Multilevel decoding for Very-Large-Size-Dictionary speech recognition

by Bernard Mérialdo

An important concern in the field of speech recognition is the size of the vocabulary that a recognition system is able to support. Large vocabularies introduce difficulties involving the amount of computation the system must perform and the number of ambiguities it must resolve. But, for practical applications in general and for dictation tasks in particular, large vocabularies are required, because of the difficulties and inconveniences involved in restricting the speaker to the use of a limited vocabulary. This paper describes a new organization of the recognition process, Multilevel Decoding (MLD), that allows the system to support a Very-Large-Size Dictionary (VLSD)—one comprising over 100 000 words. This significantly surpasses the capacity of previous speech-recognition systems. With MLD, the effect of dictionary size on the accuracy of recognition can be studied. In this paper, recognition experiments using 10 000- and 200 000-word dictionaries are compared. They indicate that recognition using a 200 000-word dictionary is more accurate than recognition using a 10 000-word dictionary (when unrecognized words are included in the error rate).

Introduction

The work described in this paper is part of an IBM France Scientific Center research project that was begun in 1982. The research is focused on the feasibility of a Voice-Activated Typewriter (VAT) for French. The task of a VAT is to transcribe speech into correctly spelled text. A VAT does not have to understand what the user dictates, at least for those portions of the text that are not semantically ambiguous. But at the same time, considering the variety of situations and topics where dictation can be used, the user must not be restricted to a limited subset of a natural language, for either vocabulary or syntax.

Two different approaches in the design of a dictation system are reasonable, based on its planned use:

1. The system is to be used in a given context—the dictation of letters inside a given company, for example. In this case, it is possible to tailor the vocabulary to the words most frequently used in the kinds of activities that predominate in the company.
2. The system is not dedicated to a specific kind of activity and should serve a large variety of users, without special adaptation of the vocabulary or syntax. That is, it is a general-purpose system.

The first approach makes the recognition problem easier, once the task has been defined. But the system has to be modified when the task changes, which may be difficult and expensive (if, for example, new task-related texts must be processed to construct a new vocabulary).

The second approach makes the recognition problem harder, but the system does not have to be changed for each task. One obvious price to pay for this facility is that the vocabulary must be larger.

A number of speech-recognition groups are becoming increasingly interested in large vocabularies (i.e., vocabularies of more than 10000 words*). Leading the field has been the recognizer developed by the Speech Recognition Group at the IBM Thomas J. Watson Research Center in Yorktown Heights, New York [1, 2]. This system recognizes sentences uttered in isolated-word mode (i.e., with a brief pause after each word), using an English dictionary of 20000 words. It works in real time and is implemented on an IBM Personal Computer AT® [3] equipped with several specialized processors. Word-recognition accuracy is about 95% (assuming that all the words are in the dictionary). The present work is very much inspired by the techniques invented by the Yorktown group.

Other groups are also working on large vocabularies, but most of the studies concern recognition of homophone sets rather than words as a part of sentence recognition (in other words, linguistic constraints on sequences of words are not taken into account). Zue and his coworkers [4, 5] have studied the use of broad phonetic classes and stress to reduce the search to a subset of a 20000-word dictionary. Gauvain [6] has compared the performance of word and syllable templates in the recognition of a 10000-homophone-set French dictionary using dynamic-programming techniques. Recognition accuracy was 94% when using word templates and 88% when using syllable templates. Gupta et al. [7] have studied a special class of Markov models to recognize items from a 60000-homophone-set English dictionary, spoken in isolated mode. Recognition accuracy varied from 52% to 76%, depending on the choice of model.

Our project's goals include the use of a Very-Large-Size Dictionary (more than 100000 words), where we can include as many words as we want. In fact, although it is known that many speakers will never utter more than a few thousand different word sets, it is impossible (or rather, we don't know how) to guess these words in advance; they are known only with hindsight. Since we do not want to build a system that recognizes what a given user has said up to now, but rather a system able to recognize what he will say in the future, we define the vocabulary as comprising not just those words that a given user has uttered (active vocabulary) but also those words that any of a group of listeners is able to understand. A dictionary which encompasses this ability contains several tens of thousands of word sets (passive vocabulary), which, taking into account all inflectional and derivational forms, represent hundreds of thousands of words.

In order to study the interest of a Very Large (as opposed to a Large) Dictionary, we have performed some computations on the coverage of dictionaries. The coverage of a dictionary is the percentage of words in the text that are found in the dictionary. For speech recognition, that is the upper bound for the recognition rate of the system, since no present system is able to recognize a word that is not found in its dictionary. (It is notable that human beings are able to do so; that is, they can, for example, identify and spell correctly—or, at least, reasonably—a proper noun that they have never heard before.)

There are, in fact, two kinds of coverage, which are computed differently:

- To compute the *static* coverage, each occurrence of a word that is not in the dictionary is counted. This is "batch" recognition, where the dictionary remains the same throughout the course of recognition.
- To compute the *dynamic* coverage, only the first occurrence of a word that is not in the dictionary is counted. This is "interactive" recognition, where each new word is added by the user to the dictionary the first time it is encountered; after that, it is, of course, no longer considered new. (In a VAT, the user should have the option of repeating a word that has been misrecognized. If the word is still misrecognized, it can be spelled out orally or input via keyboard. It would be a simple matter to check whether a spelled-out or typed word is listed in the dictionary, and, if it is not, to prompt the user for its possible inclusion.)

Obviously, dynamic coverage is always at least as great as static coverage, and will be greater with the addition of each new word.

We compared the coverage of two dictionaries, one composed of the 20000 most frequent words of a one-million-word training corpus, the other composed of a full-sized dictionary of French containing 200000 words. The coverages were computed on a collection of 50000 words of

* In the context of speech recognition, the word *word* is, unfortunately, used in at least three significantly different ways; to eliminate this ambiguity, we will use different terms for each meaning.

Everyday usage counts a stem (or, in the terminology often used by computational linguists, a "baseform" or "lemma") and its inflected forms together as one word. By this method, *table* and *tables*, for example, would count as a single word. In this paper, we will use the term *word set* for this meaning.

Most people currently doing speech recognition recognize acoustic patterns. Since the body of all acoustic patterns to be recognized can be viewed as comprising the "dictionary" or "vocabulary," the acoustic patterns themselves are sometimes called "words." In this meaning, *voix* and *voie* (which are homophones in French, like *sea* and *see* in English) are considered the same word. When this is the intended meaning, we will use the term *homophone set*.

The most common usage in speech recognition is the one based on correct transcription: Something is recognized correctly if it is spelled correctly in the final text output. Thus here, *table* and *tables* would be counted as different words (thereby yielding a larger vocabulary); however, interchanging homophones such as *voix* and *voie* in the final text would be scored as an error. We will use the term *word* for this meaning—that is, a character string which is distinct in spelling from other character strings in the vocabulary.

text, different in nature from the texts used in training. The results are indicated in **Table 1**. In this computation, proper names in the text are counted as words. The words that are not in the 200000-word dictionary are either proper names or specialized technical terms.

Of course, these results depend partially on the data used, and in particular on the relation between the training corpus used to choose the most frequent words and the test text used to compute the coverage.

Table 1 Comparison of text coverage by two dictionaries.

Dictionary size (words)	*Text coverage* (%)	
	Static	*Dynamic*
20000	94.9	98.2
200000	97.5	99.5

Speech recognition in French

Some features of the French language raise particular problems for speech recognition.

French is a highly inflected language; i.e., a typical lemma is used to produce forms with different spellings according to context and the agreement rules that apply to it. A noun generally produces two inflections, singular and plural; an adjective four, differing in gender and number; and a verb around forty, based on person, tense, and mood. The average number of inflected forms per lemma is seven in French, as compared to two in English. This makes the total list of possible words rather large; for example, our full-sized dictionary of 200000 words corresponds to a list of about 45000 lemmas (inflections with very low frequency of use do not appear in our dictionary).

Many of these derivations lead to forms that are homophones. The singular and plural forms of a noun (*table* and *tables*, for example) are usually homophones, because the plural marker (in this case, the final *s*) is generally not pronounced. With the verbs of the first conjugation (the largest group of regular French verbs), the infinitive, the second person plural of the present indicative, and all four forms of the past participle are homophones: e.g., *passer* ('to pass'), *passez* ('you pass'), *passé*, *passés*, *passée*, *passées* ('passed'). Many other derivations are comparable. This makes recognition more difficult because in most cases there are no acoustic cues to distinguish among the different spellings, so they must all be processed at the linguistic level and disambiguated by linguistic constraints alone.

Liaison is a phonetic phenomenon that inserts a consonant sound between two consecutive spoken words. For example, *les arts* is pronounced "lézar" (or, using the International Phonetic Alphabet, [lezaʀ]), although the two words are pronounced "lé" ([le]) and "ar" ([aʀ]), respectively, in isolation. There are a limited number of consonants involved in this phenomenon, and they occur in precise cases. The recognition system has to take into account the fact that the pronunciation of some words will depend on this situation.

Another such case involves what is called in French *apostrophe*. A number of function words such as *la, le, de, se* elide their final vowel when the following word begins with a phonetic vowel. The vowel is replaced by an apostrophe in spelling and the two words are concatenated. For example, the sequence of words *le* ([lə]) and *art* ([aʀ]) becomes *l'art* and is pronounced "lar" ([laʀ]). For speech recognition, an additional problem here is that the pronunciation of these short words is significantly affected by coarticulation with the first vowel of the second word. It is therefore difficult to recognize them without taking the vowel into account.

- *Syllable approach*

A long-standing problem in speech recognition is the choice of the basic unit to be used at the acoustic level. Reasonable units include phonemes (linguistically distinctive elementary sounds), allophones (phonetic variants of phonemes), subwords, and words. Phonemes are attractive, because a small set (less than 50 for most languages) is sufficient; but they are difficult to recognize accurately from the acoustic signal, because one phoneme may have very different acoustic characteristics (i.e., different allophones), depending on the context of the pronunciation. Larger units like subwords or words are interesting because they provide constraints on possible sequences of phonemes and can take into account some coarticulation phenomena. The difficulty is that there are many more of them. (Allophones are also numerous and, when used as units, do not facilitate the use of linguistic constraints and coarticulation data.)

Our approach is to consider the syllable as the basic unit for acoustic recognition. There are several reasons for this choice. Syllables are longer than phonemes, so it is easier to recognize them from the acoustic signal. No more than 5200 different phonetic syllables are required for a complete description of our 200000-word dictionary.

Another advantage of the syllable is that the problems of liaison and apostrophe in sentences can be handled easily, whereas word templates would mean having references for all possible liaison and apostrophe forms. If we take, for example, the word *artiste* ([aʀtist]), using word templates would require the addition of the following pronunciations:

- "lartist" ([laʀtist]) found in *l'artiste.*
- "dartist" ([daʀtist]) found in *d'artiste.*
- "nartist" ([naʀtist]) found in *un artiste.*
- "zartist" ([zaʀtist]) found in *les artistes.*
- and a few more . . . .

Since approximately one fourth of all French words begin

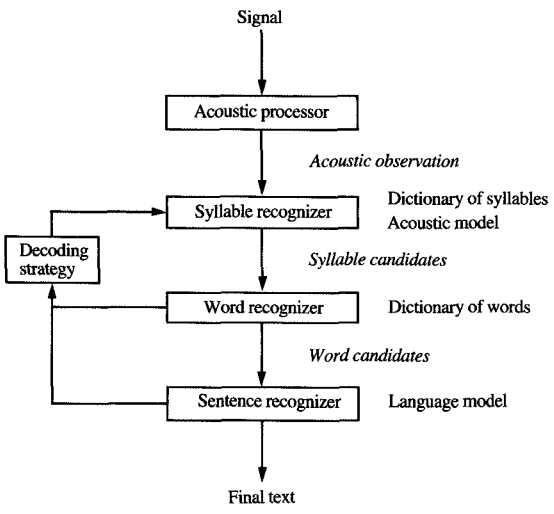

Figure 1

Diagram of MLD.

with a phonetic vowel, this would greatly multiply the number of word templates. With the syllabic approach, liaison and apostrophe require the introduction of only 1200 new syllables. Thus, 6400 phonetic syllables cover normal speech for a 200 000-word dictionary, including liaison and apostrophe.

- *Mode of pronunciation*

A common constraint in the current state of the art in speech recognition is asking the speaker to make a pause after every word. This is called isolated-word mode (IW). It facilitates recognition because coarticulation between words is reduced and pauses are strong cues in the acoustic decoding.

In our case, although our ultimate goal is continuous speech, at the moment we ask the speaker to pause between syllables, because of our choice of the syllable as the basic acoustic unit. For example, the sequence

les enfants ...

will be pronounced as

"lé zan fan ..." ([le zã fã ...]).

That is, the speech is produced in isolated-syllable (IS) mode. (It should be noted that making pauses between syllables appears not to be very difficult in French, perhaps because of the stability of word stress.) On the one hand, IS mode shares some of the difficulty of continuous-speech recognition, since (as opposed to IW mode) each syllable potentially signals a word boundary. On the other hand, this is counterbalanced by the fact that IS mode has fewer contextual phonetic effects to deal with and more frequent silences to serve as anchor points for the decoder.

For example, the sequence of phonetic syllables "lé zar" ([le zaʀ]) can correspond to the sequence of words *les arts* ('the arts') or the single word *lézard* ('lizard').

The restriction to IS mode is only a temporary constraint that facilitates development of and experimentation on recognition algorithms. It also lowers the amount of computation needed for the recognition. It is, of course, our intention to remove this constraint in the future.

Information-theory approach to speech recognition

Early work in speech recognition using information-theoretic techniques includes [8] and [9]. In accordance with the information-theory approach taken by Jelinek and his colleagues [10–12], the problem of recognizing the sentence W that corresponds to a given utterance A (also called the *acoustic observation*) can be recast as the problem of maximizing the product:

$$p(A \mid W) \cdot p(W).$$

To implement a speech-recognition system, one must therefore define

- The *acoustic observation*, i.e., what parameters are extracted from the acoustic signal captured by the microphone.
- The *acoustic model*, which defines $p(A \mid W)$ and models how sentences are pronounced in terms of this observation.
- The *language model*, which defines $p(W)$ and thus establishes which sequences are allowed by the system (sentences with zero probability will be prohibited).
- A decoding strategy to find the sentence that comes closest to realizing this maximum, since an exhaustive search is generally not feasible because of the very large number of possibilities.

- *Multilevel decoding*

We have formulated a method we call multilevel decoding (MLD), which is an organization of the recognition process that makes access to a Very-Large-Size Dictionary (VLSD) possible. Its major distinctive feature is the use of a syllable level. Although syllables have been considered as acoustic units in other speech-recognition efforts [13, 14], this is the first time that they have been integrated into a complete system and used with a VLSD.

As shown in **Figure 1**, MLD organizes recognition into four stages:

1. The speech signal is processed to provide the acoustic observation.
2. A syllable recognizer uses the acoustic model to build a list of the syllables having the highest probability of matching a part of the observation.
3. A word recognizer uses these syllables to build a list of word candidates.
4. A sentence recognizer uses the word candidates and the language model to build possible sentences.

The flow of data through these stages is mostly one-way, each level processing data from the previous level and transmitting the results to the next. The only feedback is in the decoding strategy which, depending on the status of the word and sentence recognizers, decides to which part of the utterance the syllable recognizer should be applied next.

The components of this process are described in greater detail in the following subsections.

• *The acoustic processor*

Our acoustic processor is based on a standard centisecond vector quantizer. The microphone is connected to an analog-to-digital converter that samples the speech signal at 10 kHz with 12-bit quantization. We consider consecutive windows of 128 samples each (12.8 ms). For each window we use a Hamming window and a fast Fourier transform to compute the log power spectrum of the signal. This spectrum is projected on 20 frequency bands arranged according to a mel scale (linear up to 1000 Hz, logarithmic above). The result is an acoustic vector with 20 coordinates.

In the training phase, 30000 vectors are clustered into 200 classes by a "k-means" algorithm [15]. Then, both in training and decoding, the acoustic vectors are replaced by the number of the class to which they belong. Each window of 12.8 ms of signal is then replaced by a number in the range 1 to 200, which serves as its label. The acoustic observation comprises the sequence of these labels.

Our future plans include the implementation of the Yorktown acoustic processor [1, 16], which uses a 20-kHz sampling rate and an ear model to compute the acoustic parameters.

• *The phonetic model*

The acoustic model is based on a set of phonetic Markov machines which have been used in previous work [17]. A system with 40 "phonemes" has been defined. They comprise the classical phonemes of French, plus some specific acoustic events, such as silence, bursts, and end of syllable. A Markov machine is associated with each "phoneme." These machines all have the same structure, as shown in **Figure 2**.

The machine in Figure 2 is an example of one of the models introduced by the Yorktown group. It has seven states, three null transitions (i.e., those that produce no

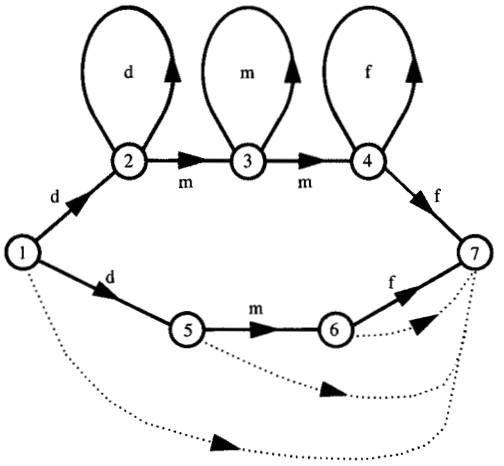

Figure 2

Phonetic Markov machine.

labels), and ten non-null transitions. Each non-null transition τ may produce any label l according to a probability distribution $q_\tau(l)$. Each transition starting from a state s has the probability $q_s(\tau)$ of being taken when the machine is in the state s. States 1–5–6–7 are used for occurrences of lengths 0, 1, 2, or 3 of the phoneme; states 1–2–3–4–7 are used for longer occurrences. To reduce the number of parameters, the emission probabilities are tied; that is, we impose the constraints

$$q_{\tau_{1-2}}(l) = q_{\tau_{2-2}}(l) = q_{\tau_{1-5}}(l) = q_d(l)$$

$$q_{\tau_{2-3}}(l) = q_{\tau_{3-3}}(l) = q_{\tau_{3-4}}(l) = q_{\tau_{5-6}}(l) = q_m(l) \quad \text{for every label } l.$$

$$q_{\tau_{4-4}}(l) = q_{\tau_{4-7}}(l) = q_{\tau_{6-7}}(l) = q_f(l)$$

Every phonetic machine has the same structure (except for the silence machine, which consists of a single loop); only the probabilities of transitions and production of labels differ from one machine to another. These probabilities are estimated during the training phase using an algorithm known as the "Forward–Backward" algorithm [18]. Training, in our case, is performed on 400 short sentences pronounced by a single speaker in IS mode.

Each phonetic syllable has a representation as a sequence of phonemes and is associated with a syllable machine obtained by concatenating the corresponding phonetic machines.

• *The syllable recognizer*

Given an acoustic observation A_1^T, we have to compute the list of the most probable syllables (candidates) matching the

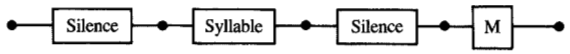

Figure 3

Syllable production model.

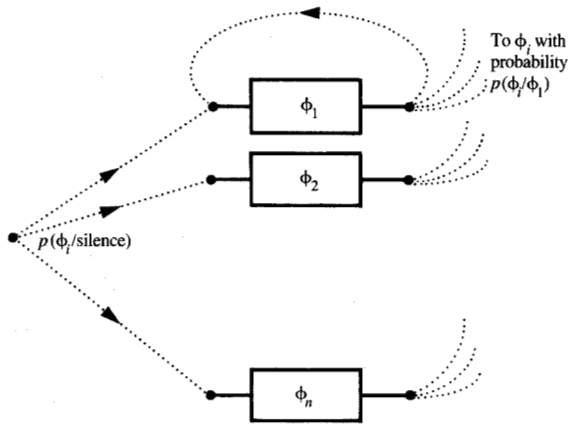

Figure 4

Looped Phonetic Model.

acoustic observation from a given time frame t. Also, for each candidate, we would like a score indicating how probable the syllable is, and an ending time frame indicating the end of the utterance of the syllable. For simplicity, we assume that $t = 1$.

So, given A_1^T, we want to compute for each syllable S the following probabilities:

- $p(S$ matches the beginning of $A_1^T)$, which is used to choose the best syllables.
- p(the end of the utterance of S is t'), which is used to choose among possible endings.

Let $\Omega(S)$ denote the event "the utterance begins with S." We then have

$$p(S \text{ matches the beginning of } A_1^T) = p[\Omega(S) \mid A_1^T]$$

$$= \frac{p[A_1^T \mid \Omega(S)] \cdot p[\Omega(S)]}{p(A_1^T)}.$$

Because we want to rely only on acoustics to choose the best syllables, we assume here that all syllables are equiprobable, so that all $p[\Omega(S)]$ are equal. The factor $p(A_1^T)$ does not depend on S, so that the choice between syllables depends only on $p[A_1^T \mid \Omega(S)]$.

To compute this probability, we build a Markov model for an utterance that begins with S. As we are in IS mode, we can say that such an utterance is composed, in sequence, of

- A pause (silence).
- The utterance of the syllable S.
- A pause (silence).
- The rest of the utterance.

A Markov model of the production of this utterance is the concatenation of the silence machine, followed by the syllable machine, followed by the silence machine, followed by a model M of the production of the rest of the utterance (**Figure 3**).

We consider here as model M, the *Looped Phonetic Model* (LPM) with diphone constraint (see **Figure 4**). The LPM with diphone constraint is constructed by placing copies of all phonetic machines in parallel and connecting their final states to the initial states by null transitions. The null transition from the final state of phoneme ϕ_i to the initial state of phoneme ϕ_j is assigned the probability $p(\phi_j \mid \phi_i)$ that phoneme ϕ_j follows phoneme ϕ_i (this probability is computed on the list of possible syllables). We set to zero the probability of omission of every phonetic machine, so that there are no null cycles in the LPM (this avoids some problems in the succeeding computations). We also add an initial state I, connected to the initial state of each phoneme ϕ_i by a null transition with probability $p(\phi_i \mid \text{silence})$, because we assume that this model produces an utterance after a pause. The final state of the model is the final state of the copy of the silence machine.

The LPM represents the production of utterances by all possible sequences of phonemes, the contribution of each sequence being weighted by its probability according to a first-order prediction on phonemes.

Now we have

$$p[A_1^T \mid \Omega(S)] = p(A_1^T \mid \text{silence} \cdot S \cdot \text{silence} \cdot \text{LPM}).$$

The probability of emission of a given observation by a Markov model is the sum of the probabilities of all paths in this model that produce this observation (a path is a sequence of consecutive transitions). So, if $E(A_1^T)$ is the set of all paths in the model "silence · S · silence · LPM" that produce A_1^T, then

$$p(A_1^T \mid \text{silence} \cdot S \cdot \text{silence} \cdot \text{LPM}) = \sum_{x \in E(A_1^T)} p(x).$$

Each such path corresponds to a certain ending of the utterance of S, the instant where the path goes through the final state F_S of the syllable machine. This allows us to compute the probability that S matches the beginning of A_1^T and finishes at time t'. If $E(A_1^T, t')$ is the set of all paths that produce A_1^T and pass through the final state F_S at time t', then

$$p(\text{the utterance of } S \text{ finishes at } t' \mid S) = \frac{\sum_{x \in E(A_1^T, t')} p(x)}{\sum_{x \in E(A_1^T)} p(x)}.$$

The previous computations allow us to compare

- Different syllables to see which ones best match the observation and are to be kept as candidates.
- For a given syllable, different possible end times of the utterance and, thereby, the selection of the most probable one.

We now examine how to perform these computations practically. Instead of considering each syllable in isolation, we take advantage of the phonetic description of the syllables to merge all the syllable machines into a single model, structured as a tree, the Syllabic Tree (ST). The ST is obtained by placing all syllable machines in parallel between an initial state I and a final state F, and merging from left to right all copies of identical phonetic machines that start from the same node.

By concatenating a silence machine, the ST, another silence machine, and the LPM, we obtain the Syllabic Tree Matching Model (see **Figure 5**).

Next we match this model to the observation; i.e., we compute the probability of being in a given state of the model and having produced a part of the observation. The method used to do this is called the forward pass of the Forward–Backward algorithm. First, we define a partial order of the states by

$s < s'$ iff there exists a null path from s to s'

(this is effectively a partial order since there are no null cycles in our models). Then, we define $\alpha(s, t)$ as the probability of being in state s and having produced observation A_1^t. The αs can be computed by the following recursions:

$$\begin{cases} \alpha(\text{initial state}, 0) = 1, \\ \alpha(s, 0) = 0 \text{ if } s \text{ is not the initial state}, \\ \alpha(s, t) = \sum_{s'} \alpha(s', t-1)\, q(s', a_t, s) \\ \qquad\qquad + \sum_{s' < s} \alpha(s', t)\, q(s', \varnothing, s), \end{cases}$$

where $q(s', a, s)$ is the probability of going from state s' to state s and producing the label a; a_t is the tth label of the observation; and $q(s', \varnothing, s)$ is the probability of going from state s' to state s without producing a label.

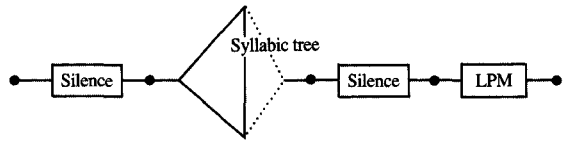

Figure 5

Syllabic Tree Matching Model.

The fact that there are no null cycles in the model guarantees that we can order the states to allow the computation of $\alpha(s, t)$ from the αs at time $t-1$ and from previously computed values of the αs at time t.

We also perform a backward pass that computes the probability of being in a given state and having produced the end of the utterance. The βs are defined as

$$\begin{cases} \beta(\text{final state}, T) = 1, \\ \beta(s, T) = 0 \text{ if } s \text{ is not the final state}, \\ \beta(s, t) = \sum_{s'} \beta(s', t+1) \cdot q(s, a_{t+1}, s') \\ \qquad\qquad + \sum_{s' > s} \beta(s', t) \cdot q(s, \varnothing, s'). \end{cases}$$

Now the probability that a syllable S matches the beginning of the utterance, as defined previously, is just the sum

$$p[A_1^T \mid \Omega(S)] = \sum_t \alpha(F_S, t) \cdot \beta(F_S, t),$$

where F_S is the final state of the syllable in the Syllabic Tree. The probability that the end of the utterance of S is at time t' is

$$p(\text{the utterance of } S \text{ finishes at } t' \mid S) = \frac{\alpha(F_S, t') \cdot \beta(F_S, t')}{\sum_t \alpha(F_S, t) \cdot \beta(F_S, t)}.$$

In practice, we do not perform the computation of αs for the entire utterance, because this would be very time-consuming and we are only interested in the match of the first syllable uttered. To reduce the computation, syllables corresponding to αs with very low values (compared to αs on other states at the same time t) are abandoned. We can thus prune the ST and continue the computation for the best candidates only. As we match the model to the observation (as t goes from 1 to T), the highest $\alpha(s, t)$s start in the first silence machine, then spread into the Syllabic Tree, and finally accumulate in the second silence machine and in the LPM. When t is high enough, all significant $\alpha(s, t)$s will be

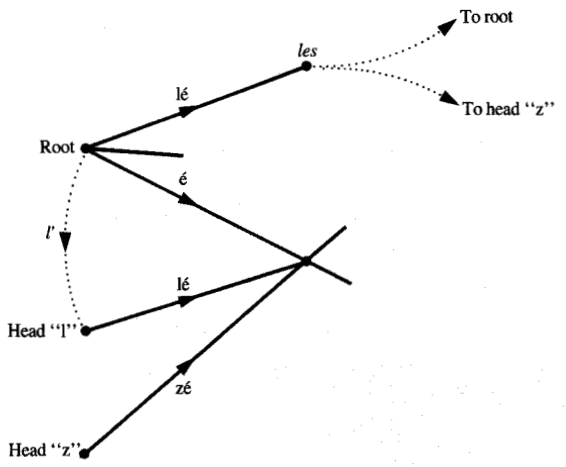

Figure 6

Head creation in W/S tree.

out of the Syllabic Tree. Then, instead of continuing the computation up to the end of the observation, we stop at that time, which we call t_F. We perform the backward pass from this time, using the initialization

$$\beta(s, t_F) = \alpha(s, t_F)$$

and the recursion already mentioned. (The choice of this initialization is heuristic and favors "good" paths which give high α values.)

Note that the only dependency of this syllable recognizer on the IS mode is in the presence of the silence machines in the Syllabic Tree Matching model. If these machines were removed, the same algorithm could be applied to match syllables in continuous speech. We plan to do so in the future.

- *The word recognizer*

The word recognizer keeps lists of partial word hypotheses and updates them with the lists of syllable candidates provided by the syllable recognizer.

This update is dependent on the dictionary, which has over 200000 entries. Each entry is composed of

- The spelling of a word.
- One or more pronunciations, transcribed as sequences of phonetic syllables.
- The possible parts of speech of this word and their corresponding frequencies.

For speech recognition, this dictionary has to be indexed on the basis of phonetic information, because we want to recover all words corresponding to a given sequence of phonetic syllables. We structure the phonetic part of the dictionary as a tree that describes words in terms of sequences of phonetic syllables (in a way similar to the Syllabic Tree that describes syllables in terms of phonetic machines). This tree is called the W/S tree. The arcs are the phonetic syllables. The nodes correspond to partial-word hypotheses. The arcs leaving a given node indicate which syllables can extend the partial word. The leaves in the W/S tree are the words themselves, and this is where we attach the information on spelling, parts of speech, and frequency.

To take liaison and apostrophe into account, the W/S tree is augmented by adding auxiliary entry points, called "heads," as shown in **Figure 6**. Each head corresponds to a consonant or consonant cluster (such as "l," "m," or "z") that may occur because of liaison or apostrophe. Each syllable S that starts at the root of the tree and begins with a vowel creates new syllables starting from each head. The new syllables are formed by the concatenation of the consonant of the head and the original syllable, and they all go to the final node of the original syllable. For example, the phonetic syllable "é" ([e]) will lead to the creation of syllables "lé," "mé," "zé" ... ([le, me, ze ...]) from different heads of the tree.

For liaison, each word (leaf of the tree) has a corresponding list of the possible heads which may start the next word. For example, the word *les* can be followed by words starting at the root of the tree or at the head corresponding to the liaison form with "z." It can also be followed by words starting at the root of the tree. For apostrophe, information at the root of the tree indicates which words may lead directly to a head of the tree. For example, the article or pronoun *le* will allow a jump from the root of the tree to the head corresponding to "l."

For the 200000-word dictionary, there are about 120000 nodes (partial words), about 340000 arcs (syllables starting from a node), and 10 different heads.

As stated earlier, the partial-word hypotheses are the nodes of the W/S tree. Each hypothesis has a beginning and an ending time frame corresponding to the part of the acoustic observation that the partial word matches. Considering partial-word hypotheses that end at a given time frame, the word recognizer calls the syllable recognizer to get a list of syllable candidates, with their score and ending time frame. Then the hypotheses are extended, by looking to see which syllables are valid arcs leaving the corresponding nodes in the W/S tree. This leads to new partial-word hypotheses, and sometimes to full-word hypotheses that are passed to the sentence recognizer. Full-word hypotheses are passed to the sentence recognizer together with their beginning, their ending time frame, and their acoustic probability (i.e., the probability of producing the corresponding part of the acoustic observation).

• *The language model*

The language model has been described elsewhere [19] and has been used for phonetic-to-text transcription. Let us just recall here its major features. It is based on a Markov model at the part-of-speech level. More precisely, we consider that sentences are messages from a source that emits words one after the other. So we can say that

$$p(W_1^n) = \prod_i p(W_i | W_1^{i-1}).$$

We make two approximations to estimate these probabilities:

- Reduction of the size of the context,

$$p(W_i | W_1^{i-1}) = p(W_i | W_{i-2}W_{i-1}).$$

- Introduction of the part-of-speech level,

$$p(W_i | W_{i-2}W_{i-1}) = p(W_i | p_i) \cdot p(p_i | p_{i-2}p_{i-1}),$$

where p_i is one of the parts of speech of the word W_i. The term $p(W_i | p_i)$ is a lexical probability that is attached to each word and stored in the dictionary; $p(p_i | p_{i-2}p_{i-1})$ is a contextual probability that is intended to model permissible part-of-speech sequences in sentences.

These probabilities are estimated as relative frequencies, using a training corpus. The contextual probability is computed as an interpolation of first-order and second-order frequencies.

• *The sentence recognizer*

The sentence recognizer keeps lists of partial-sentence hypotheses and updates them when word candidates are provided by the word recognizer.

Partial-sentence hypotheses are sequences of words W_1^n that match the acoustic observation from the beginning up to an ending time frame t. Each hypothesis is associated with a score, defined as the product,

$$p(A_1^t | W_1^n) \cdot p(W_1^n),$$

of the corresponding acoustic and linguistic probabilities.

When word hypotheses are found by the word recognizer, the sentence recognizer extends the partial-sentence hypotheses whose acoustic end is at the beginning of the word. This leads to a longer partial-sentence hypothesis, for which a new score is computed.

Eventually these longer hypotheses will match the entire utterance, in which case the score corresponds exactly to the product,

$$p(A | W) \cdot p(W),$$

that we wish to maximize, and the hypothesis is a candidate for the transcription of the utterance. The entire process of selecting syllable candidates and then word hypotheses generates a number of sentence hypotheses. The one corresponding to the highest product is kept as the sentence recognized by the system for this utterance.

• *The decoding strategy*

Decoding processes the utterance from left to right. It starts from an empty hypothesis and goes into the extension process until it reaches the end of the utterance.

The decoding strategy has to decide, according to the status of the word and the sentence recognizers (i.e., according to the existing hypotheses and their scores), to which part of the utterance the syllable decoder should be applied next.

Several different strategies are possible. The Yorktown group uses the stack decoding algorithm, which always extends the best nonextended hypothesis so far. In our case, we use a time-synchronous strategy which extends the shortest nonextended hypothesis. This strategy may lead to some extra work when compared with stack decoding, but it makes the management of partial hypotheses simpler, because when we extend the hypothesis ending at time t, we are sure that all shortest hypotheses have been processed, so that the list of hypotheses ending at time t is complete.

Recognition experiment

The MLD algorithm has been implemented on an IBM 4381 and runs in batch mode. It has been tested on a text composed of 79 short sentences, uttered in IS mode by the same speaker as in training. These sentences come from a set of letters that do not belong to the corpus used to select frequent words. The test text has 722 words and 1143 syllables.

We tested recognition using three different dictionaries:

- The first one (D_{10}) comprised the 10000 most frequent words, extracted from a large corpus. This yielded a Large dictionary which covers 94% of the test text (due to the small size of the text, static and dynamic coverage are equal).
- For the second (D_{10+43}), we added the 43 uncovered words of the text to D_{10}. This yielded a Large (artificially constructed) dictionary which, of course, covers 100% of the test text.
- The third one (D_{200}) is the full-sized dictionary, comprising 200000 words—a Very Large dictionary which also covers 100% of the test text.

Table 2 gives error rates (defined as the percentage of words not correctly recognized) for recognition experiments using the three dictionaries. These results show that recognition using D_{200} is 4.6% better than recognition using D_{10}. This improvement comes from two phenomena acting in opposition. On the one hand, D_{200} has a 6% better coverage than D_{10}. On the other hand, the greater size of D_{200} leads to more errors than D_{10}.

Table 2 Recognition with three dictionaries.

Dictionary	*Size* (words)	*Text coverage* (%)	*Error rates* (%)
D_{10}	10000	94	17.3
D_{10+43}	10043	100	10.6
D_{200}	200000	100	12.7

If we now compare recognition using D_{200} and D_{10+43} (both giving complete coverage), we see that the error rate of the former, despite its greater size, is only 2.1% more.

In other words, if we use a Very Large rather than a Large dictionary, we lose 2.1% because of a higher error rate, but more than make up for it by gaining 6% in coverage.

Obviously, the ideal dictionary would be D_{10+43}—small size but 100% coverage. Unfortunately, of course, it is impossible to construct such a dictionary independent of a particular text.

It should be noted that absolute values of the recognition rate must be considered with care when one system is compared to another, because they depend crucially on the conditions of the experiments. For example, the above results cannot be directly compared with those of Gauvain [6] since Gauvain's are in terms of homophone sets. Also, in his experiments, each word in the 10000-word dictionary has been uttered and is tested once, whereas our experiments deal with words in sentences, where frequent words (generally the shortest and the most ambiguous) occur several times. The same considerations apply to the work of Gupta et al., reported in [7].

The work at IBM Yorktown is similar enough to ours to allow meaningful comparison. Several factors may contribute to the difference between our 12.7% error rate and their 5% error rate:

- As described earlier, French is more phonetically ambiguous than English. In fact, half of the errors in our experiment are "linguistic" errors, where a homophone has been found but the wrong spelling has been chosen.
- Their 5% error rate is computed differently and does not include the effect of the coverage of their 20000-word English dictionary. This coverage is estimated to be 97.6% of text, which would give a total error rate of about 8%.
- Our part-of-speech-based language model is not as precise as the trigram language model of the Yorktown system (although our approach allows a VLSD to be handled easily, which is not the case with the trigram approach).
- Finally, our acoustic model may be more rudimentary because it uses a smaller number of machines and a smaller amount of training data.

Conclusion

In this paper, we have described a new organization for a speech recognizer based on a decomposition into syllable, word, and sentence levels. This new organization, called multilevel decoding, allows a Very-Large-Size Dictionary (in our case 200000 words) to be supported for speech recognition—a significant step toward the realization of practical voice-activated typewriters.

MLD has been implemented and tested. When comparing recognition with 200000-word and 10000-word dictionaries, we found that the 6% gain in coverage more than makes up for the 2.1% loss in recognition accuracy. This result supports the validity of VLSD for dictation systems.

As noted in the Introduction, the work reported in this paper is part of an effort toward achieving a VAT. Present research is focused on improving the quality of acoustic and language models (see Derouault's work [20], for example). We also plan to process continuous speech instead of constrained speech; and to use special processors to perform this expensive decoding in real time.

References and note

1. F. Jelinek, "The Development of an Experimental Discrete Dictation Recognizer," *Proc. IEEE* **73,** No. 11, 1616–1624 (November 1985).
2. A. Averbuch et al., "Experiments with the Tangora 20,000 Word Speech Recognizer," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing,* Dallas, April 1987, pp. 701–704.
3. Personal Computer AT is a registered trademark of International Business Machines Corporation.
4. D. P. Huttenlocher and V. W. Zue, "A Model of Lexical Access from Partial Phonetic Information," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing,* San Diego, April 1984, Vol. 2, No. 26.4.
5. A. M. Aull and V. W. Zue, "Lexical Stress Determination and Its Application to Large Vocabulary Speech Recognition," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing,* Tampa, April 1985, pp. 1549–1552.
6. J.-L. Gauvain, "A Syllable-Based Isolated Word Recognition Experiment," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing,* Tokyo, April 1986, pp. 57–60.
7. V. N. Gupta, M. Lennig, and P. Mermelstein, "Integration of Acoustic Information in a Large Vocabulary Word Recognizer," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing,* Dallas, April 1987, pp. 697–700.
8. N. Rex Dixon and C. C. Tappert, "Strategic Compromise and Modeling in Automatic Recognition of Continuous Speech: An Hierarchical Approach," *J. Amer. Soc. Cyber.* **1,** No. 2, 67–79 (1971).
9. J. K. Baker, "Stochastic Modeling for Automatic Speech Understanding," *Speech Recognition,* D. Raj Reddy, Ed., Academic Press, Inc., New York, 1975, pp. 521–542.
10. F. Jelinek, L. R. Bahl, and R. L. Mercer, "Design of Linguistic Statistical Decoder for the Recognition of Continuous Speech," *IEEE Trans. Info. Theor.* **IT-21,** No. 3, 250–256 (May 1975).
11. F. Jelinek, "Continuous Speech Recognition by Statistical Methods," *Proc. IEEE* **64,** 532–556 (April 1976).
12. L. Bahl, F. Jelinek, and R. Mercer, "A Maximum Likelihood Approach to Continuous Speech Recognition," *IEEE Trans. Pattern Anal. & Machine Intell.* **PAMI-5,** No. 2, 179–190 (March 1983).
13. Wayne A. Lea, "Speech Recognition: Past, Present, and Future," *Trends in Speech Recognition,* Wayne A. Lea, Ed., Prentice-Hall, Inc., Englewood Cliffs, NJ, 1980, pp. 39–98.

14. June E. Shoup, "Phonological Aspects of Speech Recognition," *Trends in Speech Recognition*, Wayne A. Lea, Ed., Prentice-Hall, Inc., Englewood Cliffs, NJ, 1980, pp. 125–138.
15. J. Makhoul, S. Roucos, and H. Gish, "Vector Quantization in Speech Coding," *Proc. IEEE* **73**, 1551—1558 (November 1985).
16. Jordan R. Cohen, "Application of an Adaptive Auditory Model to Speech Recognition," presented at the 110th Meeting, Acoustical Society of America, Nashville, TN, November 4–8, 1985; abstract in *J. Acoust. Soc. Amer.* **78**, Suppl. 1, S50 (Fall 1985).
17. H. Cerf-Danon, A.-M. Derouault, M. El-Bèze, B. Mérialdo, and S. Soudoplatoff, "Speech Recognition Experiment with 10,000 Word Vocabulary," presented at the NATO Advanced Institute on Pattern Recognition, Brussels, June 18–20, 1986, pp. 203–208.
18. L. Baum, "An Inequality and Association Maximization Technique in Statistical Estimation for Probabilistic Function of Markov Processes," *Inequality* **III**, 1–8 (1972).
19. A.-M. Derouault and B. Mérialdo, "Natural Language Modeling for Phoneme-to-Text Transcription," *IEEE Trans. Pattern Anal. & Machine Intell.* **PAMI-8**, No. 6, 742–749 (November 1986).
20. A.-M. Derouault, "Context-Dependent Phonetic Markov Models for Large Vocabulary Speech Recognition," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Dallas, April 1987, pp. 360–363.

Received March 19, 1987; accepted for publication October 19, 1987

Bernard Mérialdo *IBM France Scientific Center, 36 avenue Raymond Poincaré, 75116 Paris, France.* Dr. Mérialdo studied at the Ecole Normale Supérieure in Paris. In 1979, he received a "thèse de 3^{e} cycle" degree from the University of Paris 6 for his work on automatic theorem proving. From 1980 to 1981, he was an assistant professor of mathematics at the University of Rabat in Morocco. In 1981, Dr. Mérialdo joined the IBM France Scientific Center, working on language modeling. He has received an IBM Outstanding Technical Achievement Award for his work on the automatic transcription of stenotypy. He is currently the manager of the Speech Recognition Group at the Scientific Center.