IBM Internal Use Only

Speech Recognition of Italian: the IBM Rome Scientific Center Prototype

M. Brandetti, M. Ferretti, A. Fusi, G. Maltese, S. Scarci, G. Vitillaro
(BRANDET, FERRETTI, FUSI, MALTESE, SCARCI, VITILLAR at ROMESC)

IBM Rome Scientific Center
via Giorgione 159, 00147 ROME (Italy).

Abstract

A real-time speech recognition system of Italian has been developed at IBM Rome Scientific Center. It handles natural language sentences from a 20000-word dictionary, dictated with words separated by short pauses. The approach being applied to the Italian language is based on the probabilistic techniques applied to recognition of English by researchers at IBM T. J. Watson Research Center, Yorktown Heights. The architecture consists of a PC/AT equipped with signal processing hardware. The paper describes the system, shows results of decoding tests and includes descriptions of the topics in speech recognition being currently investigated.

1. Introduction

Existing speech recognition technologies have proven adequate for simple tasks, involving knowledge of a small vocabulary (tens or hundreds of words), suiting limited applications (typically recognition of a set of commands uttered in an isolated fashion by an operator whose hands are busy); they are usually independent of the target language.

Interesting applications in an office environment, such as text dictation and database query, on the other hand, must be capable of handling natural language and pronunciation. This requires large vocabularies (thousands of words), and necessitates substantially more sophisticated techniques, which take into account language-specific knowledge on phonology, syntax and (surface) semantics.

Rome Scientific Center has developed a real-time isolated-utterance speech recognition system for the Italian language, based on a 20000-word vocabulary. The recognizer architecture consists of a workstation based on a PC/AT equipped with signal processing hardware. Word-recognition accuracy for pre-recorded sentences ranges from 95% to 98%. The words must be uttered separated by short pauses.

The Speech Recognition Project started in 1985 at IBM Rome Scientific Center from a cooperation with the IBM T.J. Watson Research Center, where advanced prototypes for the English language have been developed. By July 1986 a 3000-word recognizer based on an IBM 3090 mainframe and a PC/AT was developed. By December 1986 the recognizer was implemented on *Tangora* hardware [1][2]. The 6000 and 20000-word recognizers were completed by July 1987 and April 1988, respectively.

The mathematical approach is probabilistic, based on the maximum likelihood principle [3]. The role of human knowledge is limited to the design of a basic model of speech production and perception; statistics is used as a methodology for integration of the conceived model by "automatic learning" from data.

Let $\overline{W} = w_1 w_2 \ldots w_N$ be a sequence of N words, and let $\overline{A}$ be the acoustic information, extracted from the speech signal, from which the system will try to recognize which words were uttered. The aim is to find the particular sequence of words $\hat{\overline{W}}$ which maximizes the conditional probability $P(\overline{W} \mid \overline{A})$, i.e. the most likely word sequence given the acoustic information. By Bayes' theorem,

$$P(\overline{W} \mid \overline{A}) = \frac{P(\overline{A} \mid \overline{W}) P(\overline{W})}{P(\overline{A})}$$

$P(\overline{A} \mid \overline{W})$ is the probability that the sequence of words $\overline{W}$ will produce the acoustic string $\overline{A}$, that is, the probability that the speaker, pronouncing the words $\overline{W}$, will utter sounds described by $\overline{A}$. $P(\overline{W})$ is the a priori probability of the word string $\overline{W}$, that is, the probability that the speaker will wish to pronounce the words $\overline{W}$. $P(\overline{A})$ is the probability of the acoustic string $\overline{A}$; it is not a function of $\overline{W}$, since it is fixed once $\overline{A}$ is measured, and can thus be ignored when looking for the maximum over $\overline{W}$.

A consequence of this equation is that the recognition task can be decomposed in the following problems:

1. perform **acoustic processing** to encode the speech signal into a string of values $\overline{A}$ representative of its acoustic features, and, at the same time, adequate for a statistical analysis;
2. compute the probability $P(\overline{A} \mid \overline{W})$ (for this purpose an **acoustic model** must be created);
3. evaluate $P(\overline{W})$ (for this a **language model** is needed);
4. look, among all possible sequences of words, for the most probable one, by means of an efficient **search strategy** (an exhaustive search is not feasible, even for small vocabularies).

A description of the system architecture is provided in the next section. In the following sections, acoustic and linguistic modeling of the Italian language are discussed and experimental recognition results are given; furthermore a description is given of topics in speech recognition being investigated, including automatic phone clustering for fast lexical access [4]; fast speaker adaptation [5]; speech databases [6]; automatic phonetic transcription [7]; human factors of voice-activated text-editing [8].

IBM CONFERENCE ON NATURAL LANGUAGE PROCESSING
OCTOBER 24-26 1988
THORNWOOD, NY

2. System Architecture

Recognition and transcription of speech are performed by a workstation consisting of an IBM PC-AT equipped with four signal processing cards and the IBM ECD high resolution screen. Speech is collected by either a lip microphone (providing good noise immunity) or a table pressure zone microphone (more sensitive to background noise, but very comfortable for the speaker) [9]. The digitized acoustic signal (20K samples/sec, 12 bits/sample) is processed to extract, every 10 milliseconds, a vector of 20 parameters, which represent, essentially, the signal log energy in 20 frequency bands (spaced in accordance to the frequency sensitivity of the human ear), and transformed nonlinearly to take into account the adaptation capability to different sound levels. The vector-quantization replaces each vector with an *acoustic label* identifying the closest prototype vector belonging to a speaker-dependent pre-computed codebook of 200 elements.

The search strategy is based on the *stack sequential decoding* algorithm [10]. It controls the decoding process by hypothesizing the most likely sequence of words (by means of an efficient heuristic method), and requests the evaluation of linguistic and acoustic probabilities according to the hypothesized left context of the sentence. Stack decoding proceeds from left to right, and therefore is intrinsically well suited to a real-time system, which recognizes word sequences while they are being spoken.

The human interface of the speech recognizer consists of a text editor, which allows the use of both voice and keyboard for text input and editing. Commands for text insertion and deletion, word-searching, formatting (with a "what you see is what you get") interface are included. Documents can be filed, retrieved and printed. All editor commands can be given either by keyboard or by voice. A word (or any character string) not included in the vocabulary can be input by pronouncing a keyword (which sets the system to a single- character input mode and by spelling it).

3. Acoustic Modeling

The acoustic model is based on Markov models [11][12] of Italian phonemes as fundamental building blocks. A key factor to achieve a good recognition accuracy is the definition of the topological structure of the Markov chains. It has been observed, both for English and Italian, that the same Markov structure can adequately be used for all the phonetic elements of the language, if it provides enough degrees of freedom. Differentiation among phonetic Markov sources is thus left entirely to the parameter estimation process [13]. Therefore, the essential problem is the design of the set of phonetic elements by which the language sounds are described. Phonemes, the classical units defined by the phonology of the language, are a good starting point, but don't adequately take into account the variability of the speech phenomena. On the other hand, a too detailed model, involving a large number of parameters, might require an unacceptably large statistical sample of the speaker's voice to be trained. The design of the phonetic alphabet should then look for the best trade-off between detail of modeling and brevity of training.

A systematic procedure to look for an optimal phonetic alphabet has not been developed yet. Our approach combines the results of traditional acoustic and phonetic research with analysis of statistical data. For this purpose, the speech signal is aligned to the Markov source by means of the Viterbi algorithm [14]. Comparison of several speech segments aligned to the same phonetic machine helps in identifying coarticulation effects, i.e. acoustic variability depending on the phonetic context, and inter-speaker differences due to regional inflections. A measure of the quality of the phonetic representation may be provided by the mutual information between the phonetic alphabet and the set of speech alignments. After making experiments with various phonetic alphabets (see below) we adopted a set of 56 phonetic units [15], while Italian is usually described in terms of 30 distinct phonemes.

Recognition experiments are the most reliable way to evaluate the effectiveness of a modification to the phone alphabet, but are slow and computationally expensive (they involve re-training of the speaker and decoding of a prerecorded set of sentences). We experimented some faster measures, which proved very useful. The *Kullback divergence* (or *cross-entropy*) defined as:

$$d(M_1, M_2) = \sum_{\bar{A}} P(\bar{A} \mid M_1) \log \frac{P(\bar{A} \mid M_1)}{P(\bar{A} \mid M_2)} + \sum_{\bar{A}} P(\bar{A} \mid M_2) \log \frac{P(\bar{A} \mid M_2)}{P(\bar{A} \mid M_1)}$$

can show whether utterances ($\bar{A}$) of two units (M_1, M_2) have significant statistical differences. This measure is especially convenient when considering to split a set of sounds, previously described by a single phonetic unit, into two sets described by two different units (usually depending on the phonetic context).

Exact computation of divergence requires that the summation be extended to all possible sequences of acoustic labels $\bar{A}$. As this is infeasible, approximate techniques are needed. We experimented three techniques, described in [4]: consider emission probabilities of single labels instead of sequences; extend the summation to a sample set of label sequences uttered by the speakers and aligned via the Viterbi algorithm; extend the summation to a set of label sequences obtained by a Monte Carlo simulation of the behavior of the phones.

A notable problem of Italian is the presence of inflections due to mispronunciations by speakers from some regions. A possible solution consists in describing mispronounced words with more than one word model; this requires that more than one source be matched to the incoming utterance during recognition. Our more efficient solution consists in introducing "ambiguous" phonetic units, which, after the parameter estimation performed by the training procedure, are flexible enough to model the inconsistencies of the speaker's pronunciation.

The vowel *e*, when stressed, should be pronounced closed (/e/) in some words and open (/ɛ/) in others. Table 1 shows, for one speaker, divergences between the following units: *EO* (representing the open stressed /ɛ/ sound), *EC* (the closed stressed /e/), *EX* (the ambiguous stressed *e*), *IS* (the stressed /i/). The *EO* and *EC* units, which are associated to consistently pronounced *e*, display rather high divergence, while the *EX* unit, associated to closed and open occurrences of *e*, is rather well matched to both. The *IS* unit is more similar to the closed than to the open *e*.

Table 1. Divergences between phone units.

	EO	*EC*	*EX*	*IS*
EO	0.0			
EC	6.2	0.0		
EX	4.5	2.1	0.0	
IS	13.2	4.9	8.9	0.0

The system has indeed proven capable of handling speakers from different Italian regions with essentially identical performance.

Next table shows experimental word recognition accuracy when decoding is purely acoustic (i.e., the language model gives all words the same probability), for three phone sets, using the 6000-word vocabulary recognition system. The first one, **PH45**, consists of 45 phones, obtained by augmenting the set of 30 Italian phonemes by means of basic phonetic knowledge. The above described statistical techniques were employed to further refine the set to include 55 phones (**PH55**). Finally, some experimental data on words ending with a consonant (few in Italian, but rather frequent and confusable, because of their short duration) suggested introduction of a special unit in order to model the glottal pulse often occurring at the end of these words (**PH56**). The notable improvement in accuracy is largely due to the fact that these words were often confused with similar words ending by vowel.

Table 2. Average recognition accuracy with various phonetic alphabets.

PH45	88.7
PH55	90.9
PH56	92.2

Another peculiarity of the Italian language is the high frequency of vowels. The ratio of consonants to vowels in a word, which is particularly low in all Romance languages, is only 1.12 for Italian, while for English is 1.41 and for German is 1.71 [16]. Therefore, special care was used in modeling vowels: the seven vowel phonemes of Italian are described by eighteen distinct phonetic units.

Estimation of Markov parameters is accomplished by the Baum-Welch algorithm [17], which attempts to maximize $P(\overline{A} \mid \overline{W})$ for the (known) training text uttered by the speaker.

In the standard training procedure, the user of the dictating-machine prototype is requested to read a text, which will be called L in the following, consisting of 100 meaningful sentences (1043 total words). The resulting speech sample is about 15-minute long. The text has been designed in order to provide several instances of each phone in a representative set of phonetic contexts.

During recognition, the acoustical model is used to compute the probability $P(A \mid W)$. As it is infeasible to carry out the computation for all the words in the vocabulary in real time, the acoustical match consists of two stages. A fast, rough analysis is first performed to discriminate words displaying gross mismatches to the incoming utterance [18]. In this way a small number of words is selected, for which a detailed match computation is carried out.

Sentences are uttered with short pauses between words. However, the decoder does not rely on silence detection to identify word boundaries. A probabilistic determination of the most likely end point of each word is carried out by the acoustical matcher itself. This allows very short pauses between words, while direct silence detection would require long pauses (about half a second) to avoid confusion with silence segments inside words, due to stop consonants.

4. Language modeling

The language model estimates the probability of a word sequence $W = w_1 w_2 \ldots w_N$ by evaluating the probability of each word, given the left context of the sentence:

$$P(w_1 \ldots w_N) = \prod_{i=1}^{N} P(w_i \mid w_1 \ldots w_{i-1}).$$

In accordance with the statistical approach, the estimator is built from relative frequencies extracted from a large corpus of sentences. To estimate the probability of a word, contexts with the same last $N-1$ words are considered equivalent (N-gram language model [19]):

$$P(w_i \mid w_1 \ldots w_{i-1}) = P(w_i \mid w_{i-N+1} \ldots w_{i-1})$$

A value $N = 3$ (trigram language model) was actually used. The predictive power of a probabilistic language model is measured by *perplexity*[20], defined as:

$$p = 2^{\tilde{H}}$$

where $\tilde{H}$ is an estimate of the entropy (according to the language model probability P) computed on a text $w_1 \ldots w_L$ generated by the source which is being modeled:

$$\tilde{H} = -\frac{1}{L} \times \sum_{i=1}^{L} \log_2 \tilde{P}(w_i \mid w_{i-N+1} \ldots w_{i-1}).$$

Perplexity is the average uncertainty (the *branching factor*) [20] of the model expressed by the equivalent number of equiprobable words.

The language model is built on a backing-off approach [19], combining N-gram statistics (computed from a corpus of 107 million words) and the Turing's statistical technique to estimate the probability of linguistic events not observed in the corpus [19][21]. The threshold for bigram and trigram discount factors was chosen as in [19]. Turing's formula was tested on a 10 million word corpus and showed results very close to experimental data [21].

The twenty thousand words in the system's vocabulary were chosen as the most frequent ones over a subset (44 million words) of the corpus used for language model training, which was taken from magazine and daily newspaper articles and from news-agency flashes on economy and finance, provided by "*Il Mondo*" weekly magazine, the "*Sole 24 Ore*" daily newspaper and the "*Ansa*" agency, respectively. The vocabulary gives a coverage of 96.5 % on disjoint test sets taken from the same sources as the training corpus.

We grouped into classes the words which are likely to be used in the same context such as names of towns, companies and so on. About 14% of the words in the vocabulary were put into 34 classes. The remaining words were considered as a one-word classes. For each word, the class choice was made taking into account the Italian language specific phenomena known as apostrophe to determine the possible contexts where the word can occur. The probability of occurrence of word w_i given the context w_{i-1}, w_{i-2} is given by:

$$P(w_i \mid w_{i-1}, w_{i-2}) = P(w_i \mid C_i) \times P(C_i \mid C_{i-1}, C_{i-2})$$

where C_k is the class which the word w_k belongs to. A word belongs to one class only.

The language model gives perplexities of 98 and 86 on the text used for decoding tests and on a disjoint text taken from the same sources as the training corpus, respectively.

5. Decoding tests

The following table shows the word-recognition accuracy of the decoder as measured on 62 test sentences amounting to 1043 words.

Table 3. Speech recognizer performances.. Average, best and worst recognition accuracies concerning speakers with various degrees of experience in using the decoder for 20000-word vocabulary.

Speakers			Accuracy (%)		
Experience	Gender	No. subjects	A	B	W
Good	M	5	97.5	98.2	96.4
None	M	10	96.3	98.0	94.2
None	F	6	96.3	98.2	94.8

6. Studies on fast speaker adaptation

The 15-minute training speech sample L is normally required from each speaker to find an optimal set of prototype vectors for the codebook, via k-means clustering and to compute HMM parameters, i.e. transition and emission probabilities.

Speaker-independent recognition experiments were performed (using the 6000-word vocabulary recognition system) by collecting speech samples by 10 speakers and computing common prototypes and probabilities; recognition rates ranging from 84% to 93% were achieved on new speakers. The techniques we are studying [5] are aimed at enhancing recognition accuracy by adapting the common prototypes and probabilities by a rapid analysis of a short (about 1-minute) speech sample S provided by the new speaker.

Previous works on this subject present some approaches to HMM parameters adaptation, without considering the acoustic codebook problem [22]; techniques for mapping the codebook to that of a reference speaker, in a DP matching environment, are discussed in [23]. Work on codebook adaptation was aimed at the task of isolated digits recognition, and required a sample of the whole dictionary by the speaker [24]. A Bayesian approach was applied to a feature-based system [25].

We took into consideration both the codebook and the HMM parameters estimation aspects. We rely on multi-speaker (rather than on single-speaker) references, to avoid dependency of the results on the acoustical similarity between the reference and the new speaker.

For *codebook computation*, the problem of the statistical insufficiency of the adaptation sample S is addressed according to two approaches:

1. Vector prototypes are modeled as Gaussian probability distributions. The *a priori* probability distributions of the prototypes means are estimated from sample L uttered by each of 10 speakers. Then, for each new speaker, the *a posteriori* means of the adapted prototypes, given S, are computed via Bayesian learning. For sake of computational efficiency, a diagonal covariance matrix is assumed.
2. As the recognizer performs Euclidean, rather than Gaussian, labeling of acoustic vectors, we extended the deleted-estimation technique [17][26] to an Euclidean framework, to find an optimal interpolation between the common prototypes C_k and the prototypes S_k obtained from S. The i-th component of the adapted prototype A_k is given by

$$A_{ki} = \lambda_{bi} C_{ki} + (1 - \lambda_{bi}) S_{ki}$$

where b indicates a *bin* dependent on the amount of data available for prototype k in S. λ_{bi} is estimated by minimizing total distortion.

Both techniques allow computation of adapted prototypes in few seconds. The following table shows recognition rates for 3 speakers, using clustered (from sample L), common and adapted (by technique 1 and 2 respectively) prototypes. In all cases, a complete training of the HMM parameters on sample L was performed.

Table 4. Different vector prototypes.. Recognition accuracies for 3 speakers using vector prototypes obtained with various techniques. Data refer to 6000-word vocabulary recognizer.

Spk	CLUS	COMM	ADP1	ADP2
SSS	98.0	95.7	98.0	97.7
STR	95.7	90.0	95.7	95.4
AFS	96.1	93.8	94.2	94.2

For fast *HMM parameters estimation*, we are applying deleted estimation to find the optimal (in the maximum likelihood sense) interpolation between common and speaker-dependent (obtained from S) statistics.

7. Speech database building and checking

Aligning speech to its phonetic transcription is not an easy task. Manual alignment requires an expert and is slow and expensive. Efficient and reliable automatic methods are strongly needed.

An (almost completely) automatic approach to the problem of building a very large time-aligned speech database has been developed [6]. We used this approach to collect more than 30 hours of speech uttered by 10 different speakers, corresponding to over 62000 words. The data were afterwards aligned to their phonetic transcriptions.

The system architecture is composed of: IBM PC-ATs equipped with attached A/D/A converters and signal processors [27]; optical devices which allow large, write-once, direct-access storage; a host mainframe; a token-ring network connecting the PCs and the host.

The speech collected according to the mentioned technique is stored in real time on the optical disk. The speech signal may then be transformed by techniques such as Fast Fourier Transform, Linear Predictive Coding, and cepstral analysis. For the purpose of phonetic alignment, we process the signal through the acoustic front-end of the speech recognizer (see section 2) These preliminary computations are performed by the workstation; the time-alignment and checking process then takes place on the host mainframe.

We align sequences of codewords to their phonetic transcription using the Viterbi algorithm [14].

Other automatic techniques for speech alignment found in the literature propose dynamic programming methods to align the signal to a sequence of phonetic features [28] [29], or to a reference waveform [30].

The aligned waveforms must then be analyzed in order to correct errors. These may come either from inaccuracies due the statistical nature of the Viterbi algorithm, or from problems in the recorded data, due to undesired noise or speaker mistakes. We propose a technique which overcomes

the need of a complete listening of the recorded utterances [31] and produces results of comparable accuracy.

Our technique consists in performing several statistical tests to find possibly incorrect word-aligned speech segments. Gross errors are identified by the Viterbi algorithm itself. An independent likelihood measure of the obtained alignments is provided by a statistical model of the duration of the phonemes. We also compute a more detailed likelihood measure which assumes a Poisson distribution for the probability $P(C|W)$ of the codewords produced by the Markov source associated to each word [32]. We found that is much more practical to impose a likelihood threshold on $P(W|C)$ rather than on $P(C|W)$. $P(W|C)$ is estimated through the Bayes' formula:

$$P(W|C) = \frac{P(C|W)P(W)}{P(C)}$$

where $P(W)$ is inessential, and $P(C)$ is approximated by an expression depending only on the length of the codeword string C.

This automatic process classified an average of 2.5% of the utterances as suspect. They were then manually examined by using an interactive system allowing high quality graphical display and replay of selected speech segments.

The whole process of database construction, consisting of recording, analysis, checking and correction of wrong utterances, took less than six weeks.

8. Automatic phonetic transcription

In the development of our prototype we use Automatic Phonetic Transcription (APT) [7] for the *design* of the phonetic structure of the words of the initial vocabulary as well as for its *personalization*, i.e. adding of new words by the user.
Traditional APT systems (based on rules or on automatic learning) have inadequate accuracy [33][34][35] . We followed an approach where phonotactical knowledge (well described by a set of formal rules) is separated from lexical knowledge (largely based on experience and not suitable to a formal description).

Grapheme-to-phoneme translation for the Italian language presents relatively low uncertainty. The most relevant ambiguities are:

- placement of stress;
- stressed *e* and *o* may be pronounced open (/ɛ/ and /ɔ/) or closed (/e/ and /o/);
- *i* and *u* may be either vowels (/i/ and /u/) or semi-vowels (/j/ and /w/);
- *s* may be either sonorant (/z/) or non-sonorant (/s/);
- *z* may be either sonorant (/dz/) or non-sonorant (/ts/);
- *gl* may be either a palatal liquid (/ʎ/) or a sonorant velar stop followed by an alveolar liquid (/gl/).

Other ambiguities are due to special words (such as words of foreign origin).

Our system is based on a set of rules which formalize phonotactical knowledge only, without attempting to represent lexical knowledge. Therefore the rules contain all the above mentioned ambiguities of Italian grapheme-to-phoneme translation. Each rule consists of a left part and a right part. The left part consists of a grapheme string and its (possibly empty) left and right graphemic contexts; the right part consists of the set of possible phonemic transcriptions for the grapheme string. These rules yield for a given input word a set of transcriptions. This set is then automatically pruned by means of global rules (which, for example, reject all transcriptions which do not have one and just one stressed vowel). The right phonetic translation always belongs to the resulting set.

As our phone set is subject to changes (which may be suggested by new knowledge on the pronunciation behavior of speakers), while the phoneme transcription of a word is stable, its phone transcription may vary in the future. For this reason, we perform a two-stage translation (from graphemes to phonemes first, and from phonemes to phones hereafter) according to two different sets of rules. For grapheme-to-phoneme translation the rules are 78, while for phoneme-to-phone translation the rules are currently 66.

For the grapheme-to-phoneme translation, the average number of translations per word is 5.1. The following table shows the distribution of the number of transcriptions.

Table 5. Transcription Percent Frequency.

1	7.8
2	13.9
3	18.5
4	21.3
5	10.2
6	8.9
7	1.6
8	4.9
9	1.2
10	5.7
>10	6.0

We observe that more than 80% of the words give 6 or less transcriptions.

In the *design* process, the choice of the correct transcription is currently performed manually, by means of an efficient interactive system; for *personalization*, the user is asked to provide the spelling and a sample utterance of the new word and the most likely transcription is automatically selected, by means of a statistical algorithm.

9. Voice recognizer user acceptance

In this section we describe the experiments carried on at IBM Rome Scientific Center for evaluation of voice versus keyboard as a mean for entry and editing of texts. We performed some preliminary experiments in order to assess the usability, efficiency and user acceptance of the system, and to obtain hints about possible enhancements. Previous studies on this subject were limited, because a real-time natural-language speech recognition system was not available. They dealt with voice input of small artificial languages [36], or natural language input by means of simulations [37].

Our experiments studied the task of dictating to the machine by reading a printed text. We selected an article from "*Il Sole 24 Ore*," the major Italian business newspaper, and asked several users to input it into the workstation twice: once they used the voice recognition capability of the system, and the other time they used the keyboard only. The two sessions took place in different days and in varying order.

The text to be dictated, called T in the following, consists of 553 total words, of which 290 were different, and of 3413 total characters. The number of words not included in our 20000-word vocabulary is 12: this means that the coverage of the text is 97.8%, about 1% higher than the average value computed on a large database of texts extracted from

the same newspaper. The perplexity of the language model on this text was only only slightly higher than the average perplexity for texts from the newspaper. These data suggest that T is statistically representative of the texts to which the prototype is aimed.
During the experiments, the workstation recorded in detail the behavior of the user, by keeping trace of: duration of the session; words uttered to the system in normal and in single-character mode; commands given by voice; keys pressed for character input, text manipulation, cursor movement; number of times the microphone was switched on and off.
A questionnaire was submitted to all participants to the experiment, in order to record their background in the use of keyboard and of voice recognition, their habits and wishes regarding text input, and their impressions and opinions about the usage of the system.

Participants to the experiments were 10 employees of IBM Rome Scientific Center. The sample was too small to yield statistically reliable conclusions, but still allowed some interesting observations. All participants had several years of experience of electronic text editors and used heavily the keyboard in their everyday work. Such a group of users represents an especially severe test for speech input, because of its out-of-average skills with typing.

The users can be divided into three groups according to their previous experience with voice input and to their knowledge of professional typing:

- **A** users who have some previous experience of voice input and who need to look at the keyboard when typing (three persons);
- **B** users who have no previous experience of voice input and who need to look at the keyboard when typing (five persons);
- **C** users who have no previous experience of voice input and who don't need to look at the keyboard when typing (two persons).

All users preferred to input the text in a raw way first, and then revised it and made corrections. We measured the following values:

Tag	Meaning
IT	Input Time, taken by first raw input of text;
RT	Revision Time, taken by revision and correction of text;
TT	Total Time for input and correction of text;
IE	Input Errors (percent fraction of wrong words after first input);
NE	Net Input Errors, i.e. wrong words due to speaking, typing or recognition errors, and not due to the absence of the dictated word from the recognizer vocabulary;
FE	Final Errors, i.e. wrong words due not to correcting.

The following table shows the above listed average values for the three groups, for voice and keyboard input (times are in minutes):

Table 6. Voice and keyboard input. The table shows the average values for the three groups (time in minutes). See text for tag description.

Group	Mode	IT	RT	TT	IE	NE	FE
A	VOICE	13.0	9.0	22.0	6.5	3.3	0.5
A	KEYB.	21.3	6.7	28.0	2.5	2.5	1.2
B	VOICE	17.0	17.3	34.3	8.5	5.8	1.5
B	KEYB.	23.0	6.0	29.0	1.3	1.3	0.7
C	VOICE	20.5	19.5	40.0	8.8	6.1	1.5
C	KEYB.	16.5	5.5	22.0	0.5	0.5	0.1

For all speakers, except professionally trained typists (group C), text input is faster by voice than by keyboard, even if they are using a speech recognizer for the first time. Users belonging to group C typed at a rate of 212 keys per minute, and made very few errors. We observed that all users stopped many times when dictating, in order to see what was being transcribed on the screen (the microphone was switched on and off 21 times, on average). This habit was less frequent at the end of the sessions, when users trusted the system more. This behavior was observed even in users of group A, who dictated at an overall rate of 39 words per minute, while other experiments, performed later by users who had acquired much more experience with the dictation machine, showed that a rate of 70 words per minute can be achieved. The word rate achieved in the experiments by speakers of group A by dictation was anyhow higher than that achieved by professionally trained typists when using the keyboard.

The number of errors after the first input of the text was higher for voice input than for keyboard input. This is reflected by the longer time taken by revision and correction. Some figures which describe user behavior in this phase are:

Tag	Meaning
DW	number of *Delete-Word* commands given;
DC	number of *Delete-Character* commands given;
MC	number of *Cursor-Movement* commands given;
OC	number of other commands given.

Average values are reported in the following table:

Table 7. Revision and correction of text. The table shows the average values for the given commands. See text for tag description.

Mode	DW	DC	MC	OC
VOICE	23	200	900	234
KEYB.	1	70	279	80

Users of group A were especially more efficient in the revision task, because users of groups B and C were experiencing voice editing commands for the first time and were brought to over-experiment with them.

Text revision seems the task which can benefit more from user experience and from improvements to the user interface (as well as from higher recognition accuracy). Errors found in a text input by voice are of a different kind than those produced using the keyboard: all the words transcribed by the system belong to the vocabulary. A spelling checker would be of little help. The system could provide instead, for each recognized word, upon request, a list of words very likely to be confused with it.

The indication that voice input is easier to learn and less tiring than traditional keyboard input is suggested by the answers to the questionnaire. All users found more pleasure and satisfaction in the usage of voice rather than keyboard. 60% of the subjects said that voice editing commands are more natural and easier to learn than keyboard commands, while 20% found no difference. The voice editor, as a whole, was rather simpler and more natural than traditional editors and word processors. All users learned in few minutes to insert pauses between words.

This preliminary study on the usage of a voice-activated text editor indicated that large-vocabulary speech recognition can offer a very competitive alternative to traditional text entry. Future studies on the usage of the voice-activated text editor will address the behavior of users who gained more experience in the tool, and of users who are not accustomed to word processing. Dictation for text creation will also be investigated.

References

[1] A. Averbuch et al., **Experiments with the Tangora 20000 Word Speech Recognizer,** *IEEE Int. Conf. on Acoust., Speech and Sig. Proc.*, Dallas, TX, April 1987, pp. 701-704.

[2] G. Shichman et al., **An IBM PC Based Large-Vocabulary Isolated-Utterance Speech Recognizer,** *IEEE Int. Conf. on Acoust., Speech and Sig. Proc.*, Tokyo, April 1986, pp. 53-56.

[3] F. Jelinek, **The development of an experimental discrete dictation recognizer** *Proceedings of IEEE*, vol. 73, no. 11, November 1985, pp. 1616-1624.

[4] P. D'Orta, M. Ferretti, S. Scarci, **Phoneme Classification for Real Time Speech Recognition of Italian,** *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, Dallas, TX, April 1987, pp. 81-84.

[5] P. D'Orta, M. Ferretti, S. Scarci, **Fast Speaker Adaptation for Large-Dictionary Real-Time Speech Recognition,** *IEEE Workshop on Speech Recognition*, Arden House, Harriman, NY, May 31-June 3, 1988.

[6] M. Brandetti, P. D'Orta, M. Ferretti, S. Scarci, **Building Reliable Large Speech Databases: an Automated Approach,** *EUSIPCO-88*, Grenoble, September 5-8, 1988.

[7] S. Scarci, S. Taraglio, **Automatic Phonetic Transcription for Large-Vocabulary Speech Recognition,** *Speech 88, Seventh FASE Symposium*, Edinburgh, 22-26 August 1988.

[8] M. Brandetti, P. D'Orta, M. Ferretti, S. Scarci, **Experiments on the Usage of a Voice-Activated Text Editor,** *Speech 88, Seventh FASE Symposium*, Edinburgh, 22-26 August 1988.

[9] P. D'Orta, M. Ferretti, A. Martelli, S. Melecrinis, S. Scarci, G. Volpi, **A Speech Recognition System for the Italian Language,** *ICASSP 1987*, Dallas, pp. 841-843.

[10] F. Jelinek, **A Fast Sequential Decoding Algorithm Using a Stack,** *IBM Journal of Research and Development*, vol. 13, November 1969, pp. 675-685.

[11] L.R. Rabiner, B.H. Huang, **An Introduction to Hidden Markov Models,** *IEEE ASSP Magazine*, no.1, 3 (January 1986), pp. 4-16.

[12] J.D. Ferguson, **Hidden Markov Analysis: an Introduction,** *Hidden Markov Models for Speech*, IDA-CRD, Princeton, October 1980.

[13] P. D'Orta, M. Ferretti, A. Martelli, S. Melecrinis, S. Scarci, G. Volpi, **Large-Vocabulary Speech Recognition: a System for the Italian Language,** *IBM Journal of Research and Development*, Vol. 32, No. 2, March 1988, pp.217-226.

[14] G.D. Forney, **The Viterbi Algorithm,** *Proceedings of the IEEE*, vol. 61, no. 3, March 1973, pp. 268-278.

[15] P. D'Orta, M. Ferretti, S. Scarci, **Language-Specific Knowledge in the Probabilistic Approach to Speech Recognition,** *EUSIPCO-88*, Grenoble, September 5-8, 1988.

[16] R. Carlson, et al., **Phonetic and Orthographic Properties of the Basic Vocabulary of Five European Languages,** *STL-QPSR, KTH*, Stockholm, 1985, pp. 63-94.

[17] L.R. Bahl, F. Jelinek, R.L. Mercer, **A Maximum Likelihood Approach to Continuous Speech Recognition,** *IEEE Trans. on PAMI*, vol. PAMI-5, no. 2, 1983, pp. 179-190.

[18] P. D'Orta, **Acoustic Discrimination among Words Based on Distance Measures,** *European Conference on Speech Technology*, Edinburgh, Sep. 1987, vol. 2, pp. 329-332.

[19] S. Katz, **Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer,** *IEEE Trans. on Acoust., Speech and Sig. Proc.*, vol. ASSP-34, no. 3, March 1987, pp. 400-401.

[20] F. Jelinek, R.L. Mercer, L.R. Bahl, J.K. Baker, **Perplexity - a Measure of Difficulty of Speech Recognition Tasks,** *94th Meeting Acoustical Society of America*, Miami Beach, December 1977.

[21] P. D'Orta, M. Ferretti, G. Maltese, S. Scarci, **Analisi automatica di testi per la costruzione di modelli della lingua italiana con applicazione al riconoscimento della voce,** *Atti del Convegno AICA*, Cagliari, Settembre 1988.

[22] R. Schwartz, Y. Chow, F. Kubala, **Rapid Speaker Adaptation Using a Probabilistic Spectral Mapping,** *IEEE Int. Conf. on Acoust., Speech and Sig. Proc.*, Dallas, April 1987, pp. 633-636.

[23] K. Shikano, K. Lee, R. Reddy, **Speaker Adaptation through Vector Quantization,** *Tech. Rep. CMU-CS-86-160, Carnegie Mellon University*, 1986.

[24] D. Burton, J. Shore, **Speaker-Dependent Isolated Word Recognition Using Speaker-Independent Vector Quantization Codebooks Augmented with Speaker-Specific Data,** *IEEE Trans. on ASSP*, Vol. ASSP-33, no. 2, 1985, pp. 440-442.

[25] R. Stern, M. Lasry, **Dynamic Speaker Adaptation for Feature-Based Isolated Word Recognition,** *IEEE Trans. on ASSP*, Vol. ASSP-35, no. 6, 1987, pp. 751-763.

[26] F. Jelinek, R.L. Mercer, **Interpolated Estimation of Markov Source Parameters from Sparse Data,** *Pattern Recognition in Practice*, E.L.Gelsema and L.N. Kanal, Ed., North-Holland, New York, 1980, pp. 381-387.

[27] G. Shichman, **Personal Instrument (PI) - A PC-based signal processing system,** *IBM Journal of Research and Development*, vol. 29, no.2, March 1985, pp. 158-169.

[28] M. Wagner, **Labelling of Continuous Speech with a Given Phonetic Transcription Using Dynamic Programming Algorithms,** *IEEE 1981 International Conference on Acoustics, Speech and Signal Processing.*

[29] H. C. Leung, V. W. Zue, **A Procedure for Automatic Alignment of Phonetic Transcriptions with Continuous Speech,** *IEEE International Conference on Acoustics, Speech and Signal Processing.* San Diego, CA, April 1984, 4.7.

[30] H. Hohne, C. Coker, S. E. Levinson, L. R. Rabiner, **On Temporal Alignment of Sentences of Natural and Synthetic Speech,** *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. ASSP-31, No. 4, August 1983, pp. 807-813

[31] R. Leonard, **A Database for Speaker-Independent Digit Recognition,** *IEEE International Conference on Acoustics, Speech and Signal Processing.* San Diego, CA, April 1984, 4.7.

[32] L.R. Bahl, R. Bakis, P.V. de Souza, R.L. Mercer, **Polling: A Quick Way to Obtain a Short List of Candidate Words in Speech Recognition,** *IEEE International Conference on Acoustics, Speech and Signal Processing.* New York, April 1988, 36.S11.

[33] R. Carlson, B. Granstroem, **A Text-to-Speech System Based Entirely on Rules,** *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing,* Philadelphia, PA, April 1976.

[34] D. H. Klatt, **Structure of a Phonological Rule Component for Synthesis-by-Rule Program** *IEEE Trans. on Acoust., Speech and Sig. Proc.,* vol. ASSP-24, no. 5, 1976, pp. 391-398.

[35] T. J. Sejnowski, C. R. Rosenberg, **Parallel Networks that Learn to Pronounce English Text,** *Complex Systems,* 1 (1987), pp. 145-168.

[36] J. Leggett, G. Williams, **An Empirical Investigation of Voice as 'an Input Modality for Computer Programming,** *Int. J. Man-Machine Studies* vol. 21, 1984, pp. 493-520.

[37] J. D. Gould, J. Conti, and T. Hovanyecz, **Composing Letters with a Simulated Listening Typewriter,** *Communications of ACM,* vol. 26, no.4, April 1983, pp. 295-308.