Speech Recognition in Machines

Over the past several decades, a need has arisen to enable humans to communicate with machines in order to control their actions or to obtain information. Initial attempts at providing human-machine communications led to the development of the keyboard, the mouse, the trackball, the touch-screen, and the joystick. However none of these communication devices provides the richness or the ease of use of speech, which has been the most natural form of communication between humans for tens of centuries. Hence a need has arisen to provide a voice interface between humans and machines. This need has been met, to a limited extent, by speech-processing systems that enable a machine to speak (speech synthesis systems) and that enable a machine to understand (speech recognition systems) human speech. We concentrate on speech recognition systems in this section.

Speech recognition by machine refers to the capability of a machine to convert human speech to a textual form, providing a transcription or interpretation of everything the human speaks while the machine is listening. This capability is required for tasks in which the human is controlling the actions of the machine using only limited speaking capability, such as while speaking simple commands or sequences of words from a limited vocabulary (e.g., digit sequences for a telephone number). In the more general case, usually referred to as *speech understanding*, the machine need only reliably recognize a limited subset of the user input speech—namely, the parts of the speech that specify enough about the action requested so that the machine can either respond appropriately or initiate some action in response to what was understood.

Speech recognition systems have been deployed in applications ranging from control of desktop computers, to telecommunication services, to business services.

The earliest approaches to speech recognition were based on finding speech sounds and providing appropriate labels to these sounds. This is the basis of the *acoustic-phonetic* approach (Hemdal and Hughes 1967), which postulates that there exist finite, distinctive phonetic units (phonemes) in spoken language and that these units are broadly characterized by a set of acoustic properties that are manifest in the speech signal over time. Even though the acoustic properties of phonetic units are highly variable, both with speakers and with neighboring sounds (the so-called coarticulation effect), it is assumed in the acoustic-phonetic approach that the rules governing the variability are straightforward and can be readily learned (by a machine).

The first step in the acoustic-phonetic approach is a spectral analysis of the speech combined with a feature detection that converts the spectral measurements to a set of features that describe the broad acoustic properties of the different phonetic units.

The next step is a segmentation and labeling phase in which the speech signal is segmented into stable acoustic regions, followed by attaching one or more phonetic labels to each segmented region, resulting in a phoneme lattice characterization of the speech. The last step in this approach attempts to determine a valid word (or string of words) from the phonetic label sequences produced by the segmentation to labeling. In the validation process, linguistic constraints on the task (i.e., the vocabulary, the syntax, and other semantic rules) are invoked in order to access the lexicon for word decoding based on the phoneme lattice. The acoustic-phonetic approach has not been widely used in most commercial applications.

The *pattern-matching approach* (Itakura 1975; Rabiner 1989; Rabiner and Juang 1993) involves two essential steps—namely, pattern training and pattern comparison. The essential feature of this approach is that it uses a well-formulated mathematical framework and establishes consistent speech-pattern representations, for reliable pattern comparison, from a set of labeled training samples via a formal training algorithm. A speech-pattern representation can be in the form of a speech template or a statistical model (e.g., a HIDDEN MARKOV MODEL or HMM) and can be applied to a sound (smaller than a word), a word, or a phrase. In the pattern-comparison stage of the approach, a direct comparison is made between the unknown speech (the speech to be recognized) with each possible pattern learned in the training stage in order to determine the identity of the unknown according to the goodness of match of the patterns. The pattern-matching approach has become the predominant method of speech recognition in the last decade.

The *artificial intelligence approach* (Lesser et al. 1975; Lippmann 1987) attempts to mechanize the recognition procedure according to the way a person applies intelligence in visualizing, analyzing, and characterizing speech based on a set of measured acoustic features. Among the techniques used within this class of methods are use of an expert system (e.g., a neural network) that integrates phonemic, lexical, syntactic, semantic, and even pragmatic knowledge for segmentation and labeling, and uses tools such as artificial NEURAL NETWORKS for learning the relationships among phonetic events. The focus in this approach has been mostly in the representation of knowledge and integration of knowledge sources. This method has not been used widely in commercial systems.

A block diagram of a complete system for *large vocabulary speech recognition* (Lee, Rabiner, and Pieraccini 1992; Jelinek 1985; Baker 1990) based on the pattern-matching approach is shown in Figure 1. The first step in the processing is spectral analysis to derive the feature vector used to characterize the spectral properties of the speech input. The second step in the recognizer is a combined word-level/sentence-level matching procedure. The way this is accomplished is as follows. Using a set of subword models (phoneme-like units) along with a word lexicon, a set of word models is created by concatenating each of the subword models as specified by the word lexicon. The word-level match procedure provides scores for individual words as specified by the sentence-level match procedure (which uses a word grammar—the syntax of the system) and the semantics (which specifies valid sentences in the task language). The final result is the sentence that provides the best match to the speech input according to the word vocabulary, task syntax, and task grammar.

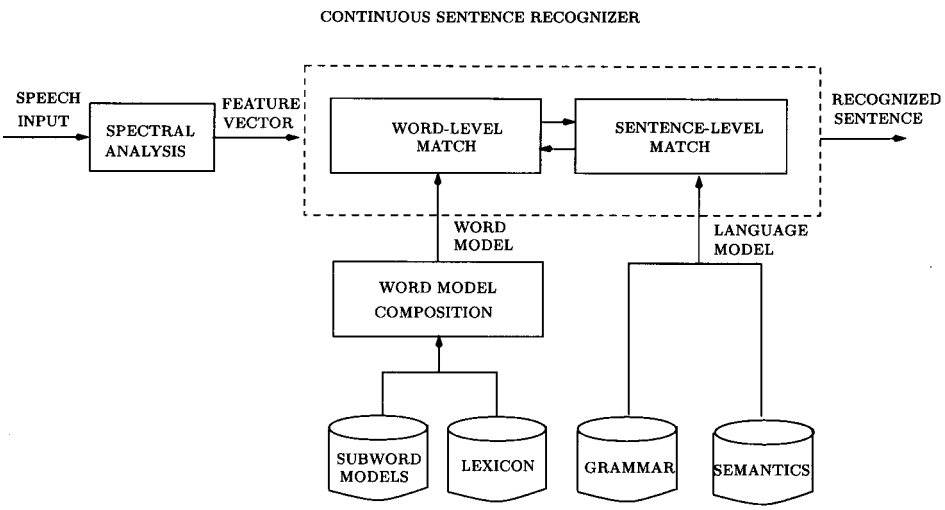

Figure 1. Overall block diagram of subwork unit-based continuous-speech recognizer.

Table 1 illustrates current capabilities in continuous speech recognition for three distinct and rather simple tasks—namely, database access (Resource Management), natural language queries (ATIS) for air-travel reservations, and read text from a set of business publications (NAB). The task syntax is the system grammar (or language model) and is realized as one of a finite-state word-pair grammar, a word-trigram grammar, or a five-gram word grammar. The systems all run in a speaker independent (SI) mode with either fluently read speech or naturally spoken dialogue.

It can be seen from Table 1 that for tasks with medium size vocabularies (1000–2500 words) and with language perplexities (average word-branching factors) significantly below that of natural language speech (perplexity of 100–200), word-error rates below 5 are easily obtainable with modern technology. Such systems could actually be utilized in limited (controlled) user environments and could be designed to work rather well. On the other hand, for more complex tasks like NAB with a 60,000-word vocabulary and perplexity comparable to that of natural-language speech, word-error rates exceed 10, thereby making these systems almost unusable in practical environments.

Table 1. Performance of continuous-speech recognition systems

Task	Syntax	Mode	Vocabulary	Word Error Rate
Resource Management	Finite State Grammar	SI Fluent Read	1000 Words	4.4%
Air Travel Information System	Backoff Trigram (Perplexity = 18)	SI Natural Language	2500 Words	3.6%
North American Business (NAB) (DARPA)	Backoff 5-gram (Perplexity = 173)	SI Fluent Read Input	60000 Words	10.8%

Speech recognition has been successfully applied in a range of systems. We categorize these applications into five broad classes.

1. *Office or business system* Typical applications include data entry onto forms, database management and control, keyboard enhancement, and dictation. Examples of voice-activated dictation machines include the IBM Tangora system and the Dragon Dictate system.
2. *Manufacturing* ASR is used to provide "eyes-free, hands-free" monitoring of manufacturing processes (e.g., parts inspection) for quality control.
3. *Telephone or telecommunications* Applications include automation of operator-assisted services (the Voice Recognition Call Processing system by ATT to automate operator service routing according to call types), inbound and outbound telemarketing, information services (the ANSER system by NIT for limited home-banking services, the stock-price quotation system by Bell Northern Research, Universal Card services by Conversant/ATT for account information retrieval), voice dialing by name/number (ATT VoiceLine, 800 Voice Calling services, Conversant FlexWord, etc.), directory-assistance call completion, catalog ordering, and telephone calling feature enhancements (ATT VIP—Voice Interactive Phone for easy activation of advanced calling features such as call waiting, call forwarding, and so on by voice rather than by keying in the code sequences).
4. *Medical* The application is primarily in voice creation and editing of specialized medical reports (e.g., Kurzweil's system).
5. *Other* This category includes voice-controlled and -operated toys and games, aids for the handicapped, and voice control of nonessential functions in moving vehicles (such as climate control and the audio system).

For the most part, machines have been successful in recognizing carefully articulated and read speech. Spontaneous human conversation has proven to be much more difficult a task. Recent performance evaluations using speech recorded off a radio station, as well as from monitoring speech of family members talking over conventional telephone lines,

shows word-error rates of from 27 to upwards of 50. These high word-error rates are an indication of how much more must be learned before machines are truly capable of recognizing human conversational speech.

See also NATURAL LANGUAGE GENERATION; NATURAL LANGUAGE PROCESSING; SPEECH PERCEPTION; SPEECH SYNTHESIS

—*Lawrence Rabiner*

References

Baker, J. M. (1990). Large vocabulary speech recognition prototype. *Proc. DARPA Speech and Natural Language Workshop*, pp. 414–415.

Hemdal, J. F., and G. W. Hughes. (1967). A feature based computer recognition program for the modeling of vowel perception. In W. Wathen-Dunn, Ed., *Models for the Perception of Speech and Visual Form.* Cambridge, MA: MIT Press.

Itakura, F. (1975). Minimum prediction residual principle applied to speech recognition. *IEEE Trans. Acoustics, Speech, and Signal Processing* ASSP-23: 57–72.

Jelinek, F. (1985). The development of an experimental discrete dictation recognizer. *IEEE Proceedings* 73(11): 1616–1624.

Lee, C. H., L. R. Rabiner, and R. Pieraccini. (1992). Speaker independent continuous speech recognition using continuous density hidden Markov models. In P. Laface and R. DeMori, Eds., *Proc. NATO-ASI, Speech Recognition and Understanding: Recent Advances, Trends and Applications.* Cetraro, Italy: Springer, pp. 135–163.

Lesser, V. R., R. D. Fennell, L. D. Erman, and D. R. Reddy. (1975). Organization of the Hearsay-II Speech Understanding System. *IEEE Trans. Acoustics, Speech, and Signal Processing* ASSP-23(1): 11–23.

Lippmann, R. (1987). An introduction to computing with neural networks. *IEEE ASSP Magazine* 4(2): 4–22.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proc.* IEEE 77(2): 257–286.

Rabiner, L. R., and B. H. Juang. (1993). *Fundamentals of Speech Recognition.* Englewood Cliffs, NJ: Prentice-Hall.

Speech Synthesis

The history of "speaking machines" goes back at least to the work of Wolfgang von Kempelen in 1791, but until the advent of the digital computer all such devices required a human operator to "play" them, rather like a musical instrument. Perhaps the best known machine of this sort was Homer Dudley's VODER, which was demonstrated at the 1939 World's Fair.

Modern speech synthesis programs, of course, can produce speechlike output on the basis of symbolic input, with no further intervention. When the symbolic input to such a program is ordinary text, the program is often called a text-to-speech (TTS) system. Most TTS systems can be viewed as having two fairly distinct halves: a first stage that analyzes the text and transforms it into some form of annotated phonetic transcription, and a second stage, which is often thought of as synthesis proper, which produces a sound wave from the phonetic transcription.

Programs that generate their own sentences, for example, automated information systems, can produce synthesizer input directly and avoid the difficulties of textual analysis. There was, at the beginning of 1998, no standard format for synthesizer input, and most systems have their own ad hoc notations. There is a move toward the development of standardized speech markup languages, on the model of text markup languages like LaTex and HTML, but considerable work remains to be done.

Text analysis in TTS systems serves two primary purposes: (1) specifying the pronunciations of individual words and (2) gathering information to guide phrasing and placement of pitch accents (see PROSODY AND INTONATION).

Word pronunciations can be looked up in dictionaries, generated by spelling-to-sound rules, or produced through a combination of the two. The feasibility of relying on spelling-to-sound rules varies from language to language. Any language will need at least a small dictionary of exceptions. English spelling is sufficiently problematic that current practice is to have a dictionary with tens of thousands—or even hundreds of thousands—of entries, and to use rules only for words that do not occur in the dictionary and cannot be formed by regular morphological processes from words that do occur. Systems vary in the extent to which they use morphology. Some systems attempt to store all forms of all words that they may be called on to pronounce. The MITalk system had a dictionary of orthographic word fragments called "morphs" and applied rules to specify the ways in which their pronunciations were affected when they were combined into words.

The parsing and morphological analysis (see NATURAL LANGUAGE PROCESSING and MORPHOLOGY) techniques used in text processing for text-to-speech are similar to those used elsewhere in computational linguistics. One reason for parsing text in text-to-speech is that the part of speech assignment performed in the course of parsing can disambiguate homographs—forms like the verb *to lead* and the noun *lead*, or the present and past tenses of the verb *to read*, which are spelled the same but pronounced differently. The other main reason is that it is possible to formulate default rules for placement of pitch accents and phrase boundaries on the basis of syntax. On the basis of such rules, markers can be placed in the annotated phonetic output of the text analysis stage that instruct the synthesis component to vary vocal pitch and introduce correlates of phrasing, such as pauses and lengthening of sounds at the ends of phrases. Such default rules tend to yield the rather unnatural and mechanical effect generally associated with synthetic speech, and improving the quality of synthetic prosody is one of the major items on the research agenda for speech synthesis.

Synthesis proper can itself be broken into two stages, the first of which produces a numerical/physical description of a sound wave, and the second of which converts the description to sound. In some cases, the sound is stored in the computer as a digitized wave form, to be played out through a general purpose digital to analog converter, whereas in other cases, the numerical/physical description is fed to special purpose hardware, which plays the sound directly without storing a waveform.