PRELIMINARY PROOFS.
Unpublished Work ©2008 by Pearson Education, Inc. To be published by Pearson Prentice Hall, Pearson Education, Inc., Upper Saddle River, New Jersey. All rights reserved. Permission to use this unpublished Work is granted to individuals registering through Melinda_Haggerty@prenhall.com for the instructional purposes not exceeding one academic term or semester.

Chapter 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 idea of giving computers the ability to process human language is as old as the idea of computers themselves. This book is about the implementation and implications of that exciting idea. We introduce a vibrant interdisciplinary field with many names corresponding to its many facets, names like **speech and language processing**, **human language technology**, **natural language processing**, **computational linguistics**, and **speech recognition and synthesis**. The goal of this new field is to get computers to perform useful tasks involving human language, tasks like enabling human-machine communication, improving human-human communication, or simply doing useful processing of text or speech.

Conversational agent

One example of a useful such task is a **conversational agent**. 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? We call programs like HAL that converse with humans via natural language **conversational agents** or **dialogue systems**. In this text we study the various components that make up modern conversational agents, including language input (**automatic speech recognition** and **natural language understanding**) and language output (**natural language generation** and **speech synthesis**).

Dialogue system

Machine translation

Let's turn to another useful language-related task, that of making available to non-English-speaking readers the vast amount of scientific information on the Web in English. Or translating for English speakers the hundreds of millions of Web pages written in other languages like Chinese. The goal of **machine translation** is to automatically translate a document from one language to another. We will introduce the algorithms and mathematical tools needed to understand how modern machine translation works. Machine translation is far from a solved problem; we will cover the algorithms currently used in the field, as well as important component tasks.

Question answering

Many other language processing tasks are also related to the Web. Another such task is **Web-based question answering**. This is a generalization of simple web search, where instead of just typing keywords a user might ask complete questions, ranging from easy to hard, like the following:

- What does "divergent" mean?
- What year was Abraham Lincoln born?
- How many states were in the United States that year?

- How much Chinese silk was exported to England by the end of the 18th century?
- What do scientists think about the ethics of human cloning?

Some of these, such as **definition** questions, or simple **factoid** questions like dates and locations, can already be answered by search engines. But answering more complicated questions might require extracting information that is embedded in other text on a Web page, or doing **inference** (drawing conclusions based on known facts), or synthesizing and summarizing information from multiple sources or web pages. In this text we study the various components that make up modern understanding systems of this kind, including **information extraction**, **word sense disambiguation**, and so on.

Although the subfields and problems we've described above are all very far from completely solved, these are all very active research areas and many technologies are already available commercially. In the rest of this chapter we briefly summarize the kinds of knowledge that is necessary for these tasks (and others like **spell correction**, **grammar checking**, and so on), as well as the mathematical models that will be introduced throughout the book.

1.1 Knowledge in Speech and Language Processing

What distinguishes 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. Sophisticated conversational agents like HAL, or machine translation systems, or robust question-answering systems, require much broader and deeper knowledge of language. To get a feeling for the scope and kind of required knowledge, consider some of what HAL would need to know to engage in the dialogue that begins this chapter, or for a question answering system to answer one of the questions above.

HAL must be able to recognize words from an audio signal and to generate an audio signal from a sequence of words. These tasks of **speech recognition** and **speech synthesis** tasks require knowledge about **phonetics and phonology**; how words are pronounced in terms of sequences of sounds, and how each of these sounds is realized acoustically.

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**, the way words break down into component parts that carry meanings like *singular* versus *plural*.

Moving beyond individual words, HAL must use 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.

(1.1) 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**.

Now consider a question answering system dealing with the following question:

(1.2) How much Chinese silk was exported to Western Europe by the end of the 18th century?

In order to answer this question we need to know something about **lexical semantics**, the meaning of all the words (*export*, or *silk*) as well as **compositional semantics** (what exactly constitutes *Western Europe* as opposed to Eastern or Southern Europe, what does *end* mean when combined with *the 18th century*. We also need to know something about the relationship of the words to the syntactic structure. For example we need to know that *by the end of the 18th century* is a temporal end-point, and not a description of the agent, as the by-phrase is in the following sentence:

(1.3) How much Chinese silk was exported to Western Europe by southern merchants?

We also need the kind of knowledge that lets HAL 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.

REQUEST:	*HAL, open the pod bay door.*
STATEMENT:	*HAL, the pod bay door is open.*
INFORMATION QUESTION:	*HAL, is the pod bay door open?*

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.*[1] This knowledge about the kind of actions that speakers intend by their use of sentences is **pragmatic** or **dialogue** knowledge.

Another kind of pragmatic or **discourse** knowledge is required to answer the question

(1.4) How many states were in the United States *that year*?

What year is *that year*? In order to interpret words like *that year* a question answering system needs to examine the earlier questions that were asked; in this case the previous question talked about the year that Lincoln was born. Thus this task of **coreference resolution** makes use of knowledge about how words like *that* or pronouns like *it* or *she* refer to previous parts of the **discourse**.

To summarize, engaging in complex language behavior requires various kinds of knowledge of language:

[1] 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.

DRAFT

- Phonetics and Phonology — knowledge about linguistic sounds
- Morphology — knowledge of the meaningful components of words
- Syntax — knowledge of the structural relationships between words
- Semantics — knowledge of meaning
- Pragmatics — knowledge of the relationship of meaning to the goals and intentions of the speaker.
- Discourse — knowledge about linguistic units larger than a single utterance

1.2 Ambiguity

Ambiguity
Ambiguous

A perhaps surprising fact about these categories of linguistic knowledge is that most 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 that can be built for it. Consider the spoken sentence *I made her duck.* Here are five different meanings this sentence could have (see if you can think of some more), each of which exemplifies an ambiguity at some level:

(1.5) I cooked waterfowl for her.
(1.6) I cooked waterfowl belonging to her.
(1.7) I created the (plaster?) duck she owns.
(1.8) I caused her to quickly lower her head or body.
(1.9) 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 (1.6), or it can be ditransitive, that is, taking two objects (1.9), meaning that the first object (*her*) got made into the second object (*duck*). Finally, *make* can take a direct object and a verb (1.8), 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 (1.5) or (1.8)) or are different entity (as in (1.6)) is an example of **syntactic disambiguation** and can

DRAFT

be addressed by **probabilistic parsing**. We will also consider ambiguities that don't arise in this particular example, such as determining whether a sentence is a statement or a question (which can be resolved 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 models are **state machines**, **rule systems**, **logic**, **probabilistic models**, and **vector-space models**. These models, in turn, lend themselves to a small number of algorithms, among the most important of which are **state space search** algorithms such as **dynamic programming**, and machine learning algorithms such as **classifiers** and Expectation-Maximization (**EM**) and other learning 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** and **finite-state transducers**.

Closely related to these models are their declarative counterparts: formal rule systems. Among the more important ones we will consider (in both probabilistic and non-probabilistic formulations) are **regular grammars** and **regular relations**, **context-free grammars**, and **feature-augmented grammars**. State machines and formal rule systems are the main tools used when dealing with knowledge of phonology, morphology, and syntax.

A third class of models that plays a critical role in capturing knowledge of language are models based on logic. We will discuss **first order logic**, also known as the **predicate calculus**, as well as such related formalisms as lambda-calculus, feature-structures, and semantic primitives. These logical representations have traditionally been used for modeling semantics and pragmatics, although more recent work has tended to focus on potentially more robust techniques drawn from non-logical lexical semantics.

Probabilistic models are crucial for capturing every kind of linguistic knowledge. Each of the other models (state machines, formal rule systems, and logic) can be augmented with probabilities. For example the state machine can be augmented with probabilities to become the **weighted automaton** or **Markov model**. We will spend a significant amount of time on **hidden Markov models** or **HMMs**, which are used everywhere in the field, in part-of-speech tagging, speech recognition, dialogue understanding, text-to-speech, and machine translation. The key advantage of probabilistic models is their ability 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".

Finally, vector-space models, based on linear algebra, underlie information retrieval

and many treatments of word meanings.

Processing language using any of these models typically involves a search through a space of states representing hypotheses about an input. In speech recognition, we search through a space of phone sequences for the correct word. In parsing, we search through a space of trees for the syntactic parse of an input sentence. In machine translation, we search through a space of translation hypotheses for the correct translation of a sentence into another language. For non-probabilistic tasks, such as tasks involving state machines, we use well-known graph algorithms such as **depth-first search**. For probabilistic tasks, we use heuristic variants such as **best-first** and **A* search**, and rely on dynamic programming algorithms for computational tractability.

Machine learning tools such as **classifiers** and **sequence models** play a significant role in many language processing tasks. Based on attributes describing each object, a classifer attempts to assign a single object to a single class while a sequence model attempts to jointly classify a sequence of objects into a sequence of classes.

For example, in the task of deciding whether a word is spelled correctly or not, classifiers such as **decision trees**, **support vector machines**, **Gaussian Mixture Models**, and **logistic regression** could be used to make a binary decision (correct or incorrect) for one word at a time. Sequence models such as **hidden Markov models**, **maximum entropy Markov models**, and conditional random fields could be used to assign correct/incorrect labels to all the words in a sentence at once.

Finally, researchers in language processing use many of the same methodological tools that are used in machine learning research—the use of distinct training and test sets, statistical techniques like **cross-validation**, and careful evaluation of trained systems.

1.4 Language, Thought, and Understanding

Turing test

To many, the ability of computers to process language as skillfully as we humans 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.

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 she is 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.

ELIZA

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_1: You are like my father in some ways.
> ELIZA_1: WHAT RESEMBLANCE DO YOU SEE
> User_2: You are not very aggressive but I think you don't want me to notice that.
> ELIZA_2: WHAT MAKES YOU THINK I AM NOT AGGRESSIVE
> User_3: You don't argue with me.
> ELIZA_3: WHY DO YOU THINK I DON'T ARGUE WITH YOU
> User_4: You are afraid of me.
> ELIZA_4: DOES IT PLEASE YOU TO BELIEVE I AM AFRAID OF YOU

As we will discuss in more detail in Ch. 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 tests 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 seem 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, 1994a). 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

> *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 startling increase in computing resources available to the average computer user, the rise of the Web as a massive source of information and the increasing availability of wireless mobile access have all placed speech and language processing applications in the technology spotlight. The following are examples of some currently deployed systems that reflect this trend:

- Travelers calling Amtrak, United Airlines and other travel-providers interact with conversational agents that guide them through the process of making reservations and getting arrival and departure information.
- Luxury car makers such as Mercedes-Benz models provide automatic speech recognition and text-to-speech systems that allow drivers to control their environmental, entertainment and navigational systems by voice. A similar spoken dialogue system has been deployed by astronauts on the International Space Station .
- Blinkx and other video search companies provide search services for million of hours of video on the Web by using speech recognition technology to capture the words in the sound track.

DRAFT

- Google provides cross-language information retrieval and translation services where a user can supply queries in their native language to search collections in another language. Google translates the query, finds the most relevant pages and then automatically translates them back to the user's native language.
- Large educational publishers such as Pearson, as well as testing services like ETS, use automated systems to analyze thousands of student essays, grading and assessing them in a manner that is indistinguishable from human graders.
- Interactive tutors, based on lifelike animated characters, serve as tutors for children learning to read (Wise et al., 2007).
- Text analysis companies such as Nielsen Buzzmetrics, Umbria, and Collective Intellect provide marketing intelligence based on automated measurements of user opinions, preferences, attitudes as expressed in weblogs, discussion forums and user groups.

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, but many of the topics listed here will be covered in more detail in subsequent chapters.

1.6.1 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 on 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 that had the highest relative correlation coefficient with the input.

1.6.2 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).[2] 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 (including 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 simple systems 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 elec-

[2] 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.

trical 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.

1.6.3 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.

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 in 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).

1.6.4 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 from speech into part-of-speech tagging, parsing and attachment ambiguities, and semantics. This empirical direction was also accompanied by a new focus on model evaluation, based on using held-out data, developing quantitative metrics for evaluation, and emphasizing the comparison of performance on these metrics with previous published research.

This period also saw considerable work on natural language generation.

1.6.5 The Field Comes Together: 1994–1999

By the last five years of the millennium it was clear that the field was undergoing major changes. 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.

DRAFT

1.6.6 The Rise of Machine Learning: 2000–2007

The empiricist trends begun in the latter part of the 1990s accelerated at an astounding pace in the new century. This acceleration was largely driven by three synergistic trends. First, large amounts of spoken and written material became widely available through the auspices of the Linguistic Data Consortium (LDC), and other similar organizations. Importantly, included among these materials were annotated collections such as the Penn Treebank (Marcus et al., 1993), Prague Dependency Treebank (Hajič, 1998), PropBank (Palmer et al., 2005), Penn Discourse Treebank (Miltsakaki et al., 2004b), RSTBank (Carlson et al., 2001) and TimeBank (Pustejovsky et al., 2003b), all of which layered standard text sources with various forms of syntactic, semantic and pragmatic annotations. The existence of these resources promoted the trend of casting more complex traditional problems, such as parsing and semantic analysis, as problems in supervised machine learning. These resources also promoted the establishment of additional competitive evaluations for parsing (Dejean and Tjong Kim Sang, 2001), information extraction (NIST, 2007a; Sang, 2002; Sang and De Meulder, 2003), word sense disambiguation (Palmer et al., 2001a; Kilgarriff and Palmer, 2000), question answering (Voorhees and Tice, 1999), and summarization Dang (2006).

Second, this increased focus on learning led to a more serious interplay with the statistical machine learning community. Techniques such as support vector machines (Boser et al., 1992; Vapnik, 1995), maximum entropy techniques and their equivalent formulation as multinomial logistic regression (Berger et al., 1996), and graphical Bayesian models (Pearl, 1988) became standard practice in computational linguistics. Third, the widespread availability of high-performance computing systems facilitated the training and deployment of systems that could not have been imagined a decade earlier.

Finally, near the end of this period, largely unsupervised statistical approaches began to receive renewed attention. Progress on statistical approaches to machine translation (Brown et al., 1990; Och and Ney, 2003) and topic modeling (Blei et al., 2003) demonstrated that effective applications could be constructed from systems trained on unannotated data alone. In addition, the widespread cost and difficulty of producing reliably annotated corpora became a limiting factor in the use of supervised approaches for many problems. This trend towards the use of unsupervised techniques will likely increase.

1.6.7 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 3, 5 and 6) the HMM/noisy channel model of speech recognition by Baker and by Jelinek, Bahl, and Mercer (Chapters 6, 9, and 10); the development of context-free grammars by Chomsky and by Backus and Naur (Chapter 12); the proof that Swiss-German has a non-context-free syntax by Huybregts and by Shieber (Chapter 15); the application of unification to language processing by

DRAFT

Colmerauer *et al.* and by Kay in (Chapter 16).

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.[3] But Merton gives a 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.

1.6.8 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.

[3] 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!

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, or build a web-based question answerer, or a machine translation engine.
- 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 number of formal models including state machines, formal rule systems, logic, and probabilistic models 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. The creation of the web, as well as significant recent improvements in speech recognition and synthesis, 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 natural language processing and computational linguistics 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, NAACL, and EACL, and the biennial COLING conference are the primary forums for work in this area. Related conferences include various proceedings of ACL Special Interest Groups (SIGs) such as the Conference on Natural Language Learning (CoNLL), as well as the conference on Empirical Methods in Natural Language Processing (EMNLP).

Research on speech recognition, understanding, and synthesis is presented at the annual INTERSPEECH conference, which is called the International Conference on

DRAFT

Spoken Language Processing (ICSLP) and the European Conference on Speech Communication and Technology (EUROSPEECH) in alternating years, or the annual IEEE International Conference on Acoustics, Speech, and Signal Processing (IEEE ICASSP). Spoken language dialogue research is presented at these or at workshops like SIGDial.

Journals include *Computational Linguistics*, *Natural Language Engineering*, *Speech Communication*, *Computer Speech and Language*, the *IEEE Transactions on Audio, Speech & Language Processing* and the *ACM Transactions on Speech and Language Processing*.

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. Artificial intelligence journals that periodically feature work on speech and language processing include *Machine Learning*, *Journal of Machine Learning Research*, and the *Journal of Artificial Intelligence Research*.

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. Manning et al. (2008) focuses on information retrieval, text classification, and clustering. NLTK, the Natural Language Toolkit (Bird and Loper, 2004), is a suite of Python modules and data for natural language processing, together with a Natural Language Processing book based on the NLTK suite. 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 and is available free online. Pereira and Shieber (1987) gives a Prolog-based introduction to parsing and interpretation. Russell and Norvig (2002) 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. A historically significant collection of foundational papers can be found in Grosz et al. (1986) (*Readings in Natural Language Processing*).

DRAFT