

Recent Advances in Speech Processing

J. Mariani

LIMSI/CNRS

BP 30

91406 Orsay Cedex (France)

On invitation from the ICASSP'89 Technical Committee, this paper aims at giving to non-specialists in Signal Processing an overview of recent advances in the domain of Speech Recognition. The paper mainly focuses on Speech Recognition, but also mentions some progress in other areas of Speech Processing (speaker recognition, speech synthesis, speech analysis and coding) using similar methodologies.

It first gives a view of what the problems related to automatic speech processing are, and then describes the initial approaches that have been followed in order to address those problems.

It then introduces the methodological novelties that allowed for progress along three axes: from isolated-word recognition to continuous speech, from speaker-dependent recognition to speaker-independent, and from small vocabularies to large vocabularies. Special emphasis centers on the improvements made possible by Markov Models, and, more recently, by Connectionist Models, resulting in progress simultaneously obtained along the above different axes, in improved performance for difficult vocabularies, or in more robust systems. Some specialised hardware is also described, as well as the efforts aimed at assessing Speech Recognition systems.

Most of the progress will be referenced with papers that have been presented at the IEEE ICASSP Conference, which is the major annual conference in the field. We will take this opportunity to produce some statistical data on the "Speech Processing" part of the conference, from its beginning in 1976 to its present fourteenth issue.

Introduction

The aim of this paper is to give non-specialists in Signal Processing an overview of recent advances in the domain of Speech Recognition. It can also be considered an introduction of the papers that will be presented in that field during this conference; especially those presenting latest results on large vocabulary, continuous speech recognition systems.

As a general comment, one may feel that in recent years, the choice between methods based on extended knowledge introduced by human experts with corresponding heuristic strategies, and self-organizing methods, based on speech data bases and learning methodologies, with little human input, has turned toward the latter. This is partly due to the results of comparative assessment trials.

Problems related to speech processing

Several problems make speech processing difficult, and unsolved at the present time:

A. There is no separator, no silence between words, comparable to spaces in written language.

B. Each elementary sound (also called phoneme) is modified by its (close) context: the phoneme which is before it, and the one which comes after it. This is related to coarticulation: the fact that when a phoneme is pronounced, the pronunciation of the next phoneme is prepared by a movement of the vocal apparatus. This cause is also referred to as the "teleological" nature of speech [90]. Other (second order) modifications of the signal corresponding to a phoneme will be caused by larger context such as its place in the whole sentence.

C. A good deal of variability is present in speech: intra-speaker variability, due to the speaking mode (singing, shouting, whispering, stuttering, with a cold, when hoarse, creakiness, voice under stress, etc.), inter-speaker variability (different timbre, male, female, child, etc.), due to the signal input device (type of microphone), or to the environment (noise, co-channel interference, etc.).

D. Because of B and C, it will be necessary to observe, or to process, a large amount of data in order to find, or to obtain, what makes an elementary sound, despite the different contexts, the different speaking modes, the different speakers and the different environments. A difficult problem for the system is to be able to decide that an "a" pronounced by an aged male adult is more similar to an "a" pronounced in a different word by a child, in a different environment, than to an "o" pronounced in the same sentence by the same male adult.

E. The same signal carries different types of information (the sounds themselves, the syntactic structure, the meaning, the sex and the identity of the person speaking, his mood, etc.). A system will have to focus on the kinds of information which are of interest for its task.

F. There are no precise rules at the present time for formalizing the information at different levels of decoding (including syntax, semantics, pragmatics), thus making it difficult to use fluent speech. Moreover, those different levels seem to be heavily linked to each other (syntax and semantics, for example). Fortunately, the problem mentioned in E. also means that the information in the signal will be redundant, and that the different types of information will cooperate with each other to make the signal understandable, despite the ambiguity and noise that may be found at each level.

First results on a simplified problem

After some overly optimistic hopes about the difficulty of the Speech Recognition task, similar to early views concerning automatic translation, a beneficial reaction in the late '60s was to consider the importance of the problem in its generality, and to try to solve a simpler problem by introducing simplifying hypotheses. Instead of trying to recognize anyone pronouncing anything, in any manner, and in fluent speech, a first sub-problem was isolated: recognizing only one person, using a small vocabulary (on the order of 20 to 50 words), and asking for short pauses between words.

The basic approach used two passes: a training pass and a recognition pass. During the training pass, the user pronounces each word of the vocabulary once. The corresponding signal is processed at the so-called "acoustic" or "parametric" level, and the resulting information, also called "acoustic image", "speech spectrogram", "template" or "reference pattern", which usually represents the signal in 3 dimensions (time, frequency, amplitude), is stored in memory, with its corresponding label. During the recognition pass, similar processing is conducted at the "acoustic" level: the corresponding pattern is then compared with all the reference patterns in memory, using an appropriate distance measure. The reference with the smallest distance is said to have been recognized, and its label can be furnished as a result. If that distance is too high, compared with a pre-defined threshold value, the decision can be non-recognition of the uttered word, thus allowing the system to "reject" a word which is not in its vocabulary.

This approach led to the first commercial systems, appearing on the market in the early '70s, such as the VIP 100 from Threshold Technology Inc. which won a US National Award in 1972. Due to those simplifications, this approach doesn't have to deal with the problems of segmenting continuous speech into words (problem A, above), of the context effect (as it deals with a complete pattern corresponding to a word always spoken in the same context - silence (B), of inter-speaker variability (C). Also, indirectly, it bypasses the problem of allowing for "natural language" speech (F), as the small size of the vocabulary, and the pronunciation in isolation prevents fluent speech ! However, the intra-speaker variability, the sound recording and the environment problems are still present.

- Pattern Matching using Dynamic Programming

In the recognition pass, the distance between the pattern to be recognized (test pattern) and each of the reference patterns in the vocabulary has to be computed. Each pattern is represented by a sequence of vectors regularly spaced along the time axis. Those vectors can represent the output of a filter bank (analog or simulated by different means, including the (Fast) Fourier Transform [33]), coefficients obtained by an autoregressive process such as Linear Prediction Coding (LPC) [157], or coefficients derived from these methods, like the Cepstral coefficients [19], or even obtained by using an auditory model [50,60,95]. Typical values are a vector of dimensions 8 to 20 (also called a spectrum or a frame), each 10 ms (for general information on speech signal processing techniques, see [117, 101, 129]).

The problem is that when a speaker pronounces the same word twice, the corresponding spectrograms will never be exactly the same. There are non-linear differences in time (rhythm), in frequency (timbre), and in amplitude (intensity). Thus, it is necessary to align the two spectrograms, so that, when the test pattern is compared to the correct reference pattern, the vectors representing the same sound in the two words correspond to each other. The distance measure between the two spectrograms will be calculated according to this alignment. Optimal alignment can be obtained by using the Dynamic Programming method

(Figure 1). If we consider the *distance matrix* D obtained by computing the distances $d(i,j)$ (for example, the Euclidian distance) between each vector of the test pattern and of the reference pattern, this method furnishes the optimal path from $(1,1)$ to (I,J) (where I and J are respectively the length of the test and of the reference pattern), and the corresponding distance measure between the two patterns. In the case of Speech Recognition, this method is also called Dynamic Time Warping, or DTW, since the main result is to "warp" the time axis. Dynamic Programming was first presented by R. Bellman [14], and first applied to speech by the Russian researchers T. Vintsjuk and G. Slutsker in the late 60s [165,160].

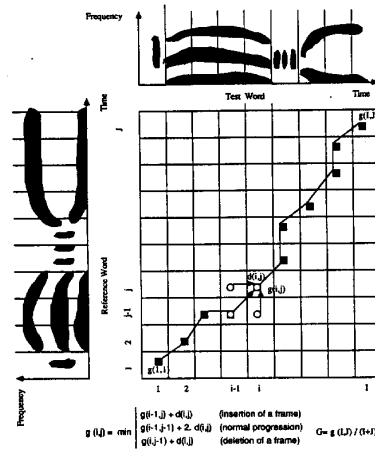


Figure 1: Example of Dynamic Time Warping between two speech patterns (the word "Paris" represented by a schematic spectrogram). G is the distance measure between the two utterances of the word. $d(i,j)$ is the distance between two frames of the reference and test patterns at instants i and j . An example of a local DP equation is given. The optimal path is represented by squares. The cumulated distances involved in the computation of the cumulated distance $g(i,j)$ are represented by circles.

- Speech and AI: the ARPA-SUR project

A different approach, mainly based on "Artificial Intelligence" techniques was initiated in 1971, in the framework of the ARPA-SUR project [114]. The idea behind it was that the use of "upper level" knowledge (lexicon, syntax, semantics, pragmatics) could produce an acceptable recognition rate, even if the initial phoneme recognition rate was poor [70]. The task was speaker-dependent, continuous speech recognition (improperly called "understanding" because upper levels were used), with a 1,000-word vocabulary. Several systems were delivered at the end of the project, in 1976. From CMU, the DRAGON system [4] was designed, using a Markov approach. The HEARSAY I and HEARSAY II systems, based on the use of a *Blackboard Model*, where each Knowledge Source can read and write information during the decoding, with a heuristic strategy, and the HARP system, which merged parts of the DRAGON and HEARSAY systems. From BBN, the SPEECHLIS and HWIM systems were developed. SDC also produced a system [76]. Although the initial requirements (which were in fact rather vague) were attained by at least one system (HARP), the systems needed so much computer power, at a time when this was expensive, and were so cumbersome to use and so non-robust, that there was no follow-up. In fact, one of the major conclusions was that there was a need for better acoustic-phonetic decoding [77].

Improvements along each of the 3 axes

From the basic IWR method, progress has been made which independently addresses the three different problems: size of the population using the system, speaking rate, size of the vocabulary.

- Speaker-Dependent (SD) to Speaker-Independent (SI):

In order to allow any speaker to use a recognition system, a multi-reference approach has been experimented. Each word of the vocabulary is pronounced by a large population, male and female, with different timbres and different dialectal origins. The distance between the different pronunciations of the same word is computed using DTW. A clustering

algorithm (such as K-means) is used to determine clusters corresponding to a certain type of pronunciation for that word. The centroid of each cluster is chosen to be the reference pattern for this type of pronunciation (Figure 2). Each word is then represented by several reference patterns. Recognition is carried out in the same way as it is in the speaker-dependent mode, with eventually, a more sophisticated decision process (like KNN (K-nearest neighbors)) [130].

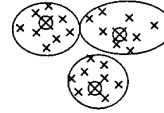


Figure 2: An illustration of Clustering. Each cross is a word. The distance between crosses represents the DTW distance between the words. Each cluster is represented by its centroid (circles).

- Isolated Word Recognition (IWR) to Connected Word Recognition (CWR) and Word Spotting:

In order to allow the user to speak continuously (without pauses between words) several problems have to be solved: how many words are in the sentence and where their boundaries are; if the training is to be done in isolation, the pattern corresponding to the beginning and to the end of the words will be modified, due to the context of the end of the previous word, and to the context of the beginning of the following word. The first two problems have been solved by using methods generalising the IWR DTW, such as "Two-Level DP matching" proposed by H. Sakoe [146], "Level building" proposed by C. Myers and L. Rabiner [108], "One-Pass DP" proposed by J. Bridle [24], also called "One-stage DP" by H. Ney [115]. It appears in fact that the initial DP approach as first described by T. Vintsjuk in 1968 [165] already had its extension to Connected Word Recognition [87]. To address the second problem, the "embedded training" method has been proposed [132], where each word is first pronounced in isolation. It is then pronounced in a sentence known to the system. The "isolated" reference templates will be used to optimally segment the sentence into its constituents, and extract the "contextual" image of the words which will be added as new reference templates.

The "Word Spotting" technique is very similar, and uses the same DTW techniques. But it should allow rejection of words in the sentence which are not in the vocabulary. Recent results on Speaker Independent Word Spotting give 61% correct detection in clean speech, and 44% when Gaussian noise is added for a Signal-to-Noise ratio of 10 dB, with a 20-word vocabulary (1 to 3 syllables long), the false alarm rate being set to 10 false alarms per hour [20].

A syntax can be used during the recognition process. The syntax represents the word sequences that are allowed for the language corresponding to the task using speech recognition. The role of the syntax is to determine which words can follow a given word (sub-vocabulary), thus accelerating the recognition process by reducing the size of the vocabulary to be recognized at each step, and improving the performance by possibly eliminating words that are acoustically similar, but do not belong to the same sub-vocabulary, and thus do not compete. Introduction of the grammar into the search procedure may be more or less difficult, depending on the CWR-DTW algorithm used. Most of the syntaxes used in DTW-type systems correspond to simple command languages (regular, or context-free grammars introduced manually by the system user).

It appears that the better the training, the better the recognition. In order to improve training, several techniques have been tried, like "embedded training" already mentioned, multireference training, where several references of a word are kept, using the same clustering techniques for representing intra-speaker variations as those used for inter-speaker variations in multi-reference speaker-independent recognition, and robust training [131].

- Small Vocabularies to Large Vocabularies:

Increasing the size of the vocabulary raises several problems: since each word is represented by its spectrogram, necessary memory size gets very large. Since matching between the test pattern and the reference patterns is done sequentially, computation time also greatly increases. If a speaker has to train the system by pronouncing all of the words, the task rapidly becomes tedious. A large vocabulary has the consequence of many acoustically similar words, thus increasing the error rate. This also implies that the speaker will want to use a natural way of speaking, without strong syntax constraints. To address these problems, there have been several improvements:

- **Vector Quantization** [53, 49, 97]: In the domain of Speech Processing, this method was first used for low bit rate speech coding [91]. Considering a reasonable amount of speech pronounced by one speaker, the method consists of computing the distances (like the Euclidian distance) between each vector of the corresponding spectrogram, and using a clustering

algorithm to determine clusters corresponding to a type of vector, which is represented by the centroid (called "prototype" or "codeword"). The set of prototypes is called a "codebook". In the training phase, after acoustic processing of the word, each spectrum is recognized to be one of the prototypes of the codebook. Thus, instead of being represented by a sequence of vectors, the word will be represented by a sequence of numbers (also called labels) corresponding to the prototypes. A *distortion measure* can be obtained by computing the average distance between the incoming vector, and the closest prototype. On a practical level, if the size of the codebook is 256, or less (that is addressable on one byte), and each vector component is coded on one byte, the reduction of information is equal to the dimension of the vectors. Also, computing time is saved during recognition for large vocabularies since, for each input vector of the test pattern, only 256 distances have to be computed, instead of computing the distances with all the vectors of all the reference templates. Moreover, the distances between prototypes can be computed after training, and kept in a distance matrix. Those codebooks can concern not only spectral information, but also energy, or variation of spectral information or of energy in time. All this can be represented by a single codebook with supervectors, constructed by including the different kinds of information [155]. It can also be represented by a different codebook for each type of information. This approach was applied with success to speaker identification [161], and to speech recognition [55]. The codebooks can also be constructed from the speech of several speakers (speaker-independent codebooks) [83].

It should be noted that similar methods have been used previously (Centisecond Model) [85]. The problem at that time was that the vectors had been labelled with a linguistic label (a phoneme), thus making a decision too early. The Vector Quantization scheme inspired much thought. One remark was that each word could have a specific set of prototypes without taking into account the chronological sequence of those prototypes. Even if some words contain the same phonemes in a different order, the transition between those phonemes is different, and the prototypes corresponding to those transitions may be different, the latter making the distinction between words. During training, a codebook is built for each reference word. The recognition process then consists of simply recognizing the incoming vectors, and choosing the reference which gives the smallest average distortion with the test [156]. A refined approach consisted of segmenting the words into multiple sections, in order to partly reflect time sequencing for words having several phonemes in common [26]. This refinement increases the computation time, without giving better results than the DTW-based approach does.

- **Sub-word units:** Another way to reduce the memory requirement is to use decision units that are shorter than the words (also called subword units). The words will then be recognized as the concatenation of such units, using a Connected "Word" DTW algorithm. These units must be chosen so that they are not too affected by the coarticulation problem at their boundaries. But they also should not be too numerous. Examples of such units are phonemes [163], diphones [98, 148, 32, 2, 149], syllables [59, 168, 48], demi-syllables [144, 139], disyllables [159].

Graphemic Word :	émigrante
Phonemic Word :	SemigRâtS
phonemes :	\$ e m i g R â t \$
diphones :	\$ e m m i g R R â t \$
syllables :	\$ e m i g R â t \$
demi-syllables :	\$ e m m i g R â t \$
disyllables :	\$ e m i g R â t \$

Figure 3: Representation of a word by subword units
(The word is "émigrante" ("emigrant") in French, \$ stands for silence)

Other approaches tend to use units with no linguistic affiliation, for example segments, obtained by a segmentation algorithm. This approach lead to Segment (or Matrix) Quantization, very similar to Vector Quantization, except that the distance between segment prototypes may need time alignment, if the segments do not have a constant length [25, 142, 84]. In a similar way, several attempts to represent the references by networks of segments can be found for speaker-independent isolated digit [74], connected digit [27], or alphabet recognition [80].

- **Time compression:** Time compression can also reduce the amount of information [75, 46]. The idea is to compress (linearly, or non linearly) the steady states, which may have very different lengths depending on speaking rate, while keeping all the vectors during the transitions, thus moving from the time space to the variation space. An algorithm like the VLTS (Variable Length Trace Segmentation) [47] halves the amount of information used. It also obtains better results when the pronunciation rate is very different between training and recognition (some often-used DTW equations, for example, do not accept speaking rate variations of more than a 2-to-1 ratio, which is easily reached between isolated word pronunciation and continuous speech). However, if duration itself carries meaning, that information may be lost.

- **Two-pass recognition:** In order to accelerate recognition, it can be processed in two passes: first there can be a rough, but fast match aimed at eliminating words of the vocabulary that are very different from the test pattern, before applying an optimal match (DTW or Viterbi) on the remaining reduced subvocabulary. In this case, the goal is not to get just the correct word, but is to eliminate as many word-candidates as possible (without eliminating the right one, of course). Simple approaches like summing the distances on the diagonal of the *distance matrix* used for DTW [48] have been tried. Other approaches are based on Vector Quantization without time alignment, the system being based on Pattern Matching [99] or on Stochastic Modeling (called "Poisson Polling") [8]. Using a phonetic classifier, based on broad [45] or usual [16] phonetic classes, and matching the recognised phoneme lattice with the reference phonemic words in the lexicon by DTW is another reported method.

- **Speaker adaptation:** The adaptation of one speaker's references to a new speaker can be obtained through their respective codebooks, if a Vector Quantization scheme is used. The reference speaker produces several sentences, which are vector quantized with his codebook. The new speaker produces the same sentences, which are vector quantized with his own codebook. Time alignment of the two sets of sentences creates a mapping between the two codebooks. This basic method has several variants [155, 21, 43].

Most of the progress related to this technique has been obtained on one aspect of the problem. Some systems addressing two aspects can also be found, like the Conversant system from AT&T [152], which allows for speaker-independent connected digit recognition over telephone lines using a multireference CWR-DTW approach. Further advances have been obtained by using more elaborated techniques: Hidden Markov Models, and Connectionist Models.

The Hidden Markov Model approach

Whereas in the previous *pattern matching* approach, a reference was represented by the pattern itself which was stored in memory, the Markov Model approach carries a higher level of abstraction, representing the reference by a model [125, 135]. To be recognized, the input is thus compared to the reference models. The first uses of this approach for speech recognition can be found at CMU [4], IBM [62] and, apparently, IDA [124].

In a stochastic approach, if we consider an acoustic signal A , the recognition process can be described as computing the probability $P(W|A)$ that any W word string (or sentence) corresponds to the acoustic signal A , and as finding the word string having the maximum probability. Using Bayes' rule, $P(W|A)$ can be represented as:

$$P(W|A) = P(W) \cdot P(A|W) / P(A)$$

where $P(W)$ is the probability of the word string W , $P(A|W)$ is the probability of the acoustic signal A , given the word string W , and $P(A)$ is the probability of the acoustic signal (which does not depend on W). Thus it is necessary to take into account $P(A|W)$ (which is the *acoustic model*), and $P(W)$ (which is the *language model*). Both models can be represented as Markov models [6]. We will first consider *Acoustic Modeling*.

- **Basic discrete approach:** Here each acoustic entity to be recognized, each reference word for example, is represented by a finite state machine, also called a Markov machine, composed of states, and of arcs between states. A *transition probability* a_{ij} is attached to the arc going from state i to state j representing the probability that this arc could be taken. The sum of the transition probabilities attached to the arcs issued from a given state i is equal to 1. There is also an *output probability* $b_{ij}(k)$ that a symbol k from a finite alphabet can be emitted when the arc from state i to state j is taken. In some variants, this output probability is attached to the state, not to the arc. When *Vector Quantization* is used, this output probability distribution (also called *output probability density function* (*pdf*))) is the probability distribution of the prototypes. The sum of the probabilities in the distribution is also equal to 1 (Figure 4). In a first-order Hidden Markov Model, it is assumed that the probability that the Markov chain is in a particular state at time t depends only on the state where it was at time $t-1$, and that the output probability at time t depends only on the arc being taken at time t .

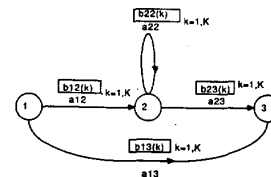


Figure 4: An example of a Hidden Markov Model
The output probability distributions $b_{ij}(k)$ are enclosed in rectangles. a_{ij} is the transition probability. This left-to-right model has 3 states and 4 arcs.

- **Continuous models:** We have just presented what are usually called "Discrete Hidden Markov Models". Another type of Markov Model is the "Continuous Markov Model". In this case, the discrete output pdf on one arc is replaced by a model of the continuous spectrum on that arc. A usual model is the multivariate Gaussian density [120], which describes the pdf by a mean vector and a covariance matrix (eventually diagonal). The use of a multivariate Gaussian mixture density seems to be more appropriate [135,66,137,122]. The Laplacian mixture density seems to allow for good quality results, with reduced computation time [116]. Several attempts to compare discrete and continuous HMMs have been reported. It seems that only complex continuous models allow for better results than discrete ones, reflecting the fact that with the usual Maximum Likelihood training, the complete model should be correct to allow for good recognition results [7]. But complex continuous models need a good deal of computation.

The number of states, the number of arcs, and the initial and final states for each arc are chosen by the system designer. The parameters of the model (transition probabilities, and output probabilities) have to be obtained through training. Three problems have to be addressed:

- the **Evaluation problem** (what is the probability that a sequence of labels has been produced by a given model?). This can be obtained by using the Forward algorithm, which gives the Maximum Likelihood Estimation that the sequence was produced by the model.
- the **Decoding problem** (which sequence of states has produced the sequence of labels?). This can be obtained by the Viterbi algorithm, which is very similar to DTW [166].
- the **Learning (or Training) problem** (how to get the parameters of the model, given a sequence of labels?). This can be obtained by the Forward-Backward (also called Baum-Welch) algorithm [12], when the training is based on Maximum Likelihood.

- Training:

- **Initialisation:** Initialisation of the parameters in the model has to be carried out before starting the training process. A hand-labelled training corpus can be used. If enough training data exists, uniform distribution will be sufficient for homogeneous units, like phone models, with discrete HMM [83]. For word models, or for continuous HMM, more sophisticated techniques have to be used [121].

- **Maximum Likelihood Estimation (MLE):** The Maximum Likelihood Estimation was the initial principle used for decoding and training the decoder [6]. The Maximum Likelihood Estimation is considered to guarantee optimality in training if the model is correct, and if the production of speech is really a Hidden Markov Model, which may not be the case [9]. We perceive that this measure will effectively guarantee optimality with regards to the training pass, but not necessarily to the recognition pass. To improve the discriminative power of the models, some alternatives have been tried:

- **Corrective training:** The model is first built on part of the training data with MLE. It is then used to recognize the training data. When there is a recognition error, or even if a wrong candidate gets too close to the right one, the initial model is modified in order to lower the probability of the labels responsible for the mistake or the "near-miss". The process is repeated with the modified parameters. It is stopped when no more modifications are observed. A list of acoustically confusable words can be used in order to reduce the duration of the process. This approach tends to minimize the whole recognition error rate related to the training data. If, the test data in operational conditions is similar to the training data, the error rate on the test data will also be minimized [9].

- **Maximum Mutual Information (MMI):** The Maximum Mutual Information approach is a similar, but more formalized method [7,104]. The goal is to determine the parameters of the model by maximizing the probability of generating the acoustic data given the right word sequence, as in MLE, but, at the same time, minimizing its probability of generating any wrong word sequence, especially the most frequent ones. Comparative results between the two methods showed that corrective training was better. This may be due to the fact that the low-probability wrong word sequences will have very little effect in MMI training while they may have some effect in corrective training [9]. Compared with ML training, the MMI training is especially more robust when the model is incorrect [7], and generally gives better results [104]. A different method, the Minimum Discriminant Information (MDI) has been proposed as a generalization of both ML and MMI [42].

- **Smoothing:** To get good results, a Markov model needs much training data. If a label never appeared at a given arc during training, it will be given zero probability in the distribution corresponding to that arc, and if it appears during recognition, this zero probability may be attributed to the whole word. A simple smoothing method is to give a very low probability to all the probabilities which are null (*floor smoothing* [86]). A more sophisticated one consists of assigning several labels instead of a single one for each frame during the training, with probabilities computed

from the distance measure, and thus defining *similar prototypes*. If the output probability of a prototype is null on an arc, it can be smoothed with the non-null probability of a *similar prototype* [150]. A third method is the *co-occurrence smoothing* [82], which smooths on all the arcs the probabilities of labels that sometimes appear on the same arcs.

- **Deleted Interpolation:** In order to smooth the estimates of the parameters obtained by 2 different methods, it is necessary to apply weights to the different estimates. Those weights will reflect the quality of each estimate, or the quantity of information used to calculate each of them. A method to automatically determine those weights is the *deleted interpolation estimation*, which splits the estimates on two arcs, and defines the weights as the transition probabilities of the arcs, as computed by the Forward-Backward algorithm [63].

- **Time modeling:** The modelisation of time in a Markov model is contained in the probabilities of the arcs. It appears that the probability to stay at a given state will decrease as a power of the probability to follow the arc looping on that state, which seems to be a poor time model in the case of the speech signal. Several attempts to improve that issue can be found.

In the *Semi-Hidden Markov Model* [44,145], a set of probability density functions $P_i(d)$ at each state i indicates the probability of staying in that state for a given duration, d . This set of probabilities is trained together with the transition and output probabilities by using a modified Forward-Backward algorithm. A simpler approach is to independently train the duration probability and the HMM parameters [134].

To allow for a more easily trainable model, continuous probability density functions can be used for duration modeling, like the Poisson distribution [145] or the gamma distribution, used by S. Levinson in his *Continuously Variable Duration Hidden Markov Model* (CVDHMM) [88].

Another way of indirectly taking time into account is to include the dynamics of the spectrum as a new parameter. It can be represented by the differenced Cepstrum coefficients corresponding to adjacent frames, and can also include the differenced power. After Vector Quantization, the *multiple codebooks* for those new parameters are built. They are introduced in the HMM with independent output pdfs on the arcs [83].

- Decision Units:

- **Word:** The natural idea is to modelise a word with a HMM. An example of a Markov word machine, from R. Bakis [62], is given in Figure 5. The number of states in the word model is equal to the average duration of the word (50 states for a 500 ms word, with a frame each 10 ms). It should be noted that the model includes the frame deletion and insertion phenomena previously detected during DTW. More recently, models with less states have been successfully tried [133]. The problem is that to get a good model of the word, there should be a large number of pronunciations of that word. Also, the recognition process considers a word as a whole, and does not focus on the information discriminating two acoustically similar words.

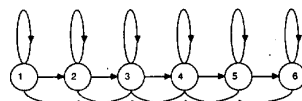


Figure 5: Example of a Bakis Model for a word.
The average length of the word is 50 ms.

In the same way as with the DTW approach, using units shorter than words has its advantages. The segmentation-by-recognition process operated by the Viterbi algorithm permits to avoid the problem of a-priori segmentation, and thus authorizes the use of subword units as small as phonemes (also called phones to allow for a less theoretical definition), diphones, syllables, demi-syllables, etc.).

- **diphone units:** The use of HMM diphone models has been compared with simple phone models, or composite models of phones and transitions. Transition models were built only for transitions corresponding to certain class pairs for specific transitions (plosive-vowel, affricate-vowel, etc.). The composite model obtained better results, with a smaller number of units than the diphone models [34].

- **context-independent phones:** Context-Independent Phone models are interesting, because they are lesser in number. An example of such a phone model is given in Figure 6. They were used in the early IBM Speech Group work on isolated and continuous speech recognition [6].

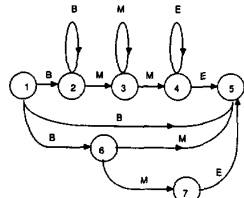


Figure 6: Example of a Phone Model in the SPHINX system
The Output probability distributions B (Beginning), M (Middle) and E (End) are tied (forced to be the same) on different arcs. The minimum length is one frame. There is no maximum length.

If the decision units are subword units, like phones, each word is represented by a string of those subword units (or a network, if the phonological variations in the pronunciation of the word are taken into account (Figure 7)). If no lexical information is used, the subword units are integrated in a *Looped Phonetic Model* (LPM), where different probabilities can be eventually affected at the succession of phonemes (Figure 8) [103].

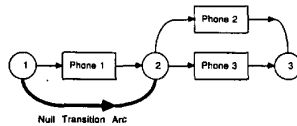


Figure 7: Example of a word model built from phone models
The word can be one or two phonemes long. The first phoneme can be deleted. There are two possibilities for the second phoneme. The probability of the different phonological variations can be put on the arcs. The null transition arc has no output symbol emission.

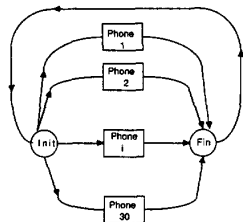


Figure 8: A Looped Phonemic Model
Each rectangle is a phone machine. The arcs from the initial state to each phone, from each phone to the final state, and from the final state back to the initial state are null transition arcs. The probability of phoneme successions can be used as a "language model".

Unfortunately, the simple phone models are much affected by the context, and the parameters of the phone model reflect many different acoustic signals for the same phoneme.

- *context-dependent phones*: To address this problem, context-dependent phones have been tried [5,150]. Different phone models are constructed for each context of the phone. If there are 30 phones used, there will be about 1,000 models for each phone, that is 30,000 models if we consider both the right and left contexts (called *triphone models*). Here also, it may be difficult to get enough training data to train all these models. Knowledge in phonetics can be used in order to reduce the number of triphone models to be trained, as some contexts will have similar effects on the middle phone [37]. Alternatively, in the *generalized triphone* approach [150], a small number of representative contexts of the HMMs are chosen. Two different context-dependent phone models are kept separately or merged), is used to determine the triphone models that have to be kept.

word-dependent phones: In the same way, a phone model can be trained in the context of each word. If the vocabulary is small, and if the number of templates for each word is large, then training is possible. The approach has been used by CNET researchers in their speaker-independent isolated word recognition system through telephone lines [39], and by BBN for a 1,000 word vocabulary [29]. At CMU, K.F. Lee has used function word phone models [83]. Function words are grammatical words, usually short and badly pronounced, and thus difficult to recognize. They are very frequent in fluent speech, and greatly affect overall recognition performance. But, as they are frequent, training can be conducted. All this justifies the need and the possibility of having special models for the phones of those words.

It is possible to mix the different models (context-independent, context-dependent, word-dependent) of the same phone by using the *deleted interpolation* method [83,37].

- **Fenones:** Other models are of acoustic nature. L. Bahl et al. are using the concept of *fenones* [10]. The idea is to represent the pronunciation of a word by the string of prototype labels obtained by vector quantization, and to create a simple Markov machine, called a fenone machine (Figure 9), for each of the labels. The parameters of these models can be obtained by training on several utterances of each word. This approach is close to DTW on word patterns. The DTW deletion and insertion phenomena for each label are included in the model. For example, the labels corresponding to a stable instant have a high transition probability to the labels corresponding to a speaker's utterance. The fenone models can be trained to a new speaker, not the word patterns. The use of speaker-independent fenone models is a way to represent the time model of each word.

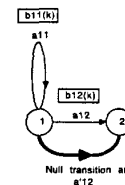


Figure 9: A Fenone Machine.
The bold arc is a null transition. The length of the machine can be 0, 1 or several frames.

- Segments: These segments are similar to the ones used in *Segment Quantization* with DTW pattern matching methods [142]. They are obtained by applying a Maximum Likelihood segmentation algorithm. A segment quantization process is then conducted. Each of the prototype segments of the resulting segment codebook is represented by an HMM, trained on the initial data. Each word of the lexicon is represented by a network of those acoustic units. The results on IWR are similar to those obtained with word models [84].

Several aspects must be taken into account in order to choose a unit:

- c. Another important aspect is the possibility of improving a detailed model built with insufficient data using the parameters obtained from a more general model, like a context-dependent model of a phone being improved by smoothing it with a context-independent model of the same phone.

Those three aspects are called **trainability**, **sensitivity** and **sharability** [83].

Speaker Adaptation: Adaptation to a new speaker can be obtained by using adaptation techniques based on codebook mapping. The approach at BBN first performed adaptation by quantizing an unknown input sentence with the reference speaker codebook [15], and applying a modified forward-backward algorithm to compute the transformation matrix presenting the conditional probability of a quantized spectrum of the new speaker, given a quantized spectrum of the reference speaker. The method was improved by building the new speaker codebook, DSW aligning a known reference sentence with the new speaker's sentence, and building new speaker codebooks and counting the co-occurrences of new and reference speaker codewords [43]. In the context of speaker-dependent recognition, speaker

adaptation from a reference speaker to a new speaker, even on a small amount of data (15 seconds of speech), allows for results close to those obtained with speaker-dependent training with 15 minutes of speech. In the context of speaker-independent recognition, the experiments conducted at CMU combining speaker-independent and speaker-dependent parameters obtained from 30 sentences with a *deleted interpolation* method showed 5 to 10% improvement by using speaker adaptation (when no grammar was used) (Table 1).

- Language model:

The *language model* can also be represented as a Markov process. In a *Bigram* model, the probability of a word, given a previous word, is computed as the frequency of two-word sequences [6]. In a *Trigram* model, the probability of a word, given the two preceding words is computed. A *Unigram* model is simply the probability of a word. A simpler model is the word pair model, where the same probability is given to all the words that can follow a given word.

Those different models must be trained on a large corpus. If the corpus is not large enough, and if the number of words in the vocabulary is high, many actually existing word successions will be absent, and the model, especially in the case of the trigrams, will have many null probabilities (if a 5,000-word vocabulary is used, the size of the trigram matrix is 5,000³). This can be improved by using smoothing techniques, similar to *floor smoothing*, like the Turing-Good estimate [110], which says that an estimate of the probability of a unseen words in a training corpus is the ratio of the number of words seen once divided by the total number of words in the corpus. Another possibility is to use *deleted interpolation* to combine unigram, bigram and trigram probabilities in the complete language model [36].

The percentage of real linguistic facts contained in the language model is called the *coverage*. Interestingly, experiments conducted on large vocabulary recognition showed that the error rate was 17.3% with a 10,000-word dictionary, including errors on 43 words of the 722-word test text that were not in the lexicon (thus having a 94% coverage). Using a 200,000-word dictionary allowed for a lower error rate of 12.7%, as, even if the task is more difficult, the coverage was then 100% [103].

In a *Biclass* or *Triclass* model, the probability of word succession is replaced by the probability of grammatical class succession [3]. The probability of a given word in a class can also be used to refine the model (*TriPOS* (as Part Of Speech) model [36]). An in-between approach is the *smoothed trigram* model where the probabilities of long words (three or more syllables long) are tied, as they are easy to recognize and do not usually have homophones (at least for English) [38].

The advantages of language models based on words is that they will contain both syntactic and semantic information. They will also be simpler to train, as the text data base does not need any initial grammatical labelling. However, the amount of data necessary to train the model, especially in the case of a trigram model will be large. In the case of using grammatical categories, the text will have to be labelled, but can be shorter. Moreover, if a new word is introduced in the dictionary it can inherit the probabilities computed for the words having the same grammatical category.

For the Dictation task, the reference is Written Language dictated by voice. In this respect, one can use a text data base to train the language model. IBM has used a 250 million word text to train their model in the Tangora system (as reported in [83]). Within the European ESPRIT programme, multilingual language models have been built [23]. For Spoken Language Understanding, there should be a need for using speech data to train the language model, thus mixing acoustical and language modeling. When written transcriptions of dialogs are available, the model can be trained on those transcriptions. In the case where only a small corpus is available (1,200 sentences of the DARPA Resource Management Database), BBN recently proposed using a model including both probabilities of phrase succession, and probabilities of word succession within a phrase [138].

The use of a language model is an absolute necessity. Experiments conducted in French on phoneme-to-grapheme conversion of error-free phoneme strings showed that a simple 9-phoneme sentence generated more than 32,000 possible segmentations into words, and graphic translations of those words [3].

HMMs have been used in many different systems:

- **Small Vocabulary Speaker-Dependent Isolated Word:** In this simple task, HMMs have been used to make the system more robust to variations in pronunciation for one speaker. At Lincoln Lab, continuous HMM word models are trained on different types of speaking modes (normal, fast, loud, soft, shouting, and with Lombard effect). This is called "Multistyle training". On the 105 word vocabulary TI speech database, the results were 0.7% errors [120]. On the difficult keyboard task, with a 62-word vocabulary including the alphabet, the digits, and punctuation marks, IBM achieved a 0.8% error rate, using *fenone* models [10].

- **Small Vocabulary Speaker-Independent Isolated Word:** CNET in France has used this approach for telephone recognition of a small amount of words, spoken in isolation. The system is robust. It has been trained with many pronunciations of each word of the vocabulary, from many speakers

obtained through telephone lines. It uses continuous word models. The results were 85% for isolated digits, and 89% for signs of the Zodiac, over the public telephone analog network [39].

- **Small Vocabulary Speaker-Dependent Continuous Speech:** Work at Lincoln Lab on robust recognition for isolated words has been enlarged to continuous speech. Preliminary tests have been conducted on a 207 word vocabulary, with a perplexity of 14 words. 10 different speaking conditions are present. The word error rate was 16.7% [121].

- **Small Vocabulary Speaker-Independent Continuous speech:** At AT&T Bell Labs, very good results on speaker-dependent, multi-speaker and speaker-independent connected digit recognition, with word-model continuous HMMs, have been reported (0.78%, 2.85%, and 2.94% string error rate, for strings of unknown length) [136]. At CNET, a system has been designed for telephone dialling using 2-digit numbers. It results in a dial-free telephone booth now being tested at different public sites [65].

- **Large Vocabulary Speaker-Dependent Isolated Word:** The IBM-Yorktown Speech Group announced a 5,000 word, speaker-dependent, isolated word recognition system on a PC in 1986 [154]. In 1987, they presented a new version with a vocabulary of 20,000 words. They use both phone and *fenone* models [64].

At the IBM Scientific Center in Paris, experiments have been conducted on a very large dictionary (200,000 words). The pronunciation mode is syllable by syllable. Although the pronunciation mode is difficult to accept, the interest is to directly process a language model corresponding to continuous speech, and including the problem of liaisons and elisions in French. They use phone models [103].

At INRS/Bell Northern, tests with a 75,000-word vocabulary, and 3 different language models were conducted. The best results (around 90%) were obtained with a trigram model which offered the lowest perplexity [38].

- **Large Vocabulary Speaker-Dependent Continuous Speech:** The IBM TJ Watson Research Center Speech Group announced a 20,000 word, speaker dependent, continuous speech recognition system for 1989 [11]. BBN presented the BYBLOS system in 1987 [30]. This system now uses context-independent, context-dependent and word-dependent phone models, and recognises a 1,000-word vocabulary in real time.

- **Large Vocabulary Speaker-Independent Continuous Speech:** The SPHINX system was developed at CMU. It has been tested on the DARPA Resource Management database, with a vocabulary of 967 words. It uses generalized triphone models, and function word models, with discrete HMM. The syntax is given by word pair, or by a bigram model [83]. The same task has been performed at Lincoln Labs with slightly worse results, using triphone models with continuous 4 Gaussian mixture HMMs [122]. At SRI, a similar system was designed with simpler phone-model discrete HMM [106].

The connectionist approach

In the connectionist approach, reference data are represented as patterns of activity distributed over a network of simple processing units [92,94].

- Perceptrons:

The ancestor of this approach is the Perceptron, a model of visual perception, proposed by F. Rosenblatt [140], that was finally abandoned after having been proved to fail in some operations [105]. More recently, there has been a renewal of interest for this system. This is due to the fact that Multi-Layer Perceptrons (MLP) have been proved to have superior classification abilities over the original perceptron [92], and that a training algorithm, called *Back-Propagation* was proposed recently for the MLP [169,143,79,119]. A Multi Layer Perceptron is composed of an input layer, an output layer, and one or several hidden layers. Each layer is composed of several cells. Each cell *i* in a given layer, is connected to each cell *j* in the next layer by links, having a weight W_{ij} that can be positive or negative, depending on the fact that the initial cell excites, or inhibits the final one. The analogy with the human brain results in calling the cells "neurons", and the links "synapses". The stimulus is introduced in the input cells (set to 0 or 1 if the model is binary), and is propagated in the network. In each cell, the sum of the weighted energy conveyed by the links arriving at that cell is computed. If it is superior to a threshold T_i , the cell reacts, and, in turn, transmits energy to the cells of the higher layer (the response of the cell to incoming energy is given by a *sigmoid function* $S()$) (Figure 10).

In the training phase, the propagated stimulus when reaching the output cells is compared with the desired output response, by computing an error value, which is *back-propagated* to the lower layers, in order to adjust the weights on the links, and the excitation threshold in each cell. This process is iterated until the parameters in the network reach enough stability. This is done for all the stimulus-response pairs.

In the recognition phase, the stimulus is propagated to the output layer. In some systems, the output cell with the highest value designates the recognized pattern. In others, the array of output cell values will be

compared with arrays representing each reference pattern, with a distance measure (like the Hamming distance for binary cells). The role of the hidden cells is to organize information in such a way that the discriminant information is activated in the network to distinguish two close elements. The hope is that the Hidden Cell corresponding to the discriminant cue will impulse on the right output cell with a strong positive weight, while impulsing on the wrong output cell with a strong negative weight.

A close look at the behavior of the Hidden layer cells during recognition has shown the fact that some of them were actually reacting to discriminating features, such as alveolar vs. velar stop detection [41], or Falling 2nd formant around 1600 Hz vs. Steady 2nd Formant around 1800 Hz [167], in the recognition of 'B', 'D', 'G' in various contexts. Comparable interesting self-organizing aspects have been found in HMMs, using a 5 state ergodic model, where all states are connected, with no a priori labeling. After training, it appeared that the states correspond to well-known acoustic-phonetic features (Strong Voicing, Silence, Nasal/Liquid, Stop Burst/Post Silence, Frication) [124].

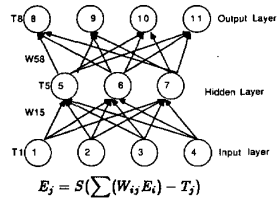


Figure 10: A Two-Layer Perceptron. Each link has a weight W_{ij} . Each cell has an activity threshold T_j ; E_i is the energy emitted by cell i .

- Time Processing:

If the discriminating power of such a network is of interest for speech recognition, the time parameter is difficult to modelize. Several ways of taking it into account can be reported:

- **Fixed length time compression:** One approach is simply to use as reference length the largest possible length, and to pad the words which are shorter with silence [123]. Another possibility is to normalize the spectrogram corresponding to a word, to a fixed length (this can be achieved by fixed length linear, or non-linear time compression). If the spectrogram is of length L , each vector being of dimension D , the network will have $D \cdot L$ input cells. If the size of the vocabulary is M words, the network will have M output cells. A word will be recognized if the corresponding output cell has the highest activation value.

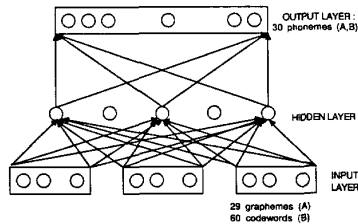


Figure 11: An example of a contextual Multi-Layer Perceptron. Each rectangle corresponds to several cells, representing different units (in the case of grapheme-to-phoneme recognition (A), or phoneme recognition (B). Here, the context is one on the right, and one on the left of the stimulus.

- **Contextual MLP:** In order to take the contextual information into account, the input can include the context in which the stimulus occurs. T. Sejnowski used that method for grapheme-to-phoneme conversion in English [153]. Let us assume that there are 26 graphemes, 3 punctuation marks and 30 phonemes in English. The input is composed of 7 groups of 29 cells. Each group represents one grapheme or punctuation mark, with the corresponding cell being set at 1, the 28 other ones at 0. The central group represents the grapheme to be translated, the 3 on the left and the 3 on the right representing respectively the left and right contexts (3 graphemes on the left of the one to be converted, and 3 on its right). The corresponding phoneme is given in the output cells (Figure 11). It means that there are 30 output cells, and that the one corresponding to the phoneme is set to 1, while the other ones are set to 0 (actually, a different coding was used for the output cells, based on 17 phonological features, 4 punctuation features and 5 stress and syllable boundary features (like vowel, voiced, etc.), resulting in a total of 26 output cells). The network

can be trained on one-word grapheme-phoneme pairs, obtained from a lexicon or from continuous text, and is able to learn some conversion rules that it can apply to new words. Depending on the size of the training corpus, the quality of the conversion on unseen words improves. With this approach, the authors achieved a 78% correct conversion test on a continuous unknown text, which is less than results obtained by using hand-made production rules.

This approach has been enlarged to speech recognition [22]. In this case, the input is made of 11 groups of 60 cells corresponding to labels obtained by vector quantization with a 60 prototype codebook. The output is made of 26 cells corresponding to the 26 phonemes used to compose the German digits. Training is carried out on a labelled speech corpus by simultaneously giving the label and its context as input, and the corresponding phoneme as output. The training corpus includes 2 pronunciations of each isolated digit. The recognition of the digits themselves, pronounced in isolation or continuously, is achieved by DTW after phoneme recognition. The comparison of this technique with a simple one-state phone-model HMM was in favor of the MLP approach (No errors for IWR, and 92.5% for 7 digit strings, against 80% and 70% (discrete (with the same VQ Codebook) HMM) and 100% and 90% (continuous HMM)). One of the striking results is the discriminating power of the MLP approach, the emergence of the right phoneme being much more apparent than in the case of HMMs where the right phoneme has a weak emergence when compared with the second best, even if the final decision of recognizing the right word is, in both cases, correct.

- **Time Delay Neural Networks (TDNN):** Another similar approach has been proposed by A. Waibel [167]. The task is the recognition of the 3 phonemes 'B', 'D', 'G' in different contexts. Here, the MultiLayer perceptron is composed of 2 Hidden Layers. The length of the input stimulus is fixed, and equal to 15 frames (150 ms). The input layer is made of 16 cells representing 16 Cepstral coefficients, each cell being connected to the cells of the first Hidden Layer by 3 arcs representing the value of a coefficient at time t , $t-10$ ms, and $t-20$ ms. The first hidden layer is composed of 8 cells. Each cell is connected to the cells of the second Hidden layer by 5 arcs representing the values of the cell at time t to $t-40$ ms. The second hidden layer has 3 cells. Each cell of the output layer receives the energy integrated over the total duration of the stimulus from one of the second Hidden layer cells. The output layer is composed of 3 cells representing each phoneme. The learning phase takes into account the fact that the arcs corresponding to a coefficient at a given time t will be observed 3 times (at time t , at time $t+10$ ms with a 10 ms delay, and at time $t+20$ ms, with a 20 ms delay). This approach has been compared with a discrete HMM approach, using 4 states, 6 transitions and a 256 prototype codebook. There is one model for each of the 3 phonemes. Results came out in favor of the MLP approach (1.5% error rate, instead of 6.3%). Here also, the emergence of the correct phoneme, when compared with the second best, shows the higher discrimination abilities of the MLP approach.

- **Neural Nets and the DTW or Viterbi algorithms:** In order to better account for the good discriminative properties of the Neural Networks approach, as well as the good time alignment properties of DTW [147] or Viterbi algorithms [93,58], some first attempts to use them in the same framework can be found.

- Boltzman machine / Simulated annealing:

The Boltzman machine is also composed of nodes, and weighted links between nodes. Unlike MLP, the nodes are not organized on different layers, but one node can be connected to any other node (fully connected). They are usually divided into visible and hidden nodes. The visible nodes can also be divided into input and output nodes. Another difference is that the nodes can usually take binary values. Each node is given a probability to be 0 or 1. This probability function depends on the "difference of energy" (equal to the weighted sum of the energy issued from the connected nodes) incoming in that node, whether it is set to 0 or 1. It also contains a term which is comparable to the temperature parameter in Thermodynamics. The higher the temperature, the more the node will be able to take the 0 or 1 values at random. The lower the temperature, the more the node will be influenced by the state of the nodes connected to it, and by the weights of the corresponding links. At the beginning of the optimization process, the temperature is high, and then it is decreased slowly. This process, known as "simulated annealing" [69], has the goal of avoiding system stabilization in a local minimum of energy (corresponding to a non-optimal solution), thus missing the true one, as it will help the system to quit that local minimum.

During the training phase, each node is first given a random value, 0 or 1. Then, the stimulus is given to the input nodes, and the desired response is given to the output nodes. The *simulated annealing* method obtains the best equilibrium, corresponding to the lower energy of the total network. For the whole training set, statistics are collected for each link on how many times the nodes at each extremity of the link were on simultaneously. The same process is used without giving any information to the output nodes. The comparison of the two allows network training, that is, updating the weights on the links, by decreasing, or increasing, their value by a fixed amount. The recognition process consists of applying the stimulus to the input nodes of the network, using

the simulated annealing method to get the optimal solution, and considering the output nodes to obtain the recognized pattern.

This approach has been used for multi-speaker vowel recognition experiments, using as input one spectrum to represent a vowel pronounced in isolation [126]. It has also been compared with a MLP having the same number of nodes (50 hidden nodes) [127]. The results showed that the Boltzman Machine was slightly better than the MLP (about 3% difference; 2% error rate against 5% on the data used for training after 25 training cycles, 39% against 42% on untrained data for the same speakers, after 15 training cycles). It has also been noticed that the MLP is about 10 times faster than the Boltzman Machine.

- Feature Maps:

The Feature Maps, or Phonotopic Maps [71], go on the hypothesis that, for speech recognition, the information that is closely related should also be topologically closely located, as it might be in the brain [72]. It is an unsupervised approach since no information is given to the system about the desired output during building of the map.

The process is similar to clustering. The network can be represented as a two-dimensional grid. Each point corresponds to a prototype spectrum. When a new spectrum of the speech data is presented, it is compared with all the existing prototypes, using a similarity measure like the Euclidean distance. When the closest one is found, the corresponding prototype is averaged with the new vector, taking into account as a weight, the number of spectra that resulted in the prototype. The eight adjacent neighbors are also modified according to the new input, with a lower influence. The same can also be applied to the 16 following neighbors (Figure 12). At the end of this process, quantization is obtained, as it would be with a clustering approach, but each prototype is close to a prototype which is similar to it. The quality of this quantizer has been compared to conventional ones [171]. The network is then labelled, by recognizing labelled sentences, and giving the corresponding labels, with an appropriate decision scheme, to the nodes in the grid. A recognition process will correspond to a trajectory in the labelled network (also called Feature Map).

This approach has been applied with success to the recognition of Finnish and Japanese (speaker-dependent, isolated words, 1000 word vocabulary). The phoneme recognition rate varies from 75% to 90%, the word recognition rate varies from 96% to 98%, the orthographic transcription of a word, using a language model, varies from 90% to 97%, depending on the vocabulary, and the speaker [73].

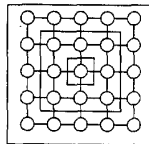


Figure 12: Example of a Feature Map architecture
Each cell corresponds to a prototype in Vector Quantization, connected to its neighbors, which are similar prototypes. If the cell in the middle is modified, its 8 close neighbors and the 16 next ones will also be modified.

- Guided Propagation:

Another system is based on a principle of *guided propagation*, supported by a topographic memory. Speech is transformed into a spectrum of discrete and localized stimulation events processed on the fly. These events feed a flow of internal signals which propagates along parallel memory pathways corresponding to speech items (i.e. words). Compared with the layered methods described above, this architecture involves a set of processing units organized in pathways between layers. Basically, each of these context-dependent units detects coincidences between the internal activation it receives (context) from the path it participates in, and stimulation events.

This approach has been used for the speaker-dependent recognition of isolated digits (0-9) in noise, on a limited speech test database. The noise is itself constituted of speech (an utterance of the number 10 pronounced by the same speaker), with a 0 dB Signal-to-Noise ratio. It has been compared with a classical DTW algorithm. The results in noise-free conditions were no errors for the DTW algorithm, and 2% errors for the connectionist model. When the noise is added, it gave 47% error for the DTW algorithm, and 10% error for the connectionist model. However, it should be noticed that the signal processing was different in the two cases (cepstral coefficients for DTW, simplified auditory model including *lateral inhibition* and *short term adaptation* for the Connectionist system) [15,78].

- Other systems:

Other connectionist systems, which can be applied to pattern recognition in general, and speech recognition in particular exist. The

Hopfield Net has a single layer, each cell being connected to all the other ones. It is used as an associative memory, and can restore noisy inputs. The Hamming net is similar to the Hopfield Net, but first computes a Hamming distance to compare the input vector with the reference patterns [94].

Other approaches are on their way, but complete results have not yet been published. J.L. Elman and J.L. McClelland proposed the TRACE model as an interesting model for speech perception, or an architecture for the parallel processing of speech. The first version, TRACE I, accepted the speech signal as input [40]. An improved version, TRACE II, accepts only acoustic features as input [96].

- Use of Neural Networks for language modeling:

The use of the self-organizing approach has proved to be efficient for Language Modeling, as it appears in Markov language models. Some trials can also be found, that use the Neural Network approach. One approach consisted of trying to extend the Bigram or Trigram models to Nigram models [113]. For a basic bigram model, the MLP which is used has 89 input cells (corresponding to 89 grammatical categories) for the word N, and 89 output cells for giving the category of the word N+1. There are two hidden layers with 16 cells in each. This MLP has been generalised to 4-grams. It was trained on 512 sentences, and tested on 512 other sentences. For a Trigram model, the results were comparable to those obtained with a Markovian approach, the information being reduced more than 130 times. Examination of the hidden cells showed that they classified the word categories into significant groups.

Although the Neural Network approach looks very appealing and quite promising, several problems are still unsolved: which architecture should be chosen, how many layers, how many cells should be used, how to deal with time processing, what should the representation of the stimulus-response pairs be, how is it possible to reduce the computation time. At present, no definite experiment in completely comparable conditions, on a large enough scale, and on a sufficiently general task, taking advantage of the interesting features of the two different approaches, has proved the superiority of the Neural Network method over statistical or pattern matching methods.

"Knowledge-Based" Methods

The "Knowledge-Based" approach became very popular when the "Expert System" technique was proposed in Artificial Intelligence. The idea is to separate the knowledge that is to be used in a reasoning process (the *Knowledge Base*), from the strategy, or reasoning mechanism on that knowledge (based on the *Inference Engine*, which fires rules). The reasoning strategy is also reflected by the way the input information (the "Facts") is processed (*left-to-right* or *Island-Driven*), and the order in which the rules are introduced, or arranged as packets of rules in the knowledge base. Most of the manipulation of information, including inputting information to be processed, is taken care of through the *Fact Base*. Knowledge is represented as "if Facts then Conclusion1 else Conclusion2" rules. It can be accompanied by a weight representing, as a heuristic, the confidence that one could apply to a given rule conclusion. The *Inference Engine* can try to match the goals to the input by applying the rules in the Knowledge Base, starting from the goals present in the conclusion of the rules, and then checking if the result of such firings is actually the input (*Backward Chaining*, *Goal Directed* or *Knowledge Driven*). Or, on the contrary, it can start from the input, find applicable rules, and fire them until a goal is obtained (*Forward Chaining*, or *Data Driven*). The strategy can change during the decoding process, on the basis of intermediate results.

This approach implies that the knowledge has to be manually entered, unless some automatic learning procedure is found. The effort for obtaining a sufficiently large amount of knowledge for speaker-independent continuous speech recognition, for large vocabularies, was measured at the early ages of this approach (beginning of the '80s) to be around 15 years.

- Spectrogram reading expert systems:

As it has been shown that some expert spectrogram readers are able to "read" speech spectrograms with a high decoding score (80% to 90%), several attempts have been made to "mimic" those experts in a "knowledge-based" expert system [31].

The expert has discussions with a "cognitive engineer" (usually a computer scientist), who has the role of extracting the facts, the knowledge, and the strategies with which the expert is applying his knowledge on the facts. Most of the time, such approaches aimed at studying a specific set of phonemes for a specific speaker [162], or a set of phonemes in a specific context, like word initial singleton stop consonants at MIT, for any speaker [174], or even some specific cues.

A problem lies in the fact that the expert, before applying his rules, uses visual cues, which are difficult to represent by rules applied on symbols. A way to avoid this visual perception problem, which deals with computer vision, is to manually verify all features measured by the system [174], or to take as entries a list of features given by the user as he "reads"

the speech spectrogram. The expert system can take the initiative of asking questions [162].

- Other approaches:

Apart from the "expert spectrogram reader" project, work was conducted at MIT for segmenting and labeling speech by using a knowledge-based approach [175]. The segmentation process produces a multi-level representation, called a "dendrogram", very similar to the scale-space filtering idea used in other areas like computer vision [170]. The speech spectrogram is segmented in units of different levels of description, from fine to coarse, the last segment being the whole sentence. This process is based on the computation of a similarity measure between adjacent segments, using an Euclidean distance on the average spectral vectors of each region previously delimited, and on the merge of similar ones. Segmentation results were 3.5% deletion, and 5% insertion errors, on 100 speakers. A phoneme lattice is then obtained by using a statistical classifier. The lexical representation has different pronunciations for each word. The result is a word lattice. On a 225-sentence test, with an average 256 word vocabulary, considering the rank order of words starting at the same place as the correct word, but having better scores, shows that the correct word is first in 32% of the cases, among the 5 top candidates in 67% of the cases, and among the 10 top in 80% of the cases. The corresponding allophone recognition rate is 70% (top choice) and 97% (5 top).

The "Angel" system was developed within the DARPA program, as a speaker-independent, continuous speech, large vocabulary recognition system. Recognition was conducted by using location and classification modules, which have the task of segmenting and identifying the corresponding segments. The output is a phoneme lattice, with label probabilities for each segment. Examples of such modules are stop, fricative, closure, or sonorant modules. The system was tested on the DARPA Resource Management Database.

Some work has aimed at integrating the knowledge-based approach with a stochastic HMM approach [56]. Others tend to use more complex knowledge-based system architectures like the *Specialist Society* structure [54], or the *Expert System Society* structure, with inductive learning [35].

Speech processing and Natural language

Now that speech recognition systems obtain acceptable results under acceptable conditions (the size of the vocabulary is large enough to allow for interesting applications, the pronunciation is continuous, and any speaker can be recognized with little, or no training), one of the major remaining difficulties is the link with the language that will be used in the application. This language may contain sentence structures which are not in the syntax, words that are not in the lexicon, hesitations, stuttering, etc. At the present time, demonstrations of advanced systems ask people to read a list containing acceptable sentences so that the sentences read follow the syntax rules, and that only the acceptable words are used.

When a real user pronounces a sentence that has not been foreseen by the syntax, the system gets into trouble. Of course, the less constrained the syntax, the more the system is able to accept sentences that deviate from an a priori human description of the possible sentences. But at the same time, the syntax gives less help in avoiding acoustic recognition errors. This is the case when a trainable word-pair or bigram grammar is used in place of a deterministic context-free grammar.

In the case of written text, training is feasible by using a large enough amount of text data. For spoken dialog, such data is difficult to obtain. Also, if in-depth "understanding" is not necessary for a text dictation task, it is mandatory in a dialog task, where the system must activate the appropriate response (answer generation, or corresponding action). The link with Natural Language Processing is a necessity. The problem then is that most of the NLP methods are usable in the case of deterministic input (a clean error-free word sequence). In the case of speech, the word sequence is ambiguous, both at the acoustic decoding level and at the level of segmentation into words, and the "understanding" process itself intervenes in solving those ambiguities! Few attempts to address this problem under realistic conditions can be identified. Also, using information relative to the semantics or the pragmatics of the task will reduce the generality of a system to the task where it is used.

Very few advanced systems integrating Speech Recognition and Natural Language Processing can be mentioned. In the MINDS system at CMU [172], the goal is to reduce the perplexity of the language by being able to generate predictions on which concepts could be conveyed in the next sentence during a dialogue, consequently making predictions on the corresponding vocabulary and syntax. The information used is the knowledge of problem-solving plans, the semantic knowledge on the application domain, the domain-independent knowledge on speaking mechanisms, the dialog history, the user expertise and also the user linguistic preferences. The test and training sets were obtained from the TONE database (NOSC, US Navy). The use of such information reduced test set perplexity from 242.4 words (when solely using grammar) to 18.3 words. Using the SPHINX system, with 10 speakers, each pronouncing 20 sentences, in a speaker-independent mode, the word accuracy went from 82.1% to 96.5%, and semantic accuracy from 80% to 100%.

In the VODIS project conducted in the framework of the UK

Alvey initiative, the task is Voice Operated Database Inquiry Systems, accessible to casual users via the telephone network [173]. The DTW-based continuous speech recogniser is linked to a frame-based dialog control. The first version, VODIS I, was based on tight syntactic constraints. In VODIS II, the goal is to weaken these constraints, following the results of field trials with naive users. The recognition process, including syntactic constraints obtained from a context free grammar, produces linked lists of words, from which a lattice of alternative words can be generated. This lattice is parsed by a bottom-up chart parser, and the best scoring alternatives resulting from that parse are converted to a frame format, including the "nextbest" solutions. The semantics and pragmatics of the task are applied to the frames to obtain the best acceptable solution. At BBN, a Chart Parser is also used on a word lattice [13].

Another project concerns the use of oral dialogue for air traffic controller training [102]. The DTW-based continuous speech recogniser is linked to a frame-based knowledge representation. At one step of the dialogue, a sentence is recognized by an optimal DTW algorithm, with a weakly constrained syntax. Analysis of the sentence determines its category and instantiates the corresponding frame, putting the words in the frame slots. A validity control process using the semantic-pragmatic constraints associated with the frame detects system or user errors. Error correction can be made either by checking in a *word confusion matrix* the words which can be confused with the ones that have been recognised, and are syntactically and semantically acceptable, or by running a new recognition process on the same speech signal with different parameters, or by generating a question to the user. Interpretation of the message gives a sequence of commands to the Air Control Simulator, updates the task context and the dialogue history.

Related progress in other areas of Speech Processing

Of course, since Vector Quantization was primarily designed for low bit rate speech coding (around 800 bits/s), many applications of VQ can be found in this area. There have also been Segment Coding experiments. Initially, the goal was to design a "phoneme vocoder", taking into account the fact that, if the initial PCM rate is in the region of 64 Kbits/s, the rate for transmitting phonemes after recognition would be in the range of 50 Bits/s! Moreover, it may not be necessary to recognize a phoneme string without errors, since the human "upper levels" could bring out the higher level information that helps to recognize the sentence, despite the phoneme recognition errors. Experiments conducted in modifying the phonemes in a text-to-speech synthesizer in French [89] have shown that an error rate of over 15% on the phoneme string, or even a grave phoneme recognition error, could prevent the recognition of a whole sentence. This gives some idea of the lowest acceptable phonemic recognition rate, in a situation where speech understanding systems have upper levels as powerful as humans (for an undefined semantic universe).

Interestingly, the first methods based on diphone recognition [149] did not result in an acceptable recognition rate, and thus good enough transmission intelligibility. While new attempts to use segment coding, without labelling the recognized segments, gave acceptable results, with slightly higher transmission rates (around 200 bits/s) [142]. As in the comparison of the *centisecond* model with VQ, it is also apparent here that labelling should be carried out only when decisions can, and must be made. Similar segments have been used for speech synthesis [112].

Vector and Segment Quantization techniques have been used in speaker verification [161,57], and in voice conversion [1]. For Voice conversion, the codebook of a speaker is mapped with the codebook of another speaker, thus giving the correspondence for each prototype. When the reference speaker pronounces a sentence, the Vector Quantization process is applied, and each label is replaced by the label of the corresponding prototype of the other speaker, thus resulting in the same sentence pronounced with the timbre of the second speaker. This approach is used by ATR to synthesize the translation to Japanese of a sentence initially pronounced in English with the voice that the English speaker would have if he actually spoke Japanese.

HMMs have been used in Formant Tracking, Pitch Estimate and Stress Marking (for speech synthesis), speaker recognition, written language recognition (in order to choose the adequate grapheme-to-Phoneme conversion rules), character recognition, automatic translation, etc.

As already mentioned, connectionist approaches have been used, not only for grapheme-to-phoneme conversion, but also speech enhancement, signal processing, prosody marker assignment in speech synthesis. A possible future application is its use in multimodal man-machine communication.

Accompanying hardware

The venue of specialised Speech Processing chips has been of major importance in the recent history of Speech Processing. Texas Instruments initiated this process, with its LPC synthesis chip in the Speak'n Spell electronic game.

Digital Signal Processing chips: DSP chips have allowed real time digital processing of speech signals, with various transforms, thus bringing consistent analysis, flexibility and higher integration. First examples of

such devices were the 2920 from INTEL, followed by the NEC 7720. More recent DSP circuits are those of the TMS 320 family, the AT&T DSP16 and DSP32, the MOTOROLA DSP56000, the ADSP-2100 from Analog Devices, etc. While the first circuits allowed only for fixed-point computation, the more recent ones, like the TMS 320C30 [164], the DSP56000 [158] or the DSP32 [18], permit floating-point computation.

- DTW chips: DTW chips are chips specialised in computing Dynamic Programming algorithms. As these algorithms require much computing power, it is good to have devices that do it fast. This allows for a larger vocabulary, or improves the quality of the systems, by permitting the use of optimal algorithms. NEC proposed its "chip set" for Isolated Word Recognition (NEC 7761-7762) in 1983. They also presented a Connected Word DTW chip (NEC 7764) at that time [61]. At Berkeley, a chip was developed for the recognition of 1,000 isolated words in 1984 [67]. A new chip is now under study, at Berkeley and SRI, for the recognition of 1,000 words, continuous speech, that should be able to execute the Viterbi algorithm for discrete HMMs with a speed of 75,000 to 100,000 arcs per frame in real time [107]. VECYSYS [128] and AT&T [51] propose DTW chips with comparable power. The MUPCD from VECYSYS is announced for 70 MOPS (Million Operations per Second), recognizes 5,000 isolated words, or 300 words, continuous speech, in real time. The GSM (Graph Search Machine) from AT&T is announced for 50 MIPS (Million Instructions per Second). It has also been tried for recognition using the Hopfield Net [109].

- special architectures: The need for computing power may lead to special architectures. By developing its proprietary "Hermes" chip, that was integrated in its PI board which could fit in a PC-AT bus, IBM was able to present in 1986 a system that ran in 1984 on 3 Array processors, an IBM 4341, an Apollo workstation (and a PCI), with greater impact, and the proof that HMMs were not only a mathematical tool reserved for mainframe addicts!

At CMU, the SPHINX system benefited from the BEAM architecture (3,000 arcs per frame in Real Time) [17]. The Level Building algorithm has been installed at AT&T on the tree structured ASPEN system [52]. The interesting parallel features of transputers also lead to new results [28]. However, the size of the effort necessary to install software on a special architecture with non-standard high level languages should be proportionate to the expected increase in performance. C or vectorised C compilers are proposed on the architectures mentioned above.

Assessment

One of the major failures of the ARPA-SUR project, conducted from 1971 to 1976, was that at the end the systems were found to be difficult to compare, since they were tested on completely different languages, having different difficulty, and completely different tasks. Only different systems coming from the same laboratory were compared on the same data (such as HEARSAY and HARPY at CMU). The problem was that in the initial call for proposals, only the size of the vocabulary was given, not the difficulty of the language. In the present DARPA project, special emphasis has been put on the definition of assessment methodology, and the corresponding speech data bases, thus resulting in regular testing and comparison of the systems on the very same data. This is true both for assessing the improvements during the development of a system, and comparing results obtained in different laboratories having slightly, or very different approaches.

Measuring the a priori difficulty of a language to be recognized is difficult. It includes both the constraints brought about by the syntax, and the acoustic similarity of the words, if they can be uttered in the same time slot, that is if they are present at the same node of the syntax. The perplexity of the language gives its difficulty regardless of the acoustic similarities between words. It can be computed from the entropy of the language. This is easily achievable for a syntax given by a finite state automaton. If the syntax is local, like bigrams or word pairs, it has to be computed according to the test data (test set perplexity) [68].

To test a recognition system, the basic idea is to build a large test corpus, and to test all the systems on that test set. Texas-Instruments (TI) has been among the first to make large speech databases. In France, the CNRS-GRECO has followed. The NATO RSG10 group built a multilingual database in 1980. Depending on the size of the database, and on the performance of the system, the accuracy of the results will be more or less meaningful.

Test methodology is also of importance. The scoring technique itself should be carefully defined. In continuous speech, different errors may occur: substitutions (a word is recognized in place of another one), insertions (a word is recognized when nothing was pronounced) and deletions (nothing is recognized, whereas something was pronounced). Two performance measures are proposed. The "Percent Correct" which refers to the input word strings, and checks how many input words are correctly recognized. Thus, it does not take into account the insertion errors. The other measure, called "Word Accuracy", considers the three different types of errors. The addition of those three types of errors can lead to negative recognition rates (the recognition of "You need" instead of "unit" counts for two insertion errors (or one substitution and one insertion) for the pronunciation of a single word, for example). Software

for the automatic extraction of scoring results has been designed by NIST (National Institute of Standards and Technology, formerly NBS (US National Bureau of Standards)) based on DTW, and is used to test the systems designed in the present DARPA project [118]. At the present time, NIST in the USA is continuing its effort in that direction. At the European level, the SAM project is aiming at the definition and use of large multilingual databases. A similar effort is under way in Japan. The following tables give examples of systems that have been tested.

Lab	System	Speaker	Voc.	Pxty	% Corr.	Word Acc.	Date
CMU	Hearsey	dept	1011	4.5	86%		1977
CMU	Harpy	dept	1011	4.5	97%		1977
IBM	Laser	dept	1000	24	91.1%	88.6%	1980
BBN	BYBLOS	dept	997	60	94.8%	92.5%	1987
BBN	BYBLOS	dept	997	997	70.1%	67.6%	1987
PSI	SPICOS	dept	917	74	92.8%	92.0%	1988
CSELT	CSELT	dept	1011	35	79.0%		1988
CMU	ANGEL	indept	997	997	45.5%	41.0%	1986
TI	TI	indept	997	997	55.5%	44.3%	1987
SRI	SRI	indept	997	997	43.6%	40.4%	1988
CMU	SPHINX	dept	997	20	96.9%	95.8%	1988
CMU	SPHINX	dept	997	60	94.9%	95.8%	1988
CMU	SPHINX	dept	997	997	76.5%	95.8%	1988
CMU	SPHINX	indept	997	20	96.2%	95.8%	1988
CMU	SPHINX	indept	997	60	94.7%	95.7%	1988
CMU	SPHINX	indept	997	997	74.2%	70.6%	1988
Lincoln	Lincoln	dept	991	60		94.5%	1988
Lincoln	Lincoln	dept	991	991		80.7%	1988
Lincoln	Lincoln	indept	991	60		89.9%	1988
Lincoln	Lincoln	indept	991	991		66.1%	1988

Table 1: Large Vocabulary Continuous Speech Recognition Systems (After K.F. Lee [83, 122], PSI: Philips-Siemens-IPO, For TI results, only Male Voice has been tested))

Lab	System	Speaker	Voc.	Pxty	Word Acc.	sent. corr.	Date
Lincoln	Lincoln	dept	991	60	94.5%	68.0%	1988
Lincoln	Lincoln	indept	991	60	89.9%	57.3%	1988

Table 2: Sentence Recognition rate compared to Word Accuracy [122]

Lab	System	Speaker	Voc.	Pxty	% Correct	Date
IBM	Tangora	dept	5000	160	97.1%	1986
IBM	Tangora	dept	20000	250	94.6%	1987
Dragon	Dragon	dept	997	21*	98.6%	1988
Dragon	writer	dept	997	21*	91.2%	1988
INRS	INRS	dept	75000	67	89.5%	1988

Table 3: Some Large Vocabulary Isolated Word Recognition System Results (with Language Model) (* for sentences obeying grammar) [38,154,64,141]

- Results on phoneme recognition:

Using a model very close to the one that was used in the SPHINX system, K.F. Lee at CMU tried on Speaker-Independent Phone recognition in continuous speech. The test database is the TIMIT database. 2830 sentences from 357 speakers were used for training, and 160 sentences by 20 speakers for testing. The overall model is a *Looped Phonetic Model* (LPM), the best results being obtained with right-context-dependent phone models. A bigram phone model can be used to give the probability of two-phoneme successions. A unigram model giving the probability of one phoneme has also been tried [81]. Comparable test experiments have been conducted at IBM-Paris Scientific Center, in speaker-dependent mode, with MMI training and a phoneme bigram model [104], and at Helsinki University, with the *Feature Map* approach [73].

Type of error	SPHINX (Indpt)	IBM-P (Dept)	Helsinki (Dept)
Correct	73.8%	84.8%	75%-90%
Substitutions	19.6%		
Deletions	6.4%		
Insertions	7.7%	4.18%	

Table 4: Phoneme Recognition Rates and types of errors

Class	% correct	Phone Lang. Mod.	% correct
sonorant	65.7%	bigram	73.8%
stop	69.9%	unigram	70.4%
fricative	78.4%	none	69.5%
closure	93.3%		

Table 5: SPHINX Phoneme Recognition Rates by broad phone class, and depending on the phone language model used [83]

How did the ICASSP conferences reflect this adventure?

Most of the referenced papers in this article have been found in the proceedings of the past 13 ICASSP conferences. This shows how much the ICASSP conference, from the first one in 1976, in Philadelphia until

the 14th in 1989 in Glasgow has been the annual meeting of the international speech community. Since the first ICASSP, 2,000 papers have been published on speech processing (analysis, synthesis, recognition, coding or enhancement, and speaker and language recognition) [10].

The variation in the total number of papers shows that each time the ICASSP conference goes outside the US, it gained in the number of papers for the following year. The number of papers on Speech Processing has always represented about one third of the total number of papers.

If one considers the individual countries, 138 countries have produced papers, but 7 countries (USA, Japan, France, UK, FRG, Italy and Canada) published 90% of the papers.

The largest share of the publication of papers goes to the US, which produced half. Europe is responsible for about one fourth, and Japan for one eighth.

This share can vary dramatically when the conference takes place outside the US, as it appears in 1982, when it first left the US for Paris (40% for the US and for Europe), in 1986 when it went to Tokyo (about 30% for US, for Europe and for Japan), or this year in Glasgow.

The number of laboratories varies from country to country. About 150 laboratories produced papers in the US, 120 in Europe 40 in Japan, and 70 in the other countries, for a total of 380 laboratories.

The 11 laboratories that have published the most (AT&T Bell Labs with 10% of the whole, BBN, MIT, Lincoln Lab, CMU, IBM-Yorktown, NTT, TI, CNET, CSELT and Georgia Tech) are responsible for 30% of the papers.

Conclusion

We have seen that interesting improvements can be reported from the recent results in Speech Recognition. To summarize, the use of large speech databases, with elaborated learning procedures, can allow for speaker-independent continuous-speech recognition with satisfactory results, if one considers word accuracy. At the same time, the phoneme recognition rate also attains a quality level that may make large performance recognition systems possible. Hidden Markov Models have proved to be powerful tools. Connectionist Models may bring new possibilities and improvements. The link with Natural Language Processing is now a necessity, in order to determine how robust the systems are when they process fluent speech in real applications, and how to make them usable.

Some important results have been found: there is no need for a priori segmentation, as an implicit segmentation is made in the recognition process itself. The more data for training, the better the recognition results. As it is easy to have data from many speakers, speaker-independent recognition can be achieved with almost as good results as speaker-dependent recognition, even if the difficulty of the task is higher.

The introductory remark on using human expertise vs self-organization may mean that we will be able to create systems with good performance, without being able to actually understand in detail how they work, just as humans are able to use their perception, action and reasoning abilities without understanding the way it works (if we did, the knowledge-based approach would be trivial). However, as the system uses a model, the study of the parameters in the model after it has been trained on the data base, may help understanding what the underlying hidden structures are.

The past results give us confidence that the next ICASSP conferences will continue to bring exciting results, as they have done in the recent history of Speech Recognition.

References

Obviously, this paper cannot present all the interesting work that has been achieved in the recent years. I have tried to use the more synthetic references on a topic, and considered mainly the articles where test experiments have been conducted on data of acceptable size. Also, IEEE publications, and publications in English were preferred. Usually, I always forget to mention my lab's work. In this paper, one may find that too many references are from LIMSI, but it was more convenient for me to get the information from my close colleagues, in the case where similar work was also conducted in a different laboratory. I would like to apologize in advance for any omissions that you may find, and for any errors which are unavoidable in such a review.

- [1] M. Abe, S. Nakamura, K. Shikano, H. Kuwabara, "Voice Conversion through Vector Quantization", IEEE ICASSP88, pp655-658, New York, 1988
- [2] M. Adda-Decker, J. Mariani, "Collaboration between Analytical and Global Methods for Continuous Speech Recognition", 7th FASE Symposium on Acoustics and Speech, SPEECH88, Vol 1, pp345-352, Edinburgh, 1988
- [3] A. Andrieu, J.P. Binquet, F. Delille, C. Fluhr, Y. Hlal, J.S. Lénard, J. Mariani, B. Poudroux, "Les dictionnaires en forme complète et leur utilisation dans la transformation lexicale et syntaxique de chaînes phonétiques correctes", 10èmes JEP du GALT, Grenoble, May 1979
- [4] J.K. Baker, "The DRAGON system - An overview", IEEE Transactions on ASSP, Vol. 23, N 1, pp 24-29, Feb. 1975
- [5] L.R. Bahl, R. Bakis, P.S. Cohen, A.G. Cole, F. Jelinek, B.L. Lewis, R.L. Mercer, "Further Results on a Continuously Read Natural Corpus", IEEE ICASSP80, pp872-875, Denver, April 1980
- [6] L.R. Bahl, F. Jelinek, R.L. Mercer, "A Maximum Likelihood approach to continuous speech recognition", IEEE Trans. on Pattern Analysis and Machine Intelligence, VOL 5, N 2, pp179-190, March 1983
- [7] L.R. Bahl, P.F. Brown, P. V. de Souza, R.L. Mercer, "Speech Recognition with Continuous-Parameter Hidden Markov Models", Computer Speech and Language, VOL 2, N 3/4, September/December 1987

- [8] L.R. Bahl, R. Bakis, P. DeSouza, R.L. Mercer, "Obtaining Candidate Words by Polling in a Large Vocabulary Speech Recognition System", IEEE ICASSP88, pp489-492, New York, 1988
- [9] L.R. Bahl, P.F. Brown, P.V. de Souza, R.L. Mercer, "A New Algorithm for the Estimation of Hidden Markov Model Parameters", IEEE ICASSP88, pp493-496, New York, April 1988
- [10] L.R. Bahl, P.F. Brown, P.V. de Souza, R.L. Mercer, M.A. Picheny, "Acoustic Markov Models used in the Tangora Speech Recognition System", IEEE ICASSP88, pp497-500, New York, April 1988
- [11] L.R. Bahl, P.F. Brown, P.V. de Souza, P.S. Gopalakrishnan, F. Jelinek, R.L. Mercer, "Large Vocabulary Natural Language Continuous Speech Recognition", IEEE ICASSP89, Glasgow, May 1989
- [12] L.E. Baum, "An Inequality and Associated Maximization Technique in Statistical Estimation of Probabilistic Functions of Markov Processes", Inequalities, Vol 3, pp1-8, 1972
- [13] "Speech and Natural Language Work at BBN", DARPA Review Meeting, Pittsburgh, June 1988
- [14] R.E. Bellman, "Dynamic Programming", Princeton Univ. Press, 1957
- [15] D. Bérrou, "Guided Propagation inside a Topographic Memory", 1st International Conference on Neural Networks, IEEE, San Diego, June 1987
- [16] R. Billi, G. Masia, F. Nesi, "Word Preselection for Large Vocabulary Speech Recognition", IEEE ICASSP86, pp65-68, Tokyo, 1986
- [17] R. Bisiani, T. Anantharaman, L. Butcher, "BEAM: An Accelerator for Speech Recognition", IEEE ICASSP89, Glasgow, 1989
- [18] J.R. Boddie, W.P. Hays, J. Tow, "The Architecture, Instruction Set and Development Support for the WE DSP32 Digital Signal Processor", IEEE ICASSP86, Tokyo, April 1986
- [19] B.P. Bogert, M.J.R. Healy, J.W. Tukey, "The Quaternary Analysis of Time Series for Echoes, Centroids, Pseudo-autocorrelations, Cross-Correlation and Spline Cracking", Proc. Symposium Time Series Analysis, M. Rosenblatt Ed., John Wiley and Sons, New York, pp 223-243, 1963
- [20] S. Boll, J. Porter, L. Bahler, "Robust Syntax Free Speech Recognition", IEEE ICASSP88, pp179-182, New York, 1988
- [21] H. Bonneau, J.L. Gauvain, "Vector Quantization for Speaker Adaptation", IEEE ICASSP87, pp1434-1437, Dallas, 1987
- [22] H. Bourlard, C.J. Wellekens, "Speech Pattern Discrimination and Multi-Layer Perceptrons", Research Report M.211, Philips Research Laboratory, Brussels, September 1987
- [23] L. Boves, M. Refice et al., "The Linguistic Processor in a Multi-Lingual Text-to-Speech and Speech-to-Text Conversion System", European Conference on Speech Technology, 1987, Edinburgh, 1987
- [24] J.S. Bridle, M.D. Brown, R.M. Chamberlain, "An Algorithm for Speech Recognition", ICASSP82, pp899-902, Paris, May 1982
- [25] B. Bregman, "Applying Matrix Quantization to Isolated Word Recognition", IEEE ICASSP85, pp23-26, Tampa, 1985
- [26] D.K. Burton, J.E. Shore, J.T. Buck, "Isolated-Word Speech Recognition Using Multisession Vector Quantization Codebooks", IEEE Trans. on ASSP, Vol. 33, N 4, August 1985
- [27] M.A. Bush, G.E. Kopec, "Network-Based Connected Digit Recognition", Trans. on ASSP, Vol. 33, N 10, October 1987
- [28] D. Chalmers, "A Transputer-Based Speech Recognition System", IEEE ICASSP89, Glasgow, 1989
- [29] Y.L. Chow, R.M. Schwartz, S. Roucos, O.A. Kimball, P. Price, G.F. Kubala, M.O. Dunham, M.A. Krasner, J. Makhoul, "The Role of Word-Dependent Correlatory Effects in a Phoneme-Based Speech Recognition System", IEEE ICASSP86, Tokyo, April 1986
- [30] Y.L. Chow, M.O. Dunham, O.A. Kimball, M.A. Krasner, G.F. Kubala, J. Makhoul, P.J. Price, S. Roucos, R.M. Schwartz, "BYBLOS: the BBN Continuous Speech Recognition System", IEEE ICASSP87, pp89-92, Dallas, 1987
- [31] R.A. Cole, A.L. Rudofsky, V. Zue, D.R. Reddy, "Speech as Patterns on Paper", in R.A. Cole, ed, Perception and Production of Fluent Speech, Lawrence Erlbaum Associates, Hillsdale, NJ, 1980
- [32] A.M. Colla, "Automatic Diphone Bootstrapping for Speaker-Adaptive Continuous Speech Recognition", IEEE ICASSP84, pp35.2.1-35.2.4, San Diego, 1984
- [33] J.W. Cooley, J.W. Tukey, "An algorithm for the Machine Calculation of the Complex Fourier Series", Math. Computation, Vol. 19, pp 297-301, 1963
- [34] M. Cravero, R. Pieraccini, F. Raineri, "Definition and Evaluation of Phonetic Units for Speech Recognition by Hidden Markov Models", IEEE ICASSP86, pp2235-2238, Tokyo, 1986
- [35] R. De Mori, L. Lam, "Plan Refinement in a Knowledge-Based System for Automatic Speech Recognition", IEEE ICASSP86, pp1217-1220, Tokyo, April 1986
- [36] A.M. Derosault, B. Mermelstein, "Natural Language Modeling for phoneme-to-text transcriptions", IEEE Trans. on PAMI, VOL 5, N 2, pp742-749, 1986
- [37] A.M. Derosault, "Context-Dependent Phonetic Markov Models for Large Vocabulary Speech Recognition", IEEE ICASSP87, pp360-363, Dallas, April 1987
- [38] P. Dumouchel, V. Gupta, M. Lennig, P. Mermelstein, "Three Probabilistic Language Models for a Large-Vocabulary Speech Recognizer", IEEE ICASSP88, pp513-516, 1988
- [39] D. Durol, "Evaluation of Speaker-Independent Isolated-Word Recognition Systems over Telephone Network", European Conference on Speech Technology, pp241-244, Edinburgh, September 1987
- [40] J.L. Elman, J.L. McClelland, "An Architecture for Parallel Processing in Speech Recognition: the TRACE Model", Bibliotheca phonet., N12, pp6-35, (Karger, Basel 1985)
- [41] J.L. Elman, D. Zipser, "Learning the Hidden Structure of Speech", UCSD Techn. Report, CS-8701, February 1987
- [42] Y. Ephraim, L.R. Rabiner, "On the Relations between Modeling Approaches for Information Sources", IEEE ICASSP88, pp24-27, New York, April 1988
- [43] M.W. Feng, F. Kubala, R. Schwartz, J. Makhoul, "Improved Speaker Adaptation using Text Dependent Spectral Mappings", IEEE ICASSP88, pp131-134, New York, 1988
- [44] J.D. Ferguson, "Variable Duration Models for Speech", Symposium on Applications of Hidden Markov Models to Text and Speech, pp143-179, Princeton, 1980
- [45] L. Fissore, G. Micca, R. Pieraccini, "Strategies for Lexical Access to Very Large Vocabularies", Speech Communication, Vol 7, N 4, December 1988
- [46] J.L. Gauvain, J. Mariani, J.S. Lénard, "On the Use of Time Compression for Word-Based Recognition", IEEE ICASSP83, pp1029-1032, Boston, 1983
- [47] J.L. Gauvain, J. Mariani, "Evaluation of Time Compression for Connected-Word Recognition", IEEE ICASSP84, pp35.12, San Diego, 1984
- [48] J.L. Gauvain, "A Syllable Based Isolated Word Recognition Experiment", IEEE ICASSP86, pp57-60, Tokyo, 1986
- [49] A. Gerstko, V. Cuperman, "Vector Quantization: A Pattern Matching Technique for Speech Coding", IEEE Communications Magazine, December 1983
- [50] O. Ghizita, "Auditory Neural Feedback as a Basis for Speech Processing", IEEE ICASSP88, pp 91-94, New York, 1988
- [51] S. Glimski, T. Mariano, D. Cassidy, T. Koh, C. Gervashi, G. Wilson, J. Kumar, "The Graph Search Machine (GSM): A Programmable Processor for Connected Word Speech Recognition and Other Applications", IEEE ICASSP87, pp519-522, Dallas, 1987
- [52] A.L. Gorin, B.R. Roe, "Parallel Level-Building on a Tree Machine", IEEE ICASSP88, pp295-298, New York, 1988
- [53] R.M. Gray, "Vector Quantization", IEEE ASSP Magazine Vol 1, N 2, April 1984
- [54] Y.F. Gong, J.P. Haton, "A Specialist Society for Continuous Speech Understanding", IEEE ICASSP88, pp627-630, New York, 1988
- [55] V.N. Gupta, M. Lennig, P. Mermelstein, "Integration of Acoustic Information in a Large Vocabulary Word Recognizer", IEEE ICASSP87, pp697-700, Dallas, 1987
- [56] J.P. Haton, N. Carbonnel, D. Fohr, J.P. Mari, A. Krouille, "Interaction between stochastic modeling and knowledge-based techniques in acoustic-phonetic decoding of speech", IEEE ICASSP87, pp668-671, Dallas, 1987
- [57] A.L. Higgins, "Speaker Recognition by Template Matching", pp273-276, New York, April 1986
- [58] W. Huang, R. Lippmann, "A Neural Net Approach for Speech Recognition", IEEE ICASSP88, pp99-102, New York, April 1988
- [59] M.J. Hunt, M. Lennig, P. Mermelstein, "Experiments in Syllable-Based Recognition of Continuous Speech", IEEE ICASSP80, pp680-683, Denver 1980
- [60] M.J. Hunt, C. Lefebvre, "Speaker Dependent and Independent Speech Recognition Experiments with an Auditory Model", IEEE ICASSP88, pp 91-94, New York, 1988

- [61] H. Ishizuka et al., "A Microprocessor for Speech Recognition", IEEE ICASSP'83, pp503-506, Boston, 1983.
- [62] F. Jelinek, "Continuous Speech Recognition by statistical methods", IEEE Proceedings, VOL 64 N 4, pp532-556, April 1976.
- [63] F. Jelinek, R. Mercer, "Interpolated Estimation of Markov Source Parameters from Sparse Data", in "Pattern Recognition in Practice", North-Holland, 1980.
- [64] F. Jelinek et al., "Experiments with the Tangora 20,000 word speech recognizer", IEEE ICASSP'87, pp701-704, Dallas, 1987.
- [65] D. Jovet, J. Monné, D. Dubois, "A New Network-Based Speaker-Independent Connected-Word Recognition System", IEEE ICASSP'86, pp1109-1112, Tokyo, 1986.
- [66] B.H. Juang, L.R. Rabiner, "Mixture Autoregressive HMM for Speech Signals", IEEE Trans on ASSP, Vol. 33, pp1404-1413, 1985.
- [67] R. Kavalier et al., "A Dynamic Time Warp IC for a One Thousand Word Recognition System", IEEE ICASSP'84, San Diego, March 1984.
- [68] O. Kinnell, P. Preis, S. Ruten, et al., "Recognition Performance and Grammar Constraints", Proceedings of the DARPA Recognition Workshop, pp33-59, February 1986.
- [69] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, "Optimization by Simulated Annealing", Science, 220, pp671-680, 1982-3, July 1983.
- [70] D.H. Klatz, "Overview of the ARPA-Speech Understanding Project", in W.A. Lea Editor, "New Trends in Speech Recognition", pp249-271, Prentice-Hall, 1980.
- [71] T. Kohonen, K. Mäkelä, T. Saarela, "Phonotopic Maps - Insightful Representation of Phonological Features for Speech Recognition", Proc. 7th ICPR, Montreal, pp182-185, July 30-August 2, 1984.
- [72] T. Kohonen, "Self-Organization and Associative Memory", Springer, 1984.
- [73] T. Kohonen, K. Torkola, M. Shozaki, J. Kangas, O. Ventä, "Microprocessor implementation of a large vocabulary speech recognizer and phonetic typewriter for Finnish and Japanese", European Conference on Speech Technology, pp377-380, Edinburgh, September 1981.
- [74] G.E. Kopec, M.A. Bush, "Network-Based Isolated Digit Recognition using Vector Quantization", Trans. on ASSP, Vol. 33, N 4, August 1985.
- [75] M. Kuhn, H. Ney, G. Tomasevich, "Fast Non Linear Time Alignment for Isolated Word Recognition", IEEE ICASSP'81, pp736-740, Atlanta, 1981.
- [76] W.A. Lea, Ed., "Trends in Speech Recognition", Prentice-Hall, 1980.
- [77] W.A. Lea, J.E. Hough, "Specific Contributions of the ARPA-SUR project, in W.A. Lea Editor, "New Trends in Speech Recognition", pp382-421, Prentice-Hall, 1980.
- [78] J. Lehoucq, D. Beroult, "Processing of Noisy Patterns with a Connectionist System using a Topographic Representation of Speech", European Conference on Speech Technology, pp191-194, Edinburgh, September 1987.
- [79] Y. Le Cun, "Une procédure d'apprentissage pour réseaux à seuil asymétrique", Proceedings Cognitive '85, pp59-604, Paris, 1985.
- [80] K.F. Lee, "Incremental Network Generation in Word Recognition", IEEE ICASSP'86, pp77-80, Tokyo, 1986.
- [81] K.F. Lee, H.W. Hon, "Speaker-Independent Phone Recognition Using Hidden Markov Models", CMU Research Report CS-88-121, March 1988.
- [82] K.F. Lee, H.W. Hon, "Large Vocabulary Speaker-Independent Continuous Speech Recognition", IEEE ICASSP'88, pp24-27, New York, April 1988.
- [83] K.F. Lee, "Large-Vocabulary Speaker-Independent Continuous Speech Recognition: the SPHINX system", CMU report, CMU-CS-88-148, April 1988.
- [84] C.H. Lee, F.K. Soong, B.H. Juang, "A Segment Linear Based Approach to Speech Recognition", IEEE ICASSP'88, pp501-504, New York, 1988.
- [85] J. LeRoux, J.P. Arrou-Vignod, M. Ivernizzi, "Correlation Data Classification for Low Bit Rate Speech Transmission and Real Time Word Recognition", Euisepe Conference, 1980.
- [86] S.E. Levinson, L.R. Rabiner, M.M. Sondhi, "An Introduction of the Application of the Theory of Probabilities Functions on a Markov Process to Automatic Speech Recognition", The Bell System Technical Journal, Vol. 62, N 4, April 1983.
- [87] S.E. Levinson, "Structural Methods in Automatic Speech Recognition", Proceedings of IEEE, Vol. 73, N 1, pp1625-1650, November 1985.
- [88] S.E. Levinson, "Continuously Variable Duration Hidden Markov Models for Automatic Speech Recognition", Computer Speech and Language, Vol. 1, pp29-45, 1986.
- [89] J.S. Liden, J. Mariani, G. Rezag, "Intelligibilité de phrases synthétiques altérées. Application à la transmission phonétique de la parole", 9th ICA, Madrid, 1977.
- [90] B. Lindblom, "On the Phonological Nature of Speech Processes", Speech Communication, Vol. 2, N 2-3, July 1983.
- [91] Y. Linde, A. Buzo, R.M. Gray, "An Algorithm for Vector Quantizer Design", IEEE Trans. on Communications, COM-28, pp84-95, January 1980.
- [92] R.P. Lippmann, "An Introduction to Computing with Neural Nets", IEEE ASSP Vol. 4, N 2, pp4-22, April 1986.
- [93] R.P. Lippmann, B. Gold, "Neural Classifiers Useful for Speech Recognition", 1st International Conference on Neural Networks, IEEE, June 1987.
- [94] R.P. Lippmann, "Neural Nets for Computing", IEEE ICASSP'88, pp1-6, New York, April 1988.
- [95] E.P. Loeb, R.F. Lyon, "Experiments in Isolated Digit Recognition with a Cochlear Model", IEEE ICASSP'87, pp 1131-1134, Dallas, 1987.
- [96] J.L. McClelland, J.L. Elman, "Interactive Processes in Speech Perception: the TRACE model, in "Parallel Distributed Processing: Exploration in the Microstructure of Cognition, Vol 1: Foundation", D.E. Rumelhart and J.L. McClelland Eds, MIT Press, 1986.
- [97] J. Makhoul, "Vector Quantization in Speech Coding", Proceedings of the IEEE, Vol. 73, N 11, November 1985.
- [98] J. Mariani, "Reconnaissance de la parole continue par diphtongues", in "Processus d'Encodage et de Décodage Phonétiques", Symposium Gail-Greco, Toulouse, 1981.
- [99] J. Mariani, "Hamlet: A Prototype of a Voice-Activated Typewriter", European Conference on Speech Technology, Edinburgh, pp222-225, September 1987.
- [100] J. Mariani, "Study of the IEEE ICASSP Conference", LIMSI Research Report, to be published (in French), 1989.
- [101] J.D. Markel, A.J. Gray, "Linear Prediction of Speech", Springer Verlag, 1976.
- [102] K. Matrouf, F. Nél, J.L. Gauvain, J. Mariani, "Adaptive-Syntax Representation in an oral task-oriented dialogue for air-traffic controller training", submitted Eurospeech'89, Paris, September 1989.
- [103] B. Méraldo, "Speech Recognition with Very Large Size Dictionary", IEEE ICASSP'87, pp364-367, Dallas, April 1987.
- [104] B. Méraldo, "Phonetic Recognition using Hidden Markov Models and Maximum Mutual Information Training", IEEE ICASSP'88, pp111-114, New York, 1988.
- [105] M. Minsky, S. Papert, "Perceptrons: An Introduction to Computational Geometry", MIT Press, 1969.
- [106] H. Murveit, M. Weintraub, "1000-Word Speaker Independent Continuous-Speech Recognition using Hidden Markov Models", IEEE ICASSP'86, pp115-118, New York, 1986.
- [107] H. Murveit, "Design for a Real-Time Large-Vocabulary Continuous-Speech Recognition System", DARPA Review Meeting, Pittsburgh, June 1988.
- [108] C.S. Myers, L.R. Rabiner, "A Level Building Dynamic Time Warping Algorithm for Connected Word Recognition", Trans. ASSP, Vol. 29, N 2, pp284-297, April 1981.
- [109] H. Na, S. Glinel, "Neural Net Based Pattern Recognition on the Graph Search Machine", IEEE ICASSP'88, pp2168-2171, New York, 1988.
- [110] A. Nadas, "On Turing's Formula for Word Probabilities", IEEE Trans. on ASSP, Vol. 33, N 6, December 1985.
- [111] A. Nadas, D. Nahamoo, M.A. Pichney, "On a Model-Robust Training Method for Speech Recognition", Trans. ASSP, Vol. 36, N 9, pp 1432-1436, September 1988.
- [112] S.Y. Nakajima, H. Hamada, "Automatic Generation of Synthesis Units Based on Context Oriented Clustering", IEEE ICASSP'88, pp659-662, New York, 1988.
- [113] M. Nakamura, K. Shikano, "A Study of English Word Category Prediction Based on Neural Networks", TR Technical Report, TR-852, November 1988.
- [114] A. Newell et al., "Speech Understanding Systems: Final Report of a Study Group", North-Holland, 1973.
- [115] H. Ney, "The Use of a One-Stage Dynamic Programming Algorithm for Connected Word Recognition", Trans. on ASSP, Vol. 32, N 2, pp263-271, April 1984.
- [116] H. Ney, A. Noll, "Phoneme Modeling Using Continuous Mixture Densities", IEEE ICASSP'88, pp374-400, New York, April 1988.
- [117] A.V. Oppenheim, R.W. Schaffer, "Digital Signal Processing", Prentice-Hall, 1975.
- [118] D.S. Paley, "Selected Test Material, Test and Scoring Procedures for the May-June 1988 DARPA Benchmark Test", DARPA Review Meeting, Pittsburgh, June 1988.
- [119] D.B. Parker, "Learning Logic", Techn. Report, TR-47, csrcms, MIT, April 1985.
- [120] D.B. Paul, R.P. Lippmann, R.P. Chen, C. Weinstein, "Robust HMM-Based Techniques for Recognition of Speech Produced under Stress and in Noise", Speech Tech 1986, New York, April 1986.
- [121] D.B. Paul, E.A. Martin, "Speaker Stress-Resistant Continuous Speech Recognition", IEEE ICASSP'88, pp283-286, New York, 1988.
- [122] D. Paul, "MIT Lincoln Lab Presentation at the DARPA Strategic Computing Speech Recognition Program Meeting", Pittsburgh, June 1988.
- [123] S.M. Peeling, R.K. Moore, "Isolated Digit Recognition Experiments Using the Multi-Layer Perceptron", Speech Communication, Vol. 7, N 4, pp403-410, December 1988.
- [124] A.B. Poritz, "Linear Prediction HMM and the Speech Signal", IEEE ICASSP'82, pp1291-1294, Paris, 1982.
- [125] A.B. Poritz, "Hidden Markov Models: A Guided Tour", IEEE ICASSP'88, pp7-13, New York, 1988.
- [126] R.W. Prager, T.D. Harrison, F. Fallside, "Boltzmann Machines for Speech Recognition", Computer Speech and Language, Vol. 1, N 1, March 1986.
- [127] R.W. Prager, F. Fallside, "A Comparison of the Boltzmann Machine and the Back Propagation Network as Recognizers of Static Speech Patterns", Computer Speech and Language, Vol. 2, N 3/4, September/December 1987.
- [128] G. Quénot, J.L. Gauvain, J.J. Gangoff, J. Mariani, "A Dynamic Time Warp Processor for Continuous Speech Recognition", IEEE ICASSP'86, pp1549-1552, Tokyo, 1986.
- [129] L.R. Rabiner, R. Schafer, "Digital Processing of Speech Signals", Prentice Hall, 1979.
- [130] L.R. Rabiner, S.C. Levinson, A.E. Rosenberg, J.C. Wilpon, "Speaker Independent Recognition of Isolated Word Using Clustering Techniques", IEEE Trans. on ASSP, Vol. 27, August 1979.
- [131] L.R. Rabiner, J.G. Wilpon, "A Simplified Robust Training Procedure for Speaker Trained Isolated Word Recognition", IJASA, Vol. 68, N 5, pp1271-1276, November 1980.
- [132] L.R. Rabiner, A. Bergh, J.G. Wilpon, "An Embedded Word Training Procedure for Connected Digit Recognition", IEEE ICASSP'82, pp1621-1624, Paris, May 1982.
- [133] L.R. Rabiner, S.E. Levinson, "A Speaker-Independent Syntax-Directed Connected Word Recognition System based on Hidden Markov Model and Level-Building", IEEE Trans. on ASSP, Vol. 33, N 3, pp561-573, June 1985.
- [134] L.R. Rabiner, B.H. Juang, S.E. Levinson, M.M. Sondhi, "Recognition of Isolated Digit using Hidden Markov Models with Continuous Mixture Densities", AT&T Techn. Journal, Vol. 64, N 6, pp1211-1233, July 1985.
- [135] L.R. Rabiner, B.H. Juang, "An Introduction to Hidden Markov Models", IEEE ASSP Magazine, Vol. 3, N 1, pp16-19, January 1986.
- [136] L.R. Rabiner, J.G. Wilpon, F.K. Soong, "High Performance Connected digit Recognition using Hidden Markov Models", IEEE ICASSP'88, New York, April 1988.
- [137] M. R. Richter, "Modeling of Continuous Speech Observations", Proceedings "Advances in Speech Processing", Conference, IBM Europe Institute, 1986.
- [138] J.R. Rohlfach, Y.L. Chow, S. Roucos, "Statistical Language Modeling using a Small Corpus from an Application Domain", IEEE ICASSP'88, pp267-270, New York, 1988.
- [139] A.E. Rosenberg, L.R. Rabiner, S.E. Levinson, J.G. Wilpon, "A Preliminary Study on the Use of Demi-Syllables in Automatic Speech Recognition", IEEE ICASSP'81, pp967-970, Paris, 1981.
- [140] R. Rosenblatt, "Principles of Neurodynamics", Spartan Books, New York, 1959.
- [141] B. Roth, "Real Time Implementations on a PC", DARPA project review, Pittsburgh, June 1988.
- [142] S. Roucos, R. Schwartz, J. Makhoul, "Segment Quantization for Very Low Rate Speech Coding", IEEE ICASSP'82, pp1563-1568, Paris, 1982.
- [143] D.E. Rumelhart, J.K. McClelland, R.J. Williams, "Learning Internal Representations by Error Propagation", in "Parallel Distributed Processing: Exploration in the Microstructure of Cognition, Vol 1: Foundation", D.E. Rumelhart and J.L. McClelland Eds, MIT Press, 1986.
- [144] G. Ruske, T. Schotola, "The Efficiency of Demi-Syllable segmentation in the Recognition of Spoken Words", IEEE ICASSP'81, pp971-974, Florida, 1981.
- [145] J.T. Russell, K. Moore, "Explicit Modeling of State Occupancy in Hidden Markov Models for Automatic Speech Recognition", IEEE ICASSP'85, pp-8, Tampa, March 1985.
- [146] H. Sakoe, "Two-Level DP Matching - A Dynamic Programming Based Pattern Matching Algorithm for Connected Word Recognition", IEEE Trans. on ASSP, Vol. 27, N 6, pp 588-595, Dec 1979.
- [147] H. Sakoe, R. Isotani, K. Yoshida, K. Ito, T. Watanabe, "Speaker-Independent Word Recognition using Dynamic Programming Neural Networks", IEEE ICASSP'89, Glasgow, May 1989.
- [148] C. Scagliola, "The Use of Diphtones as Basic Units in Speech Recognition", 4th FASE Symposium on Acoustics and Speech, Venice, April 1981.
- [149] R.M. Schwartz, J. Kivestadi, J. Makhoul, "A Preliminary Study of a Phonetic Vocoder Based on a Diphtone Model", IEEE ICASSP'80, pp32-35, Denver, 1980.
- [150] R.M. Schwartz, Y.L. Chow, S. Roucos, M. Kraemer, J. Makhoul, "Improved Hidden Markov Modeling for Connected Word Recognition", IEEE ICASSP'84, San Diego, April 1984.
- [151] R. Schwartz, Y. Chow, P. Kubala, "Rapid Speaker Adaptation using a Probabilistic Self-Mapping", IEEE ICASSP'87, pp533-536, Dallas, 1987.
- [152] D. Schinke, "Speaker Independent Recognition Applied to Telephone Access Information Systems", Speech Tech'86, pp32-33, New York, April 1986.
- [153] J.T. Sejnowski, C.R. Rosenberg, "NEATalk: A Parallel Network that Learns to Read Aloud", Technical Report, Johns Hopkins University, EEC-86/01, June 1986.
- [154] G. Shihman et al., "An IBM PC Based Large-Vocabulary Isolated Utterance Speech Recognizer", IEEE ICASSP'86, pp33-36, Tokyo, 1986.
- [155] K. Shitano, K.F. Lee, R. Reddy, "Speaker Adaptation through Vector Quantization", IEEE ICASSP'86, pp2643-2646, Tokyo, 1986.
- [156] J.E. Shore, D.K. Burton, "Discrete Utterance Speech Recognition without Time Alignment", IEEE Trans. on Information Theory, Vol. 29, pp475-491, July 1983.
- [157] M.R. Schroeder, "Predictive Coding of Speech: Historical Review and Directions for Future Research", IEEE ICASSP'86, pp 3157-3164, Tokyo, April 1986.
- [158] R. Sinar Jr., "TMS320C26 Texas Instruments' Family of Digital Signal Processor", Proceedings Speech Tech'87, New York, April 1987.
- [159] H. Singer, J.L. Gauvain, "Connected Speech Recognition using Dissolvable Segmentation", Proceedings of the Acoustical Society of Japan Conference, October 1988.
- [160] G.S. Slutsker, "Nonlinear Method Analiza Recevch Signalov", Trudy NII, N 2, 1968.
- [161] F.K. Soong, A.E. Rosenberg, "On the Use of Instantaneous and Transitional Spectral Information in Speaker Recognition", IEEE ICASSP'86, pp874-880, Tokyo, 1986.
- [162] F.E. Stern, M. Ekenazi, D. Memmi, "An expert system for speech spectrogram reading", IEEE ICASSP'86, pp1193-1196, Tokyo, April 1986.
- [163] N. Supamura, K. Shikano, S. Furui, "Isolated Word Recognition Using Phoneme-Like Templates", IEEE ICASSP'83, pp723-726, Boston, 1983.
- [164] J. Thomas, "DSP56000, A High Performance DSP Architecture", Proceedings Speech Tech'87, New York, April 1987.
- [165] T.K. Vinturik, "Recognition of Words of Oral Speech by Dynamic Programming", Kibernetika, Vol. 81, N 8, 1968.
- [166] A.J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm", IEEE Trans. on Information Theory, Vol. 13, N 2, pp260-269, April 1967.
- [167] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, K. Lang, "Phoneme Recognition using Time-Delay Neural Networks", ATR Research Report, TR-10006, October 1987.
- [168] T. Watanabe, "Segmentation-free Syllable Recognition in Continuously Spoken Japanese", IEEE ICASSP'83, pp320-323, Boston, 1983.
- [169] P. Werbos, "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences", PhD Thesis, Harvard, August 1974.
- [170] J. Witten, "Data Compression Through Statistical Modeling: A New Approach to Multi-Scale Description", IEEE ICASSP'84, San Diego, April 1984.
- [171] R. H. Wu, K. Ganesan, "Comparative Study of Algorithms for VQ Design Using Conventional (LBO) and Neural-Net Based Approaches", IEEE ICASSP'89, Glasgow, May 1989.
- [172] S.R. Young, W.H. Ward, "Towards Habitable Systems: Use of World Knowledge to Dynamically Constrain Speech Recognition", Proceedings 2nd Symposium on Advanced Man-Machine Interface through Spoken Language, pp30-32, Hawaii, November 1988.
- [173] S.J. Young, N.H. Russell, J.H.S. Thornton, "Speech Recognition in VODIS II", IEEE ICASSP'86, pp441-444, New York, April 1986.
- [174] V.W. Zue, L.F. Lamel, "An Expert Spectrogram reader: a Knowledge-Based Approach to Speech Recognition", IEEE ICASSP'86, pp1197-1200, Tokyo, April 1986.
- [175] V. Zue, "Recent Speech Recognition Results at MIT", DARPA Review Meeting, Pittsburgh, June 1988.