Lexical Access to Large Vocabularies for Speech Recognition

LUCIANO FISSORE, PIETRO LAFACE, GIORGIO MICCA, AND ROBERTO PIERACCINI

Abstract—**A large vocabulary isolated word recognition system based on the hypothesize-and-test paradigm is described. The system has been, however, devised as a word hypothesizer for a continuous speech understanding system able to answer to queries put to a geographical database. Word preselection is achieved by segmenting and classifying the input signal in terms of broad phonetic classes. Due to low redundancy of this phonetic code for lexical access, to achieve high performance, a lattice of phonetic segments is generated, rather than a single sequence of hypotheses. It can be organized as a graph, and word hypothesization is obtained by matching this graph against the models of all vocabulary words. A word model is itself a phonetic representation made in terms of a graph accounting for deletion, substitution, and insertion errors. A modified Dynamic Programming (DP) matching procedure gives an efficient solution to this graph-to-graph matching problem. Hidden Markov Models (HMM's) of subword units are used as a more detailed knowledge in the verification step. The word candidates generated by the previous step are represented as sequences of diphone-like subword units, and the Viterbi algorithm is used for evaluating their likelihood. To reduce storage and computational costs, lexical knowledge is organized in a tree structure where the initial common subsequences of word descriptions are shared, and a beam-search strategy carries on the most promising paths only. The results show that a complexity reduction of about 73 percent can be achieved by using the two pass approach with respect to the direct approach, while the recognition accuracy remains comparable.**

I. Introduction

SPEECH recognition technology is steadily growing toward its maturity. Its growth is supported not only by continuous research in all aspects of speech science, but also by the impressive advances in microelectronics that have made possible the real-time implementation of complex systems by offering general purpose digital signal processors, powerful dedicated processors, and custom VLSI circuits [37].

Significant results have been achieved in research projects by using pattern recognition and stochastic modeling methods [28]. Following these paradigms, several commercial products have been developed and marketed that perform very well for simple tasks and in constrained conditions (single speaker, limited vocabulary, isolated words) [4], [33]. Nevertheless, several difficult tasks and applications still exist in automatic speech recognition that need further research and engineering efforts to achieve systems that are really useful and widely acceptable by the end users.

While natural language continuous speech recognition seems to be a long term goal, some less ambitious tasks are currently investigated that address relevant problems such as speaker independence, telephone bandwidth speech quality, robustness in noisy environment, and access to very large vocabularies. Office dictation systems, and information access with large vocabulary over the telephone line, are emerging as realistic and useful applications. Both applications share the need of quickly accessing large vocabularies of several thousand words, a difficult task even for speaker dependent systems.

As the number of words to be discriminated is large, it is not practically feasible to collect thousands of templates, thus it is mandatory that lexical knowledge is built from a phonetic transcription of the orthographic form of the words. To this aim, subword recognition units must be defined that can be trained from a reasonably small size learning vocabulary and used as building blocks for the words of any lexicon. Furthermore, in order to reduce the computational complexity of the pattern matching process, the search for the best matching words must be as far as possible focused. The reduction of the searching space can be obtained by carefully exploiting the structural constraints that a lexicon imposes at the phonologic level [1], [31], [38], [42] by using the hypothesize-and-test paradigm. First, a vocabulary subset to which the utterance is estimated to belong to is hypothesized on the basis of a description that allows a fast search to be performed. Second, a more detailed and time consuming verification process is activated only for words belonging to that subset [19], [21], [23], [24], [32]. Different approaches can be used in the preselection step. The search can be carried out for all words in the vocabulary through a very simple and approximate description designed on the basis of heuristic knowledge [23] or by assuming that the observed label frequencies have Poisson distributions [2]. As these kinds of approaches rely on the detection of word boundaries, they cannot be directly applied to continuous speech.

A less heuristic method is reminiscent of perceptual models of word recognition such as those introduced in the Cohort Theory and in the Phonetic Refinement Theory [38]. It avoids matching all words by characterizing each

Manuscript received January 30, 1988; revised November 2, 1988.

L. Fissore, G. Micca, and R. Pieraccini are with the CSELT—Centro Studi e Laboratori Telecomunicazioni, Via G. Reiss Romoli 274, 10148 Torino, Italy.

P. Laface was with the Dipartimento di Automatica e Informatica, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy. He is now with the Dipartimento di Informatica ed Applicazioni, Universitá di Salerno, 840181 Baronissi, Salerno, Italy.

IEEE Log Number 8928734.

0096-3518/89/0800-1197$01.00 © 1989 IEEE

lexical entry by means of a partial phonetic description, so that acoustically similar words are clustered together [20], [27], [31], [45]. From the automatic recognition point of view, this is important because broad phonetic classes can be hypothesized more reliably than detailed phonetic segments. The effectiveness of the latter approach, in terms of preselection capability, has been evaluated by examining the statistical properties of large vocabularies under the assumption of a correct partial description of the words [6], [12], [41], [46]. For instance, as far as Italian language is concerned, describing a 13 747 word vocabulary by using only 6 broad phonetic classes, 7225 words can be uniquely identified, while the maximum and average size of the subset of words bearing the same description is 34 and 1.5, respectively [18]. The results of these statistical analyses, however, do not take into account segmentation and classification errors. These errors depend on the inherent variability in speech and occur even if the acoustic–phonetic module must discriminate among a limited number of gross phonetic categories. Moreover, lexicon specifications made on the basis of a reduced set of symbols can lead to small redundancy, that is, a small distortion occurring on a string of symbols that corresponds to a set of words is likely to perfectly fit the representation of a different set of words. Lexical access must be performed, therefore, through error correcting procedures that face the problem of high confusability of partial descriptions of words by generating a suitable set of likely candidates. Although word subsets larger than those predicted by an error free analysis are hypothesized, the results of several experiments, referring to different languages [5], [20], [26], [32], [41], [45], show the substantial preselection capability of the method even in the presence of classification errors. This paper is devoted to the description of an isolated word recognition system based on this hypothesize-and-test paradigm. The system has been, however, devised as a word hypothesizer, producing a lattice of lexical items, for a continuous speech understanding system able to answer to queries put to a geographical database [25], [16]. Strategies and results in continuous speech will not be addressed in the following as they are presented elsewhere [15], [14]. Envisaged applications are a voice activated directory querying system and a phonetic typewriter [7].

Words are preselected by segmenting and classifying the input signal in terms of broad phonetic classes. To achieve high performance, a lattice of phonetic segments is generated, rather than a single sequence of hypotheses. It can be organized as a graph in a structure referred to as "micro-segmentation." Words are hypothesized by matching the micro-segmentation graph against the models of all vocabulary words. A model is a phonetic representation of a word in terms of a graph accounting for deletion, substitution, and insertion errors. A modified Dynamic Programming (DP) matching procedure gives an efficient solution to this graph-to-graph matching problem.

Hidden Markov Models (HMM's) of subword units are the basis of a more detailed knowledge in the verification step. The word candidates generated by the previous step are represented as sequences of diphone-like subword units, and the Viterbi algorithm evaluates their likelihood by observing sequences of labels, associated to each centisecond of the input signal, obtained by vector quantization of 18 cepstral parameters.

To reduce storage and computational costs, lexical knowledge is organized in a tree structure where the initial common subsequences of word descriptions are shared, and a beam-search strategy carries on the most promising paths only.

This strategy of lexical access has been applied to vocabularies of different size and complexity. Large experimentation has been possible because all models can be trained without hand labeling or segmentation allowing a ready adaptation to new vocabularies and to new speakers.

The paper is organized in six sections. Section II gives an overview of the modules that compose the system. Section III is devoted to acoustic–phonetic classification and to word representation. The Dynamic Programming algorithm that solves the problem of matching two graphs is introduced in Section IV along with some definitions of the local matching costs. The lexical access strategy and the results of several experiments for assessing its performance are presented in Section V. Finally, Section VI illustrates the detailed verification module and its performance.

II. System Overview

The modules and the knowledge bases involved in the training and in the recognition process are shown in Figs. 1 and 2. A short introduction of their functions and relationship is given in the following.

Training the system from scratch requires the following four steps.

- *Codebook Generation:* The Feature Extraction module performs a Mel-based cepstral analysis of the signal. The signal is collected through a head-mounted microphone, low-pass filtered at 6 kHz, and sampled at a 12 kHz rate. An FFT analysis is performed each 10 ms frame, over 20 ms overlapping Hamming windows. At each frame, a cosine transform is applied that produces a vector of 18 cepstral coefficients. A simple endpoint detector extracts the portion of the signal corresponding to the uttered words on the basis of the energy of the frames. A fixed amount of the initial and trailing silence is kept to prevent occasional deletion of initial and final weak consonants. The Vector Quantization (VQ) module associates to every speech frame a label belonging to a finite alphabet of acoustic symbols (codebook), these symbols are used as an observation sequence by the HMM's training and verification modules. The VQ codebook is generated using the LBG clustering algorithm [30]. All ex-

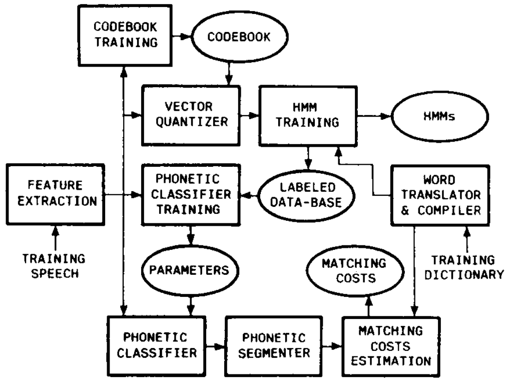


Fig. 1. Modules active in the training phase.

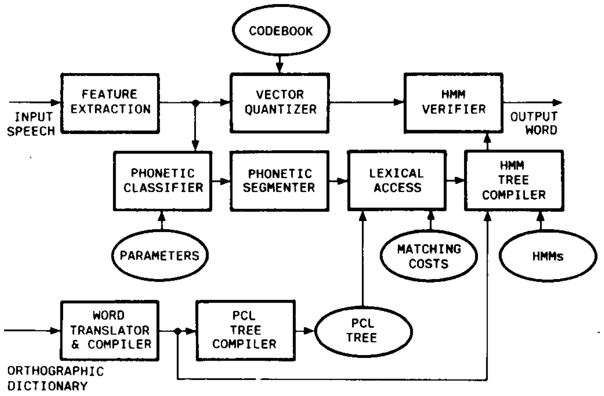


Fig. 2. Modules active in the recognition phase.

periments were performed using 7 bit speaker dependent codebooks (128 codewords).

- *Subword Units Training:* The Word Translator rewrites, according to a set of phonologic rules, the orthographic description of each word into a sequence of subword recognition units. This sequence is then compiled, by the Word Compiler module, into its corresponding HMM chain that is trained through the Forward–Backward algorithm [21]. The transition and emission probabilities of each subword unit model are obtained by processing all words of a properly designed training vocabulary. Trained units can be used as building blocks of the words of any vocabulary, and the Viterbi algorithm can estimate the likelihood that a given utterance corresponds to a word in the vocabulary. It is worth noting that all these procedures do not need labeled speech nor human interaction. On the contrary, an important byproduct of stochastic modeling of subword units is that a speech database can be automatically segmented and labeled. In fact, once the models are trained, the Viterbi algorithm can estimate the best path through the states of the HMM chain corresponding to a known utterance, and the boundaries of the units composing the word or the sentence can be detected by a traceback procedure.

- *Phonetic Classifier Training:* This module computes, from a previously labeled speech database, the parameters of the frame-by-frame Phonetic Classifier. The Phonetic Classifier estimates the likelihood that a cepstral vector belongs to a set of broad phonetic classes.

- *Estimation of Phonetic Segments Matching Costs:* Adjacent frames with the same phonetic label are collapsed into segments by the Phonetic Segmentation module. A statistical estimation procedure generates the costs for the substitution, insertion, and deletion of segments (matching costs). As will be detailed in Section III, this module describes an utterance in terms of a lattice of phonetic hypotheses rather than by a single sequence of segments.

Fig. 2 shows the overall architecture of the isolated word recognition system. A Word Tree Compiler allows lexical knowledge to be represented in a compact form by merging subword unit sequences into a tree, a convenient structure for reducing both storage and computation costs. Two representations are produced by this module: the first one is the Phonetic Class Tree (PCL Tree), whose nodes represent phonetic class labels; the second one is the HMM Tree. The lattice generated by the Phonetic Segmentation module is matched against the PCL Tree by the Lexical Access module that selects a reduced set of word candidates. These words are then represented by the HMM Tree, and the HMM verifier module evaluates the most likely candidate in the set through a beam search Viterbi algorithm.

III. Phonetic Segmentation

Phonetic segmentation is performed by two modules that work in sequence: a frame-by-frame phonetic classifier and a phonetic segmenter.

A. Phonetic Classification

The frame-by-frame labeler estimates, by means of a hierarchical cubic polynomial classifier, the likelihood that a cepstral vector belongs to the phonetic classes described by the following symbols:

$k_1 = pl$: silence or plosive consonant

$k_2 = fr$: fricative consonant

$k_3 = ln$: liquid or nasal consonant

$k_4 = fv$: front vowel

$k_5 = cv$: central vowel

$k_6 = bv$: back vowel.

This set of labels will be referred to in the following as "classification alphabet." It has been chosen as a result of a preliminary study on the discrimination of words in a large Italian lexicon by partial descriptions [25]. These phonetic features are simple enough to be extracted reliably but, at the same time, they carry sufficient information to reduce the set of words that are described by the same sequence of symbols to a reasonable size. It is worth noting that this alphabet is very close to the classification schemes proposed for lexical access of the Italian language [25] as well as of other languages [20], [45], [41] on the basis of different analyses. Similar categories, in particular, have been proposed on a linguistic basis in the pioneering work of Shipman and Zue [42]. Even more interesting, however, is the consideration that similar broad phonetic classes are produced as a result of automatic clustering of phonemes using several different statistical methods. Consider, for example, the results of Poritz's experiment on a 5-state HMM cited in [28], and classes obtained through different optimization criteria such as the maximization of the mutual information or transinformation [41]. Moreover, phonemes can be clustered into classes on the basis of the distance between phoneme HMM's [44], between cepstral parameters [35], or between more complex feature vectors [12], [34], confirming that the above-mentioned classes can be reliably discriminated. For each 10 ms speech frame, 18 Mel-based cepstral parameters ($c_0, c_1, \cdots, c_{17}$) are computed. All of them are used for Vector Quantization in order to reduce the codebook distortion, 13 coefficients are sufficient for the *verification* step, while the components of the primary pattern vector $\boldsymbol{x}$ used for *classification* are only the coefficients c_1 to c_9 and the total energy of the frame. The ideal classification of a given frame can be described by a target vector: $z = [z_1, z_2, \cdots, z_6]$ where

$$z_i = 1 \quad \text{if the frame belongs to the } i\text{th class}$$

$$z_j = 0 \quad \text{if } j \neq i.$$

The classifier gives an estimation $\boldsymbol{d} = [d_1, d_2, \cdots, d_6]$ of target vector z by using a cubic function of vector $\boldsymbol{x}: \boldsymbol{d} = K(\boldsymbol{x})$. The classifier assigns to each input frame the class k_j corresponding to the highest value component d_j of the estimation vector $\boldsymbol{d}$ [22]. Uncertainty and reject regions are also considered in the $\boldsymbol{d}$ space. If the estimated vector $\boldsymbol{d}$ falls in the neighborhoods of the nearest target vector, a single label is assigned to the analyzed frame; if its distance from two target vectors is within a given threshold, two labels are assigned, otherwise no decision is drawn.

A set of 1105 isolated Italian words (TRA dictionary), pronounced by 5 male and 2 female speakers, was collected for training the frame-by-frame classifier. These 7735 utterances were automatically labeled in terms of phonetic units as will be described in the Section VI, where training of HMM's is illustrated, and used for estimating the parameters of 7 speaker dependent classifiers.

Another set of 1011 words, belonging to the dictionary of the geographic database query application (GEO), was recorded by the same speakers and all tests were performed on this set of 7077 utterances. The classifier performance, averaged among the speakers, given in terms of percentage of frames assigned to the six phonetic classes, is summarized in the class-to-class confusion matrix of Tables I and II. Table I shows the results considering the best first decision only, while Table II considers also the possible alternative decision. A total error rate of 14 and 6.5 percent, respectively, is obtained.

B. Phonetic Segmentation

Adjacent frames that are labeled by the same single symbol are collapsed into a micro-segment. This procedure is also applied to adjacent frames that are labeled by two symbols which are the same. For a given utterance, a two level lattice of coarse phonetic micro-segments is obtained. An example of phonetic lattice is shown in Fig. 3(a), where black segments represent first decision symbol frames, while gray ones represent alternative decision symbol frames; the phonetic class symbol corresponding to a segment can be read on the left-hand side of the figure. For the sake of clarity, the micro-segmentation of the figure is that obtained as a result of the application of the majority voting filter that will be introduced in Section V-B.

Micro-segmentation can be represented, therefore, by a list of elements:

$$M(t) = (b^t, e^t, s_1^t, a_1^t, s_2^t, a_2^t); \qquad t = 1, \cdots, T \quad (1)$$

where b^t and e^t are the beginning and ending frames of the micro-segment, s_1^t and s_2^t are its first and second phonetic labels, and a_1^t and a_2^t are its classification reliabilities. Of course, s_2^t and a_2^t are missing whenever a single hypothesis is produced. The classification reliabilities a_1^t and a_2^t of $M(t)$ are defined as

$$a_i^t = \sum_{k=b^t}^{e^t} d_{s_i^t}^k; \qquad i = 1, 2 \quad (2)$$

where $d_{s_i^t}^k$ is the component related to label s_i^t of estimation vector $\boldsymbol{d}$ of the kth frame. The graph corresponding to the phonetic lattice of Fig. 3(a) is shown in Fig. 3(b).

C. Word Representation

Each word of the lexicon can be automatically translated, by means of a set of context sensitive rules, from its orthographic form into a number of possible phonemic transcriptions taking into account the main speaker variations. From the phonemic forms, a set of phonetic representations of the words with different degrees of detail can be derived. The choice of a representation alphabet depends on a tradeoff between the speedup of the lexical search due to the introduction of equivalent phonetic classes and the confusability given by a less detailed phonetic knowledge. For example, phonemes /s/ and /v/ are both fricatives, but strong fricatives like /s/ are very likely to be correctly classified as fricative consonants, while weak fricatives like /v/ are quite often classified as liquid/

TABLE I
CLASS-TO-CLASS CONFUSION MATRIX, BEST FIRST DECISION

Test	Number of Frames	pl	fr	ln	fv	cv	bv	Rejection	Error Rate
pl	173 853	86.5	4.7	3.8	1.6	1.7	1.7	0.0	13.5
fr	72 875	3.2	84.3	8.1	2.6	0.4	1.4	0.0	15.7
ln	88 473	1.2	1.9	83.4	7.6	2.9	3.0	0.1	16.7
fv	140 065	0.6	2.0	6.4	89.9	0.8	0.2	0.1	10.1
cv	94 987	0.8	1.0	3.3	1.7	91.1	2.1	0.0	8.9
bv	121 844	2.1	4.4	10.2	0.4	1.7	81.1	0.1	18.9

TABLE II
CLASS-TO-CLASS CONFUSION MATRIX, FIRST TWO BEST DECISIONS

Test	Number of Frames	pl	fr	ln	fv	cv	bv	Rejection	Error Rate
pl	173 853	94.9	1.4	1.7	0.8	0.7	0.4	0.1	5.1
fr	72 875	1.3	90.2	5.5	1.5	0.2	1.3	0.0	9.8
ln	88 473	0.5	0.9	93.4	2.8	1.3	1.1	0.0	6.6
fv	140 065	0.4	1.0	1.7	96.5	0.3	0.1	0.0	3.5
cv	94 987	0.4	0.5	1.6	0.8	95.7	0.9	0.1	4.3
bv	121 844	1.0	2.5	5.6	0.3	0.5	90.0	0.1	10.0

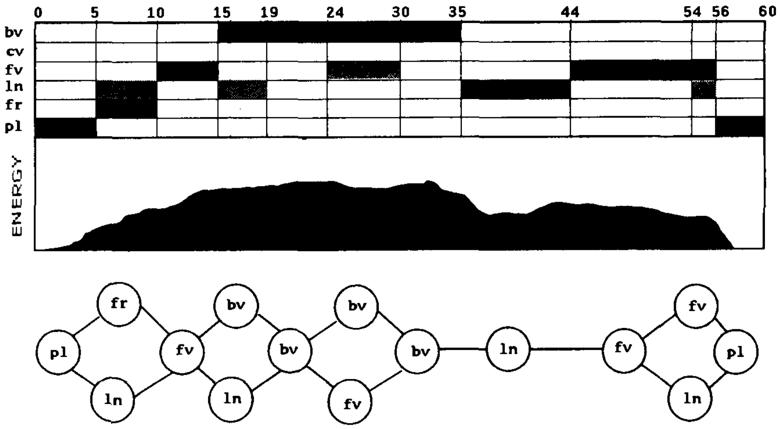


Fig. 3. A phonetic lattice.

nasals. It is possible to better account for these classification errors by representing words in terms of more detailed classes, but this advantage must be traded with an increase of the lexical search space. A compromise has been established by evaluating the results of a set of experiments, described in Section V, using the following three representation alphabets.

- A_1, described by the following 12 phonetic classes:

Consonants		Vowels	
h_1 = Spl	: plosive	h_7 = I	: unstressed front
h_2 = Lpl	: silence or geminate plosive	h_8 = II	: stressed front
h_3 = Wfr	: weak fricative	h_9 = A	: unstressed central
h_4 = Sfr	: strong fricative	h_{10} = AA	: stressed central
h_5 = Wln	: weak liquid or nasal	h_{11} = U	: unstressed back
h_6 = Sln	: strong liquid or nasal	h_{12} = UU	: stressed back

where each symbol of the classification alphabet splits into two different representation labels accounting for the difference between stressed and unstressed vowels and between strong and weak consonants;

- A_2, an alphabet of 9 classes, where the distinction between stressed and unstressed vowels has been eliminated; and
- A_3, the same alphabet used for classification (6 classes).

As an example, the Italian word FIUME (river), whose standard phonemic transcription is /fjúme/, is represented by the following strings of symbols, depending on the description alphabet:

A_1: Wfr	I	UU	Wln	I
A_2: Wfr	I	U	Wln	I
A_3: fr	fv	bv	ln	fv

The representation of a word, in terms of the symbols of a description alphabet, will be referred to as

$$W = w^1 w^2 \cdots w^M \tag{3}$$

where M is the length of the representation.

IV. Three-Dimensional DP Matching

A word representation which takes into account misclassifications can be modeled by a graph such as the one shown in Fig. 4, where the symbols of alphabet A_1 are used and each link is associated to a cost $C(op(h_i, k_j))$ corresponding to the alignment operations $op(h_i, k_j)$ below:

$sub(h_i, k_j)$: substitution of test symbol k_j for reference symbol h_i
$ins(h_i, k_j)$: insertion of test symbol k_j after reference symbol h_i
$del(h_i)$: deletion of reference symbol h_i.

The problem of finding the best matching of a reference word model against a test micro-segmentation can be stated as follows.

- Select one path in word description and one in micro-segmentation; each path corresponds to a string of symbols belonging to the representation and to the classification alphabet, respectively.
- Compute the best alignment cost between these strings by using the costs defined in Section IV.
- Repeat this procedure for all path pairs.
- Select the minimum cost path pair.

Two optimizations must be performed: the innermost computes the best alignment cost between two strings, the outermost finds out the minimum cost path pair. These optimizations are carried out in a single pass by a Dynamic Programming procedure (three-dimensional DP or 3DP) that develops warping paths in the three-dimensional space illustrated in Fig. 5. The three dimensions represent the nodes of the reference word model (dimension R), the sequence of the test micro-segments (dimension T), and the levels of the micro-segmentation lattice (dimension L). A local cost function $G(r, t, l)$ is defined in the RTL space, where r is a node of the word model associated to a symbol of the representation alphabet, t is the index of a micro-segment, and l—the lattice level—assumes the values 1 or 2 referring to the best and to the second best segmentation labels, respectively. The cost function $G(r, t, l)$ can be computed, for every r, t, and l, by the DP equations:

$$G(r,t,l) = \min_{k=1,2} \left| \begin{array}{ll} G(r-1,t-1,k) + C(sub(w^r, s_l^t)) & (4) \\ G(r,t-1,k) + IC(w^r, s_l^t) & (5) \\ G(r-1,t,k) + C(del(w^r)) & (6) \end{array} \right.$$

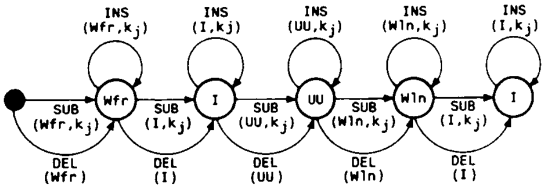


Fig. 4. Error model of the word /fjüme/.

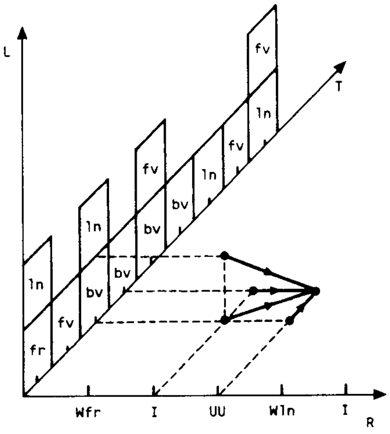


Fig. 5. Three-dimensional space.

where

$$IC(w^r, s_l^t) = \left| \begin{array}{ll} C(ins(w^r, s_l^t)) & \text{if } s_l^t \neq s_k^{t-1} \\ CC(sub(w^r, s_l^t)) & \text{otherwise} \end{array} \right. \tag{7}$$

where l and k assume the value 2 only if the tth micro-segment has two classification symbols. Equations (4), (5), and (6) account for symbol substitution, insertion, and deletion, respectively. It is worth noting that this structure can lead to "false insertion" events whenever adjacent micro-segments have the same phonetic symbol. Equation (7) solves this case by considering a micro-segment as the continuation of the preceding one if they have the same label, CC, being the "continuation cost."

For each value of r and t, 10 equations must be evaluated in the above formulation [(6) does not depend on l]. A suboptimal solution, reducing the number of equations to 4, is used instead, which, given the statistical characteristics of the segmentation process, does not substantially affect the performance of the system. In fact, the system of (4), (5), and (6) carries on all locally optimal warping paths. For any given t, two optimal alignment paths exist because both the first and the alternative phonetic label of the tth micro-segment are considered. It must be noticed, however, that if the tth micro-segment has one label only, optimal partial paths associated to point $(r, t - 1, 1)$ and to point $(r, t - 1, 2)$ in the RTL space are forced to converge, in the next step of DP, to the same point $(r, t, 1)$ and the DP algorithm keeps the best one only. As a single label is associated, on the average, to 65 percent of the micro-segments, even if the best path selection is made at each step t, the results of the matching procedure should not be appreciably affected.

A. Matching Costs

A simple function for the local matching cost is

$$C_1(op(h_i, k_j)) = -\text{Log}\left[\text{Prob}\left(op(h_i, k_j)\right)\right] \quad (8)$$

hence,

$$\begin{aligned} C_1(\text{sub}\,(h_i, k_j)) &= -\text{Log}\left[\text{Prob}\,(\text{substitution of } k_j \text{ for } h_i)\right] \\ C_1(\text{ins}\,(h_i, k_j)) &= -\text{Log}\left[\text{Prob}\,(\text{insertion of } k_j \text{ after } h_i)\right] \\ C_1(\text{del}\,(h_i)) &= -\text{Log}\left[\text{Prob}\,(\text{deletion of } h_i)\right]. \quad (9) \end{aligned}$$

These costs are estimated in the training phase by using the same phonetically balanced vocabulary (TRA) used for training the phonetic classifier. Every uttered word is aligned to its phonetic description by means of the 3DP procedure. If a word has more than one phonetic description, the model attaining the minimum alignment cost is considered. A backtracking procedure collects, for each word, the number of substitutions, deletions, and insertions of phonetic symbols:

$$\begin{aligned} N\,\text{sub}\,(h_i, k_j) &= \text{Number of substitutions of } k_j \text{ for } h_i \\ N\,\text{ins}\,(h_i, k_j) &= \text{Number of insertions of } k_j \text{ after } h_i \\ N\,\text{del}\,(h_i) &= \text{Number of deletions of } h_i. \end{aligned}$$

When all vocabulary has been processed, the alignment costs can be estimated as follows:

$$\begin{aligned} N\,\text{tot}\,(h_i) &= \sum_j \left[N\,\text{sub}\,(h_i, k_j) + N\,\text{ins}\,(h_i, k_j)\right] \\ &\quad + N\,\text{del}\,(h_i)] \\ C_1(\text{sub}\,(h_i, k_j)) &= -\text{Log}\left(N\,\text{sub}\,(h_i, k_j)/N\,\text{tot}\,(h_i)\right) \\ C_1(\text{ins}\,(h_i, k_j)) &= -\text{Log}\left(N\,\text{ins}\,(h_i, k_j)/N\,\text{tot}\,(h_i)\right) \\ C_1(\text{del}\,(h_i)) &= -\text{Log}\left(N\,\text{del}\,(h_i)/N\,\text{tot}\,(h_i)\right). \end{aligned} \quad (10)$$

These costs are reestimated by iterating the training procedure until they do not change appreciably. Two or three iterations are generally sufficient for obtaining a stable solution. The "continuation cost" CC is null using this metric. The initial costs are set as

$$C_1(\text{sub}\,(h_i, k_j))\left|\begin{aligned} &= 0 \quad \text{if } h_i \text{ belongs to class } k_j \\ &= 2 \quad \text{otherwise} \end{aligned}\right.$$

$$C_1(\text{ins}\,(h_i, k_j)) = 1$$

$$C_1(\text{del}\,(h_i)) = 1.$$

This initial setting corresponds to performing a 3DP matching using a modified Levenshtein distance [17].

TABLE III
NUMBER OF SEGMENTATION ERRORS

Speaker	Sex	Deletions	Incorrect Substitutions	Insertions
LA	f	8	176	2530
RF	f	25	279	2121
PD	m	18	399	2384
LF	m	38	257	2103
GM	m	44	320	1716
RP	m	34	311	2220
GP	m	15	174	2236

TABLE IV
PERCENTAGE OF DELETED, SUBSTITUTED, AND INSERTED SEGMENTS FOR EACH PHONETIC CLASS

Class	Deletions	Substitutions	Insertions
pl	0.47	0.23	0.54
fr	0.23	0.36	5.20
ln	0.11	0.11	7.10
fv	0.17	0.34	1.74
cv	0.06	0.00	1.41
bv	0.08	0.12	1.32
Total	1.12	1.16	17.31

The error rates of the phonetic segmentation, computed during the estimation of the alignment costs, in terms of the number of deleted, substituted, and inserted segments, are shown in Table III for 7 speakers. Deletions and substitutions of segments are not very frequent, while more than 2 insertions per word can be expected. The highest contribution to the insertions is due to fricative and liquid/nasal consonants as shown in Table IV, where the percentage of substituted, deleted, and inserted segments, averaged over all speakers, is detailed for each class.

B. Duration and Reliability of Micro-Segments

Metric C_1, defined in (8), does not take into account the micro-segmentation timing structure, a very important cue for word hypothesization. A straightforward way to include the duration of micro-segments in the matching cost is the following:

$$\begin{aligned} &C_2(h_i, k_j, op(h_i, k_j) \,|\, len(M_j)) \\ &\quad = -\text{Log}\left(\text{Prob}\left(op1\,(h_i, k_j)\right) * len(M_j)\right. \quad (11) \end{aligned}$$

where $op1\,(h_i, k_j)$ is the basic alignment operation of *one test frame*, labeled k_j, against the reference symbol h_i, and $len(M_j)$ is either the duration of micro-segment M_j, if $op1\,(h_i, k_j)$ is a substitution or an insertion operation, or it is the average duration of the h_i phonetic class corresponding to a deletion operation.

Furthermore, assuming the statistical independence of the alignment operations and of the micro-segment reliability, a matching cost function can be defined as the sum of two contributions—an alignment cost and a reliability cost as follows:

$$\begin{aligned} &C_3(h_i, k_j, op(h_i, k_j)) \\ &\quad = C_2(h_i, k_j, op(h_i, k_j)) + B(r, op(h_i, k_j)) \quad (12) \end{aligned}$$

where r is the micro-segment reliability and B is a function of the probability density $p(r \mid op(h_i, k_j))$ that can be estimated in the training phase by collecting statistics for each operation $op(h_i, k_j)$ into a histogram.

V. Lexical Access

Even if the number of phonetic micro-segments in a word is, on the average, less than the number of centisecond frames of about an order of magnitude, the complexity of matching a micro-segmentation against every vocabulary word is impractical when the lexicon size is of the order of thousands. A representation that reduces storage costs and leads to an efficient lexical access is obtained by merging the sequences of phonetic classes that describe the words in a tree in which the initial common subsequences are shared [43], [13], [25], [41]. If the nodes of the lexical tree represent phonetic classes, all words which share the same coarse phonetic description can be associated to the same node (the node representing last phoneme) as they become a set of phonetically indistinguishable lexical items. An example of a simple 12 words lexical tree is shown in Fig. 6, where all leaves and some (terminal) nodes are associated to the set of lexical items having the same phonetic structure. A tree is best suited to the lexical access task, rather than a more compact graph structure, because the former allows the N best word candidates to be easily obtained. The 3DP algorithm, in fact, can evaluate the alignment costs of all vocabulary words in parallel. This operation would be more complex and expensive if performed on a graph.

Given the micro-segmentation of an uttered word belonging to a lexicon represented by a tree, lexical access is performed by detecting the sequences of phonetic nodes, and hence the corresponding words, whose costs computed by means of the 3DP lie in a fixed range of the best one.

A. Experimental Results

A first set of experiments was devoted to the assessment of the 3DP method. The complete set of 1011 words of the GEO vocabulary pronounced by a male speaker was used as test.

Fig. 7(a) shows the rate of inclusion of the correct word in the candidate list versus the average number of candidate words for three different matching procedures, namely, optimal 3DP (curve A), suboptimal 3DP (curve B), and DP matching of the best first segmentation hypotheses only (curve C). Word models were represented by means of the symbols of alphabet A_1, and the C_1 metric was used for the evaluation of the costs. The curves were obtained as a function of the beam search threshold.

The 3DP procedure performs considerably better than classical DP: fewer candidate words and higher inclusion rates are obtained. The optimal and the suboptimal procedure give very close results, but the complexity of the suboptimal procedure is comparable to the complexity of the classical DP [see Fig. 7(b)]. In fact, for each reference node and for each micro-segment, 4 equations rather than 3 must be evaluated. Suboptimal 3DP has been, therefore, used in all remaining experiments.

A second set of experiments was carried out for selecting the best representation alphabet. The same test was performed by representing the GEO vocabulary words through the symbols of the alphabets A_1, A_2, and A_3 introduced in Section III. Table V shows the number of nodes

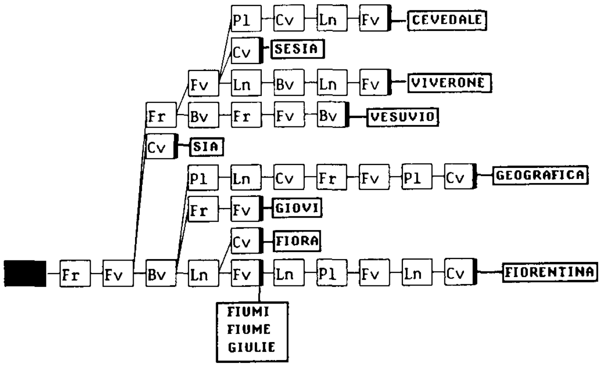
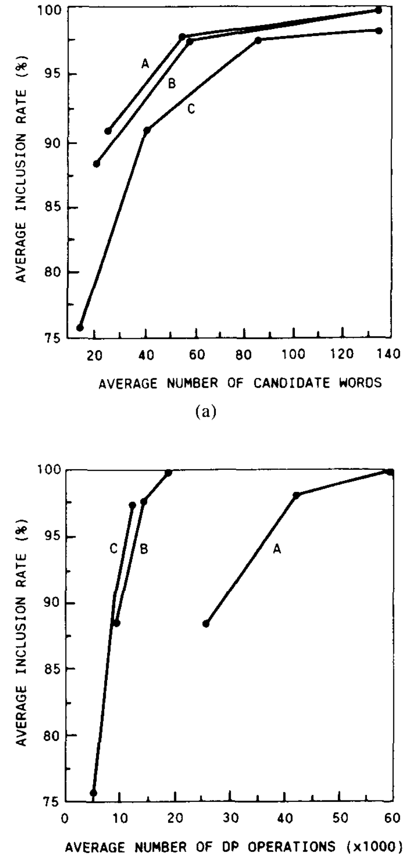


Fig. 6. A lexical tree.



(a)

(b)

Fig. 7. DP matching procedure comparison: (a) average inclusion rate versus average number of candidate words, (b) average inclusion rate versus average number of DP operations.

TABLE V
NUMBER OF NODES, LEAVES, TERMINAL NODES, AND BRANCHING FACTOR OF THE 1011 WORD GEO LEXICAL TREES USING THREE REPRESENTATION ALPHABETS

Alphabet	Number of nodes	Number of leaves	Number of terminal nodes	Branching factor
A_1	2656	801	894	1.431
A_2	2178	704	875	1.477
A_3	1739	569	749	1.485

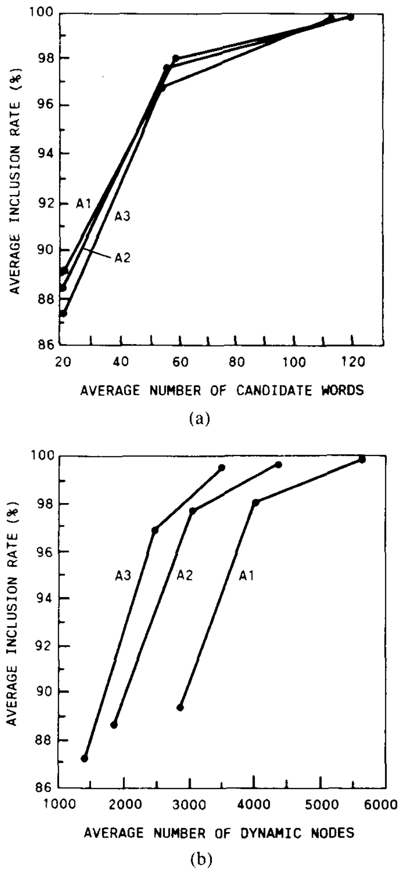


Fig. 8. Representation alphabets comparison: (a) average inclusion rate versus average number of candidate words, (b) average inclusion rate versus average number of expanded nodes.

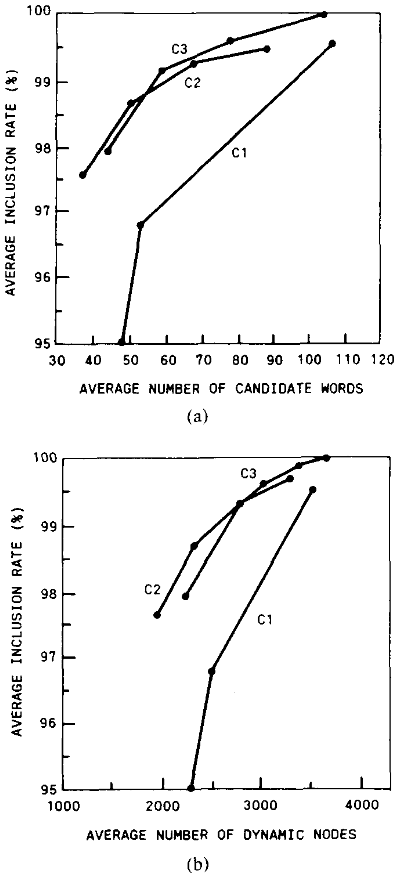


Fig. 9. Comparison of different metrics: (a) average inclusion rate versus average number of candidate words, (b) average inclusion rate versus average number of expanded nodes.

(N), the number of leaves (L), the terminal nodes (T), and the average branching factor of the obtained lexical trees.

Curves of Fig. 8(a), that present the inclusion rate versus the average number of candidates obtained by varying the beam search threshold, suggest that a more detailed specification of the lexical tree, such as that offered by alphabets A_1 and A_2, does not substantially reduce the candidate average size at inclusion rates greater than 99 percent. Better performance of alphabets A_1 and A_2, compared to alphabet A_3, for more constraining beam search thresholds, is not surprising because more information is conveyed by their alignment cost matrices. However, due to the scarce redundancy of the micro-segmentation code, large values of the beam search threshold must be used for obtaining acceptable high performance. Thus, coarseness of matching turns the accuracy of the model into unhelpful. Furthermore, the computational load increases when more detailed representation alphabets are used, as shown in Fig. 8(b), where the inclusion rate is plotted versus the average number of nodes expanded during the search. A_3 has been, therefore, used as the representation alphabet in all successive experiments.

The third experiment has been carried out to assess system performance as a function of the above-described metrics C_1, C_2, and C_3. Its results are summarized in Fig. 9(a) and (b). Timing information (metric C_2) gives substantial improvements, and further improvements are obtained by using the reliability of the phonetic labels (metric C_3).

The next set of experiments was performed for seven speakers, in the best conditions suggested by the previous experiments: suboptimal 3DP, A_3 representation alphabet, and C_3 metric. Fig. 10(a) shows, for various beam search thresholds, the inclusion rates and candidate list size for all speakers, while Fig. 10(b) presents the aver-

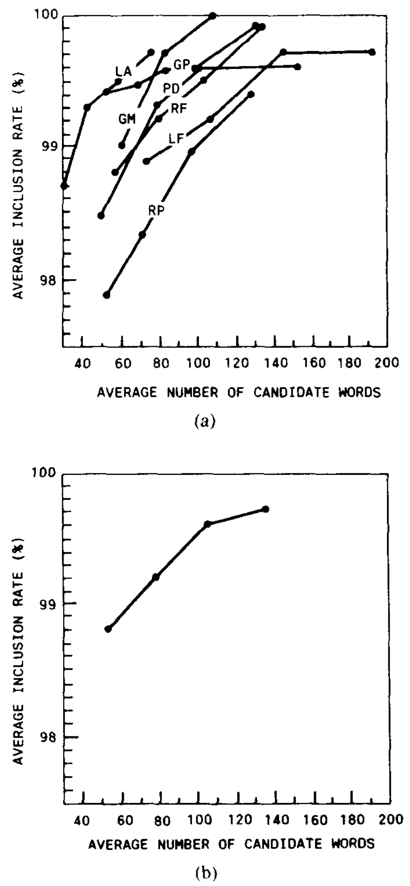


Fig. 10. Results as a function of the beam search threshold: (a) for 7 speakers (5 male, 2 female), (b) averaged results.

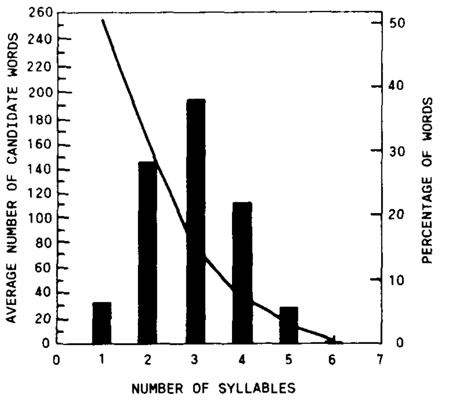


Fig. 11. Average number of hypothesized words and distribution of words in the GEO vocabulary versus their number of syllables.

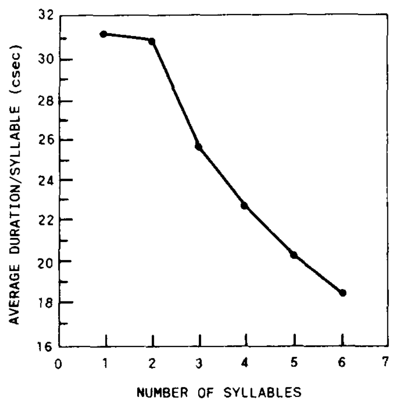


Fig. 12. Syllable duration.

aged results. The difference of the average inclusion rate among speakers is within 1 percent for the same beam search threshold value. Larger values of the threshold do not affect appreciably the accuracy of the hypotheses, while they considerably increase the average number of candidate words and the computation load. On the average, about 10 percent only of the items in the lexicon must be verified, and substantial improvement can be obtained by taking into account the heuristics introduced in Section V-B. Fig. 11 shows the average number of word candidates as a function of the number of syllables in a word; superimposed, as a bar graph, is the distribution of words in the GEO vocabulary as a function of their number of syllables. Short words generate a large number of candidates because the shorter the uttered word is, the easier it is to find, in a large vocabulary, similar or slightly different words in terms of a phonetic description into coarse classes. Errors are uniformly distributed among words composed of 2, 3, and 4 syllables. No errors were observed for monosyllabic or very long words. Monosyllabic words are generally well segmented and classified, when pronounced in isolation, because they are pronounced slowly compared to the syllables of polysyllabic words as can be observed in Fig. 12, where the average syllable duration is shown as a function of the number of syllables in a word.

Fig. 13 shows the inclusion rate as a function of the position of the correct word in the list, ordered by cost, of candidates generated by the lexical hypothesizer. In 62 percent of the cases (bottom-left of Fig. 13), the best scored word is the correct one (62 percent is the recognition rate of the system without verification). Verification can be avoided in 13 percent of the cases because only one word is hypothesized, as is shown in Fig. 14, where a histogram representing the distribution of the size of the candidate word list is reported.

B. Use of Heuristics

Robust heuristics can be introduced in the lexical access procedure to reduce the average number of hypothesized words and to speed up computation.

The first one (H_1) smoothes out the strings of phonetic labels produced by the frame-by-frame classifier through a majority voting filter. Two strings of symbols are considered: one corresponding to the best first classification, and the other one corresponding to the sequence of alternative decision labels. The second decision symbol is set to the value of the best one whenever the classifier has

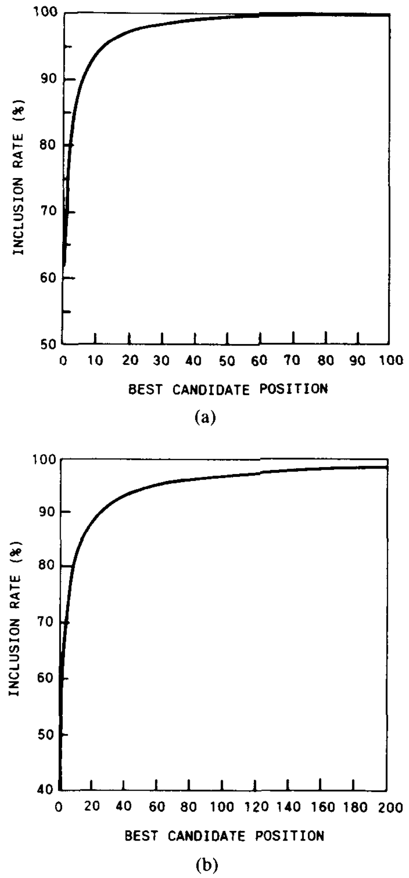


Fig. 13. Cumulative inclusion rate as a function of N-best candidate words, (a) 1011 words, (b) 18 388 words.

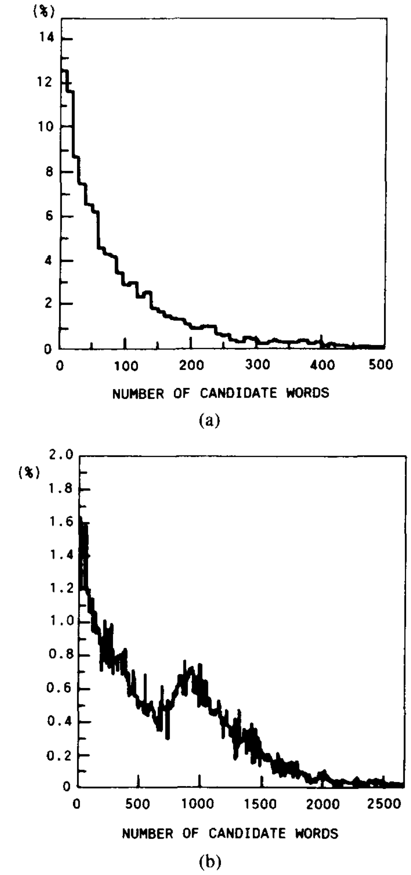


Fig. 14. Number of candidate words histogram, (a) 1011 words, (b) 18 388 words.

taken a single decision. The majority voting filter, applied to a shifting window of N (odd) frames, associates to the central frame of the window the phonetic labels that most frequently appear as the best first and the alternative decision, respectively. Fewer micro-segments are obtained because many spurious segments are eliminated. This reduction of the number of micro-segments reduces the number of operations needed for matching as well. Unfortunately, by increasing the window length, some correct segments disappear. Therefore, the number of spurious insertions decreases, but the number of deleted segments increases. The optimal window length depends on the speaking rate. Several experiments were performed for all 7 speakers varying the beam search threshold in order to achieve, for a given length of the majority voting filter window, an average inclusion rate of 99.7 percent, which is the same obtained excluding any filtering (window length equal to 1). The results are shown in Fig. 15 where the average number of candidate words (curve A), and the average number of nodes expanded per word (curve B) are plotted as a function of the filter window size. A window size value of 5 frames gives the minimum number of word candidates as well as the minimum computational complexity.

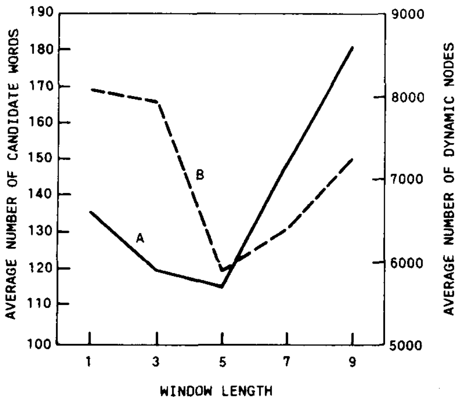


Fig. 15. Average number of candidate words and average number of expanded nodes per word versus filter window size.

A second heuristic (H_2) refers to reliable segments.

Let $R(s_i^t)$ be a function that associates a number r_i^t to the label s_i^t of a micro-segment $M(t)$. Let $R(s_i^t)$ be monotonically increasing with the probability that s_i^t is a correct classification of the micro-segment $M(t)$. If such a function exists, and if it is continuous, a threshold z and

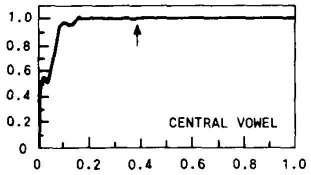


Fig. 16. Probability of correct classification of a *cv* micro-segment versus its reliability.

TABLE VI
WORD HYPOTHESIZATION MODULE PERFORMANCE

Heuristics	H_1	$H_1 + H_2$	$H_1 + H_2 + H_3$	$H_1 + H_2 + H_3 + H_4$
Insertion Rate	99.7	99.5	99.5	99.5
Average number of candidates	115.2	79.1	72.4	62.8
Average number of expanded nodes	5874	4546	4280	4280
Average number of operations	14 570	9828	8771	8771

a value $\boldsymbol{v}$ can be found such that

$$r_i^t > z \Rightarrow \text{Prob}\left[s_i^t \text{ is a correct classification } \middle| M(t)\right] > \boldsymbol{v}. \qquad (13)$$

Thus, in principle, a threshold z can be chosen such that it is possible to detect segments whose probability of being misclassified is below a fixed value or, in other words, segments that can be considered correctly classified with a given confidence value. The reliability measure associated to micro-segments can be chosen as function R according to the results shown in Fig. 16, where an estimation of the probability that a micro-segment label is correct, given its reliability, is presented for the class *cv*, each phonetic class exhibits a similar behavior. A value of the threshold z_m (shown by an arrow in the figure) was fixed for each class k_m, so that all training set segments with reliability greater than z_m were correctly classified:

$$a_i^t > z_m^t \Rightarrow s_i^t \in k_m^t. \qquad (14)$$

During lexical access, a segment satisfying the above-mentioned conditions is considered correctly classified. Hence, it cannot be inserted or substituted for a reference symbol that does not belong to the same phonetic class. This further local path constraint in the 3DP procedure has two beneficial effects: an appreciable reduction of the computational load and of the average number of word candidates, for the same inclusion rate. As can be observed in Table VI, a small reduction of the inclusion rate is traded for a sensible reduction of the average candidate word number and of the computational load expressed in terms of average number of expanded nodes.

Similar considerations lead to a third heuristic (H_3) that exploits robust cues for deciding that a particular phonetic class cannot be hypothesized for a given segment. If phonetic class k_n cannot definitely be assigned to a micro-segment, it cannot be substituted in the 3DP matching for a symbol of the representation alphabet belonging to class k_n. Frame energy, for example, has been used as a cue for deciding that high energy micro-segments cannot be substituted for a plosive sound. Table VI shows the performance obtained by using the H_2 and H_3 heuristics, and a 5 frame window majority voting filter (H_1).

As mentioned in the preceding subsection, the largest set of word candidates is generated by short words which, however, are generally well segmented. Hence, it is likely that the correct word is the first position in the candidate list. On the contrary, long words often appear at the end of their candidate word list, but the list is generally very short. The fourth heuristic (H_4) introduces, therefore, a constraint on the maximum number of active *nodes* of the lexical tree that are considered for word retrieval at the end of the search: only the N best nodes are allowed to generate word hypotheses. This constraint is not used during the search because it would be too expensive to order the best partial paths according to their cost, rather than performing a simple beam search.

Fig. 13 shows that more than 99 percent of inclusion rate can be obtained keeping only the first 60 best candidate words. This result is also illustrated in Fig. 17, which shows the inclusion rate and the average number of word candidates obtained by varying the value of the maximum number (j) of best candidate *nodes* ($Mj, j = 40, \cdots, \infty$) kept by the hypothesizer. Recall that the number of candidate *nodes* is different from the number of candidate *words*, since more than one word can be associated to a candidate node. The performance of the system using all these heuristics, constraining the maximum number of final active nodes to 140, is detailed in the last column of Table VI.

In Fig. 18(a), the average inclusion rate is shown as a

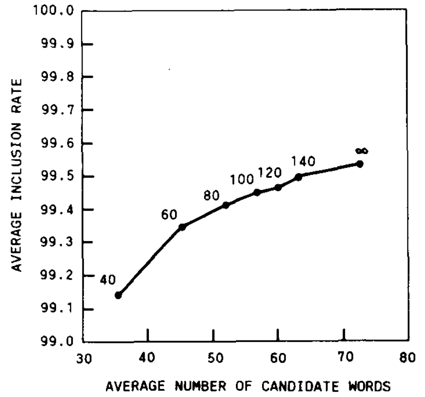
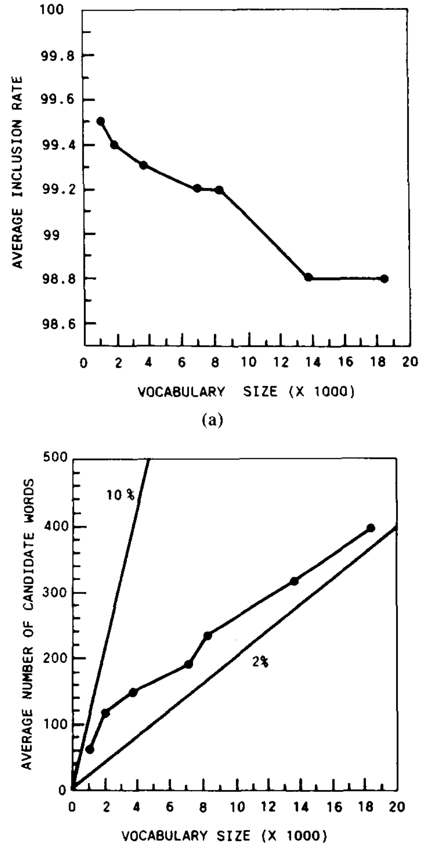


Fig. 17. Average inclusion rate and average number of candidate words as a function of the number of final active nodes.



(a)

(b)

Fig. 18. (a) Average inclusion rate, and (b) average number of candidate words as function of the vocabulary size.

function of the vocabulary size. Refer also to Fig. 13(b) and Fig. 14(b) for statistics about experiments made with a 18 388 word vocabulary that contains the union of the following sets of words:

- the 1011 word GEO vocabulary used for the recognition tests,
- the 8000 most frequent words appearing in a 4-million word corpus extracted from a political-economical magazine,
- the 13 747 words of the Collins Italian-English Pocket Dictionary.

By increasing the vocabulary size from 1011 words to 18 388, and using the same beam search threshold, a slight reduction (0.7 percent) of the average inclusion rate is observed. The increase of the average number of word candidates is presented in Fig. 18(b). It is worth noting that the percentage of the vocabulary words that must be verified decreases as vocabulary size increases: the bold right lines in the figure represents 10 and 2 percent of the vocabulary size, respectively.

VI. Verification Module

This module applies a more detailed phonetic knowledge than the phonetic classification one. A Word Translator generates, from the orthographic form of the words, one or more phonetic transcriptions through a set of rules. Multiple transcriptions are due, for example, to the ambiguity introduced by affricates and by intervocalic /s/ that, in Italian, can be voiced or unvoiced depending on the speaker regional attitude. Moreover, diphthongs and hiatuses are not discriminated by the Word Translator which always includes both these forms in the translation. The current set of rules produces 1.44 transcriptions per word, on the average. Every word phonetic transcription is then represented by a sequence of phonetic units, taking into account coarticulation phenomena occurring in the transitions between different sounds. Phonetic units are modeled by left-to-right HMM's with different numbers of states [10].

The verification module accepts as input the list of word candidates produced by the lexical access module. The HMM's sequences corresponding to this set of words are organized into a tree structure, where transcriptions with common initial parts share the same branches. Then, a beam search Viterbi procedure is performed on the tree to evaluate the most likely words.

A. *The Recognition Units*

Subword recognition units offer several advantages over whole word models in terms of storage saving and in discriminating words that include similar parts (e.g., minimal pairs) [36]. In the subword approach, the differences in the discriminant parts are enhanced because phonetic portions that are equal are represented by the same model.

This consideration suggests that steady parts of the phonemes (whenever they can be defined) be represented by the same model, and that transitions be accounted for by means of additional models only if they carry significant discriminant information [10]. This definition of the subword units was first proposed for template based systems [40], leading to satisfactory results both for Italian [8] and for English [39]. These recognition units can be considered as a tradeoff between diphones and phonemes.

TABLE VII
SOME EXAMPLES OF WORD TRANSLATION FROM ORTHOGRAPHIC FORM TO PHONETIC FORM AND TO THEIR CORRESPONDING SEQUENCE OF RECOGNITION UNITS. THE RECOGNITION UNITS "—" AND "b" ARE THE SILENCE AND THE VOICEBAR, RESPECTIVELY

Orthographic Form	Phonetic Form	Translation into Units										
SETTE	Sette	s	e	—	te	e						
APPARTIENE	Appartjene	a	—	pa	ar	r	—	ti	ie	e	n	e
AVERE	Avere	a	av	ve	e	er	re	e				
AVREBBE	Avrebbe	a	av	v	vr	r	re	e	b	be	e	
ANDARE	Andare	a	n	b	da	a	ar	re	e			

Twenty-six phonetic units were considered for the Italian language. Of the 650 (26 * 25) possible transition units, only 101 were selected according to phonetic knowledge and observing the results of recognition experiments carried out with difficult vocabularies such as minimal word pairs [10]. These 101 transition units include all plosive/vowel, affricate/vowel, and some sonorant/vowel transitions in addition to some consonant clusters. Transitions from vowel to sonorant are considered only for consonants /r/ and /v/. Twenty-two steady units complete the unit inventory: 5 vowels, 6 sonorants, 5 fricatives, 4 affricates, silence, and voicebar. Some examples of the translation from the orthographic form to the phonetic one and finally to the corresponding sequence of units are given in Table VII. Details about the context-sensitive translation rules can be found in [11] and [10].

B. Model Estimation

Hidden Markov modeling of the subword units allows model training to be automatically performed. From the orthographic form of the training vocabulary words, different phonemic transcriptions are generated according to their possible pronunciations. These alternatives are automatically converted into the unit sequence. This operation is performed for all training words. The training set is composed of one or more utterances of the training vocabulary represented as sequences of Vector Quantization codewords. For each utterance, a forward and a backward matrix is computed bootstrapping the system from untrained HMM's (uniform transition and emission matrices). For every subword unit appearing in the training database, the transition and emission probabilities are estimated by using a generalization to multiple observations of the classical reestimation formula [29]. This procedure is repeated until convergence is reached.

C. Experimental Results

Each speaker trained its set of 123 unit models, by pronouncing once the words in the TRA dictionary. Each training set consists of about 20 min of speech. These utterances were then coded by means of a speaker dependent 7 bit vector quantizer. Five iterations of the Forward-Backward algorithm were sufficient for obtaining stable estimates of the parameters of the models.

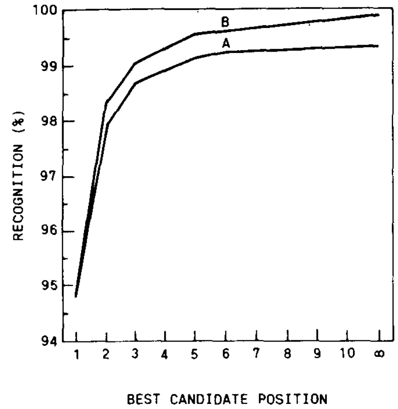


Fig. 19. Recognition rate versus N-best word scores for the 1011 word vocabulary.

Curve A in Fig. 19 shows the recognition rate, averaged over all speakers, as a function of the best candidate position for the two pass approach (hypothesis generation by partial phonetic description and successive detailed verification by stochastic decoding). Every word hypothesized by lexical access is represented by the set of its transcriptions into recognition units. All these representations are then compiled into a tree whose branches are the sequences of states of the HMM recognition units, and whose leaves identify words. A beam search Viterbi procedure operates on a tree to evaluate the best state sequences. The paths that are still active at the end of the search generate a set of word hypotheses ordered according to their likelihood; 95 percent of words attains the best first likelihood, while 99.3 percent of the uttered words are correctly included in the final set of hypotheses, whose average size is 4.4. Curve B in Fig. 19 refers, instead, to the results obtained in the direct approach, excluding the lexical access module, hence by applying the same beam search Viterbi procedure to the tree representing all vocabulary words. Obviously, slightly worse results are obtained in the former approach because the lexical access module propagates its errors (correct words missing in the candidate list) to the verification module. It is worth noting, however, that there is no difference in the recognition rate for the best first hypothesis. This means that a missing word in the candidate list produced by lexical access

TABLE VIII
COMPARISON OF THE ONE AND TWO PASS LEXICAL ACCESS STRATEGIES

	Direct Approach	Two Pass Approach
Best first recognition rate	95.0 percent	94.9 percent
Inclusion rate	99.9 percent	99.3 percent
Number of hypotheses	5.5	4.4
Number of operation/word in lexical access	—	9342
Number of operation/word in the verification module	99 795	17 200
Total number of operation/word	99 795	26 542

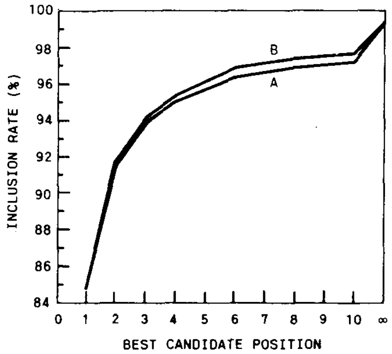


Fig. 20. Recognition rate versus *N*-best word scores, for the 18 388 word vocabulary.

is also missed as the best scored one by the direct approach. By increasing the rank of the accepted hypotheses, the difference between the two curves keeps constant and it depends only on the error of lexical access (0.5 percent). In Table VIII, a comparison of the performance of the two approaches can be found. As far as complexity is concerned, the phonetic segmentation and the generation of the hypothesis tree for verification are negligible in comparison to the matching. Matching requires a basic computation both for lexical access and for verification: the dynamic expansion of a trellis node. It consists in the evaluation of the cost of expanding a partial path from an origin node to a destination node, and in its comparison to the cost of the current best path reaching the destination node. As the complexity of cost computation is approximately equal for lexical access and for verification, a good approximation of the computational complexity of the two approaches can be given in terms of the average number of expansion operations. A complexity reduction of about 82 percent is achieved for the verification step, and of about 73 percent for the two step approach.

Fig. 20 shows the recognition rate as a function of the best candidate position for the two pass approach for the 18 388 word vocabulary. The best first recognition rate is 84.7 percent. Relevant improvements, similar to those in Fig. 19, can be observed for the best two candidates, reaching more than 91 percent of accuracy. About 99.2 percent of the words are included in the final set of hypotheses whose average size is 21.3.

VII. CONCLUSIONS

A large vocabulary isolated word recognition system has been presented. It is based on a two pass approach that relies on an efficient matching algorithm for generation of candidate words, and on HMM modeling for their verification. The main suggestions deriving from this work can be summarized as follows.

- A coarse phonetic segmentation can be more accurate than a detailed one, but few misclassifications can dramatically reduce the performance of a lexical access due to the small redundancy of the code.
- Robust phonetic segmentation can be achieved by generating, rather than a sequence of segments, a lattice of phonetic hypotheses to be matched against the vocabulary words which can be represented by a graph model including statistics about possible segmentation errors.
- Lexicon can be effectively represented as a tree, of phonetic nodes in the hypothesize step and of HMM subword units in the verification step.
- A three-dimensional DP matching algorithm has been introduced that performs better than other conventional algorithms.
- A suboptimal version of the matching procedure can be used without appreciable performance degradations.

The experimental results show the capability of the statistical models and of the lexical constraints to cope with the errors of the segmentation module. The accuracy of the HMM's of the subword phonetic units in the verification phase has also been assessed.

Over 99 percent of the correct words are within the first 5 best candidates for a 1011 word vocabulary; this accuracy reduces to about 96 percent for a 18 388 word vocabulary. A robust hypothesization system leads to interesting applications in the field of isolated and continuous speech recognition/understanding tasks.

REFERENCES

[1] G. Adda, M. Eskenazi, and P. E. Stern, "The use of rough spectral features for large vocabulary recognition," in *Proc. Euro. Conf. Speech Technol.*, Edinburgh, U.K., vol. 1, 1987, pp. 171-174.

[2] L. R. Bahl, R. Bakis, P. V. de Souza, and R. L. Mercer, "Obtaining candidate words by polling in a large vocabulary speech recognition system," in *Proc. IEEE ICASSP 1988*, pp. 489-492.

[3] L. R. Bahl, F. Jelinek, and R. L. Mercer, "A maximum likelihood approach to continuous speech recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-5, pp. 179-190, 1983.

[4] J. M. Baker, "State of the art speech recognition, U.S. research and business update," in *Proc. Euro. Conf. Speech Technol.*, Edinburgh, U.K., 1987, pp. 440-447.

[5] R. Billi, G. Massia, and F. Nesti, "Word preselection for large vocabulary speech recognition," in *Proc. IEEE ICASSP 1986*, 1986, pp. 23.6.1-23.6.4.

[6] D. M. Carter, "An information-theoretic analysis of phonetic dictionary access," *Comput. Speech and Language*, vol. 2, pp. 1-11, 1987.

[7] M. Codogno, L. Fissore, A. Martelli, and G. Volpi, "Experimental evaluation of Italian language for large dictionary speech recognition," in *Proc. Euro. Conf. Speech Technol.*, Edinburgh, U.K., 1987, pp. 159-162.

[8] A. M. Colla and D. Sciarra, "Automatic diphone bootstrapping for

speaker adaptive continuous speech recognition," in *Proc. IEEE ICASSP 1984*, pp. 35.2.1-35.2.4.

[9] M. Cravero, L. Fissore, R. Pieraccini, and C. Scagliola, "Syntax driven recognition of connected words by Markov models," in *Proc. IEEE ICASSP 1984*, pp. 35.5.1-35.5.4.

[10] M. Cravero, R. Pieraccini, and F. Raineri, "Definition and evaluation of phonetic units for speech recognition by hidden Markov models," in *Proc. IEEE ICASSP 1986*, pp. 42.3.1-42.3.4.

[11] ——, "Definition of recognition units through two levels of phonemic description," in *Proc. Montreal Symp. Speech Recognition*, Montreal, P.Q., Canada, 1986, pp. 53, 54.

[12] P. D'Orta, M. Ferretti, and S. Scarci, "Phoneme classification for real-time speech recognition of Italian," in *Proc. IEEE ICASSP 1987*, pp. 3.5.1-3.5.4.

[13] P. Demichelis, P. Laface, E. Piccolo, G. Micca, and R. Pieraccini, "Recognition of words in a large vocabulary," in *Proc. Int. Workshop on Recent Advances Appl. Speech Recognition, IWASR*, Roma, 1986, pp. 115, 123.

[14] L. Fissore, E. Giachin, P. Laface, G. Micca, R. Pieraccini, and C. Rullent, "Experimental results on large vocabulary continuous speech recognition and understanding," in *Proc. IEEE ICASSP 1988*, pp. 203-206.

[15] L. Fissore, P. Laface, G. Micca, and R. Pieraccini, "Very large vocabulary isolated utterance recognition: A comparison between one pass and two pass strategies," in *Proc. IEEE ICASSP 1988*, pp. 203-206.

[16] ——, "Interaction between fast lexical access and word verification in large vocabulary continuous speech recognition," in *Proc. IEEE ICASSP 1988*, pp. 414-417.

[17] K. S. Fu, *Syntactic Pattern Recognition and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1982.

[18] A. Giordana, P. Laface, and L. Saitta, "Discrimination of words in a large vocabulary using phonetic descriptions," *Int. J. Man-Machine Studies*, no. 24, pp. 453-473, 1986.

[19] V. N. Gupta, M. Lenning, and P. Mermelstein, "Integration of acoustic information in a large vocabulary word recognizer," in *Proc. IEEE ICASSP 1987*, pp. 17.2.1-17.2.4.

[20] D. P. Huttenlocher and V. W. Zue, "A model of lexical access from partial phonetic information," in *Proc. IEEE ICASSP 1984*, pp. 26.4.1-26.4.4.

[21] F. Jelinek, "Continuous speech recognition by statistical methods," *Proc. IEEE*, vol. 64, pp. 532-556, 1976.

[22] A. Kaltenmeier, "Acoustic/phonetic transcription using a polynomial classifier and Hidden Markov Models," in *Proc. Montreal Symp. Speech Recognition*, Montreal, P.Q., Canada, 1986, pp. 95, 96.

[23] T. Kaneko and N. R. Dixon, "A hierarchical decision approach to large-vocabulary discrete utterance recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 1061-1066, 1983.

[24] T. Kohonen, H. Riittinen, E. Reuhkala, and S. Haltsonen, "On-line recognition of spoken words from a large vocabulary," *Inform. Sci.*, vol. 33, no. 1-2, pp. 3-30, 1984.

[25] P. Laface, G. Micca, and R. Pieraccini, "Experimental results on a large lexicon access task," in *Proc. IEEE ICASSP 1987*, pp. 20.4.1-20.4.4.

[26] H. Lagger and A. Waibel, "A coarse phonetic knowledge source for template independent large vocabulary word recognition," in *Proc. IEEE ICASSP 1985*, pp. 2.7.1-2.7.4.

[27] J. N. Larar, "Lexical access using broad acoustic-phonetic classification," *Comput. Speech Language*, no. 1, pp. 47-59.

[28] S. Levinson, "Structural methods in automatic speech recognition," *Proc. IEEE*, vol. 73, pp. 1625-1649, 1985.

[29] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi, "Introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition," *Bell Syst. Tech. J.*, vol. 62, no. 4, Part 1, pp. 1035-1074, 1983.

[30] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 88-95, 1980.

[31] S. M. Marcus, "Associative models and the time course of speech," *Bibliotheca Phonetica*, no. 12, pp. 36-52, 1985.

[32] J. J. Mariani, "Hamlet: A prototype of a voice activated typewriter," in *Proc. Euro. Conf. Speech Technol.*, Edinburgh, U.K., vol. 2, 1987, pp. 222-225.

[33] ——, "Speech technology in Europe," in *Proc. Euro. Conf. Speech Technol.*, Edinburgh, U.K., vol. 1, 1987, pp. 431-439.

[34] B. Merialdo, A. M. Derouault, and S. Soudoplatoff, "Phoneme classification using Markov models," in *Proc. IEEE ICASSP 1986*, 1986, pp. 51.3.1-51.3.4.

[35] G. Micca, R. Pieraccini, P. Laface, L. Saitta, and A. Kaltenmeier, "Word hypothesization and verification in a large vocabulary," in *Proc. 3rd Esprit Tech. Week*, Brussels, 1986, pp. 845-853.

[36] R. K. Moore, M. J. Russel, and M. J. Tomlinson, "The discriminative network: A mechanism for focusing recognition in whole word pattern matching," in *Proc. IEEE ICASSP 1983*, pp. 1041-1044.

[37] H. Murveit and R. Brodersen, "An integrated-circuit-based speech recognition system," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 1465-1472, 1986.

[38] D. B. Pisoni, H. C. Nusbaum, P. A. Luce, and L. M. Slowiaczek, "Speech perception, word recognition and the structure of the lexicon," *Speech Commun.*, vol. 4, no. 1-3, pp. 75-96, 1985.

[39] A. E. Rosenberg and A. M. Colla, "A connected speech recognition system based on spotting diphone-like segments—Preliminary results," in *Proc. IEEE ICASSP 1987*, pp. 85-88.

[40] C. Scagliola, "Language models and search algorithms for real time speech recognition," *Int. J. Man-Machine Studies*, vol. 22, pp. 523-547, 1985.

[41] G. Schukat-Talamazzini and H. Niemann, "Generating word hypotheses in continuous speech," in *Proc. IEEE ICASSP 1986*, pp. 30.2.1-30.2.4.

[42] D. W. Shipman and V. Zue, "Properties of large lexicons: Implications for advanced isolated word recognition systems," in *Proc. IEEE ICASSP 1982*, pp. 546-549.

[43] A. R. Smith and L. D. Erman, "Noha—A bottom-up word hypothesizer for large-vocabulary speech understanding systems," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-3, pp. 41-51, 1981.

[44] S. Soudoplatoff, "Markov modeling of continuous parameters in speech recognition," in *Proc. IEEE ICASSP 1986*, pp. 2.2.1-2.2.4.

[45] A. Waibel, "Prosodic knowledge sources for word hypothesization in a continuous speech recognition system," in *Proc. IEEE ICASSP 1987*, pp. 20.16.1-20.16.4.

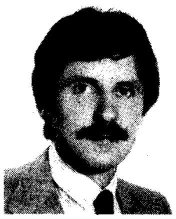
[46] V. Zue, "The use of speech knowledge in automatic speech recognition," *Proc. IEEE*, vol. 73, pp. 1602-1615, 1985.

Luciano Fissore was born in Bra, Italy, on February 22, 1955. He received the Doctorate degree in electronic engineering from the Polytechnic Institute of Turin, Turin, Italy.

Since 1982 he has been with CSELT (Centro Studi e Laboratori Telecomunicazioni) in the Speech Recognition Group as a Researcher. His current interests lie in training procedures for speaker-dependent and speaker-independent recognition systems, large vocabulary isolated and continuous speech recognition systems, and statistical language modeling for Italian.

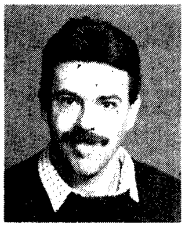
Pietro Laface was born in Reggio Calabria, Italy, in 1949. He received the Doctorate in electronic engineering from the Politecnico di Torino, Italy, in 1973.

He was Assistant and Associate Professor of Computer Science in the Dipartimento di Automatica e Informatica of the Politecnico Di Torino from 1974 to 1987. Since 1987 he has been a Professor of Computer Science at the Dipartimento di Informatica ed Applicazione of the University of Salerno, Italy. His research interests include digital signal processing, automatic speech recognition and understanding, and expert system design.



Giorgio Micca was born in Turin, Italy, on May 28, 1951. He received the Doctorate degree in computer science from the University of Turin in 1976.

Since 1977 he has been working with CSELT in the fields of packet switching networks, image analysis, and speech recognition. His actual interests include signal processing, large vocabulary continuous speech recognition systems, and statistical models of language.



Roberto Pieraccini was born in Genoa, Italy, in 1955. He received the Dr.Ing. degree in electric engineering from the University of Pisa, Italy, in 1980.

Since 1981 he has been working at CSELT (Centro Studi e Laboratori Telecomunicazioni), Torino, Italy, in speech recognition research. Since 1983 he has been involved in the design and implementation of a speech understanding system in the framework of an international joint project funded by the European Economic Community. In March 1988 he joined the Speech Research Department at AT&T Bell Laboratories, Murray Hill, NJ, as a Visiting Scientist, where he investigated stochastical subword unit modeling of English. His interests include the application of statistical pattern recognition techniques to speech understanding and dialog.