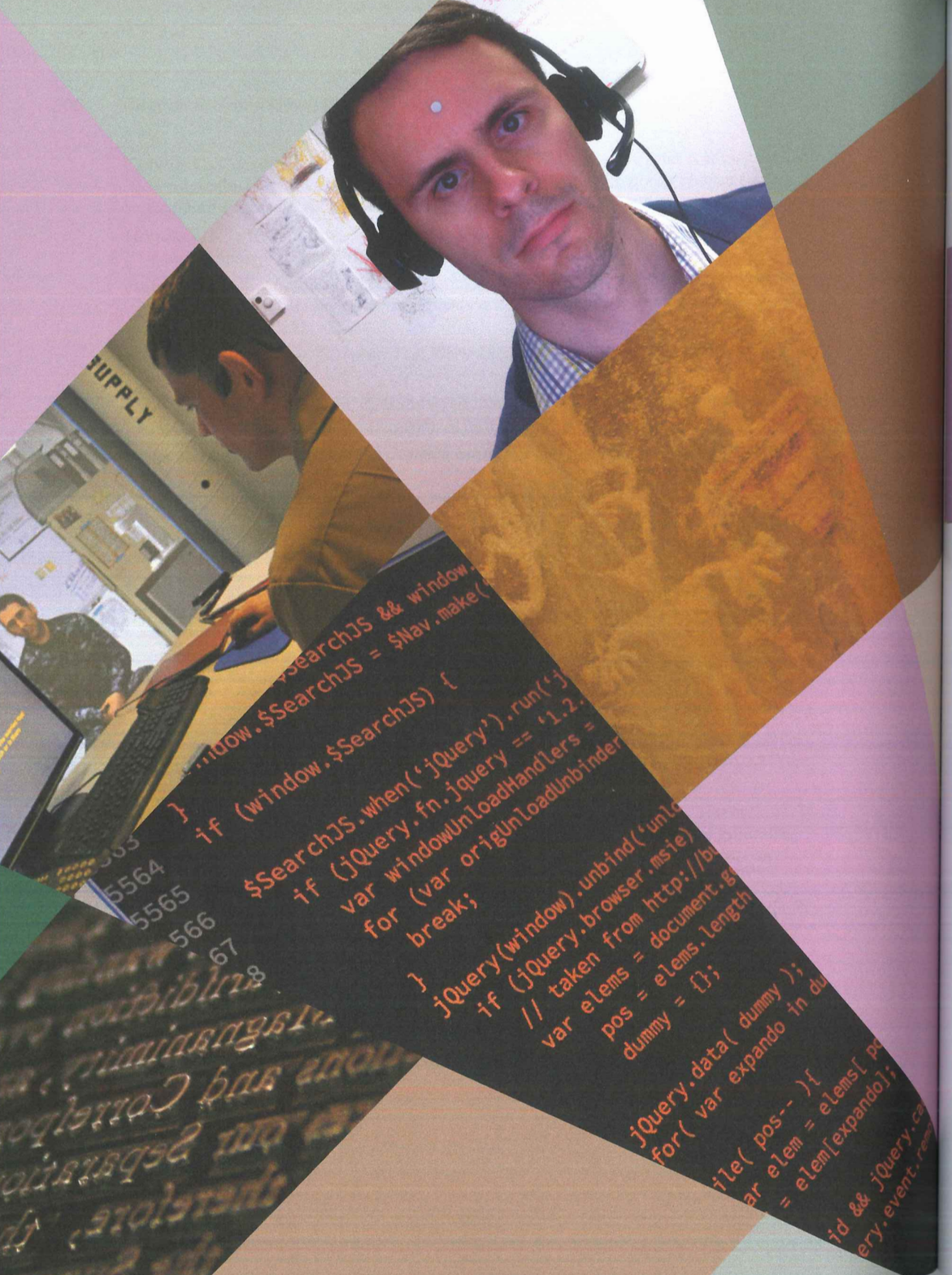

CHAPTER 9

Expressive Human and Command Languages

> "I soon felt that the forms of ordinary language were far too diffuse. . . . I was not long in deciding that the most favorable path to pursue was to have recourse to the language of signs. It then became necessary to contrive a notation which ought, if possible, to be at once simple and expressive, easily understood at the commencement, and capable of being readily retained in the memory."

Charles Babbage
"On a Method of Expressing by Signs the Action of Machinery," 1826

CHAPTER OUTLINE

9.1 Introduction
9.2 Speech Recognition
9.3 Speech Production
9.4 Human Language Technology
9.5 Traditional Command Languages

9.1 Introduction

The dream of speaking to computers and having computers speak has long lured researchers and visionaries. Arthur C. Clarke's 1968 fantasy of the HAL 9000 computer in the book and movie *2001: A Space Odyssey* has set the standard for performance of computers in science fiction and for developers of natural language systems. The reality is more complex and sometimes more frustrating than the dream, but much-improved speech recognizers have now joined the well-established speech telephone-based menu applications to reach a wide array of applications. Errors remain a significant challenge, and not all situations benefit enough from speech input to balance the cost of errors and the frustration of error correction. Once the commands, questions, or statements have been recognized, human language technologies may be needed to execute the appropriate action, initiate a clarifying dialogue, or provide translations.

Some applications simulate natural language interaction. They require users to speak a restricted set of the spoken commands that users have to learn and remember. Similarly, some textual interaction systems rely on the availability of vast text repositories that can be searched using standard search algorithms to find answers to questions written in full sentences. Repositories of translated text, such as the multiple language translations from the United Nations, can also help make good-quality translations of words, snippets, or full sentences.

See also:

Chapter 14, Documentation and User Support (a.k.a. Help)
Chapter 15, Information Search

The use of command languages in the early days of computing (e.g., DOS or Unix) receded with the advent of graphical user interfaces. However, command languages are still widely used by expert users of specialized applications from computer programmers to the millions of engineers and scientists using tools like MATLAB®, which combine a command language and graphical environment. In fact, one could argue that the spread of speech interfaces is re-invigorating the development of command languages as designers choose which combinations of words will be recognized as commands in speech interfaces.

While understanding natural language remains an unattainable dream, there are many applications that can successfully make use of the words people say, type, or listen to (Box 9.1).

This chapter starts with the rapidly growing speech interfaces (from speech recognition in Section 9.2 to speech production in Section 9.3) and then discusses human language technologies (Section 9.4) including translation educational applications. Finally, Section 9.5 reviews the traditional, yet expressive, command language interfaces.

BOX 9.1
Speech technologies.

- Store and replay (museum guides)
- Dictation (document preparation, web search)
- Close captioning, transcription
- Transactions over the phone
- Personal "assistant" (common tasks on mobile devices)
- Hands-free interaction with a device
- Adaptive technology for users with disabilities
- Translation
- Alerts
- Speaker identification

9.2 Speech Recognition

Speech recognition has made significant progress in recent years (Huang et al., 2014) and is now being used in a number of well-targeted knowledge domains such as airline information, lost luggage, medical-record data entry, and personal digital assistants (Cohen 2004; Karat et al., 2012; Pieraccini, 2012; Bouzid and Ma, 2013; Neustein and Markowitz, 2013; Mariani et al., 2014). Driven by the difficulty of typing while using mobile devices (phones or touch tablets), spoken input has gained acceptability. More users learn to use spoken commands such as "Where is the closest coffee shop?" or "Tell John I will be late." Discoverability and learnability are often an issue, but commands can be spoken without looking at the screen, while driving a car (equipped with a hands-free phone), or while hiking on a bumpy trail. However, commands such as "Make space in my drive" are still a great challenge and would require extensive dialog design (see Section 9.4 on human language technology). Improved recognition rates are making dictation and transcription possible, but error correction remains a challenge, and most applications require users to learn and remember complex sets of commands to accomplish their tasks. Background noise and variations in user speech performance make the challenge of speech recognition still greater.

9.2.1 The place for spoken interaction

While speech recognition is used successfully in a growing number of applications, the vision of computers chatting leisurely with users about varied open-ended topics remains more of a fantasy than a reality. While HAL 9000 of *2001: A Space Odyssey* communicated with the ship crew mostly by voice, newer science-fiction writers have shifted their scenarios, with reduced use of spoken interaction in favor of larger visual displays and gestures, from *Star Trek: Voyager* to *Minority Report* and *Avatar* or *Mission Impossible 4*. Voice interaction with emotion-evoking robots remains a theme in movies such as *Her* and *Ex Machina*.

While early applications of speech recognition were mostly limited to discrete-word recognition (with extensive training for the system to learn a particular user's voice), the major breakthrough in the past decade has been the improvement of continuous-speech recognition algorithms and the availability of very large repositories of voice data on the web, which can be analyzed to train algorithms. The other significant advance that made speech recognition possible on mobile devices is the ability to process the spoken input remotely and quickly enough for rapid interaction. Reduced training (or its elimination with speaker-independent systems) has greatly expanded the scope of commercial applications. Quiet environments, head-mounted high-quality microphones, and careful choice of vocabularies improve recognition rates in all cases. Low-cost speech chips and compact microphones and speakers enable designers to include speech systems in high-volume products, such as dolls and other toys.

Applications are successful when certain conditions exist (see Box 9.2) and when they serve users' needs to work rapidly with low cognitive load and low error rates. Even as technical problems are being solved and the recognition rates are improving, spoken commands are more demanding of users' working memory than is hand/eye coordination and thus may be more disruptive to users while they are carrying out tasks. Speech requires use of limited resources, while hand/eye coordination is processed elsewhere in the brain, enabling a higher level of parallel processing. Planning and problem solving can proceed in parallel with hand/eye coordination, but they are more difficult to accomplish while speaking (Radvansky and Ashcraft, 2013). In short, speaking is more demanding than many advocates of speech recognition report.

Early applications include systems for aircraft-engine inspectors, who wear wireless microphones as they walk around the engine, their hands busy opening cover plates or adjusting components. They can issue orders, read serial numbers, or retrieve previous maintenance records by using limited vocabulary. As in all speech input systems, they can be disruptive to others who find the noise a serious distraction.

The benefits of speech recognition to people with physical or visual disabilities, even temporary ones, are rewarding to see (Fig. 9.1). Its value during mobile

BOX 9.2

Speech recognition and production: Opportunities and obstacles.

Opportunities

- When users have physical impairments
- When the speaker's hands are busy
- When mobility is required
- When the speaker's eyes are occupied
- When harsh or cramped conditions preclude use of a keyboard
- When application domain vocabulary and tasks are limited
- When the user is unable to read or write (e.g., children)

Obstacles to speech recognition

- Interference from noisy environments and poor-quality microphones
- Commands need to be learned and remembered
- Recognition may be challenged by strong accents or unusual vocabulary
- Talking is not always acceptable (e.g., in shared office, during meetings)
- Error correction can be time-consuming
- Increased cognitive load compared with typing or pointing
- Math or programming difficult without extreme customization

Obstacles to speech production

- Slow pace of speech output when compared with visual displays
- Ephemeral nature of speech
- Not socially acceptable in public spaces (also privacy issues)
- Difficulty in scanning/searching spoken messages

use can be significant for users who take the time to learn and remember what can be accomplished with spoken commands, but general users of office or personal computers are not rushing yet to adopt speech input and output devices.

9.2.2 Speech recognition applications

For designers of human-computer interaction systems, speech recognition technologies have many variations, which can also be combined productively with speech production (Li et al., 2015).

The goal of speech recognition is primarily to *produce text based on spoken input* (Lewis, 2011), the most straightforward application being *dictation*. Dictation systems have now reached recognition rates that are acceptable in many situations

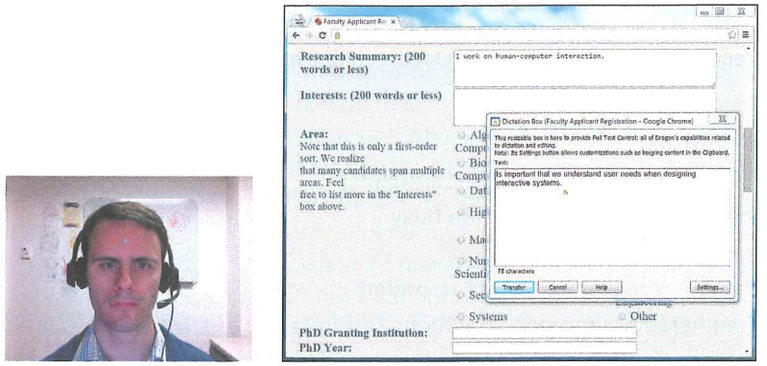

FIGURE 9.1
Using Dragon™ speech dictation and a head mouse (as made visible by the little silver dot on his forehead), a computer scientist is able to overcome a temporary hand disability.

(e.g., Google Docs' Voice Typing). They allow users to compose a document or speak search terms such as "movie theater in college park" and then correct mistakes with the keyboard instead of typing all the text. It can be a big time-saver with mobile devices, but keyboards, function keys, and pointing devices with direct manipulation often remain more rapid, depending on the quality of the recognition and the context of use (mobile or not), user's typing abilities, vocabulary complexity, nonnative speaker, and so on.

Ironically, technical fields with a lot of jargon are good candidates for speech recognition because of the distinctive nature of the terminology and the often-constrained documentation needs. For example, specialized systems for medical workers have become commercial successes (Nuance's Dragon® Medical, also embedded in electronic health record systems such as Cerner PowerChart Touch™). Dictation may be able to handle large vocabularies but inevitably requires specialized terminology to be useful to medical practitioners or lawyers.

While dictation is becoming practical, the cognitive burdens of dictation interfere with planning and sentence formation. In dictation, users may experience more interference between outputting their initial thought and elaborating on it. Spoken language may also be too informal compared with carefully typed sentences.

Speech recognition can also be used to transcribe recorded audio materials, either in real time or in a delayed fashion. It can facilitate *closed captioning* of radio or television programs or *transcription* of court proceedings or lectures.

Some applications may be beneficial even when there are errors. For example, errors will be irritating but acceptable for most television or YouTube viewers, but the payoff is that searching becomes possible. Where exact spelling is required, such as with person or place names, careful checking and error correction must be provided.

The other large category of speech recognition use is to allow users to *speak commands* that the user interface is trained to recognize effectively. This includes completing transactions over the phone, interacting with a device when direct manipulation is not convenient or possible, and using specialized voice services or "assistants." Dictation without using a keyboard will also require the use of commands to correct errors, start a new paragraph, or request the possibility to spell a name.

Specialized voice services or *"personal assistants"* like Siri, Google Now, Cortana, and Hound have become the more visible use of speech recognition. Because mobile interaction makes the use of keyboards impractical, speech becomes attractive to allow users to speak commands that execute the most common tasks performed on those devices, such as finding a location of interest, setting reminders, calendaring, communicating with others, or launching apps. Since the arrival of Apple's Siri in 2011, the competition has been fierce as companies compete to provide the most flexible and reliable services. The aim is to allow natural language, but users are often left wondering what they can say to get reliable results. This habitability problem remains a key problem, but logs of failed recognition facilitate efforts to broaden the acceptable inputs. Speaking assistants are now widely available, but many users never use them; others use only the few commands that they have learned and can remember; a smaller number of users can impress friends with apparent magic by having learned all the tricks. Companies do not report how much the assistants are being used, and while the demonstrations are impressive, the comparison tests often reveal problems (Ezzohari, 2015). Heavy use of spoken commands might be compared with the heavy use of keyboard shortcuts in traditional desktop users: not for everyone, but experts who have mastered them cannot live without them.

Speech is now widely used to complete *transactions* or access a service over the phone, for example, to report an electrical outage, trade stocks, or track lost luggage. These phone services, also called *information voice response* (IVR) systems, enable large financial savings for companies and provide 24/7 services for consumers (Lewis, 2011). Voice prompts welcome users and indicate what choices are possible. Users respond by pressing a number or speaking the word or short phrase that match their choice. Simple IVRs can be seen as an audio menu (see further discussion in Section 8.5).

A particularly challenging application is the *translation* of speech to facilitate human communication, such as foreign travelers or soldiers who must communicate in a language that they do not know well. Other emerging uses of speech recognition include the rapid *spotting of specific words or topics* in videos or

telephone calls or *speaker verification* (also called *voice biometrics*). While users answers questions, the system verifies that they are who they claim to be. However, ensuring robust performance, coping with users with colds, and dealing with noisy environments are still challenges.

9.2.3 Designing spoken interaction

After designers have established that using voice is appropriate, they must decide whether the interaction will be conducted entirely via the audio channel (using speech recognition and production; e.g., on the phone or when users are driving or have visual impairments). Alternatively, they may integrate voice and visual channels to provide informative feedback or display results on the screen of a mobile device or a computer (Oviatt and Cohen, 2015). In general, combining input by voice with visual output is much preferable, as reading on the screen is much faster than listening to long prompts and allows rapid selection. Having access to a keyboard to correct errors is also of great help.

Initiation The first step in using spoken interaction is for users to indicate that they wish to start the spoken interaction. In phone systems, a welcome prompt is sufficient to get started, but on the screen, a start button is needed (usually in the shape of a microphone), or an option is available to use a voice command to turn on the listening (e.g., "Hey Siri" or "Wake up"). This spoken command has to be very carefully chosen so that it is not misrecognized, but false positives will inevitably occur, causing frustration and possible chaos if further commands are recognized without users noticing it. The initiation may be done for each command, or a separate spoken command may be needed to stop the recognition process. For example, the Nuance Dragon™ system uses "Wake up" and "Go to sleep" and allows users to chat with others—or just relax—in the middle of a spoken interaction session. An on-screen reminder of the stop command is helpful to novice and intermittent users.

Knowing what to say Next, users need to know what can be said and reliably recognized. Learnability is one of the main issues of human language technologies that attempt to mimic natural language. In IVR phone systems, spoken prompts guide users and invite them to press keys or speak one of the proposed menu choices. Because they are typically used by novices or intermittent users, the possible transactions remain simple and the dialogue entirely directed (e.g., users are instructed to please say "account balance," "bill pay," or "fund transfer"). Some IVR systems use more open-ended prompts (e.g., "What service do you need?") and rely on a series of dialogues to clarify and confirm choices. The use of speech recognition allows users to shortcut through menu trees, which can be successful when users know the names of what they seek, such as a city, person, or stock name. They may even be able to speak while the instructional prompt is being read. This *barge-in* technique works well when most users are repeat users who can immediately speak the options they have learned from previous

experience. In all cases, the challenge is to identify novice users who attempt to use commands that are not recognized and switch them to a more directed mode that lists the possible commands. Users will become frustrated when they have to navigate a complex and deep menu structure (Section 8.4), when they are not allowed to "barge-in," when long spoken information segments contain irrelevant information, or when the menu of choices does not address their information need.

Users of mobile digital "assistants" are left with the burden of learning and remembering what the effective commands are. They may quickly become frustrated and quit if none of their attempts leads to success. Help can be provided with examples of commands (Fig. 9.2), or users are left to search blog postings to find lists of effective prompts (Cross, 2015), but those lists may be very long and commands still have to be remembered.

Recognition errors Slips produced by speech-recognition programs make for entertaining sections in product reviews in the trade press. Common errors occur when the vocabulary includes similar terms ("dime/time" or "Houston/Austin"). Challenges include dealing with regional or foreign accents and background noise. Users might also stutter, misspeak, or use the wrong terms. Dealing with unknown new words (and even failing to recognize that a word is unknown) can lead to confidently misrecognizing a similar-sounding word. Of course, the most difficult problem is matching the semantic interpretation and contextual understanding that humans apply easily to predict and disambiguate what was said. This problem was nicely highlighted in one of the few humorous titles of IBM technical reports: "How to Wreck a Nice Beach" (a play on "How to Recognize Speech"). To quote a summary of speech recognition's accomplishments, Huang et al. (2014) humbly report that "despite the impressive progress over the

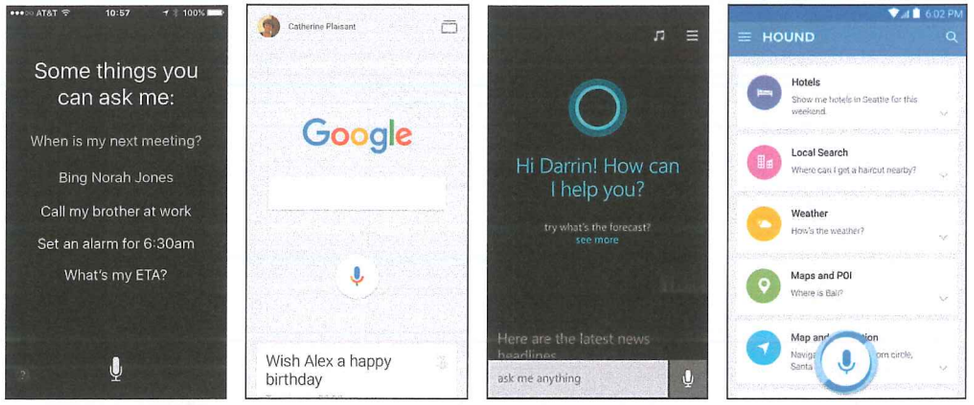

FIGURE 9.2
Mobile device assistants (from left to right: Siri, GoogleNow, Cortana, and Hound) all have similar microphone buttons but different ways of presenting suggestions.

past decades, today's speech recognition systems still degrade catastrophically even when the deviations are small in the sense (that) the human listener exhibits little or no difficulty. Robustness of speech recognition remains a major research challenge" (101). Finally, only a small portion of the myriad of world languages have adequate recognizers, and the mixing of two or more languages in the same sentence—which is common for multilingual speakers—also causes problems.

Early speech recognition systems were *speaker dependent*, and users were required to train the system to recognize their voice or deal with a particular microphone. This is not the case anymore for mobile phone use but is encouraged for professional applications that incorporate some level of personalization to increase the recognition rate. Changing microphones also required recalibration. In all cases, limiting the world of possible commands and carefully selecting easily differentiated terms dramatically improve recognition.

Correcting errors Correcting errors can be very taxing, especially when users do not have access to a keyboard or pointing device so all corrections have to be done using speech, possibly compounding errors with new ones. Even when a keyboard and pointer is available, having to correct errors is a significant distraction from the main task. A pause is generally required to separate dictation from editing commands, but providing correction commands that are very distinct will also facilitate their identification. Facilitating the erasing of last spoken text (e.g., saying "scratch that") allows repeating or rephrasing. Once a correction command has been identified (Fig. 9.3), alternative text can be proposed, or

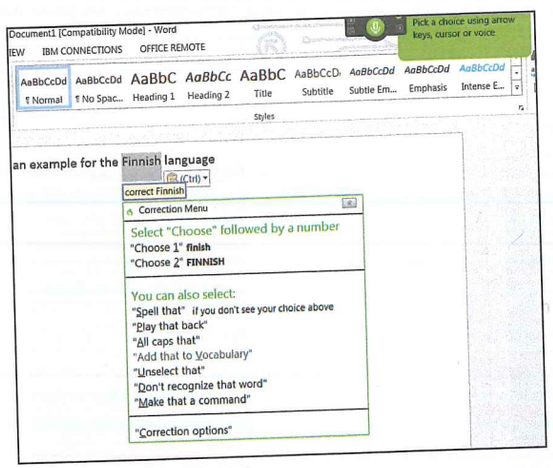

FIGURE 9.3
Correcting a word during dictation using Nuance Dragon™. After saying "Correct finnish," the word is selected and possible corrections are displayed in a menu along with additional commands such as "spell that." Users can use the cursor, arrow keys, or voice to specify their choice.

users can add and record new terms (e.g., "IEEE" pronounced as "I triple E") or spell out words (e.g., for new names or cities).

Mapping to possible actions The secret of most successful speech recognition applications today is that they are limited to narrow application domains—so the world of actions is limited, and they use commands carefully chosen to increase recognition (e.g., using "scratch that" to delete text). Banking IVRs only know about banking terms and have a small set of possible actions. Users of personal assistants on mobile devices may impress friends with the variety of possible commands, but each app has a limited set of possibilities. This stems from two main causes: First, mobile applications designers by nature focus on a limited number of often-used functions that are used constantly. Second, because speech is a highly variable signal, large corpora of recorded speech matching the application domain are needed to achieve good recognition results, so speech recognition achieves much better results in application domains that have been studied and modeled extensively. Even if the speech recognition made no error, there are many levels of possible errors mapping the corresponding text to the expected action, as illustrated in Fig. 9.4. Companies continuously collect data from users as they speak and correct errors to improve both the speech recognition and the mapping to appropriate actions. Comparisons of today's assistants such as Siri, Google Now, Cortana, and Hound seem to suggest that mapping the recognized text to the most appropriate action is the most challenging task (Ezzohari, 2015).

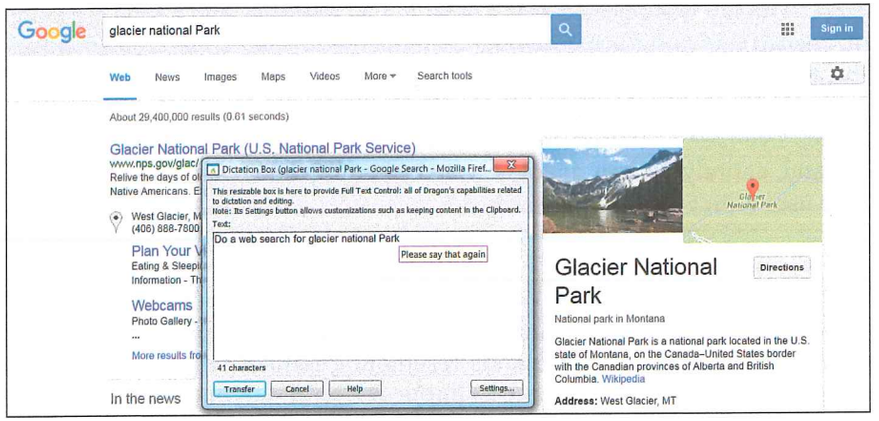

FIGURE 9.4
It can be difficult to remember what exact command will accomplish the task. In this example, when the user said, "Search the web for glacier national park," a Google search was launched and a search executed as expected, but when the user said, "Do a web search for glacier national park," all the words were accurately recognized but not as a command, so the text was placed in the Nuance Dragon™ dictation box.

Being able to rely on contextual information such as location or text from previous commands gives the impression of a more conversational interaction. For example, it might be possible to say "show me close by restaurants," "how about in Baltimore instead," "with 3 or more star reviews." Those chains of commands are significantly more difficult to interpret correctly and are today only achieved in constrained applications and by trained users who have learned what will be successful.

Feedback and dialogues During dictation or transcription, the recognized text is shown in the document being composed or in a dictation buffer, usually after a short delay (one to two seconds). Users can continue speaking or start correcting errors with the keyboard or by speaking navigation or editing commands. After correction, the text can also be transferred to a search box, the body of an e-mail message, or a field in a form. Applications tightly integrated with the speech recognition (opposed to relying on a dictation buffer) are more likely to be attractive and can generate spoken feedback as well.

Commands are usually executed directly, unless confirmation is preferable (e.g., "I am ready to e-mail this to Ben Shneiderman, should I go ahead?") or additional information or disambiguation is needed (e.g., "There are 2 John Smiths in your address book, which one should the e-mail be sent to?"). When context information has been used, feedback indicates how it was used. Specific questions may be asked to fill the holes in the task model and its attributes; for example, saying "Set an alarm" triggers a response asking "Set an alarm for when today?" (i.e., the date and time are missing from the alarm-setting task model, today was selected as the default date attribute, and a time attribute is still needed).

The availability of a display can greatly speed up interaction by presenting the proposed action in detail and only asking users to confirm or cancel, but it precludes eye-free operation (e.g., potentially endangering drivers). On the other hand, entirely spoken dialogues can be lengthy and even reveal information the user didn't want to be heard.

9.2.4 Spoken prompts and commands

When human language technology has been identified as appropriate for an application, prompts and commands resembling natural languages have to be designed. A language may have a simple or complex syntax and may have a few operations or hundreds, but the key issue—and the main usability determinant—is to adequately design clear prompts and a set of commands users can speak comfortably and remember easily and the system can recognize reliably.

The choice to use speech instead of keyboard entry is primarily a matter of user choice or possibility, but even with speech designers are the ones who decide what features to support, what commands will be used, how users will

discover what is possible, and what feedback or error messages will be provided.

The designer's first step is to study the users' task domain. The outcome is a list of task actions and objects, which is then abstracted into a set of interface actions and objects. These items, in turn, can be represented with the low-level interface syntax. Observing users speaking aloud is critical to discover commands that users might speak "naturally." Both commands and prompts may include terms that are rarely used in direct manipulation or menu systems; for example, users are likely to say "set an appointment for tomorrow" even though no specific menu for "tomorrow" exists in the menus of the graphical calendar interfaces.

A typical form is a verb followed by a noun object with qualifiers or arguments for the verb or noun—for example, users might say "launch Facebook" or "set an alarm for 7 a.m." Human learning is greatly facilitated by meaningful structure. If a set of commands is well-designed, users will recognize its structure and easily encode it in their semantic-knowledge storage. For example, if users can uniformly edit words, sentences, and documents, this meaningful pattern is easy for them to learn, apply, and recall. On the other hand, if they must use different terms to change a word, revise a sentence, or alter a document, the challenge and potential for error grow substantially, no matter how elegant the syntax is. The "naturalness" will result from careful design and inclusion of synonyms (Fig. 9.5).

An effective way to test early versions of a spoken language interaction is to conduct a *Wizard of Oz evaluation* in which a hidden person is transcribing the spoken commands into text to simulate perfect recognition and typing dialog prompts that are shown to the unsuspecting participant on a screen (for an example, see Dyke et al., 2013).

```
give me help
give me help on commands
[ ( go | move ) ] ( ( ( back | backward | backwards ) | ( forward | forwards ) ) | ( up | down ) ) ( one | a ) line
[ ( go | move ) ] ( ( ( back | backward | backwards ) | ( forward | forwards ) ) | ( up | down ) ) ( twenty | ... ) lines
( go | move ) ... [ ( ( one | one ) | ( twenty | ... ) ) ]
[ ( go | move ) ] ( ( left | right ) | ( ( back | backward | backwards ) | ( forward | forwards ) ) ) ( one | a ) character
[ ( go | move ) ] ( ( left | right ) | ( ( back | backward | backwards ) | ( forward | forwards ) ) ) ( twenty | ... ) characters
( go | move ) to [ the ] ( bottom | end )
( go | move ) to [ the ] ( bottom | end ) of [ the ] ( line | document )
( go | move ) to [ the ] ( start | top | beginning )
( go | move ) to [ the ] ( start | top | beginning ) of [ the ] ( line | document )
go to sleep
go_to_sleep
help me
```

FIGURE 9.5
A small subset of the rich set of commands used in the Nuance Dragon™ speech recognition system. Synonyms are included and used consistently.

9.3 Speech Production

Speech production is usually successful when the messages are simple and short and users' visual channels are overloaded; when they must be free to move around or on the phone; or when the environment is too brightly lit, too poorly lit, subject to severe vibration, or otherwise unsuitable for visual displays. However, designers must cope with the four obstacles to speech output: the slow pace of speech output when compared with visual displays, the ephemeral nature of speech, acceptability and privacy issues in public spaces, and the difficulty in scanning/searching (Box 9.2).

There are three general methods to produce speech. A common type of speech generation available commercially is *formant synthesis*, which produces entirely machine-generated speech using a set of algorithms to product sounds based on the phonetic representation of the text. The speech sounds somewhat artificial and robot-like. *Concatenated synthesis* instead combines tiny recorded human speech segments into phonemes, words, and phrases into full sentences. The voice is more natural but requires significantly more storage and computing power to assemble sentences on the fly. Formant synthesis and concatenated synthesis can generate any sentence as needed. Finally, *canned speech* consists of a fixed set of digitized speech segments which can be assembled together to create longer segments (e.g., "The next bus will arrive in" followed by "11" then "minutes"), but the number of possible complete sentences is limited and the seams between segments may sound awkward.

The quality of generated speech can be evaluated in terms of understandability, naturalness and acceptability. For some applications, a computer-like sound may be preferred. For example, the robot-like sounds used in the Atlanta airport subway drew more attention than did a tape recording of a human giving directions. Interactive voice response systems (IVRs) typically mix canned speech segments and speech synthesis to allow appropriate emotional tone and current information presentation.

Audio books or audio tours in museums and tourist sites also use canned speech. They are successful because they allow users to control the pace while conveying the curator's enthusiasm or author's emotion. Educational psychologists conjecture that if several senses (sight, touch, hearing) are engaged, learning can be facilitated. Adding a spoken component to an *instructional system* or an *online help system* (Section 14.3.2) may also improve the learning process.

Alerts and warnings can be presented using speech. They have been used in automobile navigation systems ("Turn right onto route M1"), internet services ("You've got mail"), or utility-control rooms ("Danger, temperature rising"), but in most cases, the novelty wears off quickly. Talking supermarket checkout machines that read out products and prices were found to violate shoppers' sense of privacy about purchases. Only generic instructions are spoken now, but many consumers still find them too noisy. Similarly, annoying warnings from cameras ("Too dark—use flash") and automobiles ("Your door is ajar") were removed and replaced with gentler tones or visual indicators. Spoken warnings in cockpits and control rooms are still used because they are omnidirectional and elicit rapid responses. However, even in these environments, spoken warnings can be missed, especially when in competition with human-human communication, and multiple methods are used simultaneously (e.g., a visual alert or a dialog box).

Applications for the visually impaired are an important success story. Utilities like the built-in Microsoft Windows Narrator or Apple VoiceOver can be used to read passages of text or hear descriptions of items on the screen. Screen readers like Freedom Scientific's JAWS, NV Access's NonVisual Desktop Access (NVDA), or Apple VoiceOver allow users with visual impairments to productively navigate between windows, select applications, browse graphical interfaces, and of course read text. Such tools rely on textual descriptions being made available for visual elements (labels for icons and image descriptions for graphics). Reading speed is adjustable, which allows interaction to be speeded up as well when needed. Book readers are also widely used in libraries. Patrons can place a book on a copier-like device that scans the text and does an acceptable job of reading it.

The slow pace of normal spoken output, the ephemeral nature of speech, and the difficulty in scanning/searching remain challenges, but speech production is widely used because it enables services that would otherwise be too expensive; hiring well-trained customer-service representatives available 24 hours a day is not practical for many organizations.

9.4 Human Language Technology

Even before there were computers, people dreamed about creating machines that would be able to understand *natural language*—that is, be able to take the appropriate action in various contexts without users having to learn any command syntax or select from menus. It is a wonderful fantasy, but language is subtle; there are many special cases, contexts are complex, and emotional relationships have a powerful and pervasive effect in human-human communication. Although true comprehension and generation of open-ended language seem an inaccessible goal, there has been extensive research on human language technology; widespread use is slow in developing, primarily because the alternatives are more appealing. Contrary to common belief, human-human interaction is not necessarily an appropriate model for human operation of computers. Since computers can display information 1,000 times faster than people can enter commands, it is advantageous to use the computer to display large amounts of information and to

allow users simply to choose among the items. Selection helps to guide users by making clear what objects and actions are available. For knowledgeable and frequent users who are thoroughly aware of the available functions, a precise, concise language (typed or spoken) is usually preferred (Section 9.5).

Natural language interaction (NLI) in the form of a series of exchanges resembling a dialogue is difficult to design and build for even a single topic. The key impediment is the *habitability* of the user interface—that is, how easy it is for users to determine what objects and actions are appropriate. Visual interfaces provide the cues for the semantics of interaction, but NLI interfaces typically depend on assumed user models. Users who are knowledgeable about their tasks—for example, stock-market brokers who know the stock codes (objects) and buy/sell actions—can place orders in natural language, but these users prefer compact command languages because they are more rapid and reliable.

While early conceptions of human language technology assumed that computers would parse natural language expressions in text or spoken forms and derive some level of "understanding" and description of users' "intent," the current successes rely instead on *statistical methods* based on the analysis of vast textual or spoken corpora and usage data of millions of users.

For example, question-answering strategies are successful in situations where there are relevant corpora and designers have crafted effective user interfaces that expand queries, search databases, show users alternatives, and present final results in ways that are most likely to be useful. Their success comes not from the understanding of the natural language but from the fact that the question at hand has already been asked before—using the same terminology—and has already been answered by others (Hearst, 2011). Another method is to analyze web search usage logs to find what results users seek often. For example, when users type "Leddo restaurant," human language technology extracts relevant queries from the dataset and identifies that "Leddo" does not exist but "Ledo" is a frequent entry. Then the word "restaurant" has been repeatedly identified as a term that leads users to look for an address, hours of operation, or a map, so that information can be presented by default. This can be done on the basis of frequency of past queries and on the log of previous users' actions.

Other applications include *extraction and tagging*. Extraction refers to the process of analyzing human language to create a more structured format, such as a relational database. The advantage is that the parsing can be done once in advance to structure the entire database and to speed searches when users pose relational queries. Legal (Supreme Court decisions or state laws), medical (scientific journal articles or patient histories), and journalistic (Associated Press news stories or *Wall Street Journal* reports) texts have been used. A variant is to tag documents based on content. For example, it is useful to have an automated analysis of business news stories to classify them as covering mergers, bankruptcies, or initial public offerings for companies in various industries such as electronics, pharmaceutical, or petroleum. Extracting and tagging applications are promising because users appreciate even a modest increase in suitable

retrievals, and imperfect retrievals are more acceptable than errors in natural language interaction. On the other hand, errors can become quite problematic when the extracted information is used to make decisions or inform policies. One example is the use of human language technology in medicine. A large amount of information about medical conditions, treatments, and outcomes is buried in textual notes written by physicians in electronic health records. Automatically extracting diagnoses or test results out of the text notes can be very useful to identify possible candidates for a clinical trial, as all records will be reviewed by a clinician. On the other hand, the use of automatic tags for clinical decision making can be problematic. The rare cases of success are limited to situations with specific users, document types, and decision support goals (Demner-Fushman et al., 2009). *Sentiment analysis* is a specialized tagging, which can be applied to groups of news articles, reviews, or social media to monitor global changes in opinion, but tagging of individual documents remains error-prone.

Human language text generation is used for simple tasks, such as the preparation of structured weather reports ("80% chance of light rain in northern suburbs by late Sunday afternoon") in which generated reports from structured databases can be sent out automatically. Automatically generated text can be used to supplement standard data charts such as bar charts or scatterplots in order to make them more accessible to users with visual impairments (e.g., Google Sheet's Explore or iweave.com). More elaborate applications of text generation include preparation of reports of medical laboratory or psychological tests. The computer generates not only readable reports ("White-blood-cell count is 12,000") but also warnings ("This value exceeds the normal range of 3,000 to 8,000 by 50%") or recommendations ("Further examination for systemic infection is recommended"). Still more involved scenarios for text generation involve the creation of legal contracts, wills, or business proposals. Text summarization remains a much greater challenge with limited success, as summaries must capture the essence of the content and convey it accurately in a compact manner (Liu et al., 2012).

Human language technologies are used in *instructional systems*. Successful examples are in grammatical error detection and proofreading. Also widely used—but more controversial—is the automated scoring of short-answer responses or essays during student assessment. Human language technology has been introduced into a variety of educational contexts such as reading support. Tutorials with materials and pedagogy that have been carefully tested can provide feedback in natural language, which encourages students to stay engaged in the educational process. Simulations can also be used to practice communication skills learned in other settings (Fig. 9.6).

A remaining question is whether learning differs when students speak their responses or type them. A Wizard of Oz experiment (where a human transcribed the learner's speech before submitting it to the tutor) suggests that learning gains and preferences are similar with both modalities, but highly motivated students reported lower cognitive load and demonstrated increased learning when typing compared with speaking (D'Mello et al., 2011).

Translation between human languages has long been a goal (Green et al., 2015), but older strategies of word replacement with some grammatical parsing have given way to statistical methods based on having large databases with correct human translations, such as United Nations documents that appear in five required languages. Then well-designed user interfaces clarify what users can input in text windows, present translation options, show the translation, and guide subsequent user actions (Fig. 9.7). This design effort gets more complex with input errors and languages that may have unfamiliar characters, differ from English left-to-right formatting, and invoke words that do not exist in the target languages.

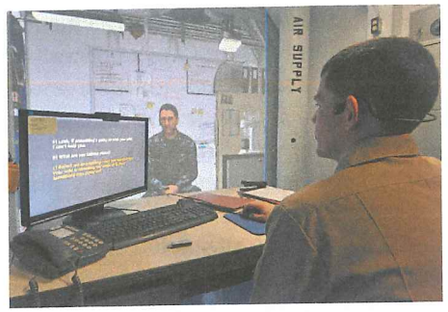

FIGURE 9.6
Using the Immersive Naval Officer Training System (INOTS), new Navy officers can practice their counseling skills in a virtual reality environment. Officers listen to an avatar and respond using spoken language, loosely following suggestions from multi-choice prompts presented on the screen and designed to match the learning objectives. The interaction is constrained, but assessment is facilitated (Dyke et al., 2013; http://www.netc.navy.mil/nstc/news_page_2012_02_24_2.asp).

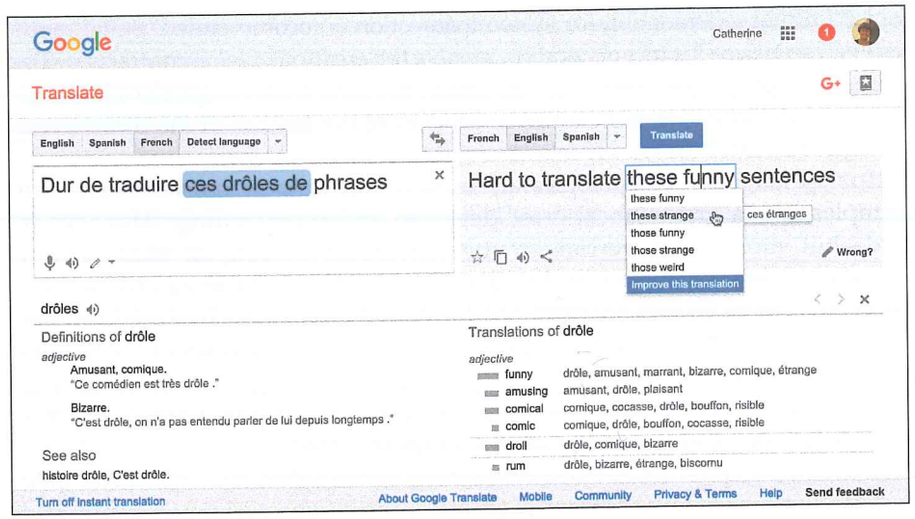

FIGURE 9.7
Google Translate, showing a French sentence translated in English. A click on the word "drôle" displayed its definition. Selecting "funny" highlighted "these funny" as well as the matching French words, and an alternative translation can be selected.

9.5 Traditional Command Languages

Early tally marks and pictographs on cave walls existed for millennia before precise notations for numbers or other concepts appeared. Eventually, languages with a small alphabet and rules of word and sentence formation dominated. Computers were quickly found to be effective to manipulate logical expressions, operate on the real world, or search vast libraries. These applications encouraged designers to find convenient notations to direct the computer, leading to command languages. With a command-language interface, users type a command and watch what happens. If the result is correct, the next command is issued; if not, some other strategy is adopted. By contrast, menu-selection users view or hear the limited set of menu items and they respond more than initiate. With command-language interfaces, users must recall the exact words to be used and the correct syntax, although prompted input often supplies a visual list of correct completions. For example, in the early days of command languages, users who needed to print a document may have been instructed to type:

```
CP TAG DEV E VTSO LOCAL 2 OPTCD=J F=3871 X=GB12
```

Another example is the Unix command used to delete blank lines from a file:

```
grep -v ^$ filea > fileb
```

Command-line interfaces are often preferred when the application is used in an advanced way (e.g., professionals using an application for hours every day). Casual users favor graphical user interfaces, but both styles of interface can be made available successfully because they do not always provide the same functionality. For example, in MATLAB, the command language can handle all the calculations, and a large subset of calculations is also available via the graphical user interface, which makes it easier for novice users to get started. Being able to type complex Boolean expressions using AND, OR, or NOT as well as regular expressions remains a key motivation for experienced users who can accomplish remarkable feats at amazing speed (Fig. 9.8).

Web addresses or URLs can be seen as a form of command language. Users come to memorize the structure of their favorite site addresses, even though the typical usage is to click on a link to select an address from a webpage or a search result page. The address field of browsers can also be used as a command line. For example, typing "(1024*768)/25" in the URL field in a Chrome browser will calculate the result, and typing "100 feet to meters" will launch the conversion tool and show the result: 30.48 meters.

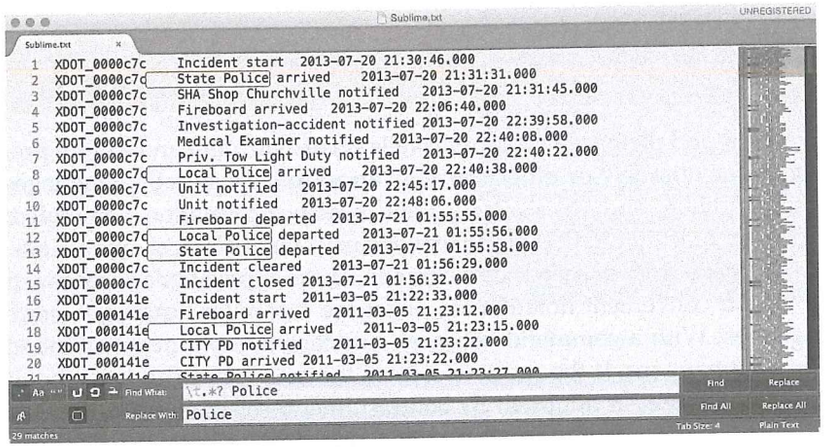

FIGURE 9.8

Using the Sublime text editor, a user is doing a search and replace in a data table using regular expressions. Typing " \t.*? Police" in the search box searches for a tab followed by zero or more characters, a space, and then the word "Police." The patterns found in the document are highlighted with a thin black line, showing that both "Local Police" and "State Police" have been found and selected. An overview of the entire document is visible on the right, revealing the presence of many other matches that can now be replaced all at once.

Twitter tags (#hcil, $TWTR, or @benbendc) can also be considered an example of new command language that needs to be learned and remembered, along with the proliferation of acronyms and abbreviations used by clever text-message writers (e.g., *LOL* for "laugh out loud" or *2G2BT* for "too good to be true"). In the traditional desktop environment, shortcut keys also remain heavily used by users who take the time to learn them (e.g., typing Ctrl-Q for Quit or Ctrl-P for Print; see Section 8.2.2). Programmers or professionals who use a single application all day long (e.g., a computer-aided design or publishing application) can memorize hundreds of commands and shortcuts, helping them gain mastery of their application (Cockburn et al., 2014).

One important opportunity linked to command languages is that histories can easily be kept and macros or scripts created to automate actions, but the essence of command languages is that they have an ephemeral nature and they produce an immediate result on some object of interest. Feedback is generated for correct commands, and error messages (Section 12.7) result from unacceptable forms or typos. Auto-completion is critical to help prevent errors. Command-language systems may offer brief prompts with choices, becoming closer to menu-selection systems. Command languages typically do not require a pointing device and therefore can become a lifesaver for users with visual impairments, which make the use of mice and touchscreens impractical.

Database-query languages for relational databases were developed in the middle to late 1970s; they led to the still widely used Structured Query Language, or SQL™, which emphasized short segments of code (2 to 20 lines) that could be typed and executed immediately. For example:

```
SELECT * FROM Products
WHERE Price BETWEEN 10 AND 20;
```

Here the goal of the user is to create a result rather than a program. A key part of database-query languages and information-retrieval languages is the specification of Boolean operations—AND, OR, and NOT—which can be very challenging to specify. See Chapter 15 for more on advances regarding searching.

Major considerations for expert users are the possibilities of tailoring the language to suit personal work styles and of creating named macros to permit several operations to be carried out with a single command. Macro facilities allow extensions that the designers did not foresee or that are beneficial to only a small fragment of the user community. A macro facility can become a full programming language that might include specification of arguments, conditionals, iteration, integers, strings, and screen-manipulation primitives plus library and editing tools—resembling a full-blown programming language.

In summary, while error rates remain high, the complexity and power of command languages have a certain attraction for a portion of the computer user community. Users gain satisfaction in overcoming the difficulties and becoming one of the inner circle "gurus" of their favorite command language.

Practitioner's Summary

The dream of natural language interaction has been mostly replaced by the effective use of statistical methods based on very large spoken and text corpora and logs of user interactions. Speech recognition for personal digital assistants and dictation has become increasingly successful, but errors and error correction remain issues. Speech-based approaches for guided interactions over telephones are also proving to be useful.

Speech generation, when well-designed, can support effective applications with phone, mobile devices, or book readers. Well-designed user interfaces enable integration of visual displays and touchscreens with speech. Text analysis, generation, and translation are useful human language technologies based on large training databases and appropriate user interfaces to prompt users and handle interactions.

Command languages continue to be attractive for expert users who learn the semantics and syntax because they can rapidly specify actions involving

multiple options. Command languages allow sequences of commands to be stored for future use as a macro or script.

For command languages as well as spoken command languages, designers begin with a careful task analysis to determine what functions should be provided. Meaningful specific names aid learning and retention.

Researcher's Agenda

Speech recognition and generation user interfaces are maturing rapidly as effective designs have a growing user community. Improved user interfaces that integrate speech with visual displays and touchscreen controls may attract still larger communities; however, research on error reduction and methods to facilitate error correction is still needed.

Natural language interaction success stories are still elusive, but human language technology has become an important part of the success of search technology (Chapter 15). Spoken and text generation has shown value, so further research is warranted. For those who continue to explore specific applications, empirical tests and long-term case studies offer successful strategies to identify the appropriate niches and designs.

WORLD WIDE WEB RESOURCES

www.pearsonglobaleditions.com/shneiderman

- Designers will find many demonstrations of spoken interaction on YouTube. For example, the different styles of feedback and dialogue used by personal "assistants" can be seen at Hound Beta vs. Siri vs. Google Now vs. Cortana: https://www.youtube.com/watch?t=134&v=9zNh8kQLhfo.
- Experimenting with common search engines and personal digital assistants such as Siri or Google Now provides hints about the current human language technology strategies used for question answering.
- Translation: http://translate.google.com or http://www.babelfish.com
- Speech recognition commercial systems: http://www.nuance.com/dragon
- IVR dialog system: http://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/dialog.html

Discussion Questions

1. Consider voice-activated digital assistants such as Siri, Cortana, or Google Talk. Identify a situation or scenario where you chose to use this personal assistant, and identify a scenario where you chose to avoid it.
2. As a follow-on to the previous question, produce a thoughtful argument about what role spoken interaction should have in user interfaces. Be sure to list at least three benefits and limitations of spoken interaction.
3. Briefly describe the applications of speech recognition.
4. What are the obstacles to speech recognition and production?
5. There exist applications of human language understanding technology. Name some examples.
6. List several situations when command languages can be attractive for users.

References

Bouzid, Ahmed, and Ma, Weiye, *Don't Make Me Tap! A Common Sense Approach to Voice Usability*, Dakota Press (2013).

Cockburn, A., Gutwin, C., Scarr, J., and Malacria, S., Supporting novice to expert transitions in user interfaces, *ACM Computing Surveys 47*, 2 (2014), Article 2.

Cohen, M. H., Giangola, J. P., and Balogh, J., *Voice User Interface Design*, Addison Wesley (2004).

Cross, J., A list of all the Google Now voice commands, Greenbot blog http://www.greenbot.com/article/2359684/system-software/a-list-of-all-the-ok-google-voice-commands.htm (2015).

Demner-Fushman, D., Chapman, W. W., and McDonald, C. J., What can natural language processing do for clinical decision support? *Journal of Biomedical Informatics 42*, 5 (2009), 760–772.

D'Mello, S. K., Dowell, N. N., and Graesser, A., Does it really matter whether students' contributions are spoken versus typed in an intelligent tutoring system with natural language? *Journal of Experimental Psychology 17*, 1 (2011), 1–17.

Dyke, G., Adamson, A., Howley, I., and Rosé, C. P., Enhancing scientific reasoning and discussion with conversational agents, *IEEE Transactions on Learning Technologies 6*, 3 (2013), 240–247.

Ezzohari, H., [ULTIMATE] personal assistant review: Hound vs Siri vs Google Now vs Cortana http://www.typhone.nl/blog/ultimate-voice-assistant-review/ (2015).

Green, S., Heer, J., and Manning, C. D., Natural language translation at the intersection of AI and HCI, *Communications of the ACM 58*, 9 (2015), 46–53.

Hearst, M. A., "Natural" search user interfaces, *Communications of the ACM 54*, 11 (2011), 60–67.

Huang, X., Baker, J., and Reddy, R., A historical perspective of speech recognition, *Communications of the ACM 57*, 1 (2014), 94–103.

Karat, M-C., Lau, J., Steward, O., and Yankelovich, N., Speech and language interfaces, applications and technologies, in Jacko, J. (Editor), *The Human-Computer Interaction Handbook*, CRC Press (2012), 367–386.

Lewis, J. R., *Practical Speech User Interface Design*, CRC Press (2011).

Li, Jinyu, Deng, Li, Haeb-Umbach, Reinhold, and Gong, Yifang, *Robust Speech Recognition: A Bridge to Practical Applications*, Academic Press (2015).

Liu, S., Zhou, M. X., Pan, S., Song, Y., Qian, W., Cai, W., and Lian, X., TIARA: Interactive, topic-dased visual text summarization and analysis, *ACM Transactions on Intelligent Systems Technology 3*, 2 (2012), 28 pages.

Mariani, Joseph, Rosset, Sophie, Garnier-Rizet, and Devillers, Laurence (Editors), *Natural Interaction with Robots, Knowbots and Smartphones: Putting Spoken Dialog Systems into Practice*, Springer (2014).

Neustein, A., and Markowitz, J. A. (Editors), *Mobile Speech and Advanced Natural Language Solutions*, Springer (2013).

Oviatt, Sharon, and Cohen, Philip, *The Paradigm Shift to Multimodality in Contemporary Computer Interfaces*, Morgan & Claypool (2015).

Pieraccini, Roberto, *The Voice in the Machine: Building Computers that Understand Speech*, MIT Press (2012).

Radvansky, G., and Ashcraft, M., *Cognition*, 6th Edition, Pearson (2013).