.

The following antibody is classified as WEAKLY CONFIRMED: antiS

The following antibody is classified as LIKELY PRESENT: antiK

The following antibodies are classified as LIKELY ABSENT: antiN antiLwa antiLea antiLwh antiI

The following antibodies are classified as UNRESOLVED: antiKpa antiLsa

Figure 12: Sample Red Composite Explanation with status of other explainers

simple pattern matching or search techniques. In [Tanner et al., 1991], Tanner and Josephson show that hypothesis matching alone is less effective than a system using abduction, that an abduction system can generate more correct answers. This can also be seen in some initial work on ARTREC [Fox et al., 1992a], and also discussed in chapter 6 (experiment 2, 3 and 4). Without an explicit abductive strategy, some of the power of explanatory coverage is lost. The power of leveraging explanation is an important concept. Systems that are not attempting to generate explanations lose this power. This notion will hopefully become more apparent in the subsections to follow as I discuss what is to be gained by using explanation explicitly.

In researching abduction, the LAIR has discovered a number of important issues
involved in abductive inference. We will consider these issues and show how they can be used in an explicit explanatory strategy in order to become more powerful (more correct, more efficient).

3.3.1 Essentials

Perhaps foremost in importance is the notion of essentials. We have termed an essential to be a hypothesis which is the only plausible means of explaining some set of data [Fischer, 1991, Josephson, 1990]. Without an essential hypothesis, some datum or data cannot be explained. If a goal for abduction is maximal or complete coverage, then essentials are required for the best explanation (that is, without the essentials, an explanation will not be complete). Essentials are, quite literally, essential for the best explanation.

Essentials are important because they are the most confident portions of the composite. That is, an essential is the only means of explaining something, and because there are no competitors for an essential, we can feel secure in accepting it. Therefore, essentials allow an abduction system to start with islands of relative certainty.

There are reasons for not including essentials as part of the best explanation. These include conditions when there was not an exhaustive search for alternative explainers, or the essential is very poorly rated (in terms of plausibility). We might not want to include an essential under such conditions because accepting the essential is less certain. In a case where an exhaustive search for competitors was not carried

\footnote{I use the term "relative" here because we can never be 100\% sure that a hypothesis is in fact essential for reasons described next.}
out, we can not be certain that the essential is truly an essential. Other means of explanation might exist, we just haven’t found them yet. In a case where an essential is poorly rated, we might not wish to accept the essential because it would weaken our overall confidence in the composite. An essential might be poorly rated if the data is noisy or if we are lacking in some knowledge which would help us score the hypothesis with more plausibility. In such a situation, we must decide whether to accept the essential based on a pragmatic consideration of “is the coverage worth the chance of being wrong?” However, in cases where we can be confident that the essential is truly the only legitimate means of explanation, then we can state with high confidence that the essential is true.

Essentials will rarely occur naturally. That is, we will not usually find a single potential cause for some finding. Instead, “rule-out” knowledge can be used to remove clearly implausible hypotheses from consideration. Rule-out knowledge can come in many forms from associational knowledge (observations we expect to see which are not present in this case) to causal knowledge (showing that the cause could never arise in the given conditions) to statistical matching knowledge (showing that the data does not statistically match with the exemplar or template knowledge). With this type of knowledge available, many alternative explanations can be ruled-out before abductive assembly begins. This creates essentials among the potential explainers.

It should be noted that essentials are an idealization. We can always come up with implausible alternative explanations such as “he really intended to miss her with the bullet from his gun but his finger slipped” or “the malfunction at the nuclear power
plant was caused by a sudden evaporation of the water in the containment vessel” or “the car problems have occurred because your engine was eaten by snakes” or even “space aliens did it”. However, in making up alternative explainers, we have to weigh the implausibility of these new hypotheses against the fact that an exhaustive search only found one plausible hypothesis. Therefore, if we thoroughly search for explainers and only find one, we can accept this explainer as essential without resorting to making up potential explainers.

In this example, $H_1$ explains $D_1$ and $D_2$. $H_2$ explains $D_1$, $D_2$ and $D_3$. If $H_1$ and $H_2$ are the only plausible explainers of the data then $H_2$ is essential as it is the only means of explaining $D_3$. $H_2$ will also be used to explain $D_1$ and $D_2$ as it is the best overall explanation.

Figure 13: Essentials

In figure 13, we see an example of an essential hypothesis. In this example there are two plausible hypotheses to explain the three data items. However, only one hypothesis, $H_2$, can explain $D_3$. Therefore $H_2$ is an essential as no other plausible
hypotheses exist (or have been found) to explain D3.

We can use essentials as leverage for further problem solving as described below.

### 3.3.2 Hypothesis Interactions

In attempting to explain data, we can make use of interacting hypotheses. Interactions among hypotheses can take on many forms from logical implications to hypothesis refinements.

Implications state that, if we know A implies B (logically or causally), and A is true, then B is true. We can use logical or causal implications where appropriate. If we have this kind of knowledge and A has been an accepted hypothesis, then we can also accept B with no doubt. We can think of implications as being truth preserving inferences.

We may have a case where A suggests B (a weak implication or suggestion rather than strong implication). This allows our confidence in B to be raised. Or, we can reconsider B and seek more evidence. If B is clearly implausible, then we may begin to doubt A. We can consider these weak implications to be expectations where we expect other hypotheses to also be true. Expectations are not truth preserving, but can aid in further problem solving.

Hypotheses may be incompatible (whether logically, such as “A” and “not A”, or causally, such as disease A cannot co-occur with disease B). If we accept one hypothesis as part of the composite, then all of the incompatible hypotheses can be ruled out of consideration. In this way, we can make a conclusion and rule out other hypotheses. This may lead to more essentials (as the alternatives were ruled-
out). Incompatibilities do occur and we will show their uses in the ARTREC system described in the next chapter. Incompatibilities can also be seen in the systems Peyer, 3-Level Machine and Creation-Evolution Theory Decider (see section 3.4.2 for an introduction to these systems, see [Josephson and Fox, 1991, Fox et al., 1991, Fox et al., 1992b] for details).

Other examples of hypothesis-interactions include refinements and generalizations. A more detailed hypothesis is a refinement of a more general hypothesis (for instance, the difference between saying that a patient has hepatitis versus saying that the patient has liver disease in general). There is a decision that can be made in accepting a hypothesis over a refined or general version which trades off explanatory power for confidence. Typically, the more refined a hypothesis is, the more detail (or findings) it can account for. The more general the hypothesis is, the stronger our confidence will be. The reason for a greater confidence in a more general hypothesis is that it is easier to recognize general concepts but when we attempt to refine them, we must seek out more detailed features. If we are unfamiliar with these features, or there is ambiguity, it becomes more difficult to differentiate. The need for specific hypotheses for the purpose of fuller coverage must be weighed against the need for confidence in the hypothesis.

Part of our strategy's strength is in weighing hypothesis-interactions against what is already known. Given that a hypothesis is accepted as part of the composite explanation, then this hypothesis can be used as leverage for further problem solving. Accepting A may cause B to be ruled out (because of incompatibilities), C to be
accepted (because of implications) and D to be reexamined in a better light (because of expectations). Based on ruling out B, accepting C and reexamining D, further purchase might be found (such as ruling out E for being incompatible with C and accepting F for being an essential since the only alternative explanation was B). And so forth.

As shown in the [Fox et al., 1991, Josephson and Fox, 1991, Fox et al., 1992b], hypothesis-interactions can solve a large portion of an abductive problem when faced with either lack of distinguishing confidence values between hypotheses, or when faced with ambiguous data.

3.3.3 Hard Decisions

A rather simple idea that has been discovered deals with hard decisions. A hard decision exists when two or more hypotheses are attempting to explain the same datum, but the hypotheses are (approximately) equal in plausibility. In such a case, the local abduction might have to pick one of the hypotheses randomly or by using some ad hoc rule. In making such a pick, this could diminish the overall confidence in the composite because there is no clear reason why one hypothesis was chosen over another. In such a case, the best strategy for the system is to not come to a local abductive conclusion. Instead, the system should delay its decision. Other problem solving may clear up the problem.

Figure 14 illustrates an example of a hard decision. In this example, there are four alternative explainers for D1. H1, H2 and H3 are all weighted equally (very likely) and H4 is almost as plausible (likely). However, if we just choose H1 over H2,
H3 and H4, we have doubted three plausible hypotheses. Even if we could doubt H2 and H3, our choice of H1 over H4 is not very convincing. However, by using hypothesis-interactions, we may find that the decision is made easily for us. Assume we have already accepted H5 to explain some other datum, and H5 implies H1, or H5 is incompatible with H2, H3 and H4. Our decision becomes simple. Or, imagine that H1 is, itself, an essential as the only way to explain some other datum, D2, then the decision is made for us.

The above is a hard decision, there are four plausible explanations for D1. However, there is no obvious choice. Based on plausibilities of hypotheses, H1, H2 and H3 are equally good. If other means (explanatory power or expectations) cannot determine which explanation is better, no choice should be made. Even the chose between H1 and H4 is a hard decision because there is only a small degree of difference in plausibility between the best and the next best choices.

Figure 14: Example of a Hard Decision

By allowing opportunistic problem solving to occur, we do not have to lessen the confidence in the composite by making hard decisions. If all other problem solving
has concluded and the datum is left unexplained, we can resort to guessing or randomly choosing a hypothesis. However, even in these cases, we may wish to avoid lessening the confidence in the composite and leave the datum unexplained. In this case, the choice is pragmatic, one of weighing the consequences of leaving the finding unexplained versus the consequences of guessing incorrectly.

This notion of hard decisions is similar to a least commitment strategy. In this case, an abductive agent will use pragmatics to decide whether or not to commit to a conclusion. Hopefully by not committing, other problem solving can take care of the hard decision. It should be noted that this idea of hard decisions is ignored in many other abductive strategies, especially those which are based on probabilities. In the case of Reggia and Peng, and Pearl, hard decisions are immaterial. A conclusion must be made, that is, all of the data must be explained, no matter how little confidence there is in the conclusion.

3.3.4 Explicit Noise Hypotheses

In many problem solving activities requiring abduction, especially in perceptual tasks, findings may contain some degree of noise. It is difficult to determine if a finding actually needs to be accounted for or whether it is simply noise. By using an explicit explanatory strategy, we are able to create a hypothesis that “this finding is noise” and use this hypothesis as an alternative explanation. We may wish to do this if a datum is questionable. We can explain this datum either in terms of a domain hypothesis or as “noise”.

To use noise hypotheses, one must know what “noise” looks like (in that domain)
in order to state this with any confidence. In using "noise" hypotheses, potential "essentials" can be dismissed because they are not legitimate. That is, if there is a mildly plausible hypothesis that the datum was caused by hypothesis A, but a much greater chance of the datum being noise, then noise is the best explanation. Rather than committing to hypothesis A, which is untrue, we accept the better explanation, noise.

Noise hypotheses are used extensively in ARTREC and will be explained in detail in the next chapter. It should be noted that ARTREC achieved much better accuracy once noise hypotheses were introduced to the system. This leads us to believe that knowledge of "noise" is very important for perceptual reasoning systems where noise is inherent.

3.3.5 Lack of Precise Plausibilities

In the description of our abductive composition strategy (discussed in the next section), we will point out that the best explanation of a local abduction is an explanation which surpasses all other alternative explainers by some large degree. If one hypothesis just barely surpasses another hypothesis as an explainer, we will not have high confidence in our conclusion. Because we are interested in making decisions primarily when we have a large gap in plausibilities between the best explanation and the alternatives, we will not need to describe the plausibilities in terms of precise statements. It is unnecessary to demand precise plausibilities (such as probabilities computed from Bayesian equations, computations of neural activations or the transition probabilities of statistical models).
In abductive systems using Bayesian probabilities, prior and conditional probabilities must be precise. A conclusion is the most probable hypothesis no matter how close the “next best” explanation is.

In other forms of problem solving where search is an issue, the means of choosing a path through the search space is based on some form of “cost function” or “goodness function”. The next choice is always the choice which yields the least cost or the most “goodness”. In such situations, there is a strong need for obtaining precise and accurate plausibilities for choosing the next step through the search space. Whether the search space is implemented as probabilities (using HMMs or Bayesian equations) or some optimality function (as in Samuel’s checker program [Samuel, 1963]), the choice of the next step is made based on a computed best score.

However, it is not always true that the best score (or highest plausibility) is the correct choice. The best choice might actually be misleading. This can occur in cases of ambiguities and noise (in the data). A system which strongly depends on the “best score” might err in situations as these. The best choice might lead the searcher down a garden path.

It is also possible that the best choice is not significantly better than other choices. In accepting a “best” choice which is not significantly better than alternatives, we weaken the overall confidence we can place in our conclusion.

We disagree with the assumption that the best choice will always be the highest rated hypothesis. If we have two (or more) hypotheses that are fairly equal in plau-

\[2^{nd} \text{In Pearl's work, he claims that exact probabilities are not necessary. His systems can achieve the correct behavior with inaccurate probabilities. However, a precise value (i.e. a real number between 0 and 1), no matter if it is correct or not, must be given for use in the Bayesian equations.}\]
sibility, it is best to come to no conclusion at all (this is a hard decision as described previously).

Therefore, coarse statements of plausibility are acceptable. We have described in [Tanner et al., 1991] a set of plausibilities which seem perfectly suitable for the task of abduction. There is no need for precise plausibilities such as probabilities used in Bayesian formulas or the statistical HMMs.

One advantage to the abductive strategy we have derived is that, when other knowledge is available (such as explanatory coverage and hypothesis-interactions) then precise plausibility values are not necessary. Our strategy for abduction considers hypothesis-interactions along with explanatory coverage. There is less need to obtain plausibilities. In fact, as shown in [Josephson and Fox, 1991, Fox et al., 1991, Fox et al., 1992b], this strategy can get by with all of the hypotheses having equal plausibilities, set in an ad hoc manner, as long as additional knowledge is available. Some of the experiments we have run clearly show that, as long as all of the hypotheses are somewhat plausible to begin with, the combination of explanatory knowledge and hypothesis-interaction knowledge is often sufficient to come to a quick and clear decision.

This notion differs entirely from other strategies for abduction where decisions are made solely based on whether one hypothesis surpasses the plausibility of another hypothesis.
3.3.6 Combined and Additive Coverage

Another benefit of explicit explanatory coverage occurs in the fact that coverage is often combinable, and sometimes additive. Combinable coverage means that two or more hypotheses might, in conjunction, explain some datum. The hypotheses separately might be insufficient to explain the datum, or to explain the entire datum. In the Red system, hypotheses could add together their explanations in order to explain a severe reaction (a 4+ reaction). So, if hypothesis A can explain a test reaction of 2, and hypothesis B can explain the same test reaction of 3, then A and B could adequately explain the 4+ reaction. Additive coverage is similar, however, the hypotheses can be added together to explain more than what either could separately explain. Without the explicit coverage, an agent may accept one hypothesis without considering the second. However, since the task is to explain a datum, it might take more than one hypothesis to adequately explain some data.

As an example of additive coverage, consider legal reasoning. We may attempt to determine motive and opportunity for the defendant. Separately, these two hypotheses (that the defendant had the motive and the defendant had the opportunity) can be used to explain things like “defendant threatened the person” and “defendant was seen in the area that night”. However, together, the hypotheses that the defendant had motive and opportunity is much more convincing to a jury than either alone. Combining both hypotheses can be used to not only explain some findings, but explain additional findings such as the murder itself.
3.3.7 Confidence in our Conclusion

Our abductive strategy creates islands of certainty. It uses hypothesis interactions to further the explanatory coverage while delaying hard decisions. This strategy allows us to have a high overall confidence in a conclusion or it gives us reasons to doubt a conclusion.

In [Josephson and Goel, 1991], Josephson and Goel show that our abductive strategy will generate the most plausible explanation of any explanation which can explain an equal or lesser amount of data. That is, if H is the output of a problem solver based on this strategy, and H explains D', a subset of D, then H will be the best possible explanation for D'. It is very important to have confidence in our abductive conclusions. Our abductive strategy can generate highly confident explanations or give reasons why we should not have confidence in an explanation.

One nice side effect of this strategy is that any system using this strategy correctly will have a high degree of correctness when it commits to a decision. As shown in [Tanner et al., 1991, Fox et al., 1992a], the strategy promotes the theme "high confidence reflects correctness", which seems like a worthwhile ability for any knowledge-based system. It should be noted that, because abductions are not truth preserving, our strategy is not infallible, but neither are humans infallible during abductive problem solving.
3.3.8 Explicit versus Implicit Abduction

As can be seen from this section, by taking an explicit explanatory approach, one is able to make use of several ideas related to explanation that might be otherwise unavailable. If one is not seeking to explain datum d, then one might not realize that hypothesis h is essential for coming to a complete conclusion. In searching for a solution to a problem, one might be willing to accept any set of hypotheses as long as the hypotheses are consistent and plausible. However, in seeking a best explanation, we will not be satisfied unless the best explanation is clearly better. This means that we will not try to make a hard decision if it leads to a questionable explanation.

Even by appealing to abduction, one might not make use of such strategies as delaying hard decisions, looking for essentials, leveraging hypothesis interactions and so forth. However, it is my feeling that using all of these strategies will make abductive problem solving much more efficient and even tractable. The following section will describe how these notions can be incorporated into a single strategy. This will be followed by some examples of using this strategy and a discussion of the efficiency of the strategy.

3.4 Peirce - a Domain Independent Abductive Assembly Tool

Based on the initial strategy for Red, several new strategies were designed. Eventually this strategy was incorporated into a problem solving shell called Peirce (named after philosopher C. S. Peirce, who first used the term "abduction"). We have embodied the above ideas into a clever strategy which, among other things, allows for more
tractable abductions and the ability for an abductive system to come to quick and confident conclusions. Our strategy is opportunistic and includes a number of ideas represented in human abductive problem solving (such as weighing the need to come to a conclusion against the confidence in making a decision at this point).

Peirce is used to generate a domain dependent problem solver called an abducer (we will use abducer as the general term for an abductive problem solver created from Peirce). Figure 15 shows the abstract task description of an abducer. The abducer's task is to generate a best explanation given a set of findings to explain and a set of plausible hypotheses to explain those findings. The abducer solves the subtask of abduction called Hypothesis Composition, also called Abductive Assembly.

The Peirce algorithm presupposes that there is a means of generating or obtaining the findings for the case. Peirce also presupposes a mechanism for hypothesis invocation (generating hypotheses) and hypothesis instantiation (scoring hypotheses and determining each hypothesis' explanatory coverage and the hypothesis-interactions that might occur). Peirce attempts to handle the hypothesis composition phase. As noted earlier, there are tools which accomplish the hypothesis invocation and part of the hypothesis instantiation subtasks. These tools are not discussed here.

3.4.1 Peirce Strategy

The steps of the algorithm are:

- Generate hypotheses (with their confidence scores) and generate or obtain findings
Best explanation - best (most plausible) subset of hypotheses (with no superfluous parts) to explain as many findings as is confidently possible. Along with the explanation is a description of each hypothesis' confidence and acceptance level (essential, clear best, weak best, etc...).

Abducer (or abductive assembler)

Plausible Hypotheses (with confidences and explanatory coverage)

Findings to be accounted for

Figure 15: Peirce Abducer Task Description

- Start the composite with any hypotheses which have been predetermined to be in the composite (this can be set up at run time when it is discovered that certain hypotheses should be included for this case, or set up by the tool user when it is determined that certain hypotheses should always be included or by the user interactively while he/she is exploring alternative hypotheses).
• Expand expectations from a higher level in the abductive process (this means that some higher level knowledge has implied, either positively or negatively, some hypotheses). The expectations will cause the hypotheses in question to be rescored by taking their current score and adapting it to reflect the positive or negative interactions.

• Propagate the effects of hypotheses being accepted into the composite. This may rule out other hypotheses incompatible to those in the composite or it may alter scores of other hypotheses which may be implied or causally connected to hypotheses in the composite.

• Loop on the following until either all findings are accounted for, or until no more progress is made in the explanatory coverage

  – Find all confirmed hypotheses and include them in the composite. A confirmed hypothesis is one which receives the highest confidence score possible and is assumed to be present in the case (this is an optional feature which can be turned off if confirmed hypotheses should not be automatically included in the composite).

  – If confirmed hypotheses are found then propagate the effects of the latest additions and go back to loop beginning, else continue.

  – Find all essential hypotheses and add them to the composite. An essential hypothesis is one that is the only plausible hypothesis to explain a finding.
- If essential hypotheses are found, then propagate the effects of their inclusion into the composite and go back to loop beginning, else continue.

- Find all clear best hypotheses. In this case, a clear best hypothesis is one which explains some finding better than any other hypothesis. To be a clear best, the hypothesis must have a score higher than a given threshold and must surpass all other hypotheses by a given distance (thresholds are given by the tool user at the time the system is built, however, they can be easily modified during or between cases. There are defaults if no thresholds are specified).

- If clear best hypotheses are found, then propagate the effects of their inclusion into the composite and go back to loop beginning, else continue.

- Find all of the weak bests hypotheses. Here, we may relax the conditions on the clear bests criteria mentioned above. This step is optional.

- If weak best hypotheses are found, then propagate the effects of their inclusion into the composite and go back to loop beginning, else continue.

- End loop

- If there are still some unaccounted findings, attempt to guess from the remaining hypotheses which have not been ruled out. This step is optional and depends upon the need to explain the remaining data over the risk of guess-
ing incorrectly. Guessing is accomplished by letting the unexplained findings vote on hypotheses which they are most likely to be explained by. This allows hypotheses which explain more to have better chances of being chosen than hypotheses which explain less, other things being equal.

- If any guessed hypotheses are found, then propagate the effects of their inclusion into the composite and go back to the loop beginning, else end. This step is of course dependent on whether guessing took place or not.

At this point, either all findings should be accounted for, or there are no more hypotheses available to explain findings, or the only remaining hypotheses are too close in plausibility and explanatory coverage to allow for a better explanation. In the last case, the unexplained findings are left unexplained because they cannot be explained with confidence due to the close plausibilities of alternative explainers. In such a case as this, guessing can take place. Guessing is a last resort attempt to explain some unexplained findings but a complete coverage can be achieved as long as there exist hypotheses to explain the remaining findings. However, guessing will definitely weaken the overall confidence that we could have in the conclusion.

It should be noted that each pass through the loop part of the algorithm is considered relative to the previous passes. This means, for example, that a hypothesis which is considered essential because a competing hypothesis is ruled out due to it being incompatible to a clear best is only an essential hypothesis relative to clear bests. Thus, an essential from the first pass through the loop is more confidently essential than an essential which is relative to a clear best. Similarly, any newly
included hypothesis which is relative to some guessing (that is, a hypothesis is included due to the effects of the inclusion of a guessed hypothesis) must be regarded as less confident than any hypotheses included before guessing took place. So, each pass through the loop in the algorithm further limits the system's confidence in any new hypothesis included into the composite as it is included relative to whatever previous propagations have occurred. Hypotheses may be confirmed, essential, clear best, weak best, disbelieved (due to incompatibility), guessed, or ruled-out because of a low confidence rating, and these statuses may be relative to confirmed, essential, clear best, weak best, disbelieved or guessed hypotheses.

We can control the above strategy and tell the abducer how confident our conclusion should be. For highly confident conclusions, we would limit the search for explainers to confirmed hypotheses, essential hypotheses and clear best hypotheses. For high coverage, we would also allow weak bests. If we are willing to tolerate more error in our conclusion, then we could also allow guessing.

The output of a Peirce abducer will list the accepted hypotheses along with their status of acceptance (i.e. if they were confirmed, essential, clear best, weak best or guessed) and their relative status (essential relative to clear best, or clear best relative to guessed). This allows us to determine how much confidence we can place in each part of the conclusion.

3.4.2 Systems Built from Peirce

Here, we show that Peirce is truly a domain-independent, problem independent, problem solving tool. We have tested Peirce out with a variety of experiments. These
experiments point out several factors in dealing with Peirce. I will first describe the various knowledge-based systems constructed from the Peirce tool. I will then discuss some important conclusions we are able to make about Peirce and the general strategy for abduction already discussed.

Speech Recognition: CV and 3-Level Machine

The initial experiments were set up as part of a pilot speech recognition project. After receiving funding for a pilot project, we constructed two small prototype speech recognition systems.

CV is a discrete speech recognition system recognizing monosyllables. It attempts to recognize one of 30 possible monosyllables, all of which are constructed from one consonant of /b/, /d/, /g/, /p/, /t/, /k/ (the six stop consonants) and one vowel of /a/, /æ/, /I/, /ow/, /er/. CV uses a feature-based recognition scheme attempting to classify the consonant and vowel portion of each utterance.

CV worked by first segmenting the speech signal into time slices, and attempted to classify each time slice as a burst, sonorance, transition, or silent region. It then grouped these regions together and attempted to derive auditory features (such as burst type, formant locations, amount of voicing, and so forth).

An abductive formant tracker was used to find the formants F1, F2 and F3 in the transition and sonorant portions of the utterance. The formant tracker traced along the spectral peaks of each time segment of the utterance and attempted to explain each peak as one of the formants or as noise. An abductive smoothing algorithm

---

3CV stands for Consonant-Vowel pairs
attempted to make the output more consistent. The tracked formants were then used to determine the vowel.

After finding F1, F2, F3, and auditory regions of interest, CV used a classification hierarchy to generate potential consonant and vowel hypotheses. Then, abduction was used to best explain the data in terms of a single consonant-vowel pair.

The problems with CV included using a single feature for place of articulation which was not very robust causing some errors in the consonant recognition. Also the formant tracker had problems with determining F3’s location in some cases. Overall results were in the upper 80% for correct CV identification.

Several enhancements and improvements to the system were suggested although never implemented. These improvements included a more robust determination of place of articulation by looking for “velar pinch” by backtracking along the formants, and an improved formant tracking algorithm to handle problematical F3 tracks.

The second speech system was a first attempt at layered abduction. Similar to the ARTREC system, this system would perform articulatory recognition. Given articulatory gestures, the 3-Level Machine would explain these in terms of phonemes, and then explain the phonemes in terms of words. This system used 2 abducers of three levels of knowledge. The processing flow for 3-Level Machine is shown in figure 16.

We again used the same set of consonant and vowels, but expanded the system to include VC, CVC, VCV, and other forms of words⁴. The articulatory gestures given as input were hand-coded. Because of this, 3-Level Machine would always

⁴VC stands for Vowel-Consonant, CVC for Consonant-Vowel-Consonant, and so on.
give a correct answer when the input was unambiguous. In order to determine the usefulness of hypothesis incompatibilities and expectations, we dirtied up the input. The result was a poorer performance without the hypothesis-interaction knowledge. However, the system was able to deal with the ambiguities in the dirtied input if either expectation or incompatibility knowledge was available.
Another experiment with the 3-Level Machine was to determine what type of top-down processing was possible. Because of the limited scope of the system, we implemented only a single form of top-down processing, one which uses plausible hypotheses at the highest level to help disambiguate close decisions at the middle level. Thus, a form of top-down expectations was implemented.

Results from the 3-Level Machine clearly indicate that multilayered abduction is possible, that abductors can interact in a positive way to promote the overall problem solution. We used a simple bottom-up control method, and when appropriate, to continue the problem solving with a top-down pass followed by another bottom-up pass. However, this simple control method is insufficient for larger problems and a more opportunistic control could solve portions of the overall problem at each level until dependencies forced some serial order. The CV and 3-Level Machine are documented in [Fox and Josephson, 1991, Fox et al., 1991].

Legal Reasoner: Peyer

We implemented a legal reasoning system. This system would explain legal evidence by hypotheses relating to the defendant's guilt or innocence. The particular case we implemented was originally implemented by Paul Thagard using his Echo system. This case pertains to a California police officer, Craig Peyer, who was charged with murdering a woman he had pulled over for a traffic ticket. The real trial ended with a hung jury. Thagard took the evidence presented during the trial and implemented it as a series of evidence and hypothesis nodes in a neural network. We took his
description and encoded it into Peirce with little difficulty. Thagard had included a
description of hypothesis explanatory coverage and hypothesis incompatibilities. This
description was easily encoded into a Peirce abducer.

One particular problem we faced was that Thagard [Thagard, 1989] did not offer to
rate hypotheses. Since Peirce depends on hypotheses having plausibilities, we decided
to opt for a very straightforward solution. Rather than assigning plausibilities in an ad
hoc manner or attempting to encode knowledge which would let the system generate
hypothesis plausibilities based on legal knowledge, we set all hypotheses equal to a
neutral value. We then let the system attempt to work based on explanatory coverage
and hypothesis-interactions.

To our satisfaction, the system came to a quick and clear answer. Peyer was
innocent based on explanatory coverage and hypothesis incompatibilities. We decided
to alter the knowledge slightly to be more realistic (based on a couple of flaws in
Thagard’s breakdown of the problem5) and ran the new set of knowledge. The result
was much the same, a clear and quick, although somewhat less confident, conclusion
that Peyer was innocent.

As an interesting side note, Thagard’s implementation in Echo’s decision of Peyer’s
status favored a guilty verdict, although its decision was not as clear cut as the decision
that our system produced. Our implementation of the Peyer case is fully documented
in [Josephson and Fox, 1991].

5These flaws pertained to using facts as pieces of evidence to be explained. The result of this
was the hypothesis “Peyer is innocent” was required as an essential. We changed these findings
to be facts not requiring explanation so that the innocence hypothesis was not accepted without
additional knowledge.
Theory Evaluator: Evolution vs Creationism

Having succeeded in the implementation of the Peyer case, we chose to implement another problem that Thagard had implemented (again from [Thagard, 1989]). This time, we used Thagard’s breakdown of the debate between Darwinian Evolution and Creationism. However, we supplemented Thagard’s own breakdown by consulting some other texts in order to update the argument to the current, hot debate between evolution and creationism [Berra, 1990, Harrold and Eve, 1987]. We implemented this problem as a layered abduction system which attempted to determine which theory was better at explaining the various findings of life on earth.

This layered abduction system attempted to explain the findings in terms of low level hypotheses (such as extinction, natural selection, artificial selection, speciation). After accepting a subset of the given low level hypotheses, these hypotheses were used as findings and one of the two theories, evolution and creationism, was used to explain them.

Similar to the Peyer system, we had no means of adequately scoring the hypotheses. Therefore, all of the low level hypotheses and the theories were instantiated with an initially neutral value. However, we allowed the system to weigh evidence in favor or against the hypotheses. When a fact was considered to favor a hypothesis, the hypothesis’ plausibility was raised. When a fact was considered to be opposed to a hypothesis, the hypothesis’ plausibility was lowered. This allows the system to come to quick and clear decisions primarily based on explanatory coverage and hypothesis-interactions with guidance from the available evidence.
We implemented four variations of this system, each with a different version of creationism. In two of the four cases, evolution was a clear winner based on explanatory coverage, hypothesis-interactions and the facts at hand. In two other cases, no theory was produced as an explanation of the findings. In one of these cases, we used the Bible as a piece of evidence to support creationism. This made creationism the theory of clear choice (based on plausibility) but it could not account for all of the findings and so evolution and creationism were both found as partially correct theories without a means of breaking the tie with any confidence. In the other case, we added a higher level theory that God started the process of evolution. This new theory (that God started the process of evolution) and the theory of evolution (without God's involvement) were equally plausible and equal in explanatory coverage, so the system did not come to a conclusion. The Evolution-Creationism theory decision making system is fully documented in [Fox et al., 1992b].

Experimental Conclusions

We were able to form several conclusions about Peirce.

- Peirce is highly useful for solving abductive problems in many very different domains.

- Peirce does not need accurate hypothesis plausibilities if other types of information are available. These types of information consist of hypothesis interactions

---

6Some of the findings that Creationism could not explain were transitional forms in the fossil records, transitional forms in nature, species' extinction, domestication to remove traits, breeding for desired traits.
and hypothesis rule-out knowledge.

- Peirce-constructed systems are able to handle noisy data by leveraging other knowledge and by explicitly using "noise" hypotheses.

- Systems built from Peirce are able to come to quick decisions. When faced with ambiguities or hard decisions, these systems will defer attempts to provide an explanation for those ambiguous or hard findings. When the system is sure it is correct, there is a high accuracy rate.

- A Peirce conclusion is made up of hypotheses accepted at some level of confidence. This level can be set by the user so that the user can specify the amount of coverage or the degree of overall confidence in the conclusion. If higher coverage is desired, the confidence may be lower whereas a higher confidence in the conclusion is possible by sacrificing some amount of coverage. This degree of coverage versus confidence is a parameter which can be set very easily, either at system design time or at system run time.

3.5 Efficiency and Plausibility of Peirce

I have introduced a problem solving method for abduction. However, why is this method any better than past abductive methods? Is it plausible to suggest that such a method be used for such a large variety of problems as theory formation, diagnosis, language understanding and perception? I will appeal to the efficiency of Peirce in order to demonstrate the practicality of this abductive method no matter what the problem. I will then discuss the psychological plausibility of such an abductive
algorithm and conclude this section by discussing the engineering applications for this algorithm.

3.5.1 Efficiency of Peirce

To defend the promotion of the Peirce algorithm as a means for accomplishing abduction, I will now devote some time to discuss the efficiency of the algorithm. In doing so, I should reiterate that the Peirce algorithm is only a method for hypothesis composition.

The tasks of evocation and instantiation must be performed separately (however, as pointed out elsewhere, hierarchical classification and hypothesis matching are efficient subtasks for hypothesis evocation and instantiation). The Peirce algorithm presupposes that certain knowledge is already available. This knowledge is a list of plausible hypotheses where each hypothesis has additional knowledge including a rough statement of plausibility, what the hypothesis can explain, and if possible, how the hypothesis interacts with other domain hypotheses.

The Peirce algorithm will begin to construct a composite hypothesis based on islands of certainty. The most certain hypotheses constitute those which are confirmed or essential. The search for these hypotheses is very simple and straightforward. To find essentials, the algorithm looks at each finding, and in turn determines if there is a single plausible explanation. To find confirmeds, the algorithm looks at the hypothesis' plausibilities to determine if that hypothesis fits the "most highly rated" category. These two steps require a linear amount of time (based on the number of findings to be explained).
If hypotheses are accepted, the propagation of acceptance occurs. Each hypothesis might have expectations, implications and incompatibilities with other hypotheses. The propagation is simply a matter of altering the remaining hypothesis pool by removing some hypotheses, altering the plausibility of some hypotheses, and accepting some hypotheses. The propagation also requires only a linear amount of time as each hypothesis will have some constant number of adjustments to make. Further, only accepted hypotheses will have their expectation, implications and incompatibilities propagated and this occurs only when the hypotheses are first accepted.

Since the propagation may cause sufficient change to the status of the remaining available hypotheses, the search for confirmeds and essentials begins anew. However, again, to search for these hypotheses takes a linear amount of time. If no new hypotheses are added, then the search for less certain but still highly plausible hypotheses commences by searching for clear best hypotheses. This step requires considering each finding, and making a decision between the alternative explainers for that finding. Again, this step requires a linear amount of time based on the number of findings to be explained. At the last step, weak bests are considered. Much like clear bests, each finding is considered and if a hypothesis seems better than other explainers, it is chosen. This step also requires a linear amount of time.

The number of cycles through this loop depends on the domain and the findings for the particular case. In the worst case, this cycle will continue, one time for each finding, bounding this main loop by a linear amount (based on the number of findings for the case). Therefore, the worst case for the Peirce algorithm is $O(n^2)$. 
However, the complexity does not indicate the efficiency of this algorithm. The Peirce strategy is highly efficient because of three strategies incorporated into it.

First, since Peirce's strategy is one which seeks local abductions and combines each best explanation into a composite explanation, there is no need to generate all possible combinations of explanations. In fact, there is no need to generate more than a single composite explanation. The propagation of effects takes care of concerns about how hypotheses interact (including the removal of any hypotheses incompatible with those that have been accepted). Therefore, a Peirce abducer will generate only one composite explanation, taking care of hypothesis interactions at each step.

Second, since Peirce's algorithm will avoid hard decisions, the algorithm will explain as much as possible by appealing to only clearly confident conclusions. The result of this is a quick composite-building mechanism which will not get slowed down when faced with dilemmas. There is no need to become stuck in a conundrum while forming the composite; any problems that arise from abduction are skipped (avoided) by leaving these problems and taking on other problems instead. If the problems (decisions) persist by the time the abducer has explained everything else, then the abducer may choose to reinvestigate these problems, or else it will leave them as unresolved. This strategy satisfies the need for the most plausible composite at the price of not coming to a complete explanation (that is, the strategy trades off coverage for a more certain answer).

Third, Peirce's strategy is one which uses a form of Least Commitment [Friedland, 1979, Sacerdoti, 1974] in that Peirce only accepts a hypothesis if that hypothesis is the
most reasonable hypothesis to accept at that point in the problem solving. This is due to the use of islands of certainty in constructing an explanation. The result of this is that a Peirce abducer will never under any circumstances attempt to backtrack. One advantage of a least commitment strategy is that backtracking will (hopefully) not be required. This is ensured in Peirce as a Peirce abducer will consider the already formed (partial) composite to be correct, and use the knowledge of the composite (e.g. hypothesis interaction knowledge) in order to obtain further purchase on more of a solution. That is, the certainty is used to find more certainty.

In addition to these efficiency strategies, we must consider another point. If knowledge is highly compiled so that hypothesis evocation and instantiation delivers forth a single hypothesis to explain some datum, then the choice for the explanation is already made. The hypothesis is essential and is included in the composite almost immediately with no deliberation. If such knowledge is available, a degenerate form of abduction occurs. A composite is still constructed, but many of the choices for local abductions are made simply. This improves the run-time efficiency of an abducer to the point of making the problem similar to one of table look-up or partial pattern matching.

It should be pointed out that Peirce can only work as well as the knowledge which is available. If there is a lack of hypotheses to work with, a Peirce abducer will be handicapped. If many plausible hypotheses are suggested (by hypothesis evocation) and they are all equally likely, then Peirce can only use explanatory knowledge and hypothesis interaction knowledge to differentiate between plausible hypotheses. If

7For a more complete discussion of this, see chapter 7.6.
this knowledge is not available, Peirce will have a very difficult time. Therefore, while Peirce does not dictate how hypothesis evocation and instantiation are performed, it is crucial that a reliable means is used for these two tasks.

As a side note, Peirce can also be implemented in parallel. I will describe later how Peirce can be used in parallel for solving layered abduction problems. However, here I will just mention that the local abductions made by Peirce can be parallelized without difficulty. If each node of a parallel processing machine contains an abducer, then the findings to the case can be divided so that one finding is given to each processor. Each processor has the simple (and linear) task of finding the local best explanation for that particular finding. The first step will have each processor seek only confirmed and essential hypotheses. If a processor finds an explanation, then it communicates this hypothesis with some central processor whose task is to propagate the effects of accepting these highly plausible hypotheses. The effects of propagation are then sent to each of the processors which have not yet explained their finding. These processors again will attempt to find a best explanation and if they fail, they can weaken the acceptance criteria (i.e. look for clear bests rather than essentials). Thus, the decision making is distributed and each processor has the task of coming to a single conclusion, leaving the hypothesis interactions to be decided by a single processor. This should speed up the processing by an order of magnitude.

3.5.2 Is Peirce a Plausible Mechanism for Abduction?

Several questions arise with the discussion of the Peirce algorithm. How realistic is it? Do humans really solve abductive problems by such a mechanism? How should
this strategy be changed for problems which make use of highly compiled knowledge (such as perceptual problems)?

Many of the ideas behind Peirce come from analyzing human problem solving. Hard decisions are typically delayed in interpretation problems. Essentials and islands of certainty are used to find purchase in difficult problems. Further, the ability to adjust a solution based on the need for coverage versus the need for accuracy is clearly exhibited in problems such as diagnosis. Because of these factors, Peirce seems perfectly reasonable as a solution.

However, I am not trying to say that we all have little Peirce algorithms running in our heads as we solve problems, but rather that the types of strategies used by Peirce seem similar to the ways that we solve abductive problems.

In perception, however, an algorithm like Peirce seems too deliberative. Is there sufficient time for a Peirce-like strategy, one which requires us to propose hypotheses, score hypotheses and then compose a single composite explanation to account for the findings we face?

The answer to this is very controversial. It has been posited that perception is inferential by many, and argued by others that perception cannot allow for any form of deliberation. If the answer is “no, there is no time for inference” then obviously a Peirce-like algorithm is not appropriate. Yet, the answer to this question remains unknown. It seems plausible to suggest that perceptual tasks require some inference, which brings about a second question: is the algorithm for Peirce suitable for perception? The thesis of this document is that perceptual problems can be solved
by using layered abduction, and the Peirce strategy in particular, as a method. As will be shown in the next chapter, we can solve portions of the speech recognition problem by using layered abduction. It remains to be seen whether the entire speech recognition problem can be solved in such a way, but the results discussed here are encouraging.

3.6 Layered Abduction

If the task of abduction is to infer a hypothesis (composite or simple) to explain findings, then layered abduction is a cascaded inference process in which a hypothesis at one level is turned into a finding to be accounted for at the next higher level. If the hypothesis at the higher level is lacking in or is of inappropriate detail, then it can be changed into a finding to be accounted for at an even higher level. This procedure can continue until a reasonable hypothesis has been found. Layered abduction has the ability to generate hypotheses of one type of knowledge and use them to come to a more abstract conclusion, or a conclusion using a different form of knowledge. This is highly useful in many types of reasoning because the world is a complicated place in which there is a wide range of knowledge types, and abstractions are found everywhere.

For example, in speech recognition, we can come to a conclusion about the speech signal. It is composed of bursts of sound, sonorant regions, noisy regions, and so forth. However, this conclusion is insufficient. It is at an entirely inappropriate level of detail. We could also conclude phonetic features such as velar pinch, voicing and formant frequency. This too is inappropriate to recognize speech. We could even
make a conclusion of the phonetic units involved and offer a string of such units. However, without word boundaries, this too would be an inappropriate conclusion. If instead, we could infer lexical items from this data (i.e. words), we would feel better about accepting a conclusion.

Similarly, in diagnosis, it is not always sufficient to explain symptoms in terms of diseases or malfunctions. We would feel better if we understood causally how the symptoms and malfunctions came about. In diagnosing why a nuclear power plant had to be shut down, we don't want to simply know that there was a rise in core temperature, we want to know how this malfunction came about. Finding deeper and deeper causes, deriving causal stories, explaining hypotheses in terms of new theories or laws, all of these tasks require a fuller problem solving environment than a simple one-layered mapping. All of these problems will require multiple inferences, across, potentially, many kinds of knowledge. Our problem solvers require the ability to reason in such a way. Layered abduction is a method for solving these types of tasks.

3.6.1 Using Peirce for Layered Abduction

The strategy discussed in the Peirce section is perfectly adaptable for accomplishing layered abduction tasks. To use Peirce for layered abduction, one would create an abducer for each level in the problem. Then, one would link abducers together in an upward flowing fashion so that the best explanation generated by one abducer would become the findings to be accounted for by the next abducer. Each level would need a separate source for instantiated hypotheses. The hypothesis composition process is
taken care of by the Peirce abducer.

Peirce does not have a built-in control strategy for layered abduction. This is both fortunate and unfortunate. It is fortunate because the control strategy for layered abduction is, in part, domain specific. For instance, in perceptual domains, little top-down processing seems to occur at the lower levels. These levels are so highly compiled that most of the processing presumably occurs only in a bottom-up manner. Top-down processing seems to come into play only when ambiguities are found to exist or enough high level knowledge is available that some lower level processing can be avoided. In theory formation, hypotheses are typically tested out by the expectations of the theories. In such a case, much of the processing occurs in a top-down fashion where expectations help score lower level hypotheses.

It is unfortunate that there is no built-in control, however, as this is additional work for the system builder. But in spite of this problem, there is a great deal of flexibility in using Peirce for layered abduction as any form of control is possible, whether the control is to allow individual abducers to run in parallel, serially in a bottom-up fashion, some complex pattern of bottom-up and top-down processing or in a middle-out fashion. The flexibility in Peirce comes from the fact that each level’s abducer is independent of all other levels’ abducers. There exists dependencies as one level’s conclusion becomes the next level’s findings, however, partial solutions can be propagated upwards or downwards while other abducers work on partial solutions. Higher level abducers that come to partial conclusions can attempt to aid lower level abducers by generating expectations. Lower level abducers can pass along partial
solutions and await either expectations from above, further solutions from below, or some form of prompting for guidance or verification.

3.6.2 Flows of Processing

There are many possible forms of processing control that can be used in a layered abduction system. They are all variations of the same theme, bottom-up and top-down.

The simplest control form is strictly bottom-up. Here, findings are introduced and explained at one level. The explanation is offered as findings for the next level to explain. This explanation is offered up to the next level. This process continues until the highest level reaches an explanation.

A variation of this control introduces top-down processing. When one level (other than the lowest level) reaches a conclusion, its conclusion is sent upwards to the next level and expectations are generated to be sent to the next lower level. These expectations are based on what should be found out of the data based on the hypotheses that were concluded. The lower level which receives expectations can use these expectations to aid problem solving if some portion of the findings have yet to be explained. Or, verification can take place where the expectations can confirm whether the lower level explanation is consistent with the expectations.

Another form of processing is a middle-out method. HWIM [Wolf and Woods, 1980] used a method where islands of certainty were found initially, and this allowed some initial conclusions at a higher level (relating to the lexical level), although not at the highest levels (relating to syntax and semantics). HWIM would then use upward
and downward flowing processing to advance all levels. Communication would then be based on the completion of solutions, flowing upwards from the low levels and downwards from the high levels.

Finally, the layered abduction strategy accommodates a parallel implementation. Initial processing would occur at the lowest level until a partial solution is found. This partial solution can then be propagated upwards while processing continues at the lowest level. As other levels reach partial conclusions, their explanations are passed upwards while their expectations are passed downwards. If each level is implemented as an abducer on a separate computer processor, then this version of the control is simple to implement and communication occurs as message passing between processors. Peirce is well-suited for this form of processing. One additional feature necessary to implement a parallel layered abduction strategy is that the abducer would go into a suspended mode if findings or expectations are not yet available. An example of parallel abducers is shown in figure 17.

It should be noted that the parallel layered abduction is especially suited for speech recognition. A portion of the acoustic signal (or articulatory signal if we consider ARTREC) can be given to the lowest level abducer. This abducer will generate a partial (or complete) explanation for the portion of the acoustic signal. This solution is passed along to higher level abducers and the lowest level abducer begins to operate on the next portion of the acoustic signal. As the lowest level abducer comes to conclusions, they are passed along to other levels. The other levels will continue to work until they also achieve a conclusion which they can pass along
Abducer3 (on processor 3)

Expectations and other information/questions to help Abducer 2 come to a complete conclusion

Partial solution sent to Abducer 3 as soon as it is available from Abducer 2

Abducer2 (on processor 2)

Expectations and other information/questions to help Abducer 1 come to a complete conclusion

Partial solution sent to Abducer 2 as soon as it is available from Abducer 1

Abducer 1 (on processor 1)

Data to be explained

Figure 17: Parallel Peirce
to higher level abducers, and/or generate expectations to pass along to the lower level abducer. Very little control is necessary in this picture because all of the intermediate abducers will have the same processing task, to achieve a (partial) conclusion for any input, and if a conclusion cannot be reached, then query the lower level abducers with expectations or questions. In principle, a parallel abduction system is not only possible, but potentially very efficient.

3.6.3 Uses of Downward-Flowing Processing

Downward flowing processing comes in the form of expectations and problem solving guidance. There are many reasons to use downward flowing processing.

- Data-seeking needs which may arise if a hypothesis at one level needs to determine if data actually exists at a lower level. This need will arise if a hypothesis’ plausibility is in question. In such a case, the higher level can prompt the lower level to reexamine the data, to see if the data exists or not. This may require test ordering (in a diagnostic situation) or reexamining the visual or acoustic input (in perceptual situations) or seeking new means of gathering data (in theory formation situations).

- Expectations based on firmly established hypotheses can be passed down to lower levels to prune away irrelevant hypotheses or to aid in hard decisions.

- Uninterpretable data can be doubted and rescoring. That is, if a finding is unaccountable at one level and it was passed up from a lower level, then this finding (which is a hypothesis at a lower level) can be reconsidered. It can
be rescored and, if necessary, removed from the composite hypothesis from the lower level.

- Jointly uninterpretable data can be considered incompatible and recomputation of the composite can be forced. This is a situation where two incompatible hypotheses are considered essential. Obviously, there is an error somewhere. The error may reside in the data that they are attempting to explain. In such a case, the data can be discarded and the composite reformed without one or both of the hypotheses.

The use of top-down knowledge is to aid the overall problem solving. This comes in the form of opportunistic problem solving so that partial solutions can be passed around, or so that problematical solutions can be caught and dealt with.

3.6.4 Implementation Issues

As stated above, Peirce is well-suited for layered abduction. In order to construct a layered abduction system, one need only define an abducer for each level, then connect the levels together by having one level look to the next lower level abducer's output for its input. This can be thought of as a series of cascaded abducers (in the same way that a ripple adder propagates results to the next adder). Communication from higher levels to lower levels is allowed by means of expectations and reinvocations of lower level abducers as the problem state changes.

In addition, each abducer will require a means of hypothesis evocation and hypothesis instantiation. Any number of methods can be used to tackle these two
subtasks.

3.7 Conclusion

In this chapter I have shown a particular strategy for abduction consisting of hypothesis evocation, hypothesis instantiation and hypothesis composition. Based on past research in the LAIR, many useful ideas have come out pertaining to abduction. These ideas can be used to construct an opportunistic, island building strategy for hypothesis composition. Peirce is exactly such a tool. I have shown the domain independence of Peirce by considering several systems constructed from Peirce. Finally, I have shown how Peirce is perfectly suitable for solving problems in both abduction and layered abduction settings.

The usefulness of Peirce cannot be overstated. Peirce is one of several generic problem solving tools that have been researched by the LAIR in the past. In conjunction with other tools such as CSRL for hierarchical classification and RA for hypothesis matching (scoring), powerful problem solving systems can be constructed which range in task from explanatory systems, diagnostic systems, design systems, perceptual systems and more. With a collection of such tools, many types of problem solving can be captured provided the domain knowledge is available.

In the next chapter, I will demonstrate a particular system constructed from Peirce and other tools to solve a perceptual problem.
CHAPTER IV

ARTREC - a Layered Abduction Speech Recognition System using Articulation as Input

4.1 Introduction

In the last chapter, I described a domain-independent strategy for both abduction and layered abduction, and a tool, Peirce, for constructing abductive problem solving agents. In this chapter I will describe a layered-abduction articulatory-recognition system, ARTREC, constructed from the Peirce tool. I will first describe the task of articulatory recognition and attempt to motivate our research. I will then discuss a source of input for this task. I will turn to ARTREC, describe the experimental data currently being used by ARTREC, give a detailed description of ARTREC's flow of processing, and a detailed look at the knowledge of ARTREC. An example of ARTREC will be given in the next chapter.

4.2 Articulatory Recognition

Articulatory recognition is a task of identifying lexical units from articulatory input. This is similar to acoustic speech recognition (or just speech recognition), however, the input differs, and to some extent, the knowledge content differs. Acoustic speech recognition uses an acoustic signal as input. Articulatory input consists of vocal-tract
motions. vocal-tract motions are the actions that create the speech signal. Figure 18 shows the comparison between the two tasks. You can see that much of the task is abstractly the same, inferring words from spoken utterances by using many types of knowledge. The differences lie in the input and the lower-level knowledge.

Acoustic speech recognition uses knowledge pertaining to auditory concepts, using acoustic features and spectral characteristics of the signal in order to derive (infer) the auditory concepts. Articulatory recognition must make use of knowledge of how sounds arise by shaping those sounds in the vocal-tract. The end result of either task is to infer or derive lexical units (words). And both tasks can make use of other types of knowledge (phonetic, syntactic).

The overall task of articulatory recognition can be broken into subtasks, just as acoustic speech recognition is broken into layers of subtasks. However, articulatory recognition is simpler, not requiring some of the lower-level details of auditory analysis or speech signal processing. The articulatory recognition subtasks are:

- Input the data (vocal-tract motions)
- Examine the data for articulatory events
- Infer articulatory gestures out of these events
- Infer phonetic units from articulatory gestures (where phonetic units can be phonemes, demisyllables or syllables)
- Combine phonetic units into lexical units (using a lexicon) and deriving word boundaries as a result
Figure 18: Acoustic Recognition versus Articulatory Recognition
Use syntactic, semantic, and other high level knowledge to critique the string of words.

It should be noted that articulatory recognition is very novel. There are only a couple of examples of a similar system, Nelson [Nelson, 1979] and later Greenewald et al [Greenewald, 1990] constructed articulatory annotation systems in order to automatically annotate Microbeam pellet data.

Greenewald's system, ArticTool, used both articulatory and acoustic input as well as a phonetic description of the utterance. Its task was, by using simple pattern matching rules, to determine where in the utterance, each word and each articulatory gesture occurred. This annotation problem was much simpler than the recognition task as the utterance was given as part of the input.

4.2.1 Why Articulation?

Articulatory recognition is not necessarily a perceptual task. Humans have no way of directly observing most articulatory motions, with the exception being lip motion and jaw motion. Other motions which are needed to infer articulation are not always visible. A natural question arises. Can articulatory recognition be of any use during acoustic speech recognition?

To answer this question, let us consider the problems with acoustic speech recognition. The speech signal is inherently variant. This has been a problem for all past speech recognition systems. Much of the variation is because of the effects that prosodic control has on the acoustic signal.
Prosody is traditionally, the study of rhythm and meter in speech. Phonetic discussion of prosodic aspects of speech signals have focused primarily on the fundamental frequency contour (F0, or voice pitch). However, prosody occurs because of linguistic and non-linguistic factors which affect the production (or articulation) of speech. Because prosody affects the way we speak, it affects the acoustic signal. Prosodic effects arise out of "real-world" situations such as speaking quickly because you are excited, speaking with a cold, speaking with emphasis on some of the words (corrective emphasis or teaching for instance), speaking loudly to be heard, speaking under the influence of alcohol, being bored or tired. All of these non-linguistic "conditions" affect the way we articulate, which results in significant changes to the acoustic signal.

One example of prosodic control causing problems in speech recognition is due to emphasis. It has long been a problem with speech recognition systems that emphasis and stress will alter the spectral characteristics of sounds making it difficult to recognize unless the emphasis is somehow captured in recognition knowledge[Waibel, 1990]. However, it seems unreasonable to contain a second set of "sound templates" which can be used for emphasized words. In fact, emphasis comes in varying degrees and manners and therefore, multiple sets of emphasized "sound templates" would be required. This would complicate any speech recognition systems' task.

To understand how prosody can affect the acoustic signal, we must see how these situations (such as emphasis, speaking quickly, etc...) affect articulation. One aspect of our research is to not only recognize lexical items from the articulatory input, but
to also attempt to recognize when and where prosodic effects come into play. We have several ideas of how prosody will affect articulation, and in order to fully determine these, implementing prosodic recognition via articulation is a good starting place.

Articulation is the process of creating sound through motions of the vocal-tract. There are dependencies which arise in articulation that cause substantial changes to the acoustic signal. In the past, these dependencies have been ignored except by implementing some form of coarticulation rules. However, these rules have never been sufficient because they are ad hoc in that the rules are designed to fix the problems that arise due to insufficiently modeling of articulation [Lowerre and Reddy, 1980, Erman, 1980, Fujimura and Lovins, 1978]. Because articulatory dependencies are poorly understood, it makes it difficult to know what knowledge is appropriate for recognition [Fujimura, 1991]. It is our hope that by explicitly modeling articulation and reasoning over this model that a speech recognition system can handle the effects of articulatory interactions and prosody on the acoustic signal.

Our primary motivation for our research is to investigate how to model knowledge pertaining to articulation and prosody. We hope that the end result is not only a better understanding of these forms of knowledge, but also, to learn how dependencies caused by articulation and prosody will affect the acoustic signal. The result of this research will be better knowledge for acoustic speech recognition.

We have two other motivations for our research. One is the desire to construct layered abductive problem solvers in the problem area of perception. We have attempted modest speech recognition systems but never got far enough to construct
a large multi-layered system. Articulatory recognition is a perfect problem to try to implement as layered abduction because it is rich in interacting hypotheses and explanatory knowledge is readily available. Also, articulatory recognition offers the same "layered-ness" as acoustic speech recognition.

More importantly, as motivation, is the attempt to aid new speech science research. Osamu Fujimura is attempting to develop a new, non-linear\(^1\) theory of speech production. This Convertor/Distributor model [Fujimura and Wilhelms, 1991] is a prime motivation for our research. It has been hoped that the model of a layered abduction system can incorporate the C/D model as its basic outline or backbone. We can use such a system to "debug" the C/D model and the C/D model to help construct the system. In this way, both sets of research can symbiotically aid each other. A discussion of the C/D model and how abduction might aid the C/D model is presented in chapter 7.4.

### 4.2.2 Articulatory Input - Microbeam Pellet Data

The best lip readers only gain access to two or three channels of articulatory information, namely the lip motions, jaw motion, and glimpses at the tongue. This is not sufficient for the articulatory recognition without other information or clues (such as syntax, discourse, semantical understanding). Fortunately, there is other, more complete, data available in the form of Microbeam pellet data.

\(^1\)The term "non-linear" refers to the aspect of the theory in which speech is not caused by a linear concatenation of phonemes. Rather, speech is created by independent motions of the speech organs interacting to create parallel strings of features. These features are the components or primitives of speech. Most past theories of speech production have used phonemes as the primitive units, and have concatenated them together to represent a string of output. Even syllable concatenation has flaws because it still restricts speech to be a linear ordering of parts.
X-ray Microbeam Pellet Data [Westbury and Fujimura, 1989] is obtained by the following process. The subject speaker has small gold pellets (2-3 millimeter in diameter) placed on various articulators in the vocal-tract. These pellets are the targets to be x-rayed. The Microbeam machine then x-rays the subject’s vocal-tract while the subject speaks. The result is digital records of vocal-tract motions. Since x-rays are dangerous, a clever tracking algorithm is used by the Microbeam machine so that it does not randomly beam x-rays at the subject, but rather, searches for the pellet based on its last location and its velocity. This minimizes subject exposure time to the x-rays. The Microbeam machine will scan the regions where the pellets were last scanned, taking x-rays at these locations.

The scanning rates for each pellet differ depending upon the pellet’s location within the vocal-tract. For example, since the tongue tip is very active, the Microbeam will scan the tongue tip more often than the other articulators such as the jaw or lips. Typically, tens of thousands of brief (10 microsecond) exposures are taken for each sample point of the subject’s vocal-tract during the course of a single utterance.

Because this technique is limited in its abilities, only certain locations of the vocal-tract can be measured. Motions of interest in the vocal-tract are primarily limited to the following:

- **Tongue Motion:** tongue motion consists of between two and four somewhat independent regions of the tongue including the tongue tip, the tongue blade and the tongue dorsum. In the case of two pellets, the tongue blade pellet might be omitted. In the case of four pellets, two tongue blade pellets might be used.
• Labial Motion: labial motion is for the lower lip, or both lower lip and upper lip.

• Mandible Motion: mandible incisor is one of the front teeth. This motion defines the motion of the jaw as the incisor moves with the lower jaw. Another pellet can be placed on the mandible molar so that a complete picture of jaw motion can be derived from the two points.

• Voicing: voicing is caused by vocal fold vibration which occur when the folds' positions bring them close to each other. For the sake of brevity, the vocal fold vibration (in the glottis) will be considered as voicing rather than the acoustic consequence of this vibration (which is also called voicing, i.e. voicing is a term which is identified in the acoustic signal as well as the act which creates the voicing). To acquire voicing data, EGG readings can be taken, or one can detect voicing from the acoustic signal.

• Velar: velar motions occur behind the nasal cavity and are responsible for nasal sounds (e.g. /n/). These motions can be x-rayed by placing a pellet on the velum, reached through the nose.

These articulators are the (primary) elements of the vocal-tract responsible for shaping sounds. The combined motions along with releases of air cause sounds which are then captured in acoustic signals. The output of the Microbeam facility is not a complete examination of the vocal-tract, but rather sample points within the vocal-tract (between four and ten) showing the vocal-tract motion across time. An example
of the pellet data for an utterance can be found in figure 19. You will note that the pellets are listed in both X and Y direction. LL stands for lower lip, TT for tongue tip, TB for tongue blade, TD for tongue dorsum and MANJ for mandible incisor.

Figure 19: Microbeam Pellet Data from the Pine Street data set

The x-ray data is stored as a collection of coordinates. Each articulator (i.e. pellet position) is stored as two separate files, one for the X direction and one for the Y direction. The X-Y coordinate plane can be thought of as running across a profile of a face. There is no data pertaining to the width of the face. Motions in
the X direction represent front-to-back motions in the mouth while motions in the Y direction represent top-to-bottom motions in the mouth. Each pellet is placed approximately in the mid-sagittal plane, except for the reference pellet placed on one of the lower molar teeth.

There are only two Microbeam facilities in the world, the first one constructed is in Japan, the other is located at the University of Wisconsin in Madison. The data we use for ARTREC has come from Madison. The Microbeam facility in Madison does not use a velum pellet and none of our current data has a voicing channel².

4.3 ARTREC - Knowledge-based System for Articulatory Recognition of Speech

Using the Peirce tool, we have constructed a multilayered abduction system for the task of articulatory recognition called ARTREC (ART for articulatory, REC for recognition). The overall task is described as follows:

The Input: Vocal-Tract motions

The Output: An English sentence consisting of ARTREC’s best explanation for the input. Included in this output is ARTREC’s confidence in each word hypothesis and, if possible, identification of the word which is (most) emphasized. ARTREC’s task description can be found in figure 20.

²Although we can obtain voicing by analyzing the acoustic signal that comes with the pellet data for voicing information. For the time being however, we do not make use of the acoustic channel at all, and so we make do without any kind of voicing information.
Figure 20: Task Description of ARTREC

4.3.1 Experimental Data

The initial set of data run on ARTREC has been dubbed the "Pine Street" data.

This is because all of the data is of the form:
"Is it five five nine pine street?"

"no, its five NINE nine pine street."

The data contains five sets of speaker data. Each set consists of approximately 80 files, each containing an utterance like the one above. All combinations of fives and nines were used (i.e. combinations are 559, 599, 999, 955, 959, and so forth). Both positive and negative answers were given (a positive answer being “Yes, its five five nine pine street”). In the case of negative answers, one of the numbers was emphasized. This is known as corrective emphasis. In correcting someone’s error, we tend to emphasize a word or words in order to indicate where the speaker erred. And so, in the above example utterance, the first “nine” in the reply is being emphasized to correct the speaker who thought the number was “five”.

The reason for accumulating this data was to determine what effects corrective emphasis might have on the articulation of the words. This data was perfectly suitable for initial attempts at articulatory recognition for the following reasons. First, we wanted a small but tricky set of words. Second, since we were not using any acoustic signal in our system, we made the problem hard but achievable. Third, while there are only 9 words in the lexicon, there is a sufficient number of articulatory gestures present that have interesting features. This was a sufficient number to adequately test the layered abduction mechanism. We envision enhancements, and have already acquired new pellet data for initial enhancements.
In our current system, we make use of 4 or 5 pellets representing the tongue tip, tongue blade, tongue dorsum, lower lip and mandible incisor (which can be considered as a jaw pellet). No other input is used such as acoustic signal, voicing, nasality (which we could only get through a velum pellet), vocal-tract size, positioning of pellets on the articulators, or speaker-dependent information (sex, dialect, age). Instead, the input to the system consists solely of the motion for the four or five pellets throughout the utterance.

4.3.2 Lexicon for Pine Street

The lexicon for ARTREC is currently very small, 9 monosyllabic words. The number of articulatory motions are limited based on the words in the lexicon. The only form of prosodic information currently sought is that of corrective emphasis. The small lexicon size reflects the limited number of words in the Pine Street data. Since we have only tuned ARTREC for the Pine Street data, only those words currently appear in ARTREC's lexicon. Those words are “is”, “it”, “five”, “nine”, “pine”, “street”, “yes”, “no”, and “its”.

While this makes the problem easier, this set of words is ideal for an initial implementation because the words are similar in many aspects, especially when considering the only source of input as being four or five points in the vocal-tract. The words “five”, “nine” and “pine” are similar in that they both contain the vowel /ay/. “Nine” and “pine” are extremely similar, differing only in the initial sound (initial alveolar

---

3 The data we have consists of 5 pellets for 3 speakers and 4 pellets for 2 speakers. In the cases where only 4 pellets were used, no tongue blade pellet was used.
closure versus initial bilabial closure). "Is", "it" and "its" are extremely similar and gave ARTREC a very hard time in differentiation between them.

4.3.3 Detailed Description of the Flow of Processing

ARTREC's flow of processing is described in detail here. Figure 21 shows ARTREC's overall flow of processing. Each step is described in its own subsection.

![Diagram of ARTREC's flow of processing]

**Figure 21:** ARTREC's flow of processing
Event Generation

ARTREC first inputs the pellet data. Each pellet's data is stored in two channels of information, one representing the X direction and one representing the Y direction. Some of the channels have different sampling rates (i.e. a different number of pellet position values are taken for different channels). All channels are made to have the same number of scans by extrapolating points in the channels where less scans were available. ARTREC then considers each channel individually. ARTREC follows the data detecting peaks and valleys. Some smoothing is done so that "bumps" are ignored\(^4\). The peaks and valleys found are then considered the vocal-tract "events" to be explained.

We can attempt to explain all of the data, however it is sufficient to explain only the peaks and valleys. Consonants are created by some form of constriction in the vocal-tract. These will appear as peaks or valleys (usually as peaks in the Y direction although sometimes as peaks in the X direction and sometimes as valleys). Vowel-like sounds are typically realized by steady positions of all pellets. However, these will also appear as peaks or valleys as the vowel sounds are surrounded by other motions so that the vowel-sounds appear at one end of the extremes while the consonantal-sounds appear at the other end of the extremes. Because of this, we need only explain the extreme motions found in the pellet data\(^5\). The motions in between

\(^4\)A bump can be considered a location where a peak occurs next to a larger peak, or a valley occurs next to a larger valley (within 10 or 20 milliseconds of each other). Only the larger peak or valley is identified as an event, the lesser peaks or valleys are ignored as unimportant.

\(^5\)It should be stated that this strategy of only explaining peaks and valleys is sufficient for the pine street data. We may have to reconsider this strategy when we move on to new sets of data.
are the articulators moving from a peak to a valley or from a valley to a peak, and can be explained as the transition from one sound to the next.

**Word Boundary Location**

After generating the events, ARTREC examines the mandible incisor events. ARTREC tries to find opening/closing motions in this channel (the mandible Y direction). ARTREC infers that each opening occurs at the beginning of a new syllable. ARTREC will then attempt to explain these syllable locations in terms of syllable hypotheses generated later.

**Articulatory Gesture Hypothesization**

Given the events (extrema) found in the pellet data, the system must then explain these individual motions in terms of articulatory actions of which the events are a part. Articulatory gestures are used to explain events. Articulatory gestures are qualitative descriptions of vocal-tract events. Such gestures include apical closure, dorsal closure, tongue lowering, labial closure, and so forth. Articulatory gestures can be thought of as primitive motions required for speech sounds. These motions occur within words or syllables and can be described as a set of features along parallel channels. Gestures will be described in fuller detail later in this chapter.

Articulatory gestures are hypothesized based on the needs of the data. Each

---

*It is not strictly true that each new syllable is reflected by jaw openings in general, but it is suitable for the monosyllabic words in our current lexicon. In fact, each word of our current lexicon is noted by jaw motion and therefore, this is a very stable feature. When ARTREC is expanded, we will need to determine whether this feature still holds true.*
event is examined, and a list of possible causes (gestures) is generated. Each gesture hypothesis is assigned a plausibility (confidence score) by local match knowledge. This score is based on how close the gesture comes to that gesture’s feature template.

After assigning plausibility scores to the gesture hypotheses, each hypothesis examines the event data and determines exactly which events it would be able to explain. This entails finding peaks or valleys that the gesture would typically expect. The events would be located nearby in time to the temporal location of the gesture. However, a gesture can also attempt to explain nearby events which might have resulted due to the influence of the gesture. Therefore, a gesture hypothesis will be able to potentially explain many events in the temporal region around the gesture hypothesis.

Noise Hypothesization

Because some motions are not created by the utterance of a sound, but created by accident or due to non-linguistic effects, the system needs a way to determine if an event should be accounted for by a gesture (as a meaningful vocal-tract motion) or as noise (some random or unintentional motion). Therefore, while events are being examined to generate gesture hypotheses, another type of hypothesis is created when necessary, a noise hypothesis.

Noise hypotheses can be thought of as alternative explainers for events which do not seem intentional. However, determining if a motion was intended or unintended is difficult, at least when examining individual motions. It is easier to detect if a motion
was intentional or not when examining many motions. Fortunately, the very nature of
the Peirce strategy encourages this behavior making it easy to implement. An event
which does not closely match the idealized gesture template can be considered as
noise and a noise hypothesis is assigned. The noise hypothesis is given a plausibility
depending upon how far away the match was from the gesture template. Events
which closely match templates will generate implausible noise hypotheses while events
which do not closely match templates will generate very plausible noise hypotheses.
A noise hypothesis will only account for a single event, which is substantially different
from a gesture hypothesis which could potentially explain many events. Also, noise
hypotheses are made to be incompatible with gesture hypotheses which attempt to
explain the same articulatory event.

Lower Level Abduction

The next step in the process is to explain the events in terms of gesture and noise
hypotheses. The abducer generated from Peirce will do this. As described in the
previous chapter, the abducer will run through cycles of explanation going from the
most certain and easiest explanations to less certain and harder explanations. The
cycle will seek out essential hypotheses followed by clear best hypotheses and finally
weak best hypotheses. If hypotheses are accepted as believed at any time, the effects
of believing the hypotheses are propagated and the cycle starts again as a search for
essentials.

One nice feature of Peirce is that the system builder has the ability to adjust the
degree of confidence versus explanatory coverage. That is, if we would like to generate a more confident answer, the result would most likely be less coverage whereas a more fully explained set of data would generate some less confident conclusions. This ability of Peirce was used in an experiment to determine how accurate ARTREC could be. This variation is discussed in the next chapter.

Syllable Hypothesization

After the first abductive explanation has been generated, the system has composed a set of plausible gesture hypotheses (and noise hypotheses) to account for the pellet data motions. This is obviously an insufficient level of detail for us. We would prefer to recognize words out of the vocal-tract motion instead of gestures. Therefore, we need additional inference.

The next step in the process is to explain the articulatory gestures. The system now has a set of semi and very plausible articulatory gestures to explain. The noise hypotheses will not need explaining as they have already been explained (as noise, unintentional motions).

As the first level used articulatory events to generate hypotheses, the second-level will use articulatory gestures to generate syllable hypotheses. The system uses knowledge of which syllables can cause which gestures. This information is straightforward as a syllable is composed of some number of gestures. To utter the syllable requires using most or all of those gestures. Given the set of gestures, ARTREC generates the list of syllables which could have caused those gestures. Currently in ARTREC, the
set of syllables is very small reflecting the small lexicon (in actuality, the syllables are the words of the system since the lexicon is composed only of monosyllabic words).

Once syllable hypotheses have been generated, they are scored in terms of plausibility in a similar fashion as the gestures. Syllables hypotheses examine the gestures in the area and generate a plausibility based on the presence and absence of gestures in the area. A syllable will require certain gestures (known as necessary gestures), expect certain gestures (known as expected gestures) and expect to not see other gestures (known as impossible gestures - which means that if this gesture appears, it is impossible or unlikely for the syllable to have occurred). Plausibility scores are based on locating or not locating these types of gestures. Naturally, if the necessary and expected gestures appear and the impossible gestures are not found, a syllable hypothesis is given a high plausibility. If the necessary gestures are not found or impossible gestures appear, then a hypothesis is given a low score (or ruled out). Finally, if some combination of necessary gestures appear, but not all of them, or some impossible gestures appear even though all of the necessary gestures are found, then the plausibility is set to a neutral, or middle value. This indicates that, based on the accepted gesture hypotheses, the analysis was very uncertain.

After scoring the syllable hypotheses, each hypothesis considers the gestures and decides what gestures it could potentially explain. A syllable hypothesis can explain either a necessary gesture or an expected gesture which is found in the region. Further, a syllable hypothesis cannot explain a gesture which is found too far away, one which seems to be within an adjacent syllable. It should be noted that syllables can overlap
each other to some extent in that. One syllable may attempt to explain a gesture located within the boundaries of a bordering syllable, but only if that gesture will not interfere with the pronunciation of that bordering syllable.

Additional Noise Hypothesization

Again, noise hypotheses are generated. These hypotheses are used to explain articulatory gestures which seem out of place. Such gestures occur either as random (unintentional) motions, or because the lower level mistakenly accepted an incorrect gesture to explain some event. Noise hypotheses are scored based on how many gestures present in one area seem implausible. And again, noise hypotheses are made to be incompatible with syllable hypotheses which are attempting to explain the same gestures.

High Level Abduction

The second level abducer is now run to generate a best explanation in terms of syllables for the gestures. Again, the abducer will run through a cycle going from very confident local abductions to less confident ones. In this abducer, the first step is to look for confirmed hypotheses. Confirmed hypotheses are those hypotheses which were rated with the highest possible plausibility in local scoring. The plausibility of "confirmed" is only given if a hypothesis finds all of the features it is seeking and none of the impossible features. After seeking confirmeds, the abducer goes on to find essentials, clear bests and weak bests, propagating the effects of any believed hypotheses and starting the cycle over with confirmeds when any new hypotheses are
added to the growing composite.

The explanation generated from this abducer represents the set of syllables, in order, which ARTREC has determined are the causes of the original motions in the pellet data. Currently, ARTREC only has monosyllabic words. Therefore, once syllables have been generated, there is no need to continue through higher levels of abductive problem solving. Words are equivalent to syllables.

Emphasis Detection

The last step in the bottom-up processing is to explain the jaw motion (mandible motion). This is accomplished in two phases. First, jaw openings are considered as syllables (or words). After finding the syllable boundaries form the earlier phase, this knowledge is left aside until actual syllables have been identified. The best explanation from ARTREC explains both the pellet data thus far examined (i.e. tongue motions, labial motions) and the jaw motion. However, some of the jaw motion might be exaggerated. The most exaggerated motion will probably be due to emphasis. To explain this exaggeration, ARTREC uses an emphasis hypothesis.

Top-Down Processing

Top-down processing may now commence. Top-down processing will use the "islands of certainty" generated by the bottom-up processing to drive expectations to the lower levels. This happens as follows. Given an accepted syllable (word) hypothesis at the top level, expectations of gestures are generated. These expectations are passed down to the articulatory gestures level.
Expectations can take on two forms, positive expectations (i.e. those hypotheses we expect to be true) and negative expectations (i.e. those hypotheses we expect to be false). Currently, ARTREC only uses positive expectations. These expectations are then used in order to clarify the middle level. It is possible that several gesture hypotheses were neither believed (accepted) nor disbelieved (ruled-out), the top-down expectations can help further determine the status. Expected gestures should be more believed while unexpected gestures should be less believed. After propagating these expectations downward, the first level abducer can be rerun to reexamine the gesture hypotheses and the events in the light of the conclusions already reached.

After coming to a new conclusion at the middle level, the entire composite is passed up to the next level and syllables are rehypothesized to account for the newly accepted composite. The higher level is then rerun. Hopefully, this additional processing will be able to further the explanation.

It should be noted that other forms of top-down processing can occur. Currently, the only form is what has been described here. However, another form of top-down processing is in using hard decisions. If a choice for explaining some gestures is difficult because only a couple of syllables are suggested, but they are equally (or nearly equally) plausible, then differentiation knowledge can be passed down to the lower levels. This form of processing has been discussed in our research group but not yet implemented. If the necessity arises for this additional form of top-down processing, it will be fairly simple to implement.
The Output

The output of ARTREC will be a sequence of syllables (words) and ARTREC’s choice for the emphasized word. Each syllable (word) will be accompanied by its temporal location within the utterance (i.e. the approximate time that the syllable was uttered).

Also, one nice feature of Peirce’s strategy is that of maintaining the acceptance criteria (or justification) of an accepted hypothesis. Each syllable is displayed with its plausibility and its justification (confirmed, essential, clear best or weak best). This allows us to examine the output of ARTREC and not only determine where ARTREC erred, but also why ARTREC erred. If ARTREC makes errors when accepting essentials or confirmeds, this alerts us to the fact that ARTREC is not correctly judging (scoring) hypotheses.

A detailed example of ARTREC, demonstrating the bottom-up processing described here is given later in this chapter.

4.3.4 Confidences

ARTREC uses what are known as the 9-valued confidence set and the 5-valued confidence set. These are the vocabularies that ARTREC will use to indicate how plausible a hypothesis is. These plausibilities can be thought of as qualitative probabilities. They are not detailed numerical values derived from Bayesian equations or statistical matching methods. They are, instead, derived based on how closely the hypothesis matches the situation at hand, but given as abstract statements of confidence (such

7It was because of too many incorrect essentials that led us to implement noise hypotheses.
A tool known as RA (for Recognition Agent) is used to capture local match knowledge. An RA is designed for each concept or hypothesis that the system use. When faced with a decision about the relevancy (or plausibility) of a hypothesis, the RA for that hypothesis is invoked. The RA returns a confidence value. For instance, an RA for the hypothesis "apical closure" will examine the events in the area and make a determination of how likely "apical closure" is. An abstract RA can be found in figure 22. In the next section, RA information from ARTREC will be given.

Figure 22: Recognition Agent

The 9-valued set uses the values: confirmed, very-likely, likely, somewhat-likely, neutral, somewhat-unlikely, unlikely, very-unlikely and ruled-out. Confirmed hypotheses are the highest possible ratings, and are considered as hypotheses which
match so well that they should automatically be placed in the composite hypothesis. Note that we do not need to automatically accept confirmed hypotheses if we feel our local match knowledge might lead us astray, but in ARTREC's second level, confirmed hypotheses are automatically included. Ruled-out hypotheses are the lowest possible ratings and are hypotheses that either found none of the expected features among data (or findings) being examined, or found data or findings which were impossible for that hypothesis. Hypotheses which are rated as ruled-out are dismissed from any future consideration.

The 5-valued set uses the integers 0, 1, 2, 3, and 4. These values are assigned to gesture hypotheses depending on how closely the hypothesis matches its template. A 0 is a very close match, close enough to be convincing that the gesture did actually occur (like confirmeds above). A 4 is a very low score similar to a "unlikely" rating and is given to any gesture which did not match the template at all. Intermediate scores of 1, 2, and 3 are given to any hypotheses which matched to some intermediate degree. Any potential gesture hypotheses which did not even match with the score of 4 is immediately discarded as "ruled-out". A 5-valued set seems sufficient, at least at present, for the gesture plausibilities.

Typically, an abducer will only want to accept highly rated hypotheses, such as any hypothesis given a rating higher than somewhat-likely. However, in the case of ARTREC, neutral hypotheses can be accepted as "weak bests" and somewhat-unlikely hypotheses can also be accepted under some circumstances. It should be noted though, that in accepting low rated hypotheses, our confidence in the composite
is weakened.

4.3.5 Types of Articulatory Knowledge

ARTREC models certain types of knowledge to accomplish the task of articulatory recognition. These kinds of knowledge are articulatory events, articulatory gestures, syllables and the prosodic information.

- **Events** are the motions that we wish to explain. Events are articulator movements. We only wish to explain extrema (peaks and valleys).

- **Gestures** are qualitative descriptions of vocal-tract motions. Gestures are groupings of events, all located in some close proximity (in time) and are interrelated.

- **Syllables** are “phonetic primitives”. Syllables are composed of groups of gestures (one or more gestures). Syllables can then be composed into words.

- **Corrective emphasis** is the single form of prosody currently being recognized. Corrective emphasis appears as exaggerated mandible (jaw) motion. This form of emphasis (and other forms of emphasis) can be detected by examining the jaw motions for large valleys.

Event Knowledge

Events are motions of interest in the pellet data. The events are the data items that we want explained. Consonants are formed by constrictions in the vocal-tract.
Vowels are formed by lengthy releases of air while the vocal-tract is relatively stable. In examining pellet data, we have found that both consonants and vowels are typically indicated by peaks and valleys in the pellet data. Therefore, we restrict our explanation of pellet data to only the events of interest, peaks and valleys.

A simple algorithm follows each pellet and generates a list of peaks and valleys. A smoothing algorithm is used so that slight bumps or depressions around peaks and valleys are not generated. That is, if a peak is found right next to another, lesser peak, the lesser peak is ignored.

The channels used in ARTREC refer to the following articulators:

- Tongue Tip in the X and Y direction
- Tongue Blade in the X and Y direction
- Tongue Dorsum in the X and Y direction
- Lower Lip in the X and Y direction
- Mandible Incisor in the Y direction

X motions refer to the articulator moving forward and backward in the mouth. A peak in the X direction is the articulator moving forward in the mouth to an extreme point, and then moving backward. A peak in lower lip refers to the lips pursing. A valley in the X direction is the articulator moving backwards in the mouth to an extreme and then moving forward again.

Y motions refer to the articulator moving up and down in the mouth. A peak in the Y direction refers to the articulator moving up in the mouth, reaching an extreme
and then moving downward. For the tongue tip, a peak is usually the tongue tip touches the alveolar ridge, or coming close to it. A peak in lower lip is usually the lower lip touching the upper teeth or the upper lip. A valley in the Y direction refers to the articulator moving downward, reaching an extreme and moving upward. A valley in tongue tip means that the tongue has moved low in the mouth. A valley in lower lip or mandible incisor means that the mouth is opening. These event types are used to generate possible articulatory gesture hypotheses.

Gesture Knowledge

Articulatory Gestures are qualitative descriptions of vocal-tract motions. They are groupings of events which together, perform some useful act towards an utterance. Several articulators can be used in conjunction to create a gesture. Gestures can persist in duration from a single motion (such as labial closure\textsuperscript{8}) or through many motions (such as a fricative-alveolar-glide motion\textsuperscript{9}). ARTREC attempts to explain the pellet data events in terms of articulatory gestures. The process for this is one of hypothesization, scoring and composition. Table 1 indicates the features sought for each gesture currently present in ARTREC. In this table, P stands for peak, HP for high peak, VHP for very high peak, MP for mid peak, V for valley, LV for low valley and VLV for very low valley. The “type” refers to the type of utterance (phoneme or syllable) that the gesture is typically found in.

Figure 23 shows an abstract gesture template. Gesture templates are stored as

\textsuperscript{8}Labial closure occurs when the lower lip touches the upper lip

\textsuperscript{9}A fricative-alveolar-glide motion is responsible for the sound /str/ as in the word “street”.
Table 1: Gesture Types

<table>
<thead>
<tr>
<th>Gesture</th>
<th>Type</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bilabial Closure</td>
<td>(/p/)</td>
<td>LLX (HP), LLY (P)</td>
</tr>
<tr>
<td>Labiodental Closure</td>
<td>(/v/,/f/)</td>
<td>LLY (HP), and upwardslope &gt; 1</td>
</tr>
<tr>
<td>Alveolar Closure</td>
<td>(/n/,/t/)</td>
<td>TTY (HP)</td>
</tr>
<tr>
<td>Final Alveolar Closure</td>
<td>(/n/,/t/)</td>
<td>TTY (MP to HP), TBY (HP)</td>
</tr>
<tr>
<td>Alveolar Partial Closure</td>
<td>(/s/)</td>
<td>TTY (P), TTX (HP), TBX (P)</td>
</tr>
<tr>
<td>Alveolar Glide</td>
<td>(/y/)</td>
<td>TTY (P), TBY (P), TDY (P)</td>
</tr>
<tr>
<td>Tongue Retroflection</td>
<td>(/t/)</td>
<td>TTY - (VHP), TTX, TBX - (LV)</td>
</tr>
<tr>
<td>Tongue Raising</td>
<td>(/s/,/n/,/t/)</td>
<td>TTY (P), TBY (P)</td>
</tr>
<tr>
<td>Tongue Lowering</td>
<td>(/ay/)</td>
<td>TTY (V)</td>
</tr>
<tr>
<td>Tongue Forward</td>
<td>(/i/,E(^{10}))</td>
<td>TTX (HP), TBX (P)</td>
</tr>
<tr>
<td>Tongue Backward</td>
<td>(/ow/)</td>
<td>TTX (VLV), TBX (V), TDX (V)</td>
</tr>
<tr>
<td>Closure-Retroflection</td>
<td>(/str/)</td>
<td>Alv. Part. Clos. &amp; Tongue Retroflec.</td>
</tr>
</tbody>
</table>

RAs. The output of a gesture template is the confidence that ARTREC has that a given set of motions is actually the gesture in question.

Hypothesization of gestures is accomplished by a cueing process. A peak or valley (a pellet data event) in one channel is caused by a gesture. ARTREC uses knowledge of which gestures cause which motions and hypothesizes the possible gestures that could be responsible for the event being examined. These gestures are then scored, based on local match knowledge. This knowledge is based on a feature template where features are types of motions in various channels. The gesture score is based on how closely the data matches the gesture template.

The types of motions that make up features are peaks and valleys of the various data channels, rough slopes of the peaks and valleys and the space between peaks and valleys, rough durations, and the height or depth of peaks and valleys. Peaks, valleys...
and slopes are stored as threshold values that must be surpassed. Both absolute threshold and relative thresholds are used. Absolute thresholds state that the motion must surpass a value referring to a position in the mouth (for example, labial closure means that the lower lip must reach the upper lip, and therefore an absolute threshold is used). Relative thresholds state that a motion must move a certain amount relative to another motion (for example, in /str/, the tongue tip must slide from a forward location in the mouth to a location behind the alveolar ridge, this requires a motion
with a relative distance).

Gesture templates were written during a process of examination of pellet data while the ARTREC system was being written. These templates are stable although future work will be able to adapt these templates during runtime so that the system can in effect, "learn" new patterns of features.

Tables 2 and 3 are two examples of gesture templates. They represent ARTREC's recognition knowledge for the gestures for Final Alveolar Closure and Tongue Retroflection. The patterns are stored in order of confident identification meaning that, if the top row's pattern matches then the hypothesis for "final alveolar closure" (or "tongue retroflection") is very confident (rated a 0, which stands for a very close match). If this top row does not match, then the next row is considered and so on until no rows match. If none of the rows match, the hypothesis is assigned automatically ruled out of consideration.

Note that the values in tables 2 and 3 are not actual measurements, but are normalized values within a range of -25000 and 25000. A value of 25000 is the most extreme high value while -25000 is the most extreme low value. Also note that these two gesture templates make use of only absolute threshold values. More complex gestures make use of slopes and relative thresholds.

Noise hypotheses are used to explain events where gesture hypotheses seem implausible. That is, in cases where gesture hypotheses do not fit the templates well at all, noise hypotheses are also considered.
Table 2: Gesture Template for “final alveolar closure”

<table>
<thead>
<tr>
<th>Features</th>
<th>Threshold Values</th>
<th>Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTY, TBY, TDX</td>
<td>&gt; 20000, &gt; 15000, &gt; 15000</td>
<td>0</td>
</tr>
<tr>
<td>TTY, TBY, TDX</td>
<td>&gt; 18000, &gt; 12000, &gt; 12000</td>
<td>1</td>
</tr>
<tr>
<td>TTY, TBY, TDX</td>
<td>&gt; 15000, &gt; 10000, &gt; 10000</td>
<td>2</td>
</tr>
<tr>
<td>TTY, TBY, TDX</td>
<td>&gt; 15000, &gt; 10000, &gt; 8000</td>
<td>3</td>
</tr>
<tr>
<td>TTY, TBY, TDX</td>
<td>&gt; 12000, &gt; 8000, &gt; 8000</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 3: Gesture Template for “tongue retroflection”

<table>
<thead>
<tr>
<th>Features</th>
<th>Threshold Values</th>
<th>Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTY, TTX, TDX, TBX</td>
<td>&gt; 23000, &lt; -10000, &gt; 10000, &lt; -5000</td>
<td>0</td>
</tr>
<tr>
<td>TTY, TTX, TDX, TBX</td>
<td>&gt; 20000, &lt; -6000, &gt; 8000, &lt; -5000</td>
<td>1</td>
</tr>
<tr>
<td>TTY, TTX, TDX, TBX</td>
<td>&gt; 18000, &lt; -2000, &gt; 6000, &lt; -3000</td>
<td>2</td>
</tr>
<tr>
<td>TTY, TTX, TDX, TBX</td>
<td>&gt; 18000, &lt; 0, &gt; 4000, &lt; 0</td>
<td>3</td>
</tr>
<tr>
<td>TTY, TBX</td>
<td>&gt; 17000, &lt; 0</td>
<td>4</td>
</tr>
</tbody>
</table>

Syllable Knowledge

The highest level of knowledge currently used in ARTREC is of syllables. We have chosen syllables as the primitive phonetic units as opposed to phonemes or other units because syllables seem more stable. Syllables are also the basic phonetic unit in Fujimura’s C/D model and so the choice was made based on these reasons\(^\text{11}\).

Since syllables are the largest units currently used in the system, the task for ARTREC is to generate the best set of syllables to explain the pellet data. In particular, the syllables will be used to explain the articulatory gestures. To infer syllables from gestures, ARTREC uses the same basic strategy as it does for inferring gestures

\(^{11}\)While we feel that phonemes are poor choices of phonetic units because of the linearity they impose, we are not trying to make any general statement about linear phonology in this case. We have decided to use syllables for convenience sake.
from events, hypothesization, scoring and composition.

Syllable hypotheses are generated based on cues using gestures generated from the previous level. A future expansion of ARTREC will be to use a classification hierarchy of syllables and to classify possible syllables rather than to cue syllables from the gestures.

Once syllable hypotheses are generated, they are scored by using a feature-based matching similar to how gestures are scored. Syllable hypotheses examine the area for appropriate gestures (necessary and expected gestures) and inappropriate gestures. The area of interest is the area denoted by the syllable boundaries with a little bit of overlap to the preceding and succeeding syllables. If the appropriate gestures appear and no inappropriate gestures appear within this region of interest, then a high score is assigned. This score is lessened if some wanted gestures do not appear and/or some unwanted gestures do appear.

The gestures are reexamined to see what a hypothesis can explain. Syllable hypotheses are able to explain any gesture in the area that they expect. However, if the gestures are located well outside of the syllable's boundary, the syllable hypothesis is unable to explain it because it would be impossible for the hypothesized syllable to have caused that gesture. Figure 24 indicates how syllable boundaries are located by examining mandible motion, and how syllable boundaries are used to determine what gestures might be part of a syllable, and what gestures might be outside of that syllable's influence.

Tables 4 and 5 display ARTREC's recognition knowledge for syllables. Table 4
indicates the "desired" or expected gestures that will appear in each syllable type. These are given as the gestures to appear in the (approximate) initial position of the syllable, middle position and final position. It should be noted that some syllables will not have 3 features but fewer, and so, in such a case, no middle gesture is given. ARTREC also looks for "undesired" gestures which can be used as rule-out knowledge for a syllable hypothesis. This knowledge is shown in table 5. This rule-out knowledge may pertain to only a portion of the syllable, and so each gesture is listed with a symbol indicating whether that rule-out knowledge should be applied to the
initial section of the syllable (I), middle section (M), or final section (F).

Table 4: Syllable Recognition Features

<table>
<thead>
<tr>
<th>Syllable</th>
<th>Initial</th>
<th>Middle</th>
<th>Final</th>
</tr>
</thead>
<tbody>
<tr>
<td>Five</td>
<td>labiodental</td>
<td>tongue low</td>
<td>labiodental</td>
</tr>
<tr>
<td>Nine</td>
<td>alveolar</td>
<td>tongue low</td>
<td>final alveolar</td>
</tr>
<tr>
<td>Pine</td>
<td>bilabial</td>
<td>tongue low</td>
<td>final alveolar</td>
</tr>
<tr>
<td>Its</td>
<td>tongue forward</td>
<td>alveolar</td>
<td>partial alveolar</td>
</tr>
<tr>
<td>Is</td>
<td>tongue forward</td>
<td></td>
<td>partial alveolar</td>
</tr>
<tr>
<td>No</td>
<td>alveolar</td>
<td></td>
<td>tongue backwards</td>
</tr>
<tr>
<td>Yes</td>
<td>tongue forward</td>
<td></td>
<td>partial alveolar</td>
</tr>
<tr>
<td>Street</td>
<td>partial alveolar</td>
<td></td>
<td>final alveolar</td>
</tr>
</tbody>
</table>

Table 5: Syllable Rule Out Knowledge

<table>
<thead>
<tr>
<th>Syllable</th>
<th>Rule Out Gestures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Five</td>
<td>alveolar (I,F), bilabial (I), tongue backwards (M)</td>
</tr>
<tr>
<td>Nine</td>
<td>labiodental (I), bilabial (I), tongue backwards (M)</td>
</tr>
<tr>
<td>Pine</td>
<td>alveolar (F), labiodental (I,F), tongue backwards (M)</td>
</tr>
<tr>
<td>Its</td>
<td>str (F), tongue backwards (I), tongue low (I)</td>
</tr>
<tr>
<td>Is</td>
<td>alveolar (I), str (F), tongue backwards (I), tongue low (I)</td>
</tr>
<tr>
<td>No</td>
<td>bilabial (I), str (I), labiodental (I) tongue high (F), tongue low (F)</td>
</tr>
<tr>
<td>Yes</td>
<td>alveolar (I), str (I, F), tongue backward (M), tongue low (M)</td>
</tr>
<tr>
<td>Street</td>
<td>labiodental (I), bilabial (I), tongue forward (M), tongue low (M)</td>
</tr>
</tbody>
</table>

There are also noise hypotheses at this level. Some gestures may have been accepted at the previous level even though they are not true. These gestures may have still been accepted if there is evidence to support them. However, it would lessen the explanation's confidence if there were mistakes due to mistaking unintentional motions for articulatory gestures. To safeguard against this, noise hypotheses are offered at the syllable level. These hypotheses are used to explain single instances of gestures which do not seem coherent with other gestures in the area. For example,
seeing both an alveolar closure and a labial closure in the same temporal proximity is not very likely as one would not try to accomplish these two constrictions to utter one sound. Therefore, one is probably noise. However, since both gestures can be accepted at the previous level (because they would not compete to explain some motion), ARTREC will appeal to noise hypotheses to explain one of the two gestures. The noise hypotheses are rated based on how well the articulatory gesture scored. If the gesture seemed very plausible, the noise hypothesis is given less plausibility. If the gesture seemed implausible, the noise hypothesis is given a higher plausibility.

Syllables which are attempting to account for gestures which are located within the same boundary are made incompatible (or mutually exclusive). This is because it is impossible to speak two different syllables (or words) at the same time. Therefore, ARTREC will make them incompatible so that, under no conditions will both syllable hypotheses be accepted into the composite.

This is a beneficial addition to ARTREC as incompatibility handling is very useful in solving problems using the Peirce strategy. If a syllable hypothesis is accepted during the hypothesis composition phase of abduction, any incompatible hypotheses are immediately ruled out from further consideration. Noise hypotheses are also incompatible with the syllable hypotheses that they are competing against. That is, if a noise hypothesis and syllable hypothesis are attempting to explain the same gesture, they are incompatible. This is because it is not possible for a gesture to be both intentional and unintentional.
Emphasis Knowledge

The last type of knowledge that the system has is knowledge pertaining to prosody. However, in our current implementation of ARTREC, the only type of prosodic information is corrective emphasis. Corrective emphasis is a form of emphasis whereby one attempts to correct someone else's question or answer. In this case, the corrective emphasis will be on one single word (syllable) in the utterance.

Corrective emphasis can be found by seeking for exaggerated jaw motion. Therefore, the mandible incisor is examined. Each word is indicated by jaw opening and closing. An emphasized word will have an even greater jaw opening.

However, if we only had to look for the largest jaw opening, then this would be a simple problem to solve. It is not the case unfortunately. There is a phenomenon in speech known as articulatory declination [Vayra and Fowler, 1992, Pierrehumbert, 1979], where a speaker will have more clarity in enunciation at the beginning of the utterance. As time goes on, the speaker will enunciate less and less. The emphasized word is always in the second half of the utterance (for our data). ARTREC must compare the jaw openings of the words in the second half of the utterance, and, by taking into account the articulatory declination, decide which word was emphasized. This is implemented as a straightforward strategy of finding the jaw openings, computing a sloping line across the locations of the jaw openings and determining which word penetrates furthest across the sloping line. The slope of the line is derived from empirical examination of pellet data.
4.4 Conclusion

In this chapter, I have described the ARTREC system, a layered abduction system solving the problem of articulatory recognition. The strategy used is a layered abductive strategy which attempts to explain vocal-tract motions in terms of words, noise and emphasis. The exact flow of processing is to explain vocal-tract motions in terms of articulatory gestures. Then, it uses articulatory gestures as input to be explained by syllables. Syllables are the highest level currently implemented in the system (as all of the words in our lexicon are monosyllables).

The knowledge types needed to solve this task pertain to articulatory events, articulatory gestures, syllables and corrective emphasis. The same basic strategy is used at both levels of processing, generate plausible hypotheses to explain the findings, score the hypotheses, set up explanatory information, and then use the abductive strategy in Peirce to generate a best explanation. The last step of the system is to seek a word which seems to be emphasized by looking for exaggerated jaw motion.

In the next chapter, I will present a detailed example of ARTREC running a typical case. The chapter after that will consider ARTREC’s experimental results.
CHAPTER V

An Example of ARTREC

5.1 Introduction

This chapter will demonstrate ARTREC in action with a typical utterance. This example will be presented as a series of pictures showing the pellet data, and the parts of the pellet data being explained.

The example being demonstrated here is from the OFJW12ppp data set, record number 30. This example will demonstrate ARTREC version 2 (i.e. from Experiment 2 from chapter 6) using the flow of processing described in chapter 4. The top-down processing discussed in chapter 4 is a simple extension to the bottom-up processing, but will not be demonstrated in this example.

Throughout this chapter, I will refer to “typical” numbers of data and hypotheses. These typical values refer to ARTREC version 2, and not the versions 1, 3 or 4 discussed in the next chapter. By “typical”, I mean the approximate number of hypotheses and data that ARTREC has to deal with for each new utterance.

The example is of the utterance “Is it Nine Five Nine Pine Street?”. For brevity, I will only focus on this portion of the utterance. The remainder of the utterance, “yes its Nine Five Nine Pine Street” is not shown here as part of the example. However,
ARTREC works in the same way on both portions of the utterance.

Figure 25: Pellet data for ARTREC example

The utterance, as is presented to ARTREC, is shown in figure 25. This figure displays the form of the data, Microbeam pellet data. You can see the 5 pellets (tongue tip denoted as TT, tongue blade denoted as TB, tongue dorsum denoted as TD, lower lip denoted as LL and mandible incisor denoted as MAN1). All of the pellets are shown in both X and Y direction (denoted as TT_X and TT_Y, and so
forth) with the exception of the mandible incisor in which ARTREC only uses the Y
direction. The acoustic signal is also given for reader's reference only. ARTREC does
not use the acoustic signal at all.

5.1.1 Events and Syllable Boundaries

The first step in ARTREC, after loading the pellet data, is to examine the pellet
channels for peaks and valleys. The peaks and valleys for the tongue and lip pellets
are used as data to be explained by gestures (and later syllables). The valleys for the
mandible pellet are used as data to be explained by syllable locations and prosodic
emphasis hypotheses.

A typical case for ARTREC has between 150 and 250 events to explain. Some of
these can immediately be dismissed as falling outside of the actual utterance (either
before the utterance commences or after the utterance ends). The remaining events
are those which ARTREC will attempt to explain. ARTREC will attempt to explain
as many as it can (although complete coverage is not necessary).

Figure 26 shows, for the example utterance, the events to explain. The events to
explain are noted with small circles in the figure. You will notice that the mandible
incisor data only requires the valleys to be explained. Valleys are explained by syllable
hypotheses and prosodic emphasis.

5.1.2 Gesture Hypothesization

The next step in ARTREC is to hypothesize the cause for the events. The cause is in
terms of articulatory gestures and noise. The typical ARTREC case will have between
Figure 26: Events indicated for ARTREC example
Figure 27: Gestures hypotheses and what they can explain
75 and 150 gestures hypotheses and between 50 and 100 noise hypotheses to explain the articulatory events. Out of these, most hypotheses are ruled out eventually leaving only 30 to 50 gesture hypotheses and 15 to 25 noise hypotheses to contribute to the composite hypotheses formed at this level.

Figure 27 shows the gesture hypotheses and what they can explain of the events. Each gesture type is listed once, the number following the gesture in brackets notes how many different gesture hypotheses of the annotated type have been found for the example. You will see that most of these hypotheses are discarded after the first abduction.

Figure 28 displays the noise hypotheses that were hypothesized to explain the events where either no gesture hypotheses were available, or where gesture hypotheses were available but they were not very convincing (i.e. highly confident).

Each hypothesis has been scored at this point by using gesture templates. Clearly implausible hypotheses are ruled out before ARTREC reaches this point.

The next step in ARTREC is to generate the first composite hypothesis. This is accomplished by the lower level abducer. The input to this abducer is the sets of noise and gesture hypotheses and the set of articulatory events, the output is a small (hopefully) list of accepted gesture and noise hypotheses. Figure 29 shows the two types of hypotheses accepted into the composite hypothesis, and what each hypothesis can explain out of the events. It should be noted that figure 29 does not show explanations for all of the events. Some events are not important and are left unexplained. These events are those which fall outside of the utterance (i.e. before or
Figure 28: Noise hypotheses and what they can explain
Figure 29: Best explanation of the events in terms of gestures
after the utterance) and those which are caused from small amounts of motion. These motions are more difficult to explain because there are less distinguishing factors to clearly promote one single hypothesis. Therefore, these lesser motions are left unexplained with the hopes that they are unimportant. We have found that ARTREC can still come to a complete conclusion (in terms of the syllables) in spite of leaving some events unexplained.

The composite from the lower level consists of noise and gesture hypotheses. ARTREC now discards the noise hypotheses and uses the gesture hypotheses. These are passed up to the next level to become findings to be explained.

5.1.3 Syllable Hypothesization

ARTREC now examines the accepted gesture hypotheses and generates syllable hypotheses to explain them.

Figure 30 shows the syllables hypotheses and noise hypotheses to explain the gestures previously accepted. In the typical case, between 40 and 60 syllables are hypothesized to explain most of the gestures. In addition, there are between 25 and 50 noise hypotheses generated to explain some of the more suspect gestures (note that there will be overlap between what syllable hypotheses and noise hypotheses will account for). In figure 30, you can see that there are many “five” and “nine” syllable hypotheses whereas there is only one hypothesis for each of “is”, “it” and “street”. Typically, there will be between two and five syllable hypotheses generated for each syllable location. Since “five”, “pine” and “nine” are all similar to a large extent, syllable hypotheses for these three syllables will typically be generated when the
Figure 30: Syllable and noise hypotheses
word is any one of five, nine or pine. Also, because of the /s/ gesture (partial alveolar closure), "street", "is", and "its" will often be hypothesized to explain occurrences of the /s/ gesture.

The syllable and noise hypotheses will have been scored at this point by using Recognition Agents. Those hypotheses which were clearly implausible will have been ruled out before this point.

The second abducer is now run, and it comes to the final explanation of the motions in terms of syllables. The best explanation for this example is shown in figure 31. Here, you can see the approximate syllable (word) locations and the syllables chosen for the best explanation.

It should be noted that this example used only half of the utterance. The complexity of the overall problem can be thought of as twice what was displayed here. The second half is similar, but in some ways more difficult. The difficulty arises due to two independent factors. First, articulatory declination might occur which would mean that the second half of the utterance is less clearly enunciated. The result of the articulatory declination is less certain hypotheses asserted by ARTREC. The other problem lies in the words "yes its" and "no its". Especially in the case of "no its", ARTREC is given problems because these two words are so rapidly spoken. ARTREC will sometimes only suggest a single hypothesis to explain the motions here, either "no" or "its". It should be noted that ARTREC has a further hypothesis for a single word "noits" which is sometimes used in such a case. In the case of "yes its", these two words are usually more clearly spoken and ARTREC has less problems.
Figure 31: Best explanation of pellet data in terms of syllables
5.2 Emphasis Detection

In the last section, I showed ARTREC running through half of a typical example. What was not shown was ARTREC's prosodic recognition. ARTREC will only attempt to locate corrective emphasis. This should appear on only a single word (if any). Here, I will show, briefly, the emphasis detection. This will be from a different example as the previous one (the previous example had a positive reply as in "yes, its nine five nine pine street" and there is no emphasis in this case).

Once ARTREC has inferred the syllables to explain the motions, ARTREC reexamines the mandible motion. If the word "no" was found, ARTREC examines the syllable locations for each word following "no". If most of these words have been found, then ARTREC will seek out corrective emphasis.

Corrective emphasis (and most types of emphasis) can be recognized by exaggerated mandible motion. Therefore, ARTREC examines the depth of each syllable's valley (in the mandible channel). However, ARTREC takes into account the factor of articulatory declination. ARTREC examines the valleys, and plots a shallow sloping line across the valleys. The valley which penetrates into the sloping line the most is considered to be the location of the emphasis. ARTREC explains this exaggerated motion as the syllable which received the corrective emphasis.

Figure 32 shows the corrective emphasis on a typical example. In this example, the words are "nine FIVE nine pine street". You can see that ARTREC has found the valley which most penetrates the sloping line. This second valley represents the word "five".
In the second half of the utterance (after the word "no"), ARTREC examines mandible motion (jaw motion). Here, you can see a sloping line representing the effects of Articulatory Declination. The valley which penetrates the sloping line the most is considered the emphasized word. In this case, the second word in the sequence penetrates the most.

Figure 32: Emphasis Detection

5.3 Performance and Complexity

ARTREC is capable of coming to quick decisions using the Peirce algorithm. In this example, I did not show each of Peirce's problem-solving steps. A typical case will require some small number of iterations through Peirce's main problem solving loop (i.e. the loop which first looks for confirmed hypotheses, then essentials, then clear
bests and then weak bests).

For event identification, each pellet is examined one time and ARTREC identifies each peak and valley. Next, ARTREC considers each type of gesture (such as alveolar closure, labial closure, tongue retroflexion) and scans the list of events for close "template" matches. Each match (whether very close or remote) is noted as a hypothesis. Those hypotheses which do not match well are immediately ruled out. All other hypotheses are scored and explanatory knowledge is generated. In addition, noise hypotheses are suggested at the same time. The first abducer generates an explanation of events in terms of gestures in one to two passes. The only form of hypothesis interaction is the incompatibilities between noise hypotheses and gesture hypotheses. ARTREC will come to a very quick explanation for this level.

Next, gestures are used to cue syllable and noise hypotheses. After scoring, each hypothesis is assigned its explanatory coverage (again by examining the gestures). Then, the second abducer generates an explanation. Here, the abducer may run through its main loop many times (potentially as many times as there are hypotheses) but the typical case requires only 4 or 5 passes through the main loop.

If top down processing is used, then expectations are generated at this point (based on the accepted syllable hypotheses) and ARTREC recompiles the lower-level composite followed by the higher-level composite. Typically, the most ARTREC will run through the main Peirce loop would be 10 times (requiring between two or four minutes on a SPARCstation SSl).
5.4 Conclusion

In this chapter, I have given a pictorial look at the processing of the ARTREC system. I have shown a typical sample case and described the complexity that ARTREC faces in terms of the numbers of findings and hypotheses at each level. By having two levels of abduction, the total number of hypotheses and findings is limited (as opposed to having syllables directly explain events which would have generated many more hypotheses and a greater overall problem complexity). I have shown that ARTREC can quickly make decisions in order to come to a conclusion.

In the next chapter, I will discuss four experiments undertaken by ARTREC and the experimental results.
CHAPTER VI

Experimental Results

6.1 Introduction

In this chapter, I will discuss four experiments conducted using the ARTREC system. I will give experimental results and offer some analyses about the results. I will also discuss how ARTREC fared on emphasis identification. I will then turn to some proposed enhancements for ARTREC.

6.2 ARTREC Experiments and Terminology

The initial implementation of ARTREC was a single-layer abduction system. I ran all five sets of speaker data. I was not satisfied with the accuracy of this version of ARTREC so I then added another abducer in hopes of improving ARTREC's accuracy. This second abducer was placed in the lower level to improve the quality of articulatory gestures used as findings for the second level. To further test layered abduction, and out of curiosity, I altered the newer version of the system by making it harder for a hypothesis to be accepted. This "pessimistic" ARTREC was created by altering global threshold values in the Peirce tool. I had hoped to achieve a higher accuracy at the expense of explaining less of the data. Finally, I added top-down processing to the pessimistic version in hopes of achieving both high accuracy and more
coverage than the pessimistic version. The results of this experiment demonstrate a fuller layered abduction process combining several forms of knowledge using both data-driven and hypothesis-driven processing.

The results will be discussed in the following terms: how many total words were involved, how many words ARTREC committed to, how many words ARTREC correctly identified, ARTREC's coverage and ARTREC's accuracy. The "committed to" value indicates how many words ARTREC actually came to a conclusion about. Remember that in Peirce, hard decisions are left open. If a Peirce abducer cannot clearly pick a hypothesis to explain a datum, that datum is left unexplained. In the best case, the datum would be explained by more processing (such as when hypotheses cause other hypotheses to be ruled out due to incompatibilities, and thus removing the hard decision). In the worst case, the datum is left unexplained.

Coverage is the number of "committed to" words divided by the number of total words. This percentage measures the amount of data that ARTREC explained. Accuracy is the number of words ARTREC identified correctly out of the words it "committed to". Each experiment's table will contain the Total Words, Committed, Correct, Coverage, and Accuracy. Committed is the number of words ARTREC committed to, and Correct is the number of words that ARTREC identified correctly. Coverage and Accuracy will be given in terms of percentages.

Later in this chapter we will look at the accuracy of the emphasis identification. Since emphasis identification was not dependent on the type of experiment, we will only discuss emphasis identification for experiment 2. This experiment was the only
one for which we accumulated emphasis accuracy statistics.

Table 6 shows the 5 sets of speaker data, how many utterances were used for that data set, and the number of words contained within all of the utterances. There were between 80 and 82 total utterances per data set, but some of these were discarded due to tracking errors in the pellet data or signal processing errors which occurred during the preprocessing phase. Each utterance is an entire question/answer pair which consists of 14 words (e.g. Is it five five nine pine street? No, its NINE five nine pine street). However, some of these words are not within the pellet data as each pellet data record only contains up to 5 seconds and some of the words may not have been recorded.

Table 6: Data Set Description

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Number of Utterances</th>
<th>Number of Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFJW5p</td>
<td>80</td>
<td>933</td>
</tr>
<tr>
<td>OFJW6pp</td>
<td>80</td>
<td>975</td>
</tr>
<tr>
<td>OFJW7p</td>
<td>76</td>
<td>877</td>
</tr>
<tr>
<td>OFJW11pp</td>
<td>78</td>
<td>961</td>
</tr>
<tr>
<td>OFJW12ppp</td>
<td>77</td>
<td>966</td>
</tr>
</tbody>
</table>

6.2.1 Experiment 1: Single Abduction

The initial version of ARTREC had a single abducer which would explain articulatory gestures in terms of syllables but would not attempt to explain the articulatory events at all. That is, the lower level of ARTREC was simple pattern matching so that all hypothesized articulatory gestures were considered as true (no matter of how poorly rated they were or if they were explanatorily superfluous). A single abducer would
then explain all of the gestures in terms of syllable and noise hypotheses. This differs from the current version which has two abducers, one to explain events in terms of gestures and one to explain gestures in terms of syllables.

Gestures were hypothesized based on cues and scored based on pattern matching. Syllable hypotheses were used to explain all of the gestures, and noise hypotheses were also used to explain those gestures which did not score very high based on pattern matching. Because there was no way to restrict the cueing process, we found that a much greater number of gestures were being suggested than were needed. For example, if an event cued three different gesture hypotheses, all three hypotheses were considered as true and required explaining. This is an unsophisticated approach to the problem. Abduction, among other things, is a means of pruning down the number of hypotheses one has to work with. The result of an abduction should be a small set of believed hypotheses rather than a large set of potential hypotheses. Without abduction, we are either stuck with the original set of generated hypotheses or must find some other means of pruning away some of the hypotheses. The former was the case in the first ARTREC.

This version of ARTREC made many mistakes in attempting to explain gestures which did not really exist. While noise hypotheses at the syllable level were able to explain some of the poorly rated gestures, ARTREC usually had problems when noise hypotheses were not suggested or when gestures were on the borderline between "real" and "noise". The incompatibility handling also caused problems because ARTREC would often make mistakes with the initial islands of certainty which would lead to
ruling out possibly good hypotheses.

The flow of processing was to generate gesture "data" from the motion data (events) and then explain the gesture "data" in terms of syllables. No noise hypotheses were generated from the motion data which meant that erroneous gestures still needed to be explained by syllable hypotheses or noise hypotheses. The syllable level was essentially the same as the current ARTREC's syllable level except that there were many more gestures to explain than in the two abducer system. There was no top-down processing used in this experiment.

The results from Experiment 1 are indicated in table 7 which shows that ARTREC-1 was able to achieve 89.47% coverage and 89.61% accuracy.

Table 7: Experiment 1 Results: Single Abducer Version

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Total Words</th>
<th>Committed</th>
<th>Correct</th>
<th>Coverage</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFJW5p</td>
<td>933</td>
<td>798</td>
<td>709</td>
<td>85.53%</td>
<td>88.85%</td>
</tr>
<tr>
<td>OFJW6pp</td>
<td>975</td>
<td>874</td>
<td>802</td>
<td>89.64%</td>
<td>91.76%</td>
</tr>
<tr>
<td>OFJW7p</td>
<td>877</td>
<td>810</td>
<td>740</td>
<td>92.36%</td>
<td>92.36%</td>
</tr>
<tr>
<td>OFJW11pp</td>
<td>961</td>
<td>851</td>
<td>768</td>
<td>88.55%</td>
<td>90.25%</td>
</tr>
<tr>
<td>OFJW12ppp</td>
<td>966</td>
<td>892</td>
<td>767</td>
<td>92.34%</td>
<td>85.99%</td>
</tr>
<tr>
<td>Overall</td>
<td>4722</td>
<td>4225</td>
<td>3786</td>
<td>89.47%</td>
<td>89.61%</td>
</tr>
</tbody>
</table>

The initial results were bothersome. I had anticipated high results. ARTREC did achieve a reasonable amount of coverage, but a 90% accuracy is not satisfactory considering the limited lexicon. We can explain a good deal of the inaccuracy because of the need to explain the gestures, many of which were not even present. These results led us to construct a second abducer for the lower level rather than just using hypothesization for the lower level.
6.2.2 Experiment 2: Bottom-Up Layered Abduction

In the updated system, I added a lower level abducer for forming a composite hypothesis of articulatory gestures in order to explain the events. Previously, any gesture that was hypothesized was automatically accepted as true. I also added noise hypotheses to the lower level so that the events in the pellet data could be explained either as articulatory gestures or noise (random or unintentional motions). This version of ARTREC would hypothesize the causes for the pellet motions, score these hypotheses and then construct a composite explanation. The remainder of the system is the same as in Experiment 1. There was no top-down processing used in this experiment.

This version of ARTREC performed better, indicating that abduction is a very useful means of pruning away unneeded hypotheses. It also shows that, where cueing is overly optimistic (that is, when too many hypotheses are cued) some other process is needed to decide which hypotheses are relevant. Abduction is perfectly suitable for this.

The results of ARTREC-2 are shown in table 8. Here, we see that overall, ARTREC-2 achieved 84.67% coverage and 94.87% accuracy.

Comparing the results of experiment 1 and experiment 2 shows slightly less coverage (5%) but a slight improvement in accuracy (5%). A 95% accuracy is respectable for the speech recognition task.
Table 8: Experiment 2 Results: Layered Abduction

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Total Words</th>
<th>Committed</th>
<th>Correct</th>
<th>Coverage</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFJW5p</td>
<td>933</td>
<td>744</td>
<td>697</td>
<td>79.74%</td>
<td>93.68%</td>
</tr>
<tr>
<td>OFJW6pp</td>
<td>975</td>
<td>851</td>
<td>813</td>
<td>87.28%</td>
<td>95.53%</td>
</tr>
<tr>
<td>OFJW7p</td>
<td>877</td>
<td>745</td>
<td>712</td>
<td>84.95%</td>
<td>95.57%</td>
</tr>
<tr>
<td>OFJW11pp</td>
<td>961</td>
<td>816</td>
<td>785</td>
<td>84.91%</td>
<td>96.20%</td>
</tr>
<tr>
<td>OFJW12ppp</td>
<td>966</td>
<td>842</td>
<td>786</td>
<td>87.16%</td>
<td>93.35%</td>
</tr>
<tr>
<td>Overall</td>
<td>4722</td>
<td>3998</td>
<td>3793</td>
<td>84.67%</td>
<td>94.87%</td>
</tr>
</tbody>
</table>

6.2.3 Experiment 3: Pessimistic Bottom-Up Layered Abduction

In this version of ARTREC, I used the same system as in Experiment 2 in that, ARTREC had two abducers in a bottom-up flow of processing. However, an attempt was made to get better accuracy by sacrificing the amount of explanatory coverage. I made it harder for ARTREC to accept hypotheses by requiring that any accepted hypotheses be very convincing. Again, there was no top-down processing used in this experiment.

Three changes made. First, essentials were accepted only if they had a minimum confidence of likely. This differs from the previous versions in which essentials could be rated as low as neutral. Second, it is more difficult to be considered a clear best hypotheses. In the earlier ARTREC, a clear best was any hypothesis which was rated at least somewhat-likely and surpassed alternative explainers by two units of confidence (i.e. if the clear best was likely, all competitors had to be at best neutral). In this experiment, clear bests had to be rated at least likely and surpass alternatives by three units or more. I also made it harder for ARTREC to accept a weak best. In
this experiment, weak bests must also have a value of likely or greater. Table 9 reflects the changes made between the “pessimistic” versions of ARTREC (both experiments 3 and 4 used the pessimistic version) and the “normal” version of ARTREC (from experiments 1 and 2).

Table 9: Pessimistic versus Normal Abduction Acceptance Criteria

<table>
<thead>
<tr>
<th>Acceptance Criteria</th>
<th>Normal Mode</th>
<th>Pessimistic Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Essentials</td>
<td>Neutral or better</td>
<td>Likely or better</td>
</tr>
<tr>
<td>Clear Best Threshold</td>
<td>Somewhat Likely or better</td>
<td>Likely or better</td>
</tr>
<tr>
<td>Clear Best Surpassing</td>
<td>2 units</td>
<td>3 units</td>
</tr>
<tr>
<td>Weak Best Threshold</td>
<td>Neutral or better</td>
<td>Likely or better</td>
</tr>
</tbody>
</table>

The results for this experiment are shown in table 10 where we can see that ARTREC-3 achieved a coverage of 66.26% with 96.07% accuracy.

Table 10: Experiment 3 Results: Pessimistic Bottom-Up Layered Abducer

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Total Words</th>
<th>Committed</th>
<th>Correct</th>
<th>Coverage</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFJW5p</td>
<td>933</td>
<td>603</td>
<td>577</td>
<td>64.63%</td>
<td>95.69%</td>
</tr>
<tr>
<td>OFJW6pp</td>
<td>975</td>
<td>629</td>
<td>606</td>
<td>64.51%</td>
<td>96.34%</td>
</tr>
<tr>
<td>OFJW7p</td>
<td>887</td>
<td>505</td>
<td>488</td>
<td>56.93%</td>
<td>96.63%</td>
</tr>
<tr>
<td>OFJW11pp</td>
<td>961</td>
<td>699</td>
<td>671</td>
<td>72.74%</td>
<td>95.99%</td>
</tr>
<tr>
<td>OFJW12ppp</td>
<td>966</td>
<td>693</td>
<td>664</td>
<td>71.74%</td>
<td>92.93%</td>
</tr>
<tr>
<td>Totals</td>
<td>4722</td>
<td>3129</td>
<td>3006</td>
<td>66.26%</td>
<td>96.07%</td>
</tr>
</tbody>
</table>

A slight improvement in accuracy was achieved. This tells us that most of ARTREC’s errors came from occurrences where ARTREC is genuinely confused about the data, and not from selecting between hypotheses in the cases of “hard decisions”. A 4% error is very slight, yet all of these errors occur because ARTREC misinterpreted the data. That is, we know that ARTREC did not err in this experiment because of
choosing the wrong hypothesis during hard decisions. With the pessimistic control, ARTREC would not even attempt to make a hard decision. Therefore, each error occurred when hypotheses’ scores or explanatory coverage were clearly incorrect.

As can be seen, the coverage of experiment 3 is only about two-thirds of the total data. ARTREC-3 would not commit to any hypothesis which had reasonable competitors.

6.2.4 Experiment 4: Full Layered Abduction

For our last experiment, I expanded the system to include top-down processing. This processing was in the form of higher-level expectation. An expectation can be used to alter the plausibilities of lower level hypotheses. ARTREC uses the “certainty” of the accepted hypothesis as leverage. In this case, an expectation will be able to increase or decrease lower level hypotheses’ plausibilities. The result of this is that some hypotheses will become clear or weak bests because their plausibilities were raised or because competing hypotheses had their plausibilities lowered.

For Experiment 4, I used the same system as Experiment 3, but added top-down expectations and gave ARTREC an altered flow of processing to include 2 bottom-up passes. The top-down expectations were in the following form. If a syllable was concluded after the first bottom-up pass, expectations were created for the gestures which should appear in that syllable. For example, a “nine” syllable would expect the gestures:

alveolar closure tongue low final alveolar closure
If these gestures occurred at the approximate time location (where the syllable hypothesis would expect each gesture to occur) then the gestures' scores would be raised. After altering gestures’ scores, ARTREC reran both levels. The amount that a score was raised depended on how confident ARTREC was in accepting the syllable hypothesis. More confident hypotheses were able to alter gesture hypothesis scores more dramatically than less confident hypotheses.

I used ARTREC-3 as a basis so that ARTREC-4 would be very pessimistic about accepting hypotheses and build upon islands of clear certainty, propagating higher level expectations to the middle level. I had hoped that experiment 4 would yield high accuracy (as in experiment 3) but also yield high coverage (as in experiment 2). The results of experiment 4 are shown in table 11. As we can see, the coverage stayed almost the same as in experiment 3 with a coverage of 66.20% and the accuracy increased a slight amount to 96.80%.

Table 11: Experiment 4 Results: Full Layered Abducer

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Total Words</th>
<th>Committed</th>
<th>Correct</th>
<th>Coverage</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFJW5p</td>
<td>933</td>
<td>609</td>
<td>588</td>
<td>65.27%</td>
<td>96.55%</td>
</tr>
<tr>
<td>OFJW6pp</td>
<td>975</td>
<td>629</td>
<td>617</td>
<td>64.51%</td>
<td>98.09%</td>
</tr>
<tr>
<td>OFJW7p</td>
<td>887</td>
<td>501</td>
<td>490</td>
<td>56.48%</td>
<td>97.80%</td>
</tr>
<tr>
<td>OFJW11pp</td>
<td>961</td>
<td>710</td>
<td>681</td>
<td>73.88%</td>
<td>95.92%</td>
</tr>
<tr>
<td>OFJW12ppp</td>
<td>966</td>
<td>677</td>
<td>650</td>
<td>70.08%</td>
<td>96.01%</td>
</tr>
<tr>
<td>Overall</td>
<td>4722</td>
<td>3126</td>
<td>3026</td>
<td>66.20%</td>
<td>96.80%</td>
</tr>
</tbody>
</table>

To my surprise, the results of experiment 4 were not what I had expected. I had hoped to see further coverage with the same accuracy. Instead, the coverage
remained almost the same (very slight drop in coverage) while accuracy increased slightly. In two of the five data sets, coverage actually went down while in the other three, coverage increased slightly. There was a greater accuracy.

For this experiment, ARTREC would make use of islands of certainty found at the syllable level to help clear up the gesture level. However, this clarification would have two results. First, it would make some hard decisions easier (that is, where ARTREC could not decide between two or more hypotheses from experiment 3, ARTREC could now make this decision). The reason that hard decisions were made easier was because the top-down expectations increased the plausibility of some hypotheses. However, this did not occur very often (i.e. the top-down processing did not help clear up too many hard decisions).

The reason for the increased accuracy was also due to hard decisions. But in this case, previously wrong hypotheses which were accepted were not accepted in this case because the top-down expectations increased some hypotheses scores such that these hypotheses created hard decisions. That is, previously incorrect hypotheses now had plausible alternatives. Thus, ARTREC did not accept the wrong hypotheses at the gesture level, but left the decisions unresolved.

What we can see from this experiment is that, ARTREC gained some coverage because the certainty helped to remove some hard decisions, but also, the certainty helped to create new hard decisions. The overall result was a slight increase in accuracy and a very slight decrease in coverage.

I was quite surprised by the results of this experiment. I feel that I was on the
right path to increased coverage, however the little bit of top-down processing was not enough to fully alter the results. Further forms of top-down processing would be required to allow this pessimistic ARTREC to achieve both high accuracy and high coverage.

6.3 Discussion of ARTREC Results

As I had hoped, ARTREC was able to achieve very high results in articulatory recognition. I was impressed with the high accuracies shown in experiments 2, 3 and 4. I had high hopes for this system, and I was not disappointed. Table 12 compares the four experiments in terms of the flow of processing, number of abducters and the "acceptance" criteria for hypotheses (i.e. pessimistic or normal). Table 13 shows the overall experimental results of ARTREC.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Flow of Processing</th>
<th>Number of Abducters</th>
<th>Acceptance Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experiment 1</td>
<td>Bottom-Up Only</td>
<td>1</td>
<td>Normal</td>
</tr>
<tr>
<td>Experiment 2</td>
<td>Bottom-Up Only</td>
<td>2</td>
<td>Normal</td>
</tr>
<tr>
<td>Experiment 3</td>
<td>Bottom-Up Only</td>
<td>2</td>
<td>Pessimistic</td>
</tr>
<tr>
<td>Experiment 4</td>
<td>Bottom-Up and Top-Down</td>
<td>2</td>
<td>Pessimistic</td>
</tr>
</tbody>
</table>

There was little surprise in our experimental results. The single abduction system suffered where the layered abduction systems achieved higher accuracy. The pessimistic experiment was to demonstrate that there is a clear tradeoff between accuracy and coverage and this was the result of experiment 3. The full abduction system surpassed the bottom-up layered abduction system in terms of accuracy, and
Table 13: Overall Experimental Results

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Coverage</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experiment 1</td>
<td>89.47</td>
<td>89.61</td>
</tr>
<tr>
<td>Experiment 2</td>
<td>84.67</td>
<td>94.87</td>
</tr>
<tr>
<td>Experiment 3</td>
<td>66.26</td>
<td>96.07</td>
</tr>
<tr>
<td>Experiment 4</td>
<td>66.20</td>
<td>96.80</td>
</tr>
</tbody>
</table>

the pessimistic abduction system in terms of coverage. If there were any surprises, it lay only in the slight improvement in accuracy between experiments 2 and 3, and the very slight drop in coverage between experiments 3 and 4.

While the accuracy here is impressive, I must remind the reader that the lexicon size is very limited. A nine word lexicon is only a minute fraction of the English language. However, to offset the small lexicon, ARTREC did not have a full set of knowledge or input to work with. In fact, ARTREC was severely handicapped as it only had access to 4 or 5 points of the vocal tract. When just considering these points, many of the words in the lexicon begin to look very similar. Had ARTREC been given data for voicing, nasality, frequencies, formants, and so forth, and knowledge of syntax and semantics, the task would become very trivial. But the task was not trivial due to the limited form of input. ARTREC achieved accuracies very respectable for speech recognition. In the past, very few systems have achieved over 95% word accuracy.

6.4 Prosodic Emphasis

Another dimension to investigate is how ARTREC performed on prosodic identification. A very simple algorithm was used in order to explain exaggerated jaw motion by
hypothesizing corrective emphasis. This algorithm looked for the jaw opening which seemed most extreme. ARTREC took into account the effects of articulatory declination. Declination is the phenomenon whereby a speaker will have less clarity in (or fullness of) pronunciation as time goes on. This is presumably due to physical factors such as tiredness and boredom. A result of articulatory declination is that an exaggerated motion towards the end of an utterance will look much like a non-exaggerated motion towards the middle or beginning of an utterance. Thus, ARTREC's algorithm for detecting exaggerated motion must take into account the problem of declination.

I implemented the declination as a sloping line. ARTREC compared all of the valleys in the mandible incisor pellet data to this sloping line. The valley that penetrated the sloping line the most was assumed to be the syllable which contained the most exaggeration. This in turn was explained as emphasis in sentences where negative responses were given. In sentences where positive responses were given, no corrective emphasis was given, and so any exaggeration was thought to be unintentional.

This simplistic algorithm worked fairly well on some of the speakers. However, I found that it did not work on all of the speakers equally. The speakers who tended to speak unclearly had much more problematical "exaggeration" in their emphasis. That is, it was much less certain which word was being emphasized. As a result, ARTREC did not fare very well in identifying the emphasized word.

The results of the prosodic identification can be seen in table 14. It is shown for each data set independently the number of utterances in which ARTREC looked for emphasis and the number of those utterances where ARTREC correctly found the
emphasized word. ARTREC did not seek out emphasis in two conditions. First, the word "no" was not found. In such a case, ARTREC assumed that the answer was a positive one and no corrective emphasis existed\(^1\). The other condition occurred if ARTREC did not commit to an answer for some of the words of the utterance. Because ARTREC was unsure of some of these words, ARTREC would not commit to an identification of the emphasized word. In any other condition, ARTREC would choose a word in the utterance to be the emphasized one.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Utterances</th>
<th>Total Emphasized Utterances</th>
<th>Correct Identification</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFJW5p</td>
<td>80</td>
<td>54</td>
<td>39</td>
</tr>
<tr>
<td>OFJW6pp</td>
<td>80</td>
<td>44</td>
<td>29</td>
</tr>
<tr>
<td>OFJW7p</td>
<td>76</td>
<td>31</td>
<td>31</td>
</tr>
<tr>
<td>OFJW11pp</td>
<td>78</td>
<td>38</td>
<td>35</td>
</tr>
<tr>
<td>OFJW12pp</td>
<td>77</td>
<td>35</td>
<td>32</td>
</tr>
<tr>
<td>Totals</td>
<td>391</td>
<td>202</td>
<td>166</td>
</tr>
</tbody>
</table>

As can be seen in the table, ARTREC varied greatly in detecting the emphasized word. For case set OFJW7p, a 100% accuracy was achieved, however, in case set OFJW6pp, a 66% accuracy was achieved. Overall, ARTREC correctly identified 82.18% of the emphasized words where the conditions described above were fulfilled. This is a far cry from "accurate" emphasis detection. We feel that, with some revision, ARTREC can locate most all of the emphasized words; however we haven't had a chance to implement any revisions. The simplistic approach taken here is not sufficient.

\(^1\)If ARTREC failed to detect the word "no" in the utterance, then ARTREC assumes that no emphasis occurred. However, this may have been a mistake as ARTREC may have erred in not finding the word "no". This strategy was probably a mistake on my part as the system builder, but by the time I realized this problem, I had gone on to other concerns.
alone, but is a beginning.

6.5 Conclusion

I have tested ARTREC on a set of approximately 400 utterances from five different male and female speakers. In order to demonstrate the usefulness of abduction, we have shown the variations of accuracy and coverage between four sets of experiments. These experiments have shown that abduction is a useful means of pruning hypotheses which were optimistically generated. I also hope to have shown the tradeoff between accuracy and coverage, and that Peirce-built systems can range between the extremes of this tradeoff by simple changes to global threshold values. I have also shown how top-down processing can aid accuracy by building on islands of certainty, and although the coverage decreased in two speakers' cases, the other three speakers had modest increases in coverage.

As for the results, ARTREC has been shown to be a very capable articulatory recognition system. In experiment 4, ARTREC achieved high accuracy of 96.8%. In experiments 2 and 3, I have shown that ARTREC without top-down processing can still achieve high accuracy. ARTREC is capable of detecting the prosodic effect of corrective emphasis, although for two of the speakers, the accuracy was not very high. If more effort had been made to create an algorithm which relied on more features, we feel that ARTREC could have achieved a much higher accuracy.

While ARTREC is very limited in scope, I will argue in the next chapter that ARTREC is scalable to a much larger lexicon size and capable of handling many more types of gestures than is currently implemented. In the next chapter I will also
discuss how ARTREC can be used in an acoustic speech recognition system as well as solving the articulatory recognition problem.
CHAPTER VII

Implications of this work

7.1 Introduction

I have described a system for articulatory recognition, ARTREC. I will consider how ARTREC might be used to aid speech recognition. I will argue why I believe that ARTREC is scalable, and I will indicate uses of ARTREC in other research.

I will discuss the usage of Peirce or a Peirce-like abductive strategy as an enhancement for Hidden Markov Model speech recognition systems, and explain how such a combination can bypass the problems associated with HMMs.

Layered abduction, and the Peirce strategy in particular, have far reaching possibilities. I will conclude with a discussion of the potential uses of Peirce to solve a variety of layered abduction problems both in perceptual areas and non-perceptual areas.

7.2 Uses of Articulation in Speech Recognition

Let us first turn to articulation and how articulation might be useful in future speech recognition work. I have shown in ARTREC that articulatory recognition is a feasible task. What I have not shown is how this task can be used to aid speech recognition.
Lexical Recognition Module

Phonetic Recognition Module

Acoustic Recognition Module

Articulatory Recognition Module

Auditory Analysis Module

Acoustic Signal for some English Utterance

Figure 33: Speech Recognition Task Decomposition Using Articulation
As can be seen in figure 33, articulatory recognition can be thought of as one step in the overall speech recognition problem. In [Liberman and Mattingly, 1985], it has been proposed that articulatory representation is necessary in the task of speech recognition. In fact, by separating knowledge of acoustic phonetics and articulatory phonetics, a system can increase its ability to cope with the problems introduced by articulatory dependencies and prosody. Acoustic phonetics can be thought of as deriving phonetic units (such as phonemes or syllables) from the acoustic signal by using auditory features. Articulatory phonetics can be thought of as deriving primitive articulatory units (either articulatory features such as closures, openings, tongue movements and so forth, or phonetic units such as phonemes or syllables) from the acoustic signal using knowledge of articulation. Making this distinction can allow a speech recognition system to reason about both the features in the acoustic signal and the interactions involved in articulation and the alterations that such interactions will have on the acoustic signal.

Some of the major stumbling blocks of acoustic speech recognition have revolved around inadequate or no explicit models of articulation, problems with identifying place of articulation, and problems with the effects that prosody has on the acoustic signal. By separating the two types of knowledge, acoustics and articulation, a speech system can use separate features of articulation rather than concatenating segments from auditory features. By concatenating segments, a system will be forced into a linear view of speech production (and thus, a linear view of speech recognition) which is seems flawed [Fowler, 1986].
7.2.1 How ARTREC can be used to aid Acoustic Speech Recognition

ARTREC, as it exists now, is not useful for acoustic speech recognition. This is because ARTREC expects Microbeam pellet data as its input. To use ARTREC as a component within an acoustic speech recognition system, some changes are needed. Mainly, additional knowledge must be brought into ARTREC so that ARTREC can use acoustic data in addition to articulatory data. Figure 34 shows a detailed view of the articulatory layer which I propose for a large speech recognition system.

Part of ARTREC’s task is to map articulatory gestures to syllables. This component of ARTREC can be used in a speech recognition system with minimal changes. If we presuppose that we can infer articulatory gestures directly from an acoustic signal, then the higher level of ARTREC can be used without adaption. Thus, ARTREC can be used to play a specific role in speech recognition, that of mapping directly from articulatory gestures to syllables.

However, what if articulatory gestures are not directly available (by inference) from the acoustic signal? Obviously, Microbeam pellet data will be unavailable in real-world situations. Will articulatory recognition be of any use?

It is possible now to infer some articulatory information from the acoustic signal. This can be done by using features detected in formant transition locations, burst analyses, spectral profiles and other auditory information. For example, one method for finding the place of articulation can be done by back extrapolation of the formant tracks from the vowel regions to the consonantal burst. Formant locations are de-
Figure 34: Articulatory Layer in Speech Recognition

rived from the acoustic signal by signal processing and auditory analysis. By using articulatory knowledge, we can make certain inferences about the tongue location at the time of the consonantal utterance, which can inform us of place of articulation.

By using a layered abduction strategy as shown in ARTREC, some initial islands of certainty can be created from auditory analysis. These auditory hypotheses can then
be used as leverage for further problem solving. First, these auditory hypotheses can be used to infer certain hypotheses about articulation (for instance, by using formant locations as mentioned above). Then, these articulatory hypotheses can be used as leverage for further articulatory and auditory hypotheses in order to infer phonetic units. ARTREC is capable of generating certain types of expectations of articulatory hypotheses based on vocal tract motions. These expectations can be for leverage in coming to more complete conclusions at these lower levels.

In this way, ARTREC becomes a module of an overall speech recognition system in which ARTREC is used to generate articulatory hypotheses from auditory data, and auditory and articulatory expectations from articulatory hypotheses. As ARTREC builds on islands of certainty, the accepted articulatory hypotheses are used to generate syllables and words at higher levels. ARTREC can avoid the problems faced by many past speech recognition systems which have had to make use of coarticulation rules and other such ad hoc techniques because ARTREC can explicitly weigh the consequences of various articulatory interactions.

While this is all speculation, I hope that the past results of ARTREC are convincing in three ways. First, the strategy for layered abduction is quite capable of pulling together disparate sources of knowledge in order to make clear decisions. Second, that top-down expectations are well suited for the auditory/articulatory mapping. Even though auditory to articulatory mapping is not well understood, islands of certainty created at both levels will allow for progress in the speech recognition task. Third, that articulatory recognition is a clearly feasible task (as shown by the results
of ARTREC). I will now discuss several uses for articulatory knowledge in a speech recognition system.

### 7.2.2 Making Speech Non-Linear

By allowing two separate representations for "phonetics" and "phonology", a system will be able to take advantage of both forms of knowledge. Past systems which have only modeled phonetics have used a linear theory of speech production. That is, in producing speech, one utters one unit (a phoneme) followed by another unit, followed by another. Speech production is thought of as concatenating units together. This view of speech production seems incorrect [Fujimura, 1992, Fujimura et al., 1991, Browman and Goldstein, 1988] and gives rise to several problems including the need for coarticulation rules and word boundary rules. These two types of rules have been used in the past to get around problems induced by appealing to linear phonology.

By explicitly modeling articulation and articulatory features (or primitives), we can get past these problems. Rather than using phonetic primitives and segmental concatenation, a system can use parallel articulatory features which can be used to make sense of articulatory dependencies. Articulatory features are fairly independent, and can occur in parallel. Thus, an articulatory model is a much closer representation of the actual speech production mechanism than a model of phonetics.

### 7.2.3 Place of Articulation

Place of articulation is the location within the vocal tract where some constriction occurred. This constriction will help "shape" the acoustic sound for consonants. For
instance, labial closure (or lip closure) will produce the labial stop consonants /p/ and /b/. Alveolar closure and partial closure will produce the alveolar consonants such as /t/, /d/, /s/, /θ/, /n/ and so forth.

Finding place of articulation from the acoustic signal is extremely difficult. The acoustic signal does not contain features that make place of articulation easily recognizable. Past systems have gotten around this problem by either doing without place of articulation (as in Hearsay) or by using some spectral profiles without specific features, and using these profiles as matching templates (as in the case of Harpy) or by training models so that place of articulation does not necessarily need to be determined (Byblos, Sphinx).

Identifying place of articulation is very important. As seen in Hearsay, finding manner of articulation is not sufficient to determine the phonetic units. Hearsay proposed too many phonetic units because of the ambiguities created by manner of articulation without place of articulation.

There are some features which are available to aid in identifying the place of articulation (such as formant locations in the transitional period between consonants and vowels, spectral tilt and spectral burst profiles). However, as indicated in [Fox and Josephson, 1991] and other places, such features do not work reliably. Articulatory context and non-linguistic effects can alter the expected values of such features. With articulatory knowledge directly available, a speech recognition system can take rough estimates from using such features, and appeal to the articulatory model to make further suggestions, inferences and predictions about what findings should appear.
7.2.4 Using Prosodic Information

Another way that ARTREC can aid speech recognition is in reasoning about prosody. At the moment, ARTREC uses exaggerated motions in the mandible incisor to determine which word is being emphasized. This is only the first type of prosodic information that could be used in ARTREC. More sophisticated algorithms should be able to detect other types of prosody occurring because of excitement, boredom and other non-linguistic effects. To derive prosody from the acoustic signal is a very difficult problem [Waibel, 1990]. However, clues can be found in the acoustic signal. These clues can be passed along to the articulatory reasoner which can make predictions about how these prosodic types can affect the acoustic signal.

For example, emphasized words will be longer and more clearly uttered. If articulatory knowledge can be used to determine that a word is being emphasized, then factors such as lengthening and clarity can be taken into account. This would lead to a better chance that the acoustic reasoner can correctly identify the word. Similarly, if articulatory reasoning can infer that a sequence of words is being spoken rapidly, the acoustic reasoner can use this knowledge. Vowels would be reduced. There are problems in determining if a vowel is reduced or non-existent. By using knowledge of prosody, a system can know when vowels might be reduced or non-existent.

Because prosody has been a problem in the past, a clear means of dealing with prosody requires first detecting where these non-linguistic effects come into play. Once this is detected, a system can then use representations of how these effects alter the acoustic signal as additional knowledge.
7.2.5 Conclusions of Using Articulatory Knowledge

Since it is unlikely that pellet data will be widely available for future speech recognition systems, we must find a new means of using articulatory knowledge of a system like ARTREC. Mapping directly from auditory features to articulatory units has not been tried. It has been proposed by [Liberman and Mattingly, 1985, Glass and Zue, 1988, Browman and Goldstein, 1986] and others that such a mapping is not only possible but necessary. The knowledge used by ARTREC is a first approximation of the knowledge needed to get around problems of co-articulation, prosodic effects, determining place of articulation, and so forth.

7.3 Scalibility of ARTREC

It is probably apparent to the reader that ARTREC solves only a small problem. Because only 9 words (syllables) and some 15 gestures are used in ARTREC, there might be concern over whether ARTREC can scale upwards to a larger lexicon. The high performance of ARTREC might be due to very limited number of hypotheses. However, I am sure that ARTREC will scale so that larger lexicons and larger numbers of gestures will not diminish ARTREC’s performance. Here are several reasons which indicate that ARTREC is scalable.

- An efficient means of generating a small number of hypotheses is possible by using hierarchical classification. This ensures that most of the potential hypotheses are ruled out before abduction even begins, which in turn reduces the complexity of the task for ARTREC. Further, by using hierarchical clas-
sification, hypotheses can be generated at different levels of specificity so that explainers can be, at first, given in a general form and refined later if necessary. This approach (of using hypotheses at different levels of specificity) has worked in medical diagnostic cases (for instance, see [Punch, 1989]).

- Because of the use of incompatibility handling and other forms of hypotheses interactions, ARTREC can handle making decisions where many potential explainers are present. Therefore, a larger number of potential hypotheses might make the task more complex, but additional knowledge is readily available.

- Hard decisions are avoided thus allowing ARTREC to build on islands of certainty. As long as some initial islands can be built, ARTREC can leverage these islands to continue building an explanation. ARTREC will only run into problems if it cannot build the initial islands of certainty.

- Layered abduction offers a means of constraining the possible combinatorial explosion of hypotheses. This is because, at one level, there will be a limited number of findings to explain because the previous level constrained the composite to only include confident members. Creating islands of certainty helps to constrain the problem no matter how many potential explainers are available.

Because of these issues, and because the Peirce strategy takes advantage of many forms of knowledge, I believe that ARTREC is scalable. Because of the constraining that results from layered abduction, there will always be a minimal number of findings to be explained at each level, which in turn should generate a minimal number of
potential explainers at each level.

In the English language, there are a finite (perhaps 50 to 100) total number of articulatory gestures. From these combinations of gestures, all English syllables can be generated. The scalability of ARTREC will be proven if we can expand ARTREC's gesture lexicon by five times its present size. This is one of the enhancements that are planned for the near future.

7.4 Abduction and the C/D Model

The Convertor/Distributor model[Fujimura and Wilhelms, 1991] is a new theory of non-linear speech production. Fujimura has suggested a means of speech production whereby the elements of the vocal tract (the articulators) will move independently of each other, but purposefully in order to create speech. The C/D Model is composed of three levels of processing, the Convertor, the Distributor, and the Articulators (or Actuators). See figure 35.

In this model, a person will have a thought to express. With this thought (in the form of a phrase or a sentence), there will be additional non-linguistic factors such as excitement or boredom, or whether the speaker is in a hurry, or under the influence of alcohol. Both the thought and these other factors are used as input to the C/D model. The output will be the movements of the vocal tract needed to create the utterance in mind.

The Convertor takes a phonological representation of the utterance and the relevant non-linguistic factors and "converts" it into a series of time indexed "pulses". This is passes on to the Distributor. Each pulse has an identity (of the phonetic units
Figure 35: The Convertor/Distributor Model of Fujimura
involved) and a magnitude. This sequence of pulses is received by the Distributor.

The Distributor then "distributes" or passes along, the magnitude and timing information to some of the articulators (or actuators). The Distributor knows, for each type of syllable (i.e. for each pulse type), which articulators are needed to generate the type of syllable. The magnitude is used to determine to what degree each articulator is needed (or to what degree the articulator should move).

Finally, each articulator receives a signal indicating when and to what extent it should "fire" (or move). The firings of an articulator occurs in conjunction with the other articulators needed for the syllable. An impulse response function dictates, for that articulator, the shape or the type of motion needed to carry out the sound. The magnitude of the pulse (to the distributor) is used as a parameter to each articulator's impulse response function. This parameter dictates the amount of muscle activity, which will yield the adequate amount of motion for that articulator. The result is parallel channels of motions in the vocal tract, each operating separately to create a unified sound.

Without going into much detail about the C/D model, we can say that, as it is a new theory, it requires "tuning" and "debugging". We can model the process of speech production and simulate the speech mechanism in order to determine if the theory is accurate. However, many of the details, or specifics, of the theory are not yet available. We need a different means of debugging this theory.

Layered abduction can help. Speech production is, abstractly, a planning or design problem. This task is one to of planning a particular set of motions in the vocal
tract to create an utterance (in the form of an acoustic signal). The plan requires determining the motions and degree of motions of each part of the vocal tract. It also requires determining when each part of the vocal tract should begin its motion.

The C/D model is basically a theory of planning, of planning the motions of the vocal tract. Abduction is a means of plan recognition [Ng and Mooney, 1989, Josephson, 1988, Josephson, 1987, Calisti, 1989, Lin, 1991]. We hope to implement an inversion of the C/D model by using layered abduction. The proposed inversion can be seen in figure 36.

Given vocal tract motions, we can use layered abduction to determine the articulators responsible for those motions. We can use this information as data to determine the actions of the distributor. In turn, we can use the actions of the distributor to infer the actions of the Convertor. Finally, we can use the actions of the Convertor and infer the intended utterance. This mapping is similar in description to what ARTREC attempts. In fact, the structure of ARTREC came from the C/D model.

Currently, ARTREC will map from pellet motions to articulatory gestures to syllables. The Distributor is responsible for generating, from articulatory units, the articulator motions. Thus, this first level in ARTREC is the inverse of the Distributor to Articulator mapping. Groupings of gestures can be used to infer syllables. The Convertor takes phonetic units (such as syllables) and generates the articulatory units needed to create the pronunciation of those syllables. Thus, the second level in ARTREC is the inverse of the Convertor to Distributor mapping.

It is, of course, not as simple as what is being presented here. The information
Determine the English Utterance from the gestures. Also, if possible, determines any non-linguistic effects that may have arisen.

Determine the articulatory gestures responsible for the sets of individual motions found in the pallet data.

Converter to convert intended utterance into pulses with magnitudes and identities.

For each event, determine the "cause" of that event. The cause will be a set of parallel motions among many articulators.

Distributor to distribute pulses to each articulator.

Determine articulators responsible for each motion. For each articulator, derive the events.

Each articulator uses an impulse response function to produce an independent motion for that articulator.

Articulatory Signal for an utterance in the form of Microbeam pallet data.

Figure 36: A Layered Abduction Description of the C/D Model Inverted
that the C/D model requires is specific knowledge pertaining to the types of gestures needed, the exact motions of the articulators, the way that the Distributor will distribute impulses among the Articulators, the degree to which each muscle must respond to the impulse, and so forth.

To obtain these types of knowledge, we can use a system like ARTREC. The equations specific to the articulators can be modeled in ARTREC and used as recognition knowledge. The types of gestures needed by the C/D model can be dictated by ARTREC's needs. The timings and magnitudes of pulses can be determined empirically by seeing where ARTREC works and where ARTREC fails. In short, we can use ARTREC, or a system like ARTREC, to determine where the specifics of the C/D model fail and where the specifics of the C/D model work.

We have already begun by determining some of the needed gestures. We have determined what articulatory motions are important for gestures. We have determined, to some extent, how corrective emphasis will alter the motions of the articulators (in this case, just the jaw).

Further research is needed to flesh out the intricacies of the C/D model. However, the two models, C/D, and ARTREC (or variations of ARTREC) can grow symbiotically. Where ARTREC requires knowledge, we turn to the C/D model for guidance. Where ARTREC fails, we turn to the C/D model for corrections. Where the C/D model requires specifics, we turn to ARTREC to implement them and see how ARTREC reacts.

Among the initial types of information that ARTREC has been able to determine
for the C/D model are as follows:

- Articulatory declination is salient in determining emphasis motions in the jaw. Articulatory declination was also noticed (less substantially) in the less enunciated words.

- Alveolar closure which occurs at the final position of a phrase includes some additional tongue blade motion.

- Alveolar closure motion differs when preceded by an /s/ sound (for instance, the first /t/ in “street” differs from the /t/ in “two” or the final /t/ in “street”). The difference lies in amount of height for the tongue tip. We have found through empirical study that /str/ requires less height for the /t/ sound than /t/ occurring at the beginning or end of a word.

- Tongue Blade motions are less important than tongue tip and tongue dorsum (at least from our data). ARTREC achieved equal (or even better) accuracy for data which did not contain a tongue blade pellet indicating that the features used in ARTREC which require tongue blade motions are not as important as other features.

- Extreme motions seem to be more “important” than the steady motions between extremes (the transitions). It seems reasonable to assume that we can explain the transitions in terms of the extreme motions (that is, by saying that transitions are the motions which bridge the utterances being pronounced). For
the pine street data, we found that we could ignore the transitional motions and explain only the extreme motions.

- Some of the motions in the articulatory data can be ignored during the recognition task. Therefore, some of the motions in speech production are unintentionally produced. The theory must take into account how these occur. Many presumably occur because of various non-linguistic factors.

The growth of both the C/D model and ARTREC are intertwined as they can both learn from each other. At the moment, both the C/D model and ARTREC are in their infancies. When the C/D model begins to produce specific impulse response functions for articulation, then ARTREC will be able to more closely match the C/D model. Currently, the two models are somewhat independent as the specific values needed for ARTREC are not yet available from the C/D model. ARTREC is also lacking in depth, stopping at the actuator motions rather than the musculature. However, in future, we expect that the C/D model will require ARTREC as an experimental implementation to determine where C/D goes wrong.

7.5 The Problems with Hidden Markov Models and how Abduction can Save the Day

Hidden Markov Models have been used to solve a large portion of the speech recognition problem. They have shown to be very successful, whereas "feature-based" knowledge has led to failure. There are still several problems in using HMMs. These problems all exist because HMM's require the linguistic knowledge (of phonemes and
words) to be represented in the form of probabilities and in no other form.

First, all of the linguistic knowledge encoded within an HMM is implicit. There is no sense of “knowing” what knowledge exists in an HMM aside from probabilities. The explanatory knowledge is only capable of explaining segments (i.e. divisions within the acoustic utterance) and is incapable of explaining features that appear in the acoustic signal. Without making the explanatory knowledge more explicit, there is no easy way of using an algorithm like Peirce to drive the HMM knowledge1.

The loss of any explicit knowledge makes the HMMs unusable for most problem solving regimes (such as heuristic approaches, task approaches, and so forth). Instead, the few means of using HMMs must lie in some form of search where the search criteria is simply the “most probable” path. This may not seem to be a flaw for those who use HMMs. But if we consider the many ideas that go into our abductive strategy, we can see that most of those are unusable by HMMs. There is clearly no notion of essentials and leveraging essentials against further problem solving. The only notions of hypothesis interactions are encoded within the probabilities themselves. We cannot say “this hypothesis becomes more likely because that hypothesis has been accepted”. The only form of expectation lies in the transition probabilities between one word model and the next (as in, this model is the most probable one to follow from that model). And there is no explicit notion of incompatibilities. Expectation and incompatibility knowledge can be stored implicitly in the transition probabilities

1Peirce could be used as a control mechanism for search through HMMs. However, Peirce attempts to explain features or individual data rather than a segment of speech. Therefore, Peirce would not generate an explanation for each feature detected in the speech signal but rather it would use chosen phoneme models to explain segmented regions of the signal.
but this knowledge cannot be used by mechanisms such as hypothesis interactions as shown in Peirce.

Hard decisions are impossible to delay or avoid in HMMs. In our problem solving, we may not wish to come to a conclusion over some datum because we want to order more tests, or give more deliberate thought to it, or even just avoid the problem (and we are all good at this form of procrastination if the problem is hard enough). HMMs do not even acknowledge hard decisions. A hard decision is the same as an easy decision. It is to simply find the most probable path. A dynamic programming algorithm will not stop to consider islands of certainty. It will not decide that an alternative solution is "nearly as plausible". Further, there is no way of knowing how much confidence we should place in an answer generated from an HMM system.

We can continue along these lines, but it should be apparent that HMMs using dynamic programming search techniques fail to make use of any of the abductive strategies discussed in chapter 3.

Second, (and related to the first), since all knowledge is encoded in the form of probabilities, HMMs are not easily combined with other forms of knowledge. Speech recognition requires bringing to bear many forms of knowledge from acoustic and articulation knowledge, to knowledge of syntax, semantics and pragmatics. It seems rather difficult to capture many of these types of knowledge as probability transitions. Therefore, this knowledge is not directly available for a system which uses HMMs.

Sphinx could not use explicit syntactic knowledge. Instead, it forced the grammar into a probabilistic framework. This is unreasonable as it assumes the availability of
grammar probabilities. Sphinx used Bigrams to capture the probability of one word following another. This form of knowledge is only useful in limited contexts. Sphinx could not use any higher level knowledge (semantics, pragmatics or discourse). While knowledge of syntax, semantics and pragmatics might be capturable in HMMs, it is not a natural way to store or use such knowledge.

Byblos was able to use some higher level knowledge (syntactic knowledge) by encoding it into a more explicit, feature-based form. However, Byblos’s grammatical knowledge and lexical knowledge were very loosely coupled. The only means of sharing knowledge between the two units was to use the grammar to prune away incorrect words. The problem with this is that the grammar could only make suggestions and could only determine when a word was not syntactically correct. Thus, the word space had to be searched for possible next words and only then did the grammatical module play a role by ruling some next words out. The grammar was limited in processing as it did not allow the lower levels to query higher levels for additional help. It was strictly a one-way flow of communication.

As Byblos shows, additional knowledge can be used by HMMs, but only in indirect ways. There is no way to use additional knowledge to affect the transition probabilities within models. The additional knowledge can only be used between models, in affect, to aid the beam search process. By using HMMs, a system is restricting the problem solving so that additional knowledge cannot be brought directly to bear. This is very unlike our layered abduction method where layers can directly interact by propagating expectations and other forms of information.
Another criticism of HMMs pertains to training HMMs. In training HMMs, an algorithm is used to update transition probabilities within a model based on how accurately that model matches with the data in the training set. For example, if we are training a phoneme model on the sound /t/, and we have the phoneme /t/ in a training sentence, we adapt the /t/ model so that it is more able to match or recognition the /t/ sound. However, training sets are usually biased in some form. The bias can come from a limited set of sentences, a limited vocabulary, a limited usage of the phonemes (or other phonetic units), a limited amount of emphasis and stress, and so forth. The result is a biased set of HMMs. This problem of biased training sets is particularly relevant in the case of Sphinx and the Bigrams used there. The Bigrams are created from the training sets. Thus, if a speaker were to utter a series of words that did not occur at all in the training set, then Sphinx would have a much greater difficulty in correctly identifying the utterance. There seems to need to have such bias in the knowledge of a speech system.

One particular problem that we envision as the result of such bias would arise due to the effects of prosody, and in particular, stress and emphasis. It is apparent from past speech recognition research that stress and emphasis are important concepts which should be used for accurate word identification. If a training set for HMMs contain "typical" pronunciation of words without stress, then the HMMs will be based on the "typical" pronunciation. If someone uses stress in their speech, an HMM speech recognition system would have difficulties in accurately identifying those stressed words. One approach would be to store both stressed and unstressed word
models in such a speech recognition system, but this might only increase the confusion of the system. It is hard to tell, based simply on probability transitions, when a word is stressed. Further, stress and emphasis occur not as a binary decision (i.e. there is stress or there is not stress) but rather, as a continuum from highly stressed to unstressed words. Thus, multiple sets of HMMs would be necessary to capture all variations of stressed and unstressed words.

Another problem with HMMs is not with the models themselves, but in what they model. I have argued, along with others [Fujimura, 1992, Browman and Goldstein, 1986], that linear phonology is the wrong way to consider speech. Most past attempts at using HMMs have been in modeling phonemes. By modeling phonemes, you inherently model linear phonology. It seems unreasonable to choose an implementation method (HMMs) which presupposes a theory of linguistics (linear phonology). HMMs make a commitment which, under ordinary implementations, does not need to be made. However, HMMs are not capable (at least in a straightforward manner) of modeling non-linear phonology because HMMs cannot model features. That is, non-linear phonology describes units in terms of distinctive features. Sounds are made up of vocal tract features, not of phonetic units. HMMs are not capable of representing features of units, only of representing units themselves. Therefore, in choosing HMMs, a researcher is forced to model some form of unit, of a phonetic unit. In the past, the choices have been for modeling phonemes or triphones. And this results in modeling a weaker theory, linear phonology rather than non-linear phonology.

I should state here that HMMs only encode knowledge, and that knowledge can
be used in an abductive system. However, systems such as Sphinx and Byblos are at best impoverished abductive systems. They are capable of explanation, but only by appealing to a very limited set of knowledge and by using a very weak method of explanation. All of the knowledge in the HMM systems is collapsed into probabilities making it difficult to distinguish the types of knowledge needed in abduction (such as differentiating between explanatory knowledge, hypothesis interaction knowledge, pragmatic concerns of the importance of data, and so forth). The only method used to drive these HMM systems is a dynamic programming algorithm using some form of beam search, a far cry from the more sophisticated strategies discussed in chapter 2 (e.g. the strategies of Pearl, Reggia and Peng, Pople and Hobbs) or the Peirce strategy discussed in chapter 3.

I have stated that HMMs capture knowledge only in the form of probabilities. HMMs collapse all knowledge into a single form. This is done so that the efficiency and optimality of dynamic programming can be used to drive the probabilities. It is also done because training HMMs is possible. However, in doing so, many things are lost. Among the losses is the ability to use other strategies, clever strategies, to seek the best explanation. Also lost is the ability to directly incorporate other forms of knowledge. And again, HMMs make it very awkward to use disparate forms of knowledge. Only by indirect means will this be possible.

One solution to this problem is to use layered HMMs in much the same approach as our layered abduction. However, in layered abduction, we can use both upward flowing and downward flowing information to build on islands of certainty. It is unclear
(at best) to see how HMMs could use any form of downward flowing information or how islands of certainty could be used for leverage for further problem solving. These advantages to an abductive strategy are completely lost in the HMM paradigm.

There is a way to use HMMs which can take advantage of hypothesis interactions, higher level knowledge, and "intelligent" problem solving. This is to use the probabilities stored in HMMs as expectation knowledge and plausibilities. However, instead of using this knowledge in a dynamic programming beam searching algorithm, we can use this knowledge to aid the selection of explanatory choices in feature-based approach to abduction. Rather than driving HMMs with a dynamic programming algorithm, create feature hypotheses and use the probabilities stored in the HMMs as some of the available knowledge. The layered abduction strategy can use probabilities just as easily as "confidence" values.

Our abductive strategy takes advantage of a notion which is ignored by using dynamic programming. When a hypothesis is much better than another hypothesis for explaining some datum, we feel secure in accepting it. If alternative explainers are almost as good (or plausible), we would feel less secure in accepting a hypothesis. Dynamic programming does not consider this. Dynamic programming will find the most probable path through the search space no matter how close the "next best" path might be. Therefore, HMM systems have no means of stating how much certainty we could place in a conclusion formed by dynamic programming. This, to us, is unacceptable.

HMMs are able to capture knowledge by training which might otherwise be un-
obtainable. This is the strength of HMMs, their easy ability to learn. We can use this learning method to store the knowledge needed in a feature-based abduction system. Thus, we combine the strengths of HMMs with the strengths of an abduction strategy.

7.6 Layered Abduction is the Proper Way to Solve Interpretation Tasks

This is quite a bold statement. But consider this: in reasoning about the world, we attempt to come to conclusions about the observations and findings that we face. Our whole perspective of the world around us is driven by perception. If perception is truly an inferential process, then what better way should we attempt to solve such problems but by inference to the best explanation?

I have argued in chapters 1 and 3 that Layered Abduction is a feasible and useful task for solving cascaded inference problems. I have given a general indication of several types of problems such as diagnosis, theory formation, story comprehension and perception. And even these problem types only scratch the surface. Others have used abduction to solve problems in all types of identification problems (medical, mechanical, sensor, concept, etc...). Therefore, when a problem calls for inference, when a problem calls for coming to a conclusion, then abduction is a method capable of supplying a conclusion. When problems call for multiple sources of input (for instance, sensors or visual or auditory input, or a combination of these) and when a problem calls for disparate knowledge types, then layered abduction should be considered.
In chapter 3, I have discussed a particular strategy for solving abductive problems. In chapter 4, I showed a particular use for the strategy. This strategy, incorporated in the tool Peirce, allows for opportunistic, island building problem solving. It is capable of solving most all problems which make use of explanatory knowledge, even in the face of uncertainty, noisy data, and lack of clear plausibility values for hypotheses. I feel that Peirce is a tool which captures a very generic form of problem solving and as such, Peirce is highly useful for constructing many types of problem solvers. The research that I have discussed has involved highly different forms of problem solving.

There is still a tension in my argument that layered abduction can be used to solve perceptual problems. The tension lies in the rapidity with which humans can process speech (or vision) versus the seemingly deliberative aspect of abductive explanation. It does seem reasonable to suggest that a process of forming an explanation need not have any deliberation. In perception, we as perceivers, have a large amount of compiled knowledge. In [Patten et al., 1992, Chandrasekaran, 1991] it is suggested that problem solving knowledge is compiled across a wide spectrum of types which can be categorized into at least four distinct classes, table look-up, pattern matching, generic tasks and first principles.

Table look-up is the most highly compiled form of knowledge where we simply provide input (in the form of findings or stimuli) and our minds look up (in some large table of knowledge) what the solution is. This form of knowledge may be thought of as a chunk [Newell, 1990] of knowledge, or as a hard-wired piece of knowledge which is nearly instantly retrieved. It seems reasonable that some amount of perceptual
processing uses such highly compiled knowledge. At this level, there is no need for deliberation of any kind.

The next category is at the level of partial pattern matching. In the past, partial pattern matching has taken the form of production systems or rule-based systems [Shortliffe, 1976]. At this level, input is provided to the problem solver, and small steps are taken in order to determine an answer. Again, no or little deliberation takes place.

The next category might be the Generic Task level [Chandrasekaran, 1987, Chandrasekaran, 1988, Chandrasekaran, 1986]. Generic tasks are at a level of problem solving where one subdivides the problem into subproblems, each of which can be solved by some generic method. Examples of generic tasks include hypothesis matching (which is a subtask of hypothesis instantiation as described in chapter 3), abduction, hierarchical classification. These task-level solutions require particular types of knowledge (such as parent-child relationships in classification, features of interest in hypothesis matching, and explanatory knowledge in abduction). As these tasks require more complex steps, some generic tasks may occur at a more deliberative level of processing.

The last category is first principles [Sembugamoorthy and Chandrasekaran, 1986, Reiter, 1987, M.Keuneke, 1988, Allemang, 1990, de Kleer, 1986]. At this level, knowledge is found in the form of models, and reasoning over this knowledge requires understanding physics, simulation, and so forth so that one can trace through a model in order to infer the consequences. This form of knowledge is used when the more
compiled forms of knowledge are unavailable. For example, in theory formation, one might need to derive an entire new theory from first principles. A beginning diagnostician will require to form a diagnostic conclusions from first principles knowledge because the diagnostician has yet to compile his or her experience into more highly compiled knowledge.

It seems reasonable to say that knowledge begins at a first principles level, and that the more we use the knowledge, the more compiled it becomes. Diagnosis may occur at any of the four levels described above depending upon the experiences of the diagnostician. Perception can similarly occur using knowledge of any of these, however as time goes on in an individual’s lifetime, he (or she) is more and more likely to use highly compiled knowledge for perceptual problems.

And here is where the friction arises in my argument. I am endorsing a generic task solution of perception (by using layered abduction) and yet most of the perceptual knowledge seems to lie at the partial pattern matching or table look-up levels. Here is how this tension can be relieved. First, abduction can use highly compiled knowledge. Knowledge of explanatory coverage, evocation and instantiation of hypotheses, and hypothesis interactions can all be very highly compiled. This is, in fact, most likely considering how much we use perception. And when hypothesis evocation brings forth a single hypothesis to explain some datum, then the abductive explanation is made very simply, by accepting the hypothesis (as an essential). In such a case, there is no need to make use of a full-blown abduction algorithm. but to simply accept the hypothesis. When such a situation does not arise (that is, when more than a
single hypothesis is suggested by evocation) then a decision is required. Therefore, the Peirce strategy, as described in chapter 3, is still sufficient for solving perceptual problems. The strategy will come to a very quick decision when a single hypothesis is plausible (which seems to be the case quite often in perception) and deeper strategies take over when more than a single hypothesis is deemed as plausible.

We can propose a method to solve a problem by using very efficient generic tasks while still using more highly compiled knowledge. If anything, by considering the perceptual tasks as abductive, we can determine the types of knowledge needed to solve the task (i.e. we require explanatory knowledge, knowledge of hypothesis interactions, etc...). And if the knowledge comes in a highly compiled form, abduction can still make use of this knowledge.

Given some input to the auditory tract (i.e. the ears), we attempt to determine what was said. We do so by hypothesization, and by combining individual hypotheses (say of phonemes, syllables or words). This is exactly what abduction accomplishes. The need for an explanation does not require any amount of deliberation, in fact, when highly compiled and efficient knowledge is used, the abductive mechanism will form a composite hypothesis very rapidly. Only when ambiguities arise will there be any need for making use of less compiled knowledge in order to make sense of the ambiguities.

In conclusion then, I have described a very efficient method for solving the problem of abductive inference. This method is equally suitable for solving single layered abductive problems and multi-layered abductive problems. I have given a plausibility
argument that most inferential problems are solved by abduction, and I have indicated, by argument and example, how a single method for abduction can be used to solve problems ranging from theory formation to perception. It seems reasonable to suggest that this highly efficient (and in principle, parallelizable) method could play a significant role in human speech recognition. It seems rather certain that this highly efficient method can play a significant role in automated speech recognition as shown in this document.

7.7 Conclusion

In this chapter I have discussed the merits of using articulatory knowledge for speech recognition. In particular, I have discussed how ARTREC might be used in large acoustic speech recognition systems. I have argued that ARTREC is scalable into a much larger set of articulatory gestures and lexicon.

I have discussed in detail some problems with Hidden Markov Models. HMMs are not panacea's for speech recognition. They are useful ways to encode hard-to-obtain knowledge by using training. However, this training technique could also be used to obtain the knowledge in different formats. Whether to use probabilities or not, is not really a concern. The concern is that current speech recognition research using HMMs is obscuring the problems associated with current linguistic theory with the benefits of using a "learnable" mechanism which can find features for itself. Training has many uses. However explicit knowledge of linguistics is also necessary to solve speech recognition problems. HMMs demonstrate the usefulness of training but obscure the need for feature knowledge. Without explicit feature-based knowledge, we lose the
power of explanation (i.e. abduction). Without abduction, we lose many of the advantages for increasing the accuracy of speech recognition and many other types of problems.

I have concluded this chapter with a brief restatement of my beliefs in the usefulness of layered abduction. As has been shown, layered abduction can be used to solve a variety of problem types. Peirce itself can be used to construct knowledge-based systems to solve these (inferential, explanatory) problems.

In the next, and final, chapter, I will review the contributions of the work presented here. I will discuss limitations of our approach. I will also offer some very possible extensions to the work.
CHAPTER VIII
Future Directions and Conclusion

8.1 Introduction

This dissertation has examined a problem solving task called layered abduction, and a particular method or strategy for solving this task, incorporated into a problem solving tool called Peirce. It is my thesis that layered abduction can be used to solve most (or all) types of interpretation tasks, and in particular, layered abduction can be used to solve the problem of speech recognition.

This chapter reexamines the work presented in this dissertation. I will discuss the contributions of this work. I will discuss the limitations of the ARTREC system and discuss possible future enhancements. I will also entertain some possible expansions for the Peirce tool. Finally, I will give a future research agenda for advancing abduction and layered abduction in Artificial Intelligence.

8.2 The Intention of this Research

We originally began this research in hopes of constructing a full-blown layered abduction speech recognition system (see figure 3 in chapter 1). This research dates back over three years when a proposal was granted a small amount of initial research money. The results of that research (documented in [Fox and Josephson, 1991, Fox
et al., 1991]) led to two prototype speech recognition systems. One of these prototypes was a consonant-vowel isolated syllable recognizer. It used a feature-based abductive approach to recognize the syllables. The other prototype was a small multilayered articulatory recognition system which input articulatory gestures and inferred words. See chapter 3.4.2 for more details on these systems. The research was far from complete when we had to terminate the project. However, we continued to pursue the idea of a layered abductive speech recognition system. We then met Professor Osamu Fujimura who was attempting to define a new theory of speech production; one based on non-linear phonology. His new model, the Convertor/Distributor model, was to give new life to our own research.

One novel idea that was involved in our original research was to use explicit articulatory knowledge. By appealing to such knowledge, a speech system would not need to rely on the "coarticulation" rules used in past systems based on linear phonology. Prosody could be more directly represented. While that research never reached fruition, the C/D model reinspired us. Our ideas for using articulation for speech recognition could be revived.

There are three lines of motivation behind our research. We want to construct a layered abduction speech recognition system. A second motivation is to center our approach on articulatory knowledge, appealing to articulation in order to solve some of the problems found in speech recognition. Our third motivation is to help Fujimura in his creation of the C/D model. We hope to implement a reverse mapping of the C/D model by using layered abduction. These goals have not yet been reached.
However, ARTREC was a good start in the right direction having accomplished the initial step towards all three of the goals. We have learned a lot during the research on ARTREC and we hope to continue to have success as ARTREC grows.

As far as contributions go, this work has shown the following. First, I have constructed a layered abduction system which works! I have demonstrated the usefulness of our abductive strategy in a series of experiments (both in speech recognition and outside of speech recognition). I have demonstrated the power of explanation as opposed to simple pattern matching. Next, I have constructed a system for articulatory recognition and shown its feasibility. I have argued (hopefully in a convincing manner) how this system could be used for a full-fledged speech recognition system. Finally, I have shown how ARTREC can be used to help in the creation and debugging of Fujimura’s new theory of speech production.

8.3 Limitations of ARTREC

ARTREC is a modest attempt at an articulatory speech recognition system. It is a far cry from what is needed to solve speech recognition. As pointed out throughout this document, ARTREC makes no attempt whatsoever to solve the acoustic speech recognition problem. I have only suggested ways to alter or combine ARTREC with other mechanisms for the purpose of speech recognition (in chapter 7.2). While I believe in these suggestions, I have not proven them. This single limitation is very staggering. I cannot prove the assumption that perception is abductive, without first implementing a full speech recognition system with layered abduction.

Further, I have only implemented a modest sized layered abductive system. A
three level system using two abducors is not very complex. I have yet to answer ques-
tions pertaining to the overall control of a large layered abductive system. Can we 
bypass the control problems that Hearsay had to confront? Can we implement a lay-
ered abductive system in parallel? Can we implement the individual abducors so that 
they contribute to the overall solution rather than constantly offsetting each other? I 
have shown that layered abduction is feasible. In the fourth ARTREC experiment, I 
showed how top-down processing can improve overall system performance. But when 
should top-down processing come into play during the problem solving?

These questions show that I have only scratched the surface in our research. I 
have high hopes and we are very encouraged by the success of ARTREC thus far. 
But I have many ideas which have yet to implement. These ideas are for expansions 
to both ARTREC and the Peirce tool.

8.4 Future Expansions of ARTREC

There are many dimensions in which ARTREC is to be expanded. This work has 
already commenced. We have acquired new pellet data from the Madison Microbeam 
facility. This data is similar to the “Pine Street” data in that it consists of numbers 
and street names. However, we intentionally constructed the set of new data to include 
most of the types of articulatory gestures found in the English language although the 
gestures in the new data are not used in all possible conjunctions with each other.

The lexicon size for the new data is around 65 words, about a quarter of which 
are polysyllabic words. We hope to determine if our method for finding syllable 
boundaries will work as well in polysyllabic words.
This new data also features several forms of non-linguistic effects in the pronunciation of speech. Besides emphasis, we have evoked other emotional responses such as boredom, humor and annoyance. While we might not be able to identify the exact form of effect, we can detect that such an effect has arisen and use this to determine how the effect will alter the pronunciation. We will continue to pursue the effects of prosody on articulation.

The first expansion is to make ARTREC run on the new data which requires two changes. First, ARTREC must be able to handle new gesture types and new words. This means enlarging the gesture and syllable knowledge to include the new forms. The second expansion allows ARTREC to use polysyllabic words by adding a new level of problem solving, separating the syllable level from the word level. This new level will take strings of syllables and attempt to infer, using a lexicon, what word or words could have caused those syllable strings. ARTREC can also use the lexicon to infer word boundaries if they are not clear from the mandible motion. We know already that layered abduction is a useful means of deriving word boundaries from strings of phonetic units [Fox et al., 1991] without additional knowledge other than a lexicon. We feel that adding a new level will be relatively straightforward.

We have several further enhancements in mind for ARTREC.

• Automated learning of gestures. This is a process whereby ARTREC will derive its own gesture templates from input data. We will present examples to

---

1 ARTREC will require some means of determining word boundaries. Currently, ARTREC uses mandible motion to infer syllable boundaries. There is no reason to assume that this measure will be successful in determining all syllable boundaries or any word boundaries.
ARTREC, and using simple mathematical manipulations of prestored gesture templates, have ARTREC adapt these feature templates into more accurate versions. This process of adapting feature templates has thus far been accomplished by hand. However, we envision simple methods which will allow ARTREC to do this itself. Once this is accomplished, ARTREC will be able to teach itself new gesture forms with only a modest amount of guidance by the system designers. This learning method would be similar, in some ways, to neural network or HMM training. However, we would make the features explicit which would allow us to better correct ARTREC when it errs. Also, explicit features are necessary for Peirce. We can gain the advantage of training without losing the flexibility or power of the abductive algorithm (unlike HMMs which must use a dynamic programming algorithm guided by some form of beam search, or neural networks which have no explicit algorithm at all).

- Syntactic knowledge is highly useful in speech recognition. Using a grammar in speech recognition, we are able to rule out many words without giving them much consideration. While the original "Pine Street" data for ARTREC was simplistic and did not require the use of any grammar, future data might require this extra knowledge. We envision constructing an abductive parser to accomplish this. Research into abductive parsing has been done by [Dasigi, 1988, Venugopala and Reggia, 1988] and we are interested in using Peirce to implement such a parser. Thus, a syntactic level will be placed on top of the word level. This should be a fairly easy addition to ARTREC.
• Adding an acoustic signal to ARTREC is also envisioned. Eventually, we would like to construct a full-fledged speech recognition system in which an articulatory level is a part. To advance our research, we plan on adding an acoustic level to ARTREC. This level will input an acoustic signal and infer possible articulatory knowledge. While we will still make use of pellet data, at some point in the expansion of ARTREC, it is hoped that the pellet data will no longer be needed. This last expansion is the most ambitious and most certainly the most difficult.

We feel that the results from ARTREC are encouraging enough to continue examining articulatory recognition. Articulatory recognition is an important, and typically missing, element of speech recognition. The initial enhancements (more words and gestures, more abductive levels) will prove that large scale articulatory recognition is feasible. We have already shown layered abduction to be an efficient and powerful means of implementing articulatory recognition.

8.5 Expansions to Peirce

Layered abduction is a mechanism which can be applied to numerous problems. Peirce is a flexible tool creating a rich programming environment. Even so, we envision some enhancements to the Peirce tool. These enhancements include (but are not limited to) the following:

• A greater number of levels to produce conclusions during the hypothesis composition (or assembly) cycle. Currently Peirce has confirmeds, essentials, clear
bests and weak bests. Essentials are those hypotheses which seem to be the only plausible means of explaining some datum. Essentials are an idealization, there can always be alternative explanations found no matter how implausible. Clear bests are superior explainers, and weak bests are mildly better explainers. However, we could have many in-between levels. Essentials may be rated as absolute (if there are no other explainers at all), reasonable (if all alternative explainers have been ruled out or rated low) or just potential (if the search for alternative explainers was not exhaustive). Clear bests may come in different levels such as incredibly-clear-best, very-clear-best, mildly-clear-best, marginally-clear-best. What is envisioned is a cycle in which the criteria for "bestness" is decremented by some present amount on each pass through the algorithm. At first, only the most superior hypotheses are used. Then, if needed, we resort to less superior, but still very good hypotheses. And so forth until, if necessary, we resort to using weak best hypotheses. The preset "decrement" amount and the amount of tradeoff between desired coverage and desired accuracy can be set by the system builder or the system user either at system construction time or at run time.

- Changing the propagation of incompatibilities. In our current system, if a hypothesis is incompatible to a believed hypothesis, it is ruled out. However, if that believed hypothesis was a weak best, should we rule out the incompatible hypothesis? What if our decision to accept the weak best was erroneous? In ruling out an incompatible, the propagations of this action might be far reaching
such as new essentials or clear bests being accepted into the composite. If the original choice to accept a hypothesis is tenuous (and this would be true in accepting a weak best), then any actions taken as a result of the acceptance should also be tenuous. Therefore, when we accept a hypothesis, we should not rule out incompatible hypotheses, we should adjust the plausibilities of the incompatible hypotheses depending upon how much confidence we have in the accepted hypothesis. If the accepted hypothesis was essential, we can rule out incompatible hypotheses. If the accepted hypothesis was a clear best, we could greatly weaken the plausibilities of incompatible hypotheses. If the accepted hypothesis was a weak best, we could slightly weaken the plausibilities of incompatible hypotheses.

- Provide a more flexible control for layered abduction. As it stands now, there is no built-in mechanism for control multiple abducters in Peirce. Yet, we have ideas of how abducters should interact with each other. In past and current layered abductive systems, the control between the multiple abducters must be provided at the time the systems are constructed. If an opportunistic control strategy can be provided (such as in [Punch, 1989]) more time can be spent on encoding the proper knowledge rather than adjusting the processing between abducters. Punch, [Punch et al., 1986], has developed an opportunistic abducer, also called Peirce\(^2\) which displays some of the flexible that we would like out of

---

\(^2\)Punch's Peirce was constructed here at the LAIR while he was working on his dissertation. His version of Peirce is very similar in ways to our Peirce, however, our Peirce was written in Common Lisp as part of the Integrated Generic Task Toolset. Many of the same ideas behind Punch's Peirce can be found in our Peirce. The greatest exception is the flexible control which was not part of
8.6 Future Work

The work presented here is based on using layered abduction to solve perceptual problems. There are many *types* of problems that can make use of layered abduction. My own personal research will continue the pursuit of layered abduction. I would like to continue to examine various abductive problems and recast them in a *layered* abductive framework. The perceptual research will continue as ARTREC is expanded along the various dimensions discussed earlier. Also, ARTREC will continue to be used in conjunction with the C/D model so that we can continue to aid in Fujimura's research.

I suggest here some further research for layered abduction in non-perceptual areas. We have implemented a small system to decide between two theories, evolution and creationism. This system uses static findings and hypotheses, and determines which theory has more explanatory power. There are several dimensions which could make this decision making system more realistic. For one, none of the hypotheses are rated in terms of plausibility. Such knowledge is not easily available. Another problem is that it takes a shallow look at both theories. A more realistic implementation would include much deeper knowledge about the theories and what they can explain. One possible means of a deeper representation is to use Functional Representation as a means of storing knowledge [Sembugamoorthy and Chandrasekaran, 1986]. A functional representation can be used to evoke hypotheses, determine plausibility *our Peirce.*
of hypotheses, and create expectations [M. Keuneke, 1988, Allemang, 1990, Punch, 1989]. All of these tasks are extremely useful for an abductive reasoner. Functional Representation seems a natural way to encode much of the knowledge that a system might need. Figure 37 shows a functional model used as a module which interacts with an abducer.

![Diagram](image)

**Figure 37: Functional Model with an Abducer**

Our work on legal reasoning is very limited. We would like to implement a different legal case, one which would make use of all of the evidence presented and generate
and score hypotheses related to the case. A several month long legal case cannot be easily captured in a small knowledge-based system. Our Peyer example is very primitive and only covers a portion of the knowledge needed for legal reasoning. More research into the legal process and into particular cases is necessary to implement a large legal reasoning system.

The work we have put into these two systems is not substantial but enough to lead us to believe that large systems are possible and reasonable. Future work lies in further researching such problems and implementing larger systems using the same framework as the layered abduction strategy presented here.

As shown in this document, layered abduction is a useful means of data interpretation. The articulatory recognition task can be thought of as just that, data interpretation. Given the Microbeam pellet data, one must infer the causes of those motions. Many other problems involve similar data interpretation tasks such as red blood cell antibody identification based on test reactions or character identification or plan recognition or vehicle identification based on sensor values [Josephson et al., 1984, Marquis, 1991, Lin, 1991, Carver and Lesser, 1991]. Further, there is a great need for systems which can take disparate sources of input, large but different types of knowledge, and form coherent explanations (or conclusions) for the input. Speech recognition is one such problem. Visual understanding, story comprehension, large scale diagnosis, theory formation are all such problems.

We feel that with a layered abduction strategy as discussed in chapter 3 and as shown in chapter 4, we can build problem solvers which can solve any type of
data inferencing problem. Abstractly, Peirce is a tool for constructing agents which generate explanations for the causes of the input. Our future research will revolve around implementing large systems using Peirce. We will continue to examine layered abduction and improve on Peirce. Eventually, we will (hopefully) construct a large scale speech recognition system using layered abduction and articulation.


decision making: A case of evolution vs. creationism. Technical report, The Ohio
State University.


An iceberg model. In Perkell, J. S. and Klatt, D. H., editors, Invariance and

search. Language and Speech.


Prosodic effects on articulatory gestures - a model of temporal organization. In
Proceedings of Interactional Congress on Phonetic Sciences.

catenative phonetic units. Syllables and Segments, pages 107-120.

functions for elemental articulatory events. In Journal of Acoustic Society of Amer­
ica, ASA Meeting.

mentation of continuous speech. In Proceedings of the International Conference for
Acoustics Speech and Signal Processing. IEEE.

Events Using Articulatory And Acoustic Data. PhD thesis, The Ohio State Uni­
versity.

editor, The Oxford Companion to the Mind, pages 608-611. Oxford University
Press.

London.


University of Iowa Press.


