

# 1

## INTRODUCTION

*Dave Bowman: Open the pod bay doors, HAL.*

*HAL: I'm sorry Dave, I'm afraid I can't do that.*

Stanley Kubrick and Arthur C. Clarke,  
screenplay of *2001: A Space Odyssey*

The HAL 9000 computer in Stanley Kubrick's film *2001: A Space Odyssey* is one of the most recognizable characters in twentieth-century cinema. HAL is an artificial agent capable of such advanced language-processing behavior as speaking and understanding English, and at a crucial moment in the plot, even reading lips. It is now clear that HAL's creator Arthur C. Clarke was a little optimistic in predicting when an artificial agent such as HAL would be available. But just how far off was he? What would it take to create at least the language-related parts of HAL? Minimally, such an agent would have to be capable of interacting with humans via language, which includes understanding humans via **speech recognition** and **natural language understanding** (and, of course, **lip-reading**), and of communicating with humans via **natural language generation** and **speech synthesis**. HAL would also need to be able to do **information retrieval** (finding out where needed textual resources reside), **information extraction** (extracting pertinent facts from those textual resources), and **inference** (drawing conclusions based on known facts).

Although these problems are far from completely solved, much of the language-related technology that HAL needs is currently being developed, with some of it already available commercially. Solving these problems, and others like them, is the main concern of the fields known as Natural Language Processing, Computational Linguistics, and Speech Recognition and Synthesis, which together we call **Speech and Language Processing**. The goal of this book is to describe the state of the art of this technology

at the start of the twenty-first century. The applications we will consider are all of those needed for agents like HAL as well as other valuable areas of language processing such as **spelling correction**, **grammar checking**, **information retrieval**, and **machine translation**.

## 1.1 KNOWLEDGE IN SPEECH AND LANGUAGE PROCESSING

By speech and language processing, we have in mind those computational techniques that process spoken and written human language, *as language*. As we will see, this is an inclusive definition that encompasses everything from mundane applications such as word counting and automatic hyphenation, to cutting edge applications such as automated question answering on the Web, and real-time spoken language translation.

What distinguishes these language processing applications from other data processing systems is their use of *knowledge of language*. Consider the Unix `wc` program, which is used to count the total number of bytes, words, and lines in a text file. When used to count bytes and lines, `wc` is an ordinary data processing application. However, when it is used to count the words in a file it requires *knowledge about what it means to be a word*, and thus becomes a language processing system.

Of course, `wc` is an extremely simple system with an extremely limited and impoverished knowledge of language. More-sophisticated language agents such as HAL require much broader and deeper knowledge of language. To get a feeling for the scope and kind of knowledge required in more-sophisticated applications, consider some of what HAL would need to know to engage in the dialogue that begins this chapter.

To determine what Dave is saying, HAL must be capable of analyzing an incoming audio signal and recovering the exact sequence of words Dave used to produce that signal. Similarly, in generating its response, HAL must be able to take a sequence of words and generate an audio signal that Dave can recognize. Both of these tasks require knowledge about **phonetics and phonology**, which can help model how words are pronounced in colloquial speech (Chapters 4 and 5).

Note also that unlike Star Trek's Commander Data, HAL is capable of producing contractions like *I'm* and *can't*. Producing and recognizing these and other variations of individual words (e.g., recognizing that *doors* is plural) requires knowledge about **morphology**, which captures information about the shape and behavior of words in context (Chapters 2 and 3).

Moving beyond individual words, HAL must know how to analyze the structure underlying Dave's request. Such an analysis is necessary among other reasons for HAL to determine that Dave's utterance is a request for action, as opposed to a simple statement about the world or a question about the door, as in the following variations of his original statement.

HAL, the pod bay door is open.

HAL, is the pod bay door open?

In addition, HAL must use similar structural knowledge to properly string together the words that constitute its response. For example, HAL must know that the following sequence of words will not make sense to Dave, despite the fact that it contains precisely the same set of words as the original.

I'm I do, sorry that afraid Dave I'm can't.

The knowledge needed to order and group words together comes under the heading of **syntax**.

Of course, simply knowing the words and the syntactic structure of what Dave said does not tell HAL much about the nature of his request. To know that Dave's command is actually about opening the pod bay door, rather than an inquiry about the day's lunch menu, requires knowledge of the meanings of the component words, the domain of **lexical semantics**, and knowledge of how these components combine to form larger meanings, **compositional semantics**.

Next, despite its bad behavior, HAL knows enough to be polite to Dave. It could, for example, have simply replied *No* or *No, I won't open the door*. Instead, it first embellishes its response with the phrases *I'm sorry* and *I'm afraid*, and then only indirectly signals its refusal by saying *I can't*, rather than the more direct (and truthful) *I won't*.<sup>1</sup> The appropriate use of this kind of polite and indirect language comes under the heading of **pragmatics**.

Finally, rather than simply ignoring Dave's command and leaving the door closed, HAL chooses to engage in a structured conversation relevant to Dave's initial request. HAL's correct use of the word *that* in its answer to Dave's request is a simple illustration of the kind of between-utterance device common in such conversations. Correctly structuring these such conversations requires knowledge of **discourse conventions**.

To summarize, the knowledge of language needed to engage in complex language behavior can be separated into six distinct categories.

<sup>1</sup> For those unfamiliar with HAL, it is neither sorry nor afraid, nor is it incapable of opening the door. It has simply decided in a fit of paranoia to kill its crew.

- Phonetics and Phonology — The study of linguistic sounds
- Morphology — The study of the meaningful components of words
- Syntax — The study of the structural relationships between words
- Semantics — The study of meaning
- Pragmatics — The study of how language is used to accomplish goals
- Discourse — The study of linguistic units larger than a single utterance

## 1.2 AMBIGUITY

AMBIGUITY

A perhaps surprising fact about the six categories of linguistic knowledge is that most or all tasks in speech and language processing can be viewed as resolving **ambiguity** at one of these levels. We say some input is ambiguous if there are multiple alternative linguistic structures than can be built for it. Consider the spoken sentence *I made her duck*. Here's five different meanings this sentence could have (there are more), each of which exemplifies an ambiguity at some level:

- (1.1) I cooked waterfowl for her.
- (1.2) I cooked waterfowl belonging to her.
- (1.3) I created the (plaster?) duck she owns.
- (1.4) I caused her to quickly lower her head or body.
- (1.5) I waved my magic wand and turned her into undifferentiated waterfowl.

These different meanings are caused by a number of ambiguities. First, the words *duck* and *her* are morphologically or syntactically ambiguous in their part-of-speech. *Duck* can be a verb or a noun, while *her* can be a dative pronoun or a possessive pronoun. Second, the word *make* is semantically ambiguous; it can mean *create* or *cook*. Finally, the verb *make* is syntactically ambiguous in a different way. *Make* can be transitive, that is, taking a single direct object (1.2), or it can be ditransitive, that is, taking two objects (1.5), meaning that the first object (*her*) got made into the second object (*duck*). Finally, *make* can take a direct object and a verb (1.4), meaning that the object (*her*) got caused to perform the verbal action (*duck*). Furthermore, in a spoken sentence, there is an even deeper kind of ambiguity; the first word could have been *eye* or the second word *maid*.

We will often introduce the models and algorithms we present throughout the book as ways to **resolve** or **disambiguate** these ambiguities. For

example deciding whether *duck* is a verb or a noun can be solved by **part-of-speech tagging**. Deciding whether *make* means “create” or “cook” can be solved by **word sense disambiguation**. Resolution of part-of-speech and word sense ambiguities are two important kinds of **lexical disambiguation**. A wide variety of tasks can be framed as lexical disambiguation problems. For example, a text-to-speech synthesis system reading the word *lead* needs to decide whether it should be pronounced as in *lead pipe* or as in *lead me on*. By contrast, deciding whether *her* and *duck* are part of the same entity (as in (1.1) or (1.4)) or are different entity (as in (1.2)) is an example of **syntactic disambiguation** and can be addressed by **probabilistic parsing**. Ambiguities that don’t arise in this particular example (like whether a given sentence is a statement or a question) will also be resolved, for example by **speech act interpretation**.

### 1.3 MODELS AND ALGORITHMS

One of the key insights of the last 50 years of research in language processing is that the various kinds of knowledge described in the last sections can be captured through the use of a small number of formal models, or theories. Fortunately, these models and theories are all drawn from the standard toolkits of Computer Science, Mathematics, and Linguistics and should be generally familiar to those trained in those fields. Among the most important elements in this toolkit are **state machines**, **formal rule systems**, **logic**, as well as **probability theory** and other machine learning tools. These models, in turn, lend themselves to a small number of algorithms from well-known computational paradigms. Among the most important of these are **state space search** algorithms and **dynamic programming** algorithms.

In their simplest formulation, state machines are formal models that consist of states, transitions among states, and an input representation. Some of the variations of this basic model that we will consider are **deterministic** and **non-deterministic finite-state automata**, **finite-state transducers**, which can write to an output device, **weighted automata**, **Markov models**, and **hidden Markov models**, which have a probabilistic component.

Closely related to these somewhat procedural models are their declarative counterparts: formal rule systems. Among the more important ones we will consider are **regular grammars** and **regular relations**, **context-free grammars**, **feature-augmented grammars**, as well as probabilistic variants of them all. State machines and formal rule systems are the main tools

used when dealing with knowledge of phonology, morphology, and syntax.

The algorithms associated with both state-machines and formal rule systems typically involve a search through a space of states representing hypotheses about an input. Representative tasks include searching through a space of phonological sequences for a likely input word in speech recognition, or searching through a space of trees for the correct syntactic parse of an input sentence. Among the algorithms that are often used for these tasks are well-known graph algorithms such as **depth-first search**, as well as heuristic variants such as **best-first**, and **A\* search**. The dynamic programming paradigm is critical to the computational tractability of many of these approaches by ensuring that redundant computations are avoided.

The third model that plays a critical role in capturing knowledge of language is logic. We will discuss **first order logic**, also known as the **predicate calculus**, as well as such related formalisms as feature-structures, semantic networks, and conceptual dependency. These logical representations have traditionally been the tool of choice when dealing with knowledge of semantics, pragmatics, and discourse (although, as we will see, applications in these areas are increasingly relying on the simpler mechanisms used in phonology, morphology, and syntax).

Probability theory is the final element in our set of techniques for capturing linguistic knowledge. Each of the other models (state machines, formal rule systems, and logic) can be augmented with probabilities. One major use of probability theory is to solve the many kinds of ambiguity problems that we discussed earlier; almost any speech and language processing problem can be recast as: “given  $N$  choices for some ambiguous input, choose the most probable one”.

Another major advantage of probabilistic models is that they are one of a class of **machine learning** models. Machine learning research has focused on ways to automatically learn the various representations described above; automata, rule systems, search heuristics, classifiers. These systems can be trained on large corpora and can be used as a powerful modeling technique, especially in places where we don’t yet have good causal models. Machine learning algorithms will be described throughout the book.

## 1.4 LANGUAGE, THOUGHT, AND UNDERSTANDING

To many, the ability of computers to process language as skillfully as we do will signal the arrival of truly intelligent machines. The basis of this belief is

the fact that the effective use of language is intertwined with our general cognitive abilities. Among the first to consider the computational implications of this intimate connection was Alan Turing (1950). In this famous paper, Turing introduced what has come to be known as the **Turing Test**. Turing began with the thesis that the question of what it would mean for a machine to think was essentially unanswerable due to the inherent imprecision in the terms *machine* and *think*. Instead, he suggested an empirical test, a game, in which a computer's use of language would form the basis for determining if it could think. If the machine could win the game it would be judged intelligent.

TURING TEST

In Turing's game, there are three participants: two people and a computer. One of the people is a contestant and plays the role of an interrogator. To win, the interrogator must determine which of the other two participants is the machine by asking a series of questions via a teletype. The task of the machine is to fool the interrogator into believing it is a person by responding as a person would to the interrogator's questions. The task of the second human participant is to convince the interrogator that the other participant is the machine, and that they are human.

The following interaction from Turing's paper illustrates the kind of interactions he had in mind. Clearly a convincing simulation of humanity doesn't require being an expert in all domains:

Q: Please write me a sonnet on the topic of the Forth Bridge.

A: Count me out on this one. I never could write poetry.

Q: Add 34957 to 70764.

A: (Pause about 30 seconds and then give answer as) 105621.

Turing predicted that roughly by the end of the twentieth century a machine with 10 gigabytes of memory would have around a 30% chance of fooling a human interrogator after 5 minutes of questioning. Given the fact that you can fool some of the people all the time, it is not clear how rigorous this particular standard is. Nevertheless, the critical issue for Turing was that using language as humans do is sufficient, by itself, as an operational test for intelligence.

Surprisingly, many of the issues raised by Turing became relevant as early as 1966 in the context of the ELIZA program (Weizenbaum, 1966). ELIZA was an early natural language processing system capable of carrying on a limited form of conversation with a user. Consider the following session with a version of ELIZA that imitated the responses of a Rogerian psychotherapist.

User<sub>1</sub>: You are like my father in some ways.  
ELIZA<sub>1</sub>: WHAT RESEMBLANCE DO YOU SEE  
User<sub>2</sub>: You are not very aggressive but I think you don't want me to notice that.  
ELIZA<sub>2</sub>: WHAT MAKES YOU THINK I AM NOT AGGRESSIVE  
User<sub>3</sub>: You don't argue with me.  
ELIZA<sub>3</sub>: WHY DO YOU THINK I DON'T ARGUE WITH YOU  
User<sub>4</sub>: You are afraid of me.  
ELIZA<sub>4</sub>: DOES IT PLEASE YOU TO BELIEVE I AM AFRAID OF YOU

As we will discuss in more detail in Chapter 2, ELIZA is a remarkably simple program that makes use of pattern-matching to process the input and translate it into suitable outputs. The success of this simple technique in this domain is due to the fact that ELIZA doesn't actually need to *know* anything to mimic a Rogerian psychotherapist. As Weizenbaum notes, this is one of the few dialogue genres where the listener can act as if they know nothing of the world.

ELIZA's deep relevance to Turing's ideas is that many people who interacted with ELIZA came to believe that it really *understood* them and their problems. Indeed, Weizenbaum (1976) notes that many of these people continued to believe in ELIZA's abilities even after the program's operation was explained to them. In more recent years, Weizenbaum's informal reports have been repeated in a somewhat more controlled setting. Since 1991, an event known as the Loebner Prize competition has attempted to put various computer programs to the Turing test. Although these contests have proven to have little scientific interest, a consistent result over the years has been that even the crudest programs can fool some of the judges some of the time (Shieber, 1994). Not surprisingly, these results have done nothing to quell the ongoing debate over the suitability of the Turing test as a test for intelligence among philosophers and AI researchers (Searle, 1980).

Fortunately, for the purposes of this book, the relevance of these results does not hinge on whether or not computers will ever be intelligent, or understand natural language. Far more important is recent related research in the social sciences that has confirmed another of Turing's predictions from the same paper.

Nevertheless I believe that at the end of the century the use of words and educated opinion will have altered so much that we will be able to speak of machines thinking without expecting to be contradicted.

It is now clear that regardless of what people believe or know about the inner workings of computers, they talk about them and interact with them as

social entities. People act toward computers as if they were people; they are polite to them, treat them as team members, and expect among other things that computers should be able to understand their needs, and be capable of interacting with them naturally. For example, Reeves and Nass (1996) found that when a computer asked a human to evaluate how well the computer had been doing, the human gives more positive responses than when a different computer asks the same questions. People seemed to be afraid of being impolite. In a different experiment, Reeves and Nass found that people also give computers higher performance ratings if the computer has recently said something flattering to the human. Given these predispositions, speech and language-based systems may provide many users with the most natural interface for many applications. This fact has led to a long-term focus in the field on the design of **conversational agents**, artificial entities that communicate conversationally.

## 1.5 THE STATE OF THE ART AND THE NEAR-TERM FUTURE

We can only see a short distance ahead, but we can see plenty there that needs to be done.

Alan Turing.

This is an exciting time for the field of speech and language processing. The recent commercialization of robust speech recognition systems, and the rise of the Web, have placed speech and language processing applications in the spotlight, and have pointed out a plethora of exciting possible applications. The following scenarios serve to illustrate some current applications and near-term possibilities.

A Canadian computer program accepts daily weather data and generates weather reports that are passed along unedited to the public in English and French (Chandioux, 1976).

The *Babel Fish* translation system from Systran handles over 1,000,000 translation requests a day from the AltaVista search engine site.

A visitor to Cambridge, Massachusetts, asks a computer about places to eat using only spoken language. The system returns relevant information from a database of facts about the local restaurant scene (Zue et al., 1991).

These scenarios represent just a few of applications possible given current technology. The following, somewhat more speculative scenarios, give

some feeling for applications currently being explored at research and development labs around the world.

A computer reads hundreds of typed student essays and grades them in a manner that is indistinguishable from human graders (Landauer et al., 1997).

An automated reading tutor helps improve literacy by having children read stories and using a speech recognizer to intervene when the reader asks for reading help or makes mistakes (Mostow and Aist, 1999).

A computer equipped with a vision system watches a short video clip of a soccer match and provides an automated natural language report on the game (Wahlster, 1989).

A computer predicts upcoming words or expands telegraphic speech to assist people with a speech or communication disability (Newell et al., 1998; McCoy et al., 1998).

## 1.6 SOME BRIEF HISTORY

Historically, speech and language processing has been treated very differently in computer science, electrical engineering, linguistics, and psychology/cognitive science. Because of this diversity, speech and language processing encompasses a number of different but overlapping fields in these different departments: **computational linguistics** in linguistics, **natural language processing** in computer science, **speech recognition** in electrical engineering, **computational psycholinguistics** in psychology. This section summarizes the different historical threads which have given rise to the field of speech and language processing. This section will provide only a sketch; see the individual chapters for more detail on each area and its terminology.

### **Foundational Insights: 1940s and 1950s**

The earliest roots of the field date to the intellectually fertile period just after World War II that gave rise to the computer itself. This period from the 1940s through the end of the 1950s saw intense work on two foundational paradigms: the **automaton** and **probabilistic** or **information-theoretic models**.

The automaton arose in the 1950s out of Turing's (1936) model of algorithmic computation, considered by many to be the foundation of modern computer science. Turing's work led first to the **McCulloch-Pitts neuron** (McCulloch and Pitts, 1943), a simplified model of the neuron as a kind of

computing element that could be described in terms of propositional logic, and then to the work of Kleene (1951) and (1956) on finite automata and regular expressions. Shannon (1948) applied probabilistic models of discrete Markov processes to automata for language. Drawing the idea of a finite-state Markov process from Shannon's work, Chomsky (1956) first considered finite-state machines as a way to characterize a grammar, and defined a finite-state language as a language generated by a finite-state grammar. These early models led to the field of **formal language theory**, which used algebra and set theory to define formal languages as sequences of symbols. This includes the context-free grammar, first defined by Chomsky (1956) for natural languages but independently discovered by Backus (1959) and Naur et al. (1960) in their descriptions of the ALGOL programming language.

The second foundational insight of this period was the development of probabilistic algorithms for speech and language processing, which dates to Shannon's other contribution: the metaphor of the **noisy channel** and **decoding** for the transmission of language through media like communication channels and speech acoustics. Shannon also borrowed the concept of **entropy** from thermodynamics as a way of measuring the information capacity of a channel, or the information content of a language, and performed the first measure of the entropy of English using probabilistic techniques.

It was also during this early period that the sound spectrograph was developed (Koenig et al., 1946), and foundational research was done in instrumental phonetics that laid the groundwork for later work in speech recognition. This led to the first machine speech recognizers in the early 1950s. In 1952, researchers at Bell Labs built a statistical system that could recognize any of the 10 digits from a single speaker (Davis et al., 1952). The system had 10 speaker-dependent stored patterns roughly representing the first two vowel formants in the digits. They achieved 97–99% accuracy by choosing the pattern which had the highest relative correlation coefficient with the input.

### **The Two Camps: 1957–1970**

By the end of the 1950s and the early 1960s, speech and language processing had split very cleanly into two paradigms: symbolic and stochastic.

The symbolic paradigm took off from two lines of research. The first was the work of Chomsky and others on formal language theory and generative syntax throughout the late 1950s and early to mid 1960s, and the work of many linguistics and computer scientists on parsing algorithms, initially top-down and bottom-up and then via dynamic programming. One of the earliest

complete parsing systems was Zelig Harris's Transformations and Discourse Analysis Project (TDAP), which was implemented between June 1958 and July 1959 at the University of Pennsylvania (Harris, 1962).<sup>2</sup> The second line of research was the new field of artificial intelligence. In the summer of 1956 John McCarthy, Marvin Minsky, Claude Shannon, and Nathaniel Rochester brought together a group of researchers for a two-month workshop on what they decided to call artificial intelligence (AI). Although AI always included a minority of researchers focusing on stochastic and statistical algorithms (include probabilistic models and neural nets), the major focus of the new field was the work on reasoning and logic typified by Newell and Simon's work on the Logic Theorist and the General Problem Solver. At this point early natural language understanding systems were built. These were simple systems that worked in single domains mainly by a combination of pattern matching and keyword search with simple heuristics for reasoning and question-answering. By the late 1960s more formal logical systems were developed.

The stochastic paradigm took hold mainly in departments of statistics and of electrical engineering. By the late 1950s the Bayesian method was beginning to be applied to the problem of optical character recognition. Bledsoe and Browning (1959) built a Bayesian system for text-recognition that used a large dictionary and computed the likelihood of each observed letter sequence given each word in the dictionary by multiplying the likelihoods for each letter. Mosteller and Wallace (1964) applied Bayesian methods to the problem of authorship attribution on *The Federalist* papers.

The 1960s also saw the rise of the first serious testable psychological models of human language processing based on transformational grammar, as well as the first on-line corpora: the Brown corpus of American English, a 1 million word collection of samples from 500 written texts from different genres (newspaper, novels, non-fiction, academic, etc.), which was assembled at Brown University in 1963–64 (Kučera and Francis, 1967; Francis, 1979; Francis and Kučera, 1982), and William S. Y. Wang's 1967 DOC (Dictionary on Computer), an on-line Chinese dialect dictionary.

### **Four Paradigms: 1970–1983**

The next period saw an explosion in research in speech and language processing and the development of a number of research paradigms that still dominate the field.

---

<sup>2</sup> This system was reimplemented recently and is described by Joshi and Hopely (1999) and Karttunen (1999), who note that the parser was essentially implemented as a cascade of finite-state transducers.

The **stochastic** paradigm played a huge role in the development of speech recognition algorithms in this period, particularly the use of the Hidden Markov Model and the metaphors of the noisy channel and decoding, developed independently by Jelinek, Bahl, Mercer, and colleagues at IBM's Thomas J. Watson Research Center, and by Baker at Carnegie Mellon University, who was influenced by the work of Baum and colleagues at the Institute for Defense Analyses in Princeton. AT&T's Bell Laboratories was also a center for work on speech recognition and synthesis; see Rabiner and Juang (1993) for descriptions of the wide range of this work.

The **logic-based** paradigm was begun by the work of Colmerauer and his colleagues on Q-systems and metamorphosis grammars (Colmerauer, 1970, 1975), the forerunners of Prolog, and Definite Clause Grammars (Pereira and Warren, 1980). Independently, Kay's (1979) work on functional grammar, and shortly later, Bresnan and Kaplan's (1982) work on LFG, established the importance of feature structure unification.

The **natural language understanding** field took off during this period, beginning with Terry Winograd's SHRDLU system, which simulated a robot embedded in a world of toy blocks (Winograd, 1972a). The program was able to accept natural language text commands (*Move the red block on top of the smaller green one*) of a hitherto unseen complexity and sophistication. His system was also the first to attempt to build an extensive (for the time) grammar of English, based on Halliday's systemic grammar. Winograd's model made it clear that the problem of parsing was well-enough understood to begin to focus on semantics and discourse models. Roger Schank and his colleagues and students (in what was often referred to as the *Yale School*) built a series of language understanding programs that focused on human conceptual knowledge such as scripts, plans and goals, and human memory organization (Schank and Albelson, 1977; Schank and Riesbeck, 1981; Cullingford, 1981; Wilensky, 1983; Lehnert, 1977). This work often used network-based semantics (Quillian, 1968; Norman and Rumelhart, 1975; Schank, 1972; Wilks, 1975c, 1975b; Kintsch, 1974) and began to incorporate Fillmore's notion of case roles (Fillmore, 1968) into their representations (Simmons, 1973).

The logic-based and natural-language understanding paradigms were unified on systems that used predicate logic as a semantic representation, such as the LUNAR question-answering system (Woods, 1967, 1973).

The **discourse modeling** paradigm focused on four key areas in discourse. Grosz and her colleagues introduced the study of substructure in discourse, and of discourse focus (Grosz, 1977a; Sidner, 1983), a number of

researchers began to work on automatic reference resolution (Hobbs, 1978), and the **BDI** (Belief-Desire-Intention) framework for logic-based work on speech acts was developed (Perrault and Allen, 1980; Cohen and Perrault, 1979).

### **Empiricism and Finite State Models Redux: 1983–1993**

This next decade saw the return of two classes of models which had lost popularity in the late 1950s and early 1960s, partially due to theoretical arguments against them such as Chomsky’s influential review of Skinner’s *Verbal Behavior* (Chomsky, 1959b). The first class was finite-state models, which began to receive attention again after work on finite-state phonology and morphology by Kaplan and Kay (1981) and finite-state models of syntax by Church (1980). A large body of work on finite-state models will be described throughout the book.

The second trend in this period was what has been called the “return of empiricism”; most notably here was the rise of probabilistic models throughout speech and language processing, influenced strongly by the work at the IBM Thomas J. Watson Research Center on probabilistic models of speech recognition. These probabilistic methods and other such data-driven approaches spread into part-of-speech tagging, parsing and attachment ambiguities, and connectionist approaches from speech recognition to semantics.

This period also saw considerable work on natural language generation.

### **The Field Comes Together: 1994–1999**

By the last five years of the millennium it was clear that the field was vastly changing. First, probabilistic and data-driven models had become quite standard throughout natural language processing. Algorithms for parsing, part-of-speech tagging, reference resolution, and discourse processing all began to incorporate probabilities, and employ evaluation methodologies borrowed from speech recognition and information retrieval. Second, the increases in the speed and memory of computers had allowed commercial exploitation of a number of subareas of speech and language processing, in particular speech recognition and spelling and grammar checking. Speech and language processing algorithms began to be applied to Augmentative and Alternative Communication (AAC). Finally, the rise of the Web emphasized the need for language-based information retrieval and information extraction.

## On Multiple Discoveries

Even in this brief historical overview, we have mentioned a number of cases of multiple independent discoveries of the same idea. Just a few of the “multiples” to be discussed in this book include the application of dynamic programming to sequence comparison by Viterbi, Vintsyuk, Needleman and Wunsch, Sakoe and Chiba, Sankoff, Reichert *et al.*, and Wagner and Fischer (Chapters 5 and 7); the HMM/noisy channel model of speech recognition by Baker and by Jelinek, Bahl, and Mercer (Chapter 7); the development of context-free grammars by Chomsky and by Backus and Naur (Chapter 9); the proof that Swiss-German has a non-context-free syntax by Huybregts and by Shieber (Chapter 13); the application of unification to language processing by Colmerauer *et al.* and by Kay in (Chapter 11).

Are these multiples to be considered astonishing coincidences? A well-known hypothesis by sociologist of science Robert K. Merton (1961) argues, quite the contrary, that

all scientific discoveries are in principle multiples, including those that on the surface appear to be singletons.

Of course there are many well-known cases of multiple discovery or invention; just a few examples from an extensive list in Ogburn and Thomas (1922) include the multiple invention of the calculus by Leibnitz and by Newton, the multiple development of the theory of natural selection by Wallace and by Darwin, and the multiple invention of the telephone by Gray and Bell.<sup>3</sup> But Merton gives an further array of evidence for the hypothesis that multiple discovery is the rule rather than the exception, including many cases of putative singletons that turn out be a rediscovery of previously unpublished or perhaps inaccessible work. An even stronger piece of evidence is his ethnomethodological point that scientists themselves act under the assumption that multiple invention is the norm. Thus many aspects of scientific life are designed to help scientists avoid being “scooped”; submission dates on journal articles; careful dates in research records; circulation of preliminary or technical reports.

---

<sup>3</sup> Ogburn and Thomas are generally credited with noticing that the prevalence of multiple inventions suggests that the cultural milieu and not individual genius is the deciding causal factor in scientific discovery. In an amusing bit of recursion, however, Merton notes that even this idea has been multiply discovered, citing sources from the 19th century and earlier!

### **A Final Brief Note on Psychology**

Many of the chapters in this book include short summaries of psychological research on human processing. Of course, understanding human language processing is an important scientific goal in its own right and is part of the general field of cognitive science. However, an understanding of human language processing can often be helpful in building better machine models of language. This seems contrary to the popular wisdom, which holds that direct mimicry of nature's algorithms is rarely useful in engineering applications. For example, the argument is often made that if we copied nature exactly, airplanes would flap their wings; yet airplanes with fixed wings are a more successful engineering solution. But language is not aeronautics. Cribbing from nature is sometimes useful for aeronautics (after all, airplanes do have wings), but it is particularly useful when we are trying to solve human-centered tasks. Airplane flight has different goals than bird flight; but the goal of speech recognition systems, for example, is to perform exactly the task that human court reporters perform every day: transcribe spoken dialog. Since people already do this well, we can learn from nature's previous solution. Since an important application of speech and language processing systems is for human-computer interaction, it makes sense to copy a solution that behaves the way people are accustomed to.

## **1.7 SUMMARY**

This chapter introduces the field of speech and language processing. The following are some of the highlights of this chapter.

- A good way to understand the concerns of speech and language processing research is to consider what it would take to create an intelligent agent like HAL from 2001: A Space Odyssey.
- Speech and language technology relies on formal models, or representations, of knowledge of language at the levels of phonology and phonetics, morphology, syntax, semantics, pragmatics and discourse. A small number of formal models including state machines, formal rule systems, logic, and probability theory are used to capture this knowledge.
- The foundations of speech and language technology lie in computer science, linguistics, mathematics, electrical engineering and psychology. A small number of algorithms from standard frameworks are used

throughout speech and language processing,

- The critical connection between language and thought has placed speech and language processing technology at the center of debate over intelligent machines. Furthermore, research on how people interact with complex media indicates that speech and language processing technology will be critical in the development of future technologies.
- Revolutionary applications of speech and language processing are currently in use around the world. Recent advances in speech recognition and the creation of the World-Wide Web will lead to many more applications.

## BIBLIOGRAPHICAL AND HISTORICAL NOTES

Research in the various subareas of speech and language processing is spread across a wide number of conference proceedings and journals. The conferences and journals most centrally concerned with computational linguistics and natural language processing are associated with the Association for Computational Linguistics (ACL), its European counterpart (EACL), and the International Conference on Computational Linguistics (COLING). The annual proceedings of ACL and EACL, and the biennial COLING conference are the primary forums for work in this area. Related conferences include the biennial conference on Applied Natural Language Processing (ANLP) and the conference on Empirical Methods in Natural Language Processing (EMNLP). The journal *Computational Linguistics* is the premier publication in the field, although it has a decidedly theoretical and linguistic orientation. The journal *Natural Language Engineering* covers more practical applications of speech and language research.

Research on speech recognition, understanding, and synthesis is presented at the biennial International Conference on Spoken Language Processing (ICSLP) which alternates with the European Conference on Speech Communication and Technology (EUROSPEECH). The IEEE International Conference on Acoustics, Speech, and Signal Processing (IEEE ICASSP) is held annually, as is the meeting of the Acoustical Society of America. Speech journals include *Speech Communication*, *Computer Speech and Language*, and the *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Work on language processing from an Artificial Intelligence perspective can be found in the annual meetings of the American Association for Artificial Intelligence (AAAI), as well as the biennial International Joint Conference on Artificial Intelligence (IJCAI) meetings. The following artificial intelligence publications periodically feature work on speech and language processing: *Artificial Intelligence*, *Computational Intelligence*, *IEEE Transactions on Intelligent Systems*, and the *Journal of Artificial Intelligence Research*. Work on cognitive modeling of language can be found at the annual meeting of the Cognitive Science Society, as well as its journal *Cognitive Science*. An influential series of invitation-only workshops was held by ARPA, called variously the *DARPA Speech and Natural Language Processing Workshop* or the *ARPA Workshop on Human Language Technology*.

There are a fair number of textbooks available covering various aspects of speech and language processing. Manning and Schütze (1999) (*Foundations of Statistical Language Processing*) focuses on statistical models of tagging, parsing, disambiguation, collocations, and other areas. Charniak (1993) (*Statistical Language Learning*) is an accessible, though older and less-extensive, introduction to similar material. Allen (1995) (*Natural Language Understanding*) provides extensive coverage of language processing from the AI perspective. Gazdar and Mellish (1989) (*Natural Language Processing in Lisp/Prolog*) covers especially automata, parsing, features, and unification. Pereira and Shieber (1987) gives a Prolog-based introduction to parsing and interpretation. Russell and Norvig (1995) is an introduction to artificial intelligence that includes chapters on natural language processing. Partee et al. (1990) has a very broad coverage of mathematical linguistics. Cole (1997) is a volume of survey papers covering the entire field of speech and language processing. A somewhat dated but still tremendously useful collection of foundational papers can be found in Grosz et al. (1986) (*Readings in Natural Language Processing*).

Of course, a wide-variety of speech and language processing resources are now available on the World-Wide Web. Pointers to these resources are maintained on the home-page for this book at:

<http://www.cs.colorado.edu/~martin/slp.html>.