An Introduction to Speech and Speaker Recognition

Richard D. Peacocke and Daryl H. Graf
Bell-Northern Research

Being able to speak to your personal computer, and have it recognize and understand what you say, would provide a comfortable and natural form of communication. It would reduce the amount of typing you have to do, leave your hands free, and allow you to move away from the terminal or screen. You would not even have to be in the line of sight of the terminal. It would also help in some cases if the computer could tell *who* was speaking.

If you want to use voice as a new medium on a computer workstation, it is natural to explore how speech recognition can contribute to such an environment. Here, we will review the state of speech and speaker recognition, focusing on current technology applied to personal workstations.

Limited forms of speech recognition are available on personal workstations. Currently there is much interest in speech recognition, and performance is improving. Speech recognition has already proven useful for certain applications, such as telephone voice-response systems for selecting services or information, digit recognition for cellular phones, and data entry while walking around a railway yard or clambering over a jet engine during an inspection.

Nonetheless, comfortable and natural communication in a general setting (no constraints on what you can say and how you say it) is beyond us for now, posing a problem too difficult to solve. Fortunately, we can simplify the problem to allow the creation of applications like the examples just mentioned. Some of these simplifying constraints are discussed in the next section.

Speech recognition, the ability to identify spoken words, and speaker recognition, the ability to identify who is saying them, are becoming commonplace applications of speech processing technology.

Speaker recognition is related to work on speech recognition. Instead of determining what was said, you determine who said it. Deciding whether or not a particular speaker produced the utterance is called *verification*, and choosing a person's identity from a set of known speakers is called *identification*. The most general form of speaker recognition (text-independent) is still not very accurate for large speaker populations, but if you constrain the words spoken by the user (text-dependent) and do not allow the speech quality to vary too wildly, then it too can be done on a workstation.

See the sidebar "Applications" for a description of typical speech and speaker recognition applications.

Factors affecting speech recognition

Modern speech recognition research began in the late 1950s with the advent of the digital computer. Combined with tools to capture and analyze speech, such as analog-to-digital converters and sound spectrograms, the computer allowed researchers to search for ways to extract features from speech that allow discrimination between different words. The 1960s saw advances in the automatic segmentation of speech into units of linguistic relevance (such as phonemes, syllables, and words) and on new pattern-matching and

0018-9162/90/0800-0026$01.00 © 1990 IEEE

Applications

Although the performance of speech and speaker recognition systems is far from perfect, these systems have already proven their usefulness for certain applications.

Speech recognition. Currently, speech recognition is most often applied in manufacturing for companies needing voice entry of data or commands while the operator's hands are otherwise occupied. Related applications occur in product inspection, inventory control, command/control, and material handling. Speech recognition also finds frequent application in medicine, where voice input can significantly accelerate the writing of routine reports.

Speech recognition over the telephone network, although less used, has the greatest potential for growth. Automating the telephone operator's job can greatly reduce operating costs for telephone companies. Furthermore, speech recognition can help users control the personal workstation or interact with other applications remotely when touch-tone keypads are not available. (Telephone network applications are described in articles by Matthew Lennig and Ryohei Nakatsu elsewhere in this issue.)

Finally, speech recognition offers greater freedom to the physically handicapped.

Typical real-world applications:

- Delco electronics employs IBM PC/AT-Cherry Electronics and Intel RMX86 recognition systems to collect circuit board inspection data while the operator repairs and marks the boards.
- Southern Pacific Railway inspectors now routinely use a PC-based Votan recognition system to enter car inspection information from the field by walkie-talkie.
- Michigan Bell has installed a Northern Telecom recognition system to automate collect and third-number billed calls. AT&T has also put in field trial systems to automate call-type selection in its Reno, Nevada, and Hayward, California, offices.

Speaker recognition. Speaker recognition has been applied most often as a security device to control access to buildings or information. One of the best known examples is the Texas Instruments corporate computer center security system. Security Pacific has employed speaker verification as a security mechanism on telephone-initiated transfers of large sums of money. In addition to adding security, verification is advantageous because it reduces the turnaround time on these banking transactions. Bellcore uses speaker verification to limit remote access of training information to authorized field personnel. Speaker recognition also provides a mechanism to limit the remote access of a personal workstation to its owner or a set of registered users.

In addition to its use as a security device, speaker recognition could be used to trigger specialized services based on a user's identity. For example, you could configure an answering machine to deliver personalized messages to a small set of frequent callers.

classification algorithms. By the 1970s, a number of important techniques essential to today's state-of-the-art speech recognition systems had emerged, spurred on in part by the Defense Advanced Research Projects Agency speech recognition project. These techniques have now been refined to the point where very high recognition rates are possible, and commercial systems are available at reasonable prices.

Five factors can be used to control and simplify the speech recognition task[1]:

(1) *Isolated words.* Speech consisting of isolated words (short silences between the words) is much easier to recognize than continuous speech because word boundaries are difficult to find in continuous speech. Also, coarticulation effects in continuous speech cause the pronunciation of a word to change depending on its position relative to other words in a sentence. For example, "did you?" is not the same as "did" + short silence + "you?" Other effects depend on the rate of speaking as well, such as our tendency to drop the "t" in want when saying "want to" casually and quickly.

Error rates can definitely be reduced by requiring the user to pause between each word. For example, in a study by Bahl et al.,[2] error rates of 9 percent for continuous recognition decreased to 3 percent for isolated-word recognition. However, this type of restriction places a burden on the user and reduces the speed with which information can be input to the system (from a range of about 150-250 words per minute down to about 20-100 words per minute).

(2) *Single speaker.* Speech from a single speaker is also easier to recognize than speech from a variety of speakers because most parametric representations of speech are sensitive to the characteristics of the particular speaker. This makes a set of pattern-matching templates for one speaker perform poorly for another speaker. Therefore, many systems are speaker dependent — trained for use with each different operator. Relatively few speech recognition systems can be used by the general public. A rule of thumb used by many researchers is that, for the same task, speaker-dependent systems will have error rates roughly three to five times smaller than speaker-independent ones.

One way to make a system speaker independent is simply to mix training templates from a wide variety of speakers. A more sophisticated approach will attempt to look for phonetic features that are relatively invariant between speakers.

(3) *Vocabulary size.* The size of the vocabulary of words to be recognized also strongly influences recognition accuracy. Large vocabularies are more likely to contain ambiguous words than small vocabularies. Ambiguous words are those whose pattern-matching templates appear similar to the classification algorithm used by the recognizer. They are therefore harder to distinguish from each other. Of course, small vocabularies composed of many ambiguous words can be particularly difficult to recognize. A famous example is the E-set, which consists of a subset of the English alphabet and digits: "B," "C,"

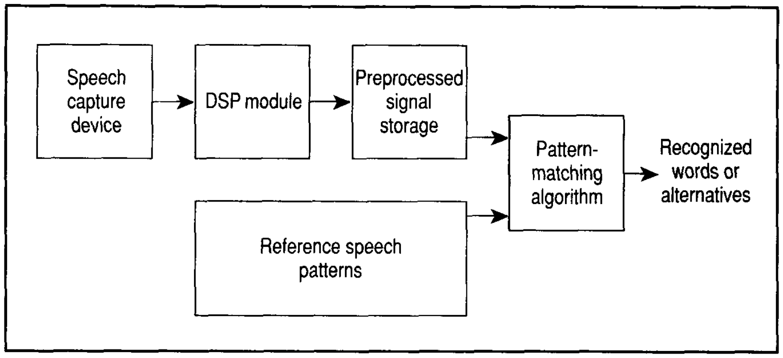

Speech capture device
DSP module
Preprocessed signal storage
Pattern-matching algorithm
Recognized words or alternatives
Reference speech patterns

Figure 1. Components of a typical speech recognition system.

"D," "E," "G," "P," "T," "V," "Z," and "three."

The amount of time it takes to search the speech model database also relates to vocabulary size. Systems containing many pattern templates typically require pruning techniques to cut down the computational load of the pattern-matching algorithm. By ignoring potentially useful search paths, pruning heuristics can also introduce recognition errors.

(4) *Grammar*. The grammar of the recognition domain defines the allowable sequences of words. A tightly constrained grammar is one in which the number of words that can legally follow any given word is small. The amount of constraint on word choice is referred to as the *perplexity* of the grammar. Systems with low perplexity are potentially more accurate than those that give the user more freedom because the system can limit the effective vocabulary (and search space) to those words that can occur in the current input context. For example, a system described in Kimbal et al.[3] had an error rate of 1.6 percent with perplexity 19 (tightly constrained), while the error rate hit about 4.5 percent with perplexity 58 (more loosely constrained).

(5) *Environment*. Background noise, changes in microphone characteristics, and loudness can all dramatically affect recognition accuracy. Many recognition systems are capable of very low error rates as long as the environmental conditions remain quiet and controlled. However, performance degrades when noise is introduced or when conditions differ from the training session used to build the reference templates. To compensate, the user must almost always wear a head-mounted, noise-limiting microphone with the same response characteristics as the microphone used during training.

Components of a speech recognition system

Most computer systems for speech recognition include the following five components (see Figure 1):

(1) A *speech capture device*. This usually consists of a microphone and associated analog-to-digital converter, which digitally encodes the raw speech waveform.

(2) A *digital signal processing module*. The DSP module performs endpoint (word boundary) detection to separate speech from nonspeech, converts the raw waveform into a frequency domain representation, and performs further windowing, scaling, filtering, and data compression.[4] The goal is to enhance and retain only those components of the spectral representation that are useful for recognition purposes, thereby reducing the amount of information that the pattern-matching algorithm must contend with. A set of these speech parameters for one interval of time (usually 10-30 milliseconds) is called a *speech frame*.

(3) *Preprocessed signal storage*. Here, the preprocessed speech is buffered for the recognition algorithm.

(4) *Reference speech patterns*. Stored reference patterns can be matched against the user's speech sample once it has been preprocessed by the DSP module. This information is stored as a set of speech templates or as generative speech models.

(5) A *pattern matching algorithm*. The algorithm must compute a measure of goodness-of-fit between the preprocessed signal from the user's speech and all the stored templates or speech models. A selection process chooses the template or model (possibly more than one) with the best match.

Two major types of pattern matching in use are template matching by dynamic time warping and hidden Markov models. Artificial neural networks applied to speech recognition have also had some success, but this work is still in the early stages of research.[5] Moreover, linguistic knowledge incorporated into the pattern-recognition algorithm can enhance performance. However, such sophisticated techniques lie outside of the scope of this article (see, for example, O'Shaughnessy[4] and Mariani[6]).

Template matching by dynamic time warping became very popular in the 1970s. Template matching is conceptually simple. You want to compare the preprocessed speech waveform directly against a reference template by summing the distances between respective speech frames. However, biological limitations tend to produce nonlinear variations in timing from utterance to utterance. Consequently, the various frames of a word may be out of alignment with the corresponding frames of the given template. Since the order of speech events is fairly constant, you correct the misalignment by stretching the template in some places and compressing it in others to find an optimum match. Dynamic programming helps compute the optimum match. The sidebar "Dynamic time warping" illustrates the resulting time warp process.

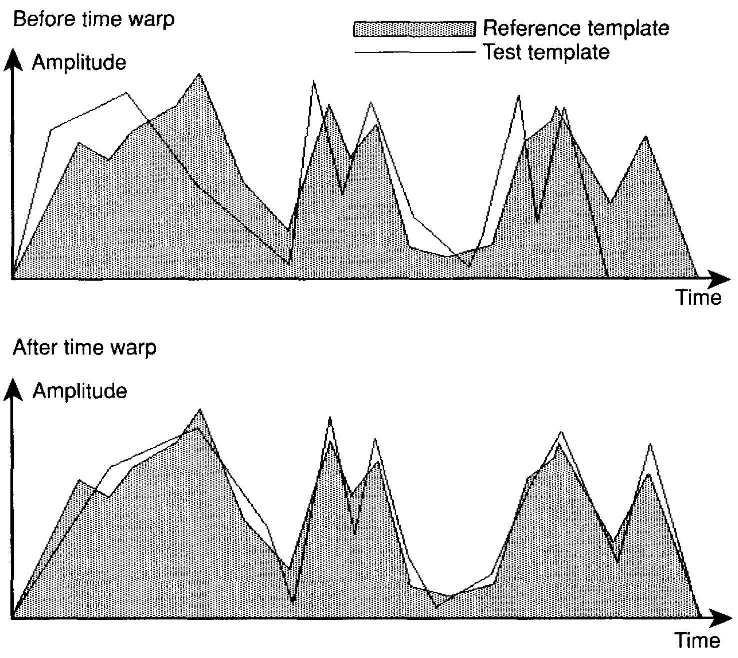
Hidden Markov models are used in most current research systems because this technique produces better results for continuous speech with moderate-size vocabularies. HMMs are stochastic state machines that associate probabilities of producing sounds with transitions from state to state. An ideal HMM models speech with the same variations that occur in human speech due to coarticulation and other effects. Speech generated by a human being is matched against an HMM by computing the probability that the HMM would have generated the same utterance or by finding the state sequence through the HMM that has the highest probability of producing the utterance. The fact that HMMs generate poor-quality speech explains why recognition based on HMMs is still not perfect.

The sidebar "Hidden Markov models" further details the use of HMMs. Markov chains, although known about for almost a century, have only been successfully used in the context of speech recognition for the past 15 years or so. Until recently, no method existed for optimizing the model

Dynamic time warping

Frame distances between the processed speech frames and those of the reference templates are summed to provide an overall distance measure of similarity. But, instead of taking frames that correspond exactly in time, you would do a time "warp" on the utterance (and scale its length) so that similar frames in the utterance line up better against the reference frames. A dynamic programming procedure finds a warp that minimizes the sum of frame distances in the template comparison. The distance produced by this warp is chosen as the similarity measure.

In the illustration here, the speech frames that make up the test and reference templates are shown as scalar amplitude values plotted on a graph with time as the *x* axis. In practice, they are multidimensional vectors, and the distance between them is usually taken as the Euclidean distance. The graphs show how warping one of the templates improves the match between them. (For further information, see chapter 10 of O'Shaughnessy.[4])



parameters to generate observed speech patterns. (The US Department of Defense actually suppressed publication of the advances in HMM algorithms for a while in the mid-1970s, probably because of their use in cryptanalysis.) As well as representing low-level speech segments and transitions, hidden Markov models provide a framework on which you can model higher level structures in continuous speech signals and incorporate other knowledge about the communication.

Current speech recognition systems

Current speech recognition systems can be categorized according to the types of constraint they place on the speech. At one end of the spectrum fall speaker-independent, continuous, unconstrained-grammar, large-vocabulary systems. These systems are still very much in the research stage.

Several systems among those representing the state of the art were trained and tested on the same speech data — the DARPA resource management database — and are easily compared. The DARPA resource management task involves queries and commands to a database of warships. The associated database consists of a 997-word vocabulary and grammars with various complexities. Sphinx, a recognizer developed at Carnegie Mellon University, has a maximum word-recognition accuracy of 93.7 percent for a grammar of perplexity 60 and 70.6 percent for a grammar of perplexity 997.[1] BBN's Byblos[7] and a system developed at Lincoln Labs[8] have word accuracies of 88.7 percent and 87.4 percent, respectively, for the perplexity 60 grammar (BBN's system requires about two minutes of speech to adapt to a particular speaker before reaching this level of performance). Texas Instruments* and Stanford Research Institute[9] have reported systems with 44.3 percent and 40.4 percent accuracy on the perplexity 997 grammar. These systems have considerably lower *sentence* accuracies.

Representative of the state of the art in speaker-dependent, isolated-word, large-vocabulary recognizers are systems like IBM's Tangora recognizer, which is capable of 97 percent accuracy for a 20,000-word vocabulary[10] and NEC's 97.5 percent accurate, 1,800-word system.[11]

A variety of other systems trade off constraints on the input speech for higher recognition accuracies. Among these are the AT&T Bell Labs telephone-grade, speaker-independent, connected-digit recognizer (98.5 percent accurate when the number of digits is known[12]) and a speaker-dependent version of BBN's Byblos, which measured 94.8 percent accurate on the perplexity 60 DARPA resource management task.

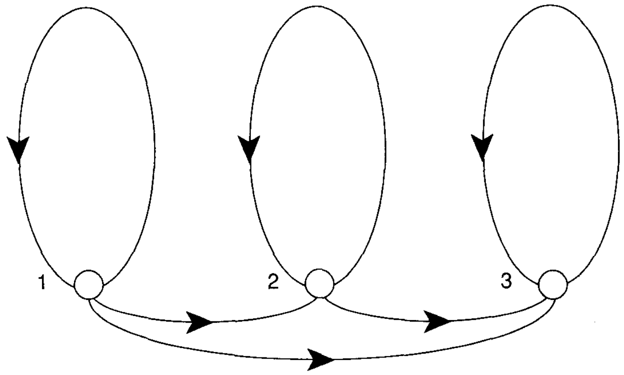
At the highly constrained speech end of the spectrum fall speaker-dependent, single-word, small-vocabulary recognition systems. A variety of such systems developed can achieve accuracies above 99 percent.

Various commercial systems have appeared for Sun workstations and IBM-compatible PCs over the past few years. Table 1 summarizes the capabilities, costs, and manufacturers' claimed accuracies of a sample of these commercial products. Although several companies advertise speaker-independent, continuous, large-

* See Kai-fu Lee,[1] p. 133.

Hidden Markov models

A hidden Markov model (HMM) is a doubly stochastic process for producing a sequence of observed symbols. An underlying *stochastic finite state* machine (FSM) drives a set of stochastic processes, which produce the symbols. When a state is entered after a state transition in the FSM, a symbol from that state's set of symbols is selected probabilistically for output. The term "hidden" is appropriate because the actual state of the FSM cannot be *observed* directly, only through the symbols emitted. In the example here, the sequence of symbols AAaaB could have been produced by any of three different state transition sequences.



State	Possible Outputs
1	A,a
2	a
3	B

AAaaB could be produced by the following state sequences:

→ 1 → 1 → 1 → 1 → 3
or → 1 → 1 → 1 → 2 → 3
or → 1 → 1 → 2 → 2 → 3

Although not shown in the example, probabilities are attached to the finite state transitions, and discrete probability distributions control the symbol output for each state (continuous density HMMs also exist). In the case of isolated word recognition, each word in the vocabulary has a corresponding HMM. These HMMs might actually consist of HMMs that model subword units such as phonemes connected to form a single word-model HMM. In the case of continuous word recognition, a single HMM corresponds to the domain grammar. This grammar model is constructed from word-model HMMs. The observable symbols correspond to (quantized) speech frame measurements.

An algorithm known as the forward/backward (or Baum-Welch) algorithm finds a set of state transition probabilities and symbol output distributions for each HMM. This gradient descent algorithm uses training data to iteratively refine an initial (possibly *random*) set of model parameters such that the HMM is more likely to generate patterns from the training set.

After this initial training stage, a word or sentence to be recognized is spoken, and speech measurements are made that reduce the utterance to a sequence of symbols. In the case of isolated word recognition, the forward algorithm computes the probability that each word model produced the observed sequence of symbols — the model with the highest probability represents the recognized word. In the case of continuous recognition, the Viterbi algorithm finds the state transition path, through the grammar model, *with the maximum likelihood* of generating the set of measurements. The sequence of word models on this path corresponds to the recognized sentence. (For further information see "Introduction to Hidden Markov Models" by L.R. Rabiner and B.H. Juang, published in *IEEE Trans. Acoustics, Speech, and Signal Processing*, Jan. 1986, pp. 4-16.)

vocabulary speech recognition, they carefully avoid making strong claims about the accuracy of their products. With commercial systems, you typically get what you pay for. Products available for less than $1,000 US are isolated-word, small-vocabulary recognizers. Speaker-dependent, isolated-word, large-vocabulary recognizers for automated dictation are available for a few thousand dollars. You'll see an order of magnitude leap in price when you move to large-vocabulary, speaker-independent, continuous-speech recognizers.

Speaker recognition — the voice, not just the words

Speaker recognition is related to speech recognition. When the task involves identifying the person talking rather than what is said, the speech signal must be processed to extract measures of speaker variability instead of being analyzed by segments corresponding to phonemes or pieces of text one after the other. For speaker recognition, only one classification is made, based on part or all of an input test utterance. Although various studies have shown that certain acoustical features work better than others in predicting speaker identity, few recognizers examine specific sounds because of difficulties in phone segmentation and identification.

Both automatic speaker verification and speaker identification use a stored database of reference patterns (templates) for *N* known speakers. Both involve similar analysis and decision techniques. Verification is simpler because it only requires comparing the test pattern against one reference pattern and it involves a binary decision: Is there a good enough match against the template of the claimed speaker? The error rate for speaker identification can be much greater because it requires choosing which of the *N* voices known to the system best matches the test voice or "no match" if the test voice differs sufficiently from all the reference templates.

Comparing test and reference utterances for speaker identity is much simpler for identical underlying texts, as in text-dependent speaker recognition. With cooperative speakers you can apply speaker recognition straightforwardly by using the same words to train the system and then test it. This usually happens in verification, but speaker identification often requires

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase®
Smarter legal research.