17

Speech Recognition and Understanding Systems

17.1 Speech Processing

The NLP systems I have already described required that their English input be in text format. Yet, there are several instances in which speaking to a computer would be preferable to typing at one. People can generally speak faster than they can type (about three words per second versus about one word per second), and they can speak while they are moving about. Also, speaking does not tie up hands or eyes.

In discussing the problem of computer processing of speech, it is important to make some distinctions. One involves the difference between recognizing an isolated spoken word versus processing a continuous stream of speech. Most AI research has concentrated on the second and harder of these problems. Another distinction is between speech *recognition* and speech *understanding*.

By speech recognition is meant the process of converting an acoustic stream of speech input, as gathered by a microphone and associated electronic equipment, into a text representation of its component words. This process is difficult because many acoustic streams sound similar but are composed of quite different words. (Consider, for example, the spoken versions of "There are many ways to recognize speech," and "There are many ways to wreck a nice beach.") Speech understanding, in contrast, requires that what is spoken be *understood*. An utterance can be said to be understood if it elicits an appropriate action or response, and this might even be possible without recognizing *all* of its words.

Understanding speech is more difficult than understanding text because there is the additional problem of processing the speech waveform to extract the words being uttered. Speech, as it is captured by a microphone, is converted into an electronic signal or waveform, which can be displayed on an oscilloscope. In Fig. 17.1, I show a waveform generated by a person saying "This is a test." This diagram shows the amplitude (voltage) of the speech signal plotted against time. The sections of the waveform corresponding to the words are demarcated by the boxes at the top of the diagram. The boxes at the bottom show acoustical elements of these words, which are called "phones."

In general, phones are the sounds that correspond to vowels or consonants. English speech is thought to be composed of forty or so different phones. Special alphabets have been devised to represent phones. One is the International Phonetic Alphabet (IPA), which contains the phones of all known languages. IPA uses several special characters that do not have standard computer (ASCII) codes. Another, containing just the phones used in American English and using only standard characters, is

https://doi.org/10.1017/CBO9780511819346.021 Published online by Cambridge University Press

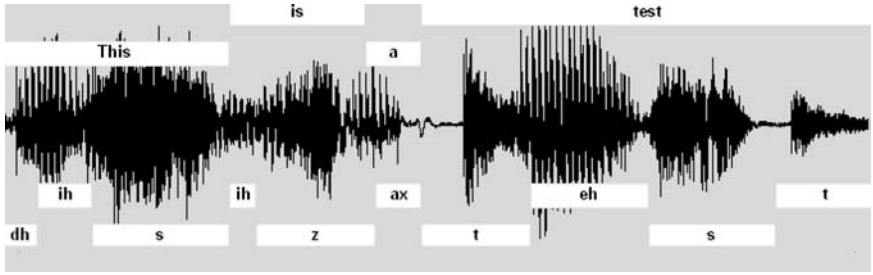

Figure 17.1. A speech waveform. (Used with permission of Gunish Rai Chawla.)

ARPAbet, which was developed during speech-processing research sponsored by DARPA. The phones boxed in Fig. 17.1 use the ARPAbet notation. The table in Fig. 17.2 shows the ARPAbet phones and sample words containing them.

Early speech recognition systems attempted first to segment the speech waveform into its constituent phones and then to assemble the phones into words. To do so,

Symbol	Example Sound	Symbol	Example Sound
Consonants		**Vowels**	
[p]	**p**at	[iy]	lil**y**
[t]	**t**om	[ih]	m**i**ss
[k]	**c**at	[ey]	l**a**zy
[b]	**b**oy	[eh]	m**e**ss
[d]	**d**ip	[ae]	**a**fter
[g]	**g**arment	[aa]	p**o**p
[m]	**m**at	[ao]	**o**rchestra
[n]	**n**ut	[uh]	w**oo**d
[ng]	si**ng**	[ow]	l**o**tus
[f]	**f**ive	[uw]	t**u**lip
[v]	do**v**e	[uh]	b**u**tter
[th]	**th**istle	[er]	b**ir**d
[dh]	fea**th**er	[ay]	**i**tem
[s]	**s**at	[aw]	flower
[z]	ha**z**e	[oy]	t**oi**l
[sh]	sma**sh**	[y uw]	**few**
[zh]	ambro**s**ia	[ax]	r**u**ffian
[ch]	**ch**ic	[ix]	l**i**p
[jh]	pa**g**e	[axr]	leath**er**
[l]	**l**ick	[ux]	d**u**de
[w]	ki**w**i		
[r]	pa**r**se		
[y]	**y**ew		
[h]	**h**orse		
[q]	uh-oh (glottal stop)		
[dx]	bu**tt**er		
[nx]	wi**n**ter		
[el]	thist**le**		

Figure 17.2. Consonants and vowels in the ARPAbet phonetic alphabet.

https://doi.org/10.1017/CBO9780511819346.021 Published online by Cambridge University Press

the speech signal was first digitized, and various parameters, such as the frequency or pitch, were extracted. The ways in which the values of these parameters change in time were used to segment the waveform into units containing phones. Using dictionaries that associate the values of waveform parameters with phones and phones with words, the waveform was finally converted into text. The process sounds simple but it is actually quite complex because, among other things, the beginnings and endings of spoken words and their component phones overlap in complex patterns, and people often pronounce the same words in different ways. For example, the word "you" might be pronounced differently in "are you" [aa r y uw] and "did you" [d ih d jh uh].

Attempts to recognize speech began at Bell Laboratories as far back as the 1930s. In 1952, engineers at Bell Labs built a system for recognizing the numbers "zero" through "nine" uttered by a single speaker.[1] Other work was done in the 1950s and 1960s at RCA Laboratories, at MIT, in Japan, in England, and in the Soviet Union.[2] Work accelerated in the 1970s, some of which I'll describe next.

17.2 The Speech Understanding Study Group

Larry Roberts, who went to DARPA in late 1966 as "chief scientist" in the Information Processing Techniques Office (IPTO) and later became its director, became intrigued with the idea of building systems that could understand speech. Cordell Green, by then serving as a lieutenant in the U.S. Army, was assigned to IPTO under Roberts in early 1970 and was put in charge of funding and monitoring AI research projects. According to Green, Roberts told him "Do a feasibility study on a system that can recognize speech."[3]

So, at the end of March 1970, Green organized a meeting at Carnegie Mellon University of several of the DARPA contractors and others interested in speech processing to discuss the feasibility of speech understanding by computer. Among those attending the meeting were researchers from SDC, Lincoln Laboratory, MIT, CMU, SRI, and BBN. It was decided at the meeting to form a "study group" to assess the state of the art and to make recommendations concerning the launching of a major DARPA-supported project in speech understanding. The group was to be chaired by Allen Newell of CMU.[4]

During the March meeting, Roberts was persuaded to talk about the kind of speech-understanding system that he had in mind. According to the study group's rendition of his remarks, Roberts was thinking about a system that could accept continuous speech from many cooperative users, over a telephone, using a vocabulary of 10,000 words, with less than 10% semantic error, in a few times real time, and be demonstrable in 1973.

The study group held its first meeting at BBN on May 26 and 27, 1970. At that meeting, the group considered some specific tasks that the understanding system would be able to engage in. Among these were answering questions about data management, answering questions about the operational status of a computer, and consulting about a computer operating system.

A final meeting of the group was held at SDC in Santa Monica on July 26–28, 1970. The recommendation of the group (in brief) was to aim for a system that could accept continuous speech, from many cooperative speakers of the "general

American dialect," over a good quality microphone (not a telephone), using a selected vocabulary of 1,000 words (not 10,000 words), with a "highly artificial syntax," involving tasks such as data management or computer status (but not consulting), with less than 10% error, in a few times real time, and be demonstrable in 1976 (not 1973) with a moderate chance of success. A final report of the group was drafted after the meeting, delivered to DARPA, and eventually published in 1973.[5]

Although there had been much prior research in speech processing by computer (nicely summarized in the study group's report), not everyone was optimistic about success. One naysayer was John R. Pierce, a researcher at Bell Laboratories, where much speech-recognition work had already taken place. In 1969, Pierce wrote a letter[6] to the *Journal of the Acoustical Society of America* in which he claimed that most people working on speech recognition were acting like "mad scientists and untrustworthy engineers. The typical recognizer gets it into his head that he can solve 'the problem.'" In the same letter, though, he also wrote that

> . . . performance would continue to be very limited unless the recognizing device *understands* what is being said with something of the facility of a native speaker (that is, better than a foreigner who is fluent in the language). If this is so, should people continue work toward speech recognition? Perhaps this is for people in the field to decide. [My italics.]

17.3 The DARPA Speech Understanding Research Program

In fact, people in the field did decide. In October 1971, Roberts established at DARPA a five-year Speech Understanding Research (SUR) program based largely on the study group's report. Its budget was about $3 million per year. CMU, Lincoln Laboratory, BBN, SDC, and SRI were contracted to build systems. Complementary research would be performed at Haskins Laboratories, the Speech Communications Research Laboratory, the Sperry Univac Speech Communications Department, and the University of California at Berkeley.

In 1976, some of these efforts resulted in systems that were demonstrated and tested against the program's goals. CMU developed two of these, HARPY and HEARSAY-II. BBN produced HWIM (Hear What I Mean). SRI and SDC formed a partnership in which SDC developed the acoustic processing components and SRI developed the parsing and semantic components. However, the SDC effort ran into difficulties with computer access, so the combined SRI/SDC system was never formally tested. I'll briefly summarize the BBN work and then describe the CMU work in more detail.[7]

17.3.1 *Work at BBN*

SPEECHLIS was the first speech understanding system developed at BBN. It was designed to answer spoken questions about the moon rocks database (the one used in BBN's earlier LUNAR system). It was rather slow and was not systematically tested.[8]

HWIM was designed to be a travel budget manager's automated assistant and was able to respond to spoken questions such as "How much is left in the speech understanding budget?"[9] In its final version, HWIM was tested on two versions,

each of sixty-four different utterances by three male speakers. Thirty-one of these sentences had previously been used by the system as it was being designed, so there might have been some implicit (if unintentional) built-in extra capability for dealing with those sentences. The sentences ranged in length from three to thirteen words. HWIM was able to respond correctly to 41% of the sentences and "close" to correctly to 23% more of them. The system did not respond at all to 20% of the sentences. Although both SPEECHLIS and HWIM pioneered new and important methods in speech understanding, HWIM's performance was generally regarded as not meeting the original DARPA objectives. (Their designers claimed that the test was not indicative of HWIM's potential and that they could have done better with more time.)

17.3.2 *Work at CMU*

In 1969, Raj Reddy left Stanford to become a faculty member at Carnegie Mellon University. One of the first speech systems he and colleagues worked on at CMU was called HEARSAY (later renamed HEARSAY-I).[10] It used a number of independent computational processes to recognize spoken moves in chess from a given board position, such as "king bishop pawn moves to bishop four." It was during the early stages of this work, that DARPA formed the Speech Understanding Study Group and initiated work in speech understanding. A public demonstration of HEARSAY-I recognizing connected speech was given in June 1972.

Three different speech recognition and understanding systems were developed at CMU under the umbrella of the DARPA speech understanding research effort. These were DRAGON, HARPY, and HEARSAY-II, and they all contributed important AI ideas. Work on these systems was led by Allen Newell, Raj Reddy, James Baker, Bruce Lowerre, Lee Erman, Victor Lesser, and Rick Hayes-Roth.[11]

A. DRAGON

During the early days of CMU's speech understanding research, a Ph.D. student, James K. Baker, began work on a speech understanding system he called "DRAGON."[12] (According to Allen Newell, the name DRAGON was meant "to indicate that it was an entirely different kind of beast from the AI systems being considered in the rest of the speech effort."[13]) Like HEARSAY-I, DRAGON was designed to understand sentences about chess moves.

DRAGON introduced powerful new techniques for speech processing – elaborations of which are used in most modern speech recognition systems. It used statistical techniques to make guesses about the most probable strings of words that might have produced the observed speech signal. It was an early example of the importation of probabilistic representations and associated computational methods into AI. We'll see a good deal more of these in later chapters.

I'll try to explain the main ideas without using much mathematics. Using the notation introduced in Section 2.3.2, suppose we let x stand for a string of words and y stand for the speech waveform that is produced when x is spoken. (Actually, we'll let y be some information-preserving representation of the waveform in terms of its easily measurable properties such as the amounts of energy the waveform contains

in various frequency bands. For simplicity, I'll continue to call y a waveform, even though I mean its representation, which might be different for different speech understanding systems.)

Because the same speaker may say the same words somewhat differently on different occasions, and different speakers certainly will say them differently, the word string x does not completely determine what the speech waveform y will be. That is, given any x, we can only say what the probabilities of the different y's might be. As described in Chapter 2, these probabilities are written in functional form as $p(y \mid x)$ (read as the "probability of y given x"). In principle, the actual values of $p(y \mid x)$ for some particular x, say $x = X$, could be estimated, for example, by having a number of speakers utter the word string X many different times and tabulating how frequently different speech waveforms y occur. This process would have to be repeated for many different word strings. DRAGON avoided this tedious tabulation in a way to be explained shortly.

For speech recognition, however, we want to know the probability of a word string x, given the speech signal y, so that we can select the most probable x. That is, we want $p(x \mid y)$ rather than $p(y \mid x)$. We could use Bayes's rule as before, to produce the desired probability as follows:

$$p(x \mid y) = p(y \mid x)p(x)/p(y).$$

Upon observing a particular waveform, say $y = Y$, here is how we would use the quantities in this formula to decide what word string x was most probably uttered:

1. Look up all the values of $p(Y \mid x)$ for all of the values of x we are considering. (We don't have to do this for *all* possible strings of words, but only for those allowed by the vocabulary and syntax of the specialized area appropriate to the speech understanding task – chess moves in the case of DRAGON.)
2. Multiply each of these values by $p(x)$. (The decision should be biased in favor of likely word strings.)
3. Select that x, say X, for which the product is the largest. [We can ignore dividing by $p(y)$ because its value does not affect which $p(x \mid Y)$ is largest.]

Although this process would work in principle, it is quite impractical computationally. Instead, DRAGON and other modern speech-recognition systems exploit the hierarchical structure involved in what is presumed to be the way a speech waveform is generated. There are various levels in this hierarchy that could be identified. To oversimplify a bit, at the top of the hierarchy a given semantic idea is expressed by a string of words obeying the syntactic rules of the language. The string of words, in turn, gives rise to a string of phones – the phonetic units. Finally, the phone string is expressed by a speech waveform at the bottom of the hierarchy.

At each level, we have a sequence of entities, say, $x_1, x_2, \ldots x_n$, producing a sequence of other entities, say, $y_1, y_2, \ldots, y_n$. We can diagram the process as shown in Fig. 17.3.

The DRAGON system made some simplifying assumptions. It assumed that each x_i in the sequence of x's is influenced only by its immediate precedent, x_{i-1}, and not

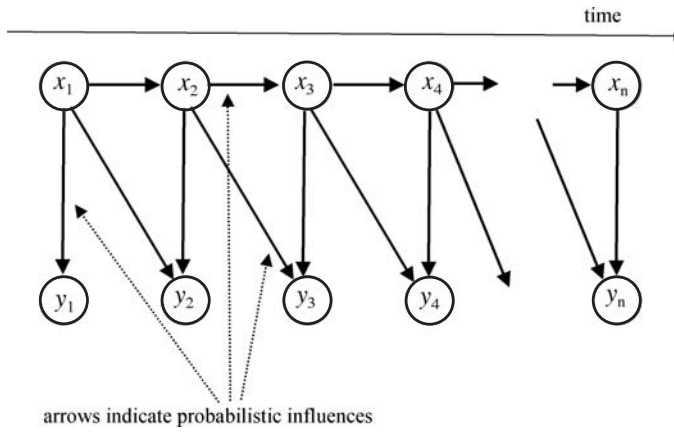

Figure 17.3. Two hierarchical levels in speech generation.

by any other of the x_i. This assumption is called the Markov assumption. [Andrey Andreyevich Markov (1856–1922) was a Russian mathematician. He used (what was later called) a Markov model to analyze the statistics of a sequence of 20,000 Russian letters taken from Pushkin's novel *Eugene Onegin*.[14] Markov models are used extensively in physics and engineering. Google uses the Markov assumption, for example, in its computation of page rank.] Of course, we know that each word in a sequence depends on more than just the immediately preceding word. Even so, the Markov assumption makes computations simpler and still allows good performance.

Further, it was assumed that each y_i was influenced only by x_i and x_{i-1}. All of these "influences" are probabilistic. That is, given quantities like x_3 and x_4, for example, the value of y_4 is not completely determined. One can only say what the probabilities of the values of y_4 might be; these are given by the functional expression $p(y_4 \mid x_3, x_4)$. Probability values for the y's are thus given by what is called a "probabilistic function of a Markov process." To produce estimates of these probabilities, statistics can be gathered during a "learning process" (in which a speaker utters a training set of sentences).

DRAGON combined these separate levels into a network consisting of a hierarchy of probabilistic functions of Markov processes. Entities representing segments of the speech waveform were at the bottom, entities representing phones were in the middle, and entities representing words were at the top. At each level, Bayes's rule was used to compute probabilities of the x's given the y's. Because only the speech waveform at the bottom level was actually observed, the phones and words were said to be "hidden." For this reason, the entire network employed hidden Markov models (HMMs). DRAGON was the first example of the use of HMMs in AI. They had been developed previously for other purposes.[15]

Using this network, recognition of an utterance was then achieved by finding the highest probability path through the network. Computing the probabilities for syntactically valid word sequences, given the sequence of segments of the observed speech waveform, is a problem that is similar to one I described earlier, namely,

computing the confidences of strings of characters on FORTRAN coding sheets (see p. 72). Again, a method based on dynamic programming was used. As Baker wrote, "The optimum path is found by an algorithm which, in effect, explores all possible paths in parallel."[16] At the end of the process, the most probable syntactically legal string of words is identified. The mathematical operations for making these computations are too complex to explain here, but they can be performed efficiently enough to make speech recognition practical.

Although the DRAGON system was not among those that were finally tested against DARPA's speech understanding system objectives, Baker claimed that its initial results were "very promising" and that in "its first test with live speech input, the system correctly recognized every word in all nine sentences in the test."[17] DRAGON became the basis for a commercial product, "Dragon Naturally Speaking," first developed and marketed by Dragon Systems, a company founded by Baker and his wife, Janet.

B. HARPY

HARPY was a second system produced at CMU under DARPA's speech understanding research effort. Bruce T. Lowerre designed and implemented the system as part of his Ph.D. research.[18] HARPY combined some of the ideas of HEARSAY-I and DRAGON. Like DRAGON, it searched paths through a network to recognize a spoken sentence, but it did not annotate the links between nodes in the network with transition probabilities like DRAGON did. Like HEARSAY-I, HARPY used heuristic search methods.

Versions of HARPY were developed for understanding spoken sentences about several different task areas. The main one involved being able to answer questions about, and to retrieve documents from, a database containing summaries (called "abstracts") of AI papers. Here are some examples:

"Which abstracts refer to theory of computation?"
"List those articles."
"Are any by Feigenbaum and Feldman?"
"What has McCarthy written since nineteen seventy-four?"

HARPY could handle a vocabulary of 1,011 words. Instead of using a grammar with the conventional syntactic categories such as `Noun`, `Adjective`, and so on, HARPY used what is called a "semantic grammar," one that has expanded categories such as `Topic`, `Author`, `Year`, and `Publisher` that were semantically related to its subject area, namely, data about AI papers. HARPY's grammar was limited to handle just the set of sentences about authors and papers that HARPY was supposed to be able to recognize.

The network was constructed from what were called "knowledge sources" (KSs), which consisted of information needed for the recognition process.[19] The first of these encoded syntactic knowledge about the grammar.

A second knowledge source used by HARPY described how each word in HARPY's vocabulary might be pronounced. And, because in spoken language word boundaries overlap in ways that depend on the words involved, successful recognition requires a third knowledge source dealing with such phenomena. A fourth knowledge source

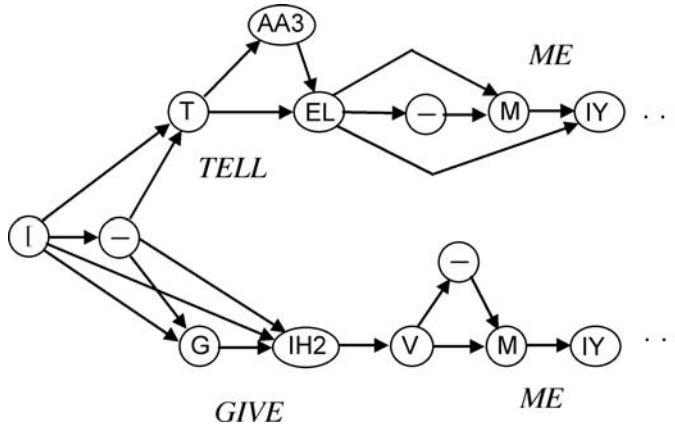

Figure 17.4. A partial network of the phones that might occur in a spoken sentence.

specified the phones involved in the pronunciation of words and transitions between words.

HARPY combined all of this knowledge into a giant network of phones representing all the possible ways that syntactically legal sentences might be spoken. Each "phone node" in the network was paired with a representation of a segment of a speech waveform, called a "spectral template," expected to be associated with that particular phone. These templates were obtained initially by having a speaker read about 700 sentences. They could be "tuned" for a new speaker by having the speaker read about 20 selected sentences during a "learning" session. A partial network of phones is shown in Fig. 17.4 to illustrate the general idea. HARPY's actual network had 15,000 nodes. The network is for those parts of the sentences that begin with "Tell me . . ." and "Give me. . . ." The symbols inside the nodes represent phones, using DRAGON's notation for them. Arrows represent possible transitions from one phone to the next. Note that there are multiple paths, corresponding to different ways to pronounce the words.

To recognize the words in a spoken sentence, the observed speech waveform was first divided into variable-length segments that were guessed to correspond to the sequence of phones in the waveform. A spectral template was computed for each of these segments. The recognition process then proceeded as follows: The spectral template corresponding to the first spectral segment in the speech waveform was compared against all of the templates corresponding to the phones at the beginning of the network. In reference to Fig. 17.4, these would include comparisons against templates for –, T, G, and IH2 because they were among the nodes in the network that could be reached in one step from the start node, namely, [. (Of course, in using the complete network rather than just the partial example just illustrated, several more comparisons would be made against templates of additional phone nodes reachable in one step from the start node.) The best few matches were noted, and the paths to these nodes were designated to be the best one-step partial paths through the network. At the next stage, the spectral template of the next waveform segment was compared against the templates of all of those phone nodes reachable by

extending the best one-step paths one more step. Using the values of the comparisons computed so far, a set of best two-step partial paths was identified. This process continues until the end of the network was reached. At that time the very best path found so far could be associated with the words associated with the nodes along that path. This word sequence was then produced as HARPY's recognition decision.

HARPY's method of searching for a best path through the network can be compared with the A* heuristic search process described earlier. Whereas A* kept the entire search "frontier" available for possible further searching, HARPY kept on its frontier only those nodes on the best few paths found so far. (The number of nodes kept on the frontier was a parameter that could be set as needed to control search.) HARPY's designers called this technique "beam search" because the nodes visited by the search process were limited to a narrow beam through the network. Because nodes not in the beam were eliminated as the process went on, it is possible that the best complete path found by HARPY might not be the overall best one in the network. (One of the eliminated nodes might be on this overall best path.) Even so, the path found usually corresponded to a correct interpretation of the spoken sentence.

At the end of the DARPA speech understanding project, HARPY was tested on 100 sentences spoken by three male and two female speakers. It was able to understand over 95% of these sentences correctly, thereby meeting DARPA's goal of less than 10% error. On average, HARPY executed about 30 million computer instructions to deal with one second of speech. Using a 0.4-million instructions per second (0.4 MIPS) machine (a DEC PDP-KA10), it would take over a minute to process a second of speech; although this is quite a bit worse than real-time performance, it achieved DARPA's goal of "a few times real time" (if we interpret "a few" somewhat accommodatingly). To put the real-time matter in perspective, today's computers process billions of instructions per second. HARPY was the only system to meet DARPA's goals.

C. HEARSAY-II

Finally, HEARSAY-II, a redesigned and improved version of HEARSAY-I, was perhaps the most ambitious of CMU's speech projects.[20] Like HARPY, HEARSAY-II was designed to answer questions about, and to retrieve documents from, a database containing abstracts of AI papers. (An earlier task considered was to retrieve wire-service news stories.) It too was limited to a vocabulary of 1,011 words and used a semantic grammar specialized to its subject area.

The first steps in HEARSAY's processing of an utterance involved segmenting the speech waveform and labeling the phones estimated to be present in each segment. HEARSAY then used a novel method of gradually building these components into syllables, the syllables into words, the words into word sequences, and finally word sequences into phrases. The phrases were then converted into appropriate routines for accessing the database of AI papers.[21]

The processing method used by HEARSAY involved a layered structure called a "Blackboard." The labels of the phones estimated to be present, along with numbers related to their probabilities of occurrence, were "written" in one of the lower layers of the Blackboard. Specialized knowledge-source routines that "knew about" how syllables were constructed from phones "read" these labels and computed guesses

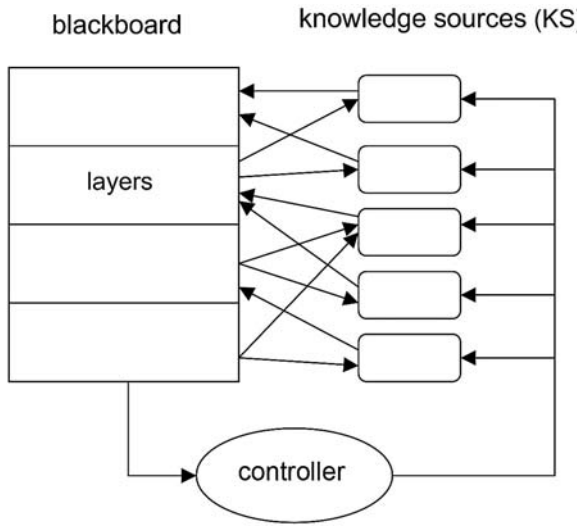

Figure 17.5. The Blackboard architecture.

about what syllables were in the utterance. These guesses, along with numbers measuring their confidences or likelihoods, were then written in the syllable layer of the Blackboard. Other knowledge-source routines that knew about how words were constructed from syllables read information already on the Blackboard and wrote guesses about words in the word layer of the Blackboard. And so on. HEARSAY-II had around 40 of these knowledge sources. The general idea is illustrated in Fig. 17.5.

In principle, a knowledge source could read or write information on any layer of the Blackboard that was relevant to it. Moreover, it could do so in what is called an "asynchronous" manner – not dependent on when other knowledge sources were doing their reading and writing. There were some knowledge sources that could write predictions about new words based on words already written in the word layer and on information in other layers. Knowledge sources could even write guesses about words in the word layer based on word sequences already written (with high confidence) in the sequence layer. This process of inferring what must be present in a lower layer (even though missed by initial processing) from what (from other evidence) is present in a higher layer is a theme that recurs often in later AI research. As far as I know, this extremely important AI innovation was first manifest in the HEARSAY-II system.

According to Raj Reddy,[22] one of the inventors of the Blackboard architecture (along with Victor Lesser, Lee Erman, and Frederick Hayes-Roth), Herbert Simon often used the word "blackboard" to describe the "working memory" component of the production system architecture he and Allen Newell were working with (see p. 468). A production system used IF–THEN rules (called productions), which were triggered by contents of the working memory and wrote new data in it. Reddy and team, recognizing the variety of different sources of knowledge relevant to speech processing, generalized the production system idea, extending the production rules

into larger programs, renaming them "knowledge sources," and elaborated working memory into the layered Blackboard structure.

At the end of the DARPA speech understanding project, HEARSAY-II was tested on twenty-three spoken sentences, brand new to the system, having an average of seven words per sentence, and 81% of these were recognized word-for-word correctly, although 91% led to the same database query as would have a word-for-word correct sentence. HEARSAY's designers claimed that this performance "comes close to meeting the ambitious goals . . . established for the DARPA program in 1971." Although HEARSAY-II came close the results were not quite as good as those of HARPY.

Although the Blackboard architecture is no longer used in modern speech recognition systems, it was adopted by several other AI programs. (We'll see one of these later in the book.) According to Russell and Norvig, "Blackboard systems are the foundation of modern user interface architectures."[23]

17.3.3 *Summary and Impact of the SUR Program*

CMU's HEARSAY-II and HARPY were demonstrated at CMU on September 8, 1976, and BBN's HWIM was demonstrated at BBN on September 10. In a summary report of the projects, MIT's Dennis Klatt wrote that "it is unclear whether there are large differences in ability among [these] three systems. However, only [HARPY] was able to meet the ARPA goals."[24]

The developers of HEARSAY-II attributed HARPY's superior performance to three factors: its more thorough search of potential solutions (permitted by its precomputed network of all the sentences that might have been spoken), its more thorough built-in knowledge of transition phenomena between adjacent words, and its more thorough testing, tuning, and debugging.[25]

Some researchers and DARPA program managers, however, argued about the way in which the tests were carried out and claimed that none of the systems met the SUR program objectives. In any case, DARPA decided not to fund a proposed follow-on program. The program did show, however, that speech understanding was a reasonable technical goal and stimulated progress in speech processing technologies, notably in system organization, syntax and semantics, and acoustic processing. A National Research Council report concluded that "DARPA's funding of research on understanding speech has been extremely important. . . . the results of this research have been incorporated into the products of established companies, such as IBM and BBN, as well as start-ups such as Nuance Communications (an SRI spinoff) and Dragon Systems. . . . The leading commercial speech-recognition program on the market today, the Dragon "NaturallySpeaking" software [now sold by Nuance], traces its roots directly back to the work done at CMU between 1971 and 1975 as part of SUR. . . ."[26]

17.4 Subsequent Work in Speech Recognition

Speech recognition research was also being carried out in other laboratories besides those that were directly involved with DARPA's SUR program. For example,

Frederick Jelinek of the Speech Processing Group in IBM's Computer Sciences Department at the Thomas J. Watson Research Center in Yorktown Heights, New York, is credited with being an early proponent of the use of statistical methods (including hidden Markov models) in speech recognition.[27] The HMM approach was ultimately adopted by all the leading speech recognition companies.

In 1984, DARPA began funding speech recognition work again as part of its "Strategic Computing" program (a program that will be described in a later chapter). Participants included CMU, SRI, BBN, MIT, IBM, and Dragon Systems. Among the systems developed at CMU over the next several years, for example, were SPHINX by Kai-Fu Lee and others and JANUS, a multilingual speech recognition and translation system, by Alex Waibel and others. (These and other systems are available as open-source software from the "Speech at CMU" Web page, http://www.speech.cs.cmu.edu/. The page also has links to many other speech recognition laboratories.)

Based on their work on DRAGON at CMU, James and Janet Baker founded Dragon Systems in 1982. In 1997, Dragon introduced "Dragon NaturallySpeaking," a speech recognition program for personal computers. It had a vocabulary of 23,000 words.[28] IBM followed with ViaVoice, and other companies, including Microsoft, also have speech recognition software.

The transcription of spoken sentences to their textual equivalents is now largely a solved problem. For example, high-quality speech recognition is commonly employed today in many automated telephone response systems. However, *understanding* natural language speech (or text) to permit general dialogs with computer systems, for example, remains a long-term research problem. I'll continue my discussion of work on that problem in a later chapter.

Notes

1. K. H. Davis, R. Biddulph, and S. Balashek, "Automatic Recognition of Spoken Digits," *Journal of the Acoustical Society of America*, Vol. 24, No. 6, pp. 627–642, 1952. **[211]**
2. For a history of early work see B. H. Juang and Lawrence R. Rabiner, "Automatic Speech Recognition – A Brief History of the Technology Development," available online at http://www.ece.ucsb.edu/Faculty/Rabiner/ece259/Reprints/354_LALI-ASRHistory-final-10-8.pdf; or Sadaoki Furui, "50 Years of Progress in Speech and Speaker Recognition," available online at http://www.furui.cs.titech.ac.jp/publication/2005/SPCOM05.pdf. **[211]**
3. C. Cordell Green, "AI During IPTO's Middle Years," in Thomas C. Bartee (ed.), *Expert Systems and Artificial Intelligence: Applications and Management*, p. 240, Indianapolis: Howard W. Sams & Co., 1988. **[211]**
4. Other members of the group were Jeffrey Barnett of the Systems Development Corporation, James Forgie of Lincoln Laboratory, C. Cordell Green, then a lieutenant in the U.S. Army stationed at DARPA, Dennis Klatt of MIT, J. C. R. Licklider, then at MIT, John Munson of SRI, Raj Reddy of CMU, and William Woods of BBN. **[211]**
5. The report was published as a special issue of the journal *Artificial Intelligence*: Allen Newell *et al.*, *Speech Understanding Systems: Final Report of a Study Group*, New York: American Elsevier Publishing Co., Inc., 1973. A draft of the report is available online in the Newell collection at http://diva.library.cmu.edu/webapp/newell/item.jsp?q=box00105/fld08162/bdl0001/doc0001/. **[212]**

6. J. R. Pierce, "Whither Speech Recognition?," *Journal of the Acoustical Society of America*, Vol. 46, No. 4, pp. 1049–1051, Part 2, 1969. Also see a rebuttal by Arthur Samuel and Pierce's response to Samuel and to other rebuttals in *Journal of the Acoustical Society of America*, Vol. 47, No. 6, Part 2, pp. 1616–1617, 1970. **[212]**
7. For a description of the SRI work, see Donald E. Walker (ed.), *Understanding Spoken Language*, New York: Elsevier North-Holland, Inc., 1978. **[212]**
8. For more details, see William A. Woods, "Motivation and Overview of BBN SPEECHLIS: An Experimental Prototype for Speech Understanding Research," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-23, No. 1, pp. 2–9, February 1975. **[212]**
9. See J. Wolf and William A. Woods, "The HWIM Speech Understanding System," *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '77*, Vol. 2, pp. 784–787, May 1977; also (for full details) William A. Woods *et al.*, *Speech Understanding Systems – Final Report*, BBN Report No. 3438, Vols. I–V, Bolt, Beranek, and Newman, Inc., Cambridge, MA, 1976. **[212]**
10. D. Raj Reddy, Lee D. Erman, and Richard B. Neely, "A Model and a System for Machine Recognition of Speech," *IEEE Transactions on Audio and Electroacoustics,* Vol. AU-21, No. 3, pp. 229–238, June 1973; and D. Raj Reddy, Lee D. Erman, R. D. Fennell, and Richard. B. Neely, "The HEARSAY Speech Understanding System: An Example of the Recognition Processes," in *Proceedings of the 3rd International Joint Conference on Artificial Intelligence*, pp. 185–183, Stanford, CA, August 1973. **[213]**
11. For background on the speech processing work at CMU during this period, see Lee D. Erman, "Overview of the HEARSAY Speech Understanding Research," *SIGART Newsletter*, No. 56, pp. 9–16, February 1976. **[213]**
12. James K. Baker, "Stochastic Modeling as a Means of Automatic Speech Recognition," doctoral dissertation, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, 1975, and James K. Baker, "The DRAGON System – An Overview," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-23, No. 1, February 1975. **[213]**
13. Allen Newell, "Harpy, Production Systems and Human Cognition, in Ronald A. Cole (ed.), *Perception and Production of Fluent Speech*, Hillsdale, NJ: Lawrence Erlbaum Associates, 1980. Available online as Carnegie Mellon University Technical Report CMU-CS-78-140 at http://diva.library.cmu.edu/webapp/newell/item.jsp?q=box00089/fld06145/bdl0001/doc0001/. **[213]**
14. For a translation see A. A. Markov, "An Example of Statistical Investigation of the Text *Eugene Onegin* Concerning the Connection of Samples in Chains," *Science in Context*, Vol. 19, No. 4, pp. 591–600, 2006. **[215]**
15. See L. E. Baum and J. A. Eagon, "An Inequality with Applications to Statistical Estimation for Probabilistic Functions of a Markov Process and to a Model for Ecology," *Bulletin of the American Medical Society*, Vol. 73, pp. 360–363, 1967. Baker credits Baum with introducing him to the theory of a probabilistic function of a Markov process. **[215]**
16. James K. Baker, "The DRAGON System – An Overview," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-23, No. 1, p. 24, February 1975. **[216]**
17. *Ibid*, p. 29. **[216]**
18. Bruce T. Lowerre, "The HARPY Speech Recognition System," doctoral dissertation, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, April 1976. **[216]**
19. I am basing my description of HARPY on Bruce Lowerre and Raj Reddy, "The HARPY Speech Understanding System," *Trends in Speech Recognition*, Prentice Hall. Reprinted

in A. Waibel and K. Lee (eds.), *Readings in Speech Recognition*, pp. 576–586, San Mateo, CA: Morgan Kaufmann Publishers, Inc., 1990. **[216]**

20. Lee D. Erman, Frederick Hayes-Roth, Victor R. Lesser, and D. Raj Reddy, "The HEARSAY-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty," *Computing Surveys*, Vol. 12, No. 2, June 1980. **[218]**
21. For a detailed summary of how HEARSAY processed an example sentence, see *ibid*. **[218]**
22. Telephone conversation, August 14, 2008. **[219]**
23. Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*, second edition, p. 580, Upper Saddle River, NJ: Prentice Hall, 2003. **[220]**
24. Dennis H. Klatt, "Review of the ARPA Speech Understanding Project," *Journal of the Acoustical Society of America*, Vol. 62, No. 2, pp. 1345–1366, December 1977. **[220]**
25. Lee D. Erman, Frederick Hayes-Roth, Victor R. Lesser, and D. Raj Reddy, *op. cit*. **[220]**
26. *Funding a Revolution: Government Support for Computing Research*, Chapter 9, Committee on Innovations in Computing and Communications: Lessons from History, Computer Science and Telecommunications Board, Commission on Physical Sciences, Mathematics, and Applications, National Research Council, Washington, DC: National Academy Press, 1999. Available online at http://books.nap.edu/openbook.php?record_id=6323&page=15. **[220]**
27. See, for example, Frederick Jelinek, "Continuous Speech Recognition by Statistical Methods," *Proceedings of the IEEE*, Vol. 64, No. 4, pp. 532–556, April 1976. **[221]**
28. Dragon NaturallySpeaking is now available through Nuance. **[221]**