Speech Recognition in French With a Very Large Dictionary

H. Cerf-Danon, A-M. Derouault, M. Elbeze, B. Merialdo
IBM France Scientific Center 5, Place Vendôme, 75001, Paris, France

Abstract

A key factor in the development of Listening Typewriters is the ability to support Very Large Size Dictionaries (VLSD, containing several hundred of thousands words), because any restriction on the vocabulary is a restriction on potential users. This is even more important for inflected languages, such as French.

This paper reports some experiments in the development of a speech recognition system using a Very Large Size French dictionary and gives an overview of our decoding strategy and our prototype. The particularity of our system is that it is based on syllable recognition. In the first phase of our research, the speaker is asked to speak with short pauses between syllables. Our next goal will be Continuous Speech Dictation.

INTRODUCTION

Our approach is based on the probabilistic formulation introduced by F. Jelinek [1] where the item \( W \) recognized from a given utterance \( A \) is selected by a Maximum A Posteriori (MAP) rule:

\[
\hat{W} = \text{Argmax} \ p(W|A)
\]

The design of our system is then composed of a modeling problem to define \( p(W|A) \) and a computational problem to compute the Argmax. Equation (1) can be reformulated as:

\[
\hat{W} = \text{Argmax} \ p(A|W) \cdot p(W)
\]

which identifies the acoustic model \( p(A|W) \) and the linguistic model \( p(W) \).

We report here some experiments in the different fields which arise from this formulation of the problem.

In a first section, we shall explain why, due to different specificities of French, the syllable is a good unit for Automatic Dictation in French with a Very Large Vocabulary. We shall then describe our Acoustic Phonetic modeling based on phonetic Hidden Markov Models, which take into account some coarticulation effects. A third section concerns our work on a syllabic fast-match. Next we give an overview of our Language model, based on a Markov model of sequences of parts-of-speech, then of our decoding strategy and finally we give a description of our actual prototype.

PARTICULARITIES OF FRENCH

The “special” difficulties of French.

They are mostly due to three kinds of problems particular to French.

- French is a highly inflected language: on average, the number of different forms for each lemma is 7 (while it is only 2 for English). Our 200,000 words dictionary corresponds to about 45,000 lemmas (unfrequent inflections are ignored).
- There are many homophones in French: on average there are 2 different possible spellings for a given phonetic form (while for English the average is close to 1).
- We have then two different phenomena which would lead us to multiply the size of our phonetic dictionary by up to 2, if we did not introduce a level between phoneme and word in our recognition system.
  1. The apostrophe where a word such as le, when followed by a word beginning by a vowel, is transformed into l’ and concatenated (both for spelling and phonetic) to the next word, as in l’enfant.
  2. The liaison inserts a consonant between two consecutive words, when spoken, as in les enfants pronounced as les-z-enfants.

The advantage of the syllable approach.

A constant problem in Speech Recognition is the choice of the basic unit to be used at the acoustic level. This unit can be either a phoneme (elementary sound), a sub-word or a word. A small list of phonemes (around 30 or 40) is sufficient to describe any dictionary. But they are difficult to recognize accurately from the acoustic signal, because one phoneme may have very different acoustic characteristics, depending on the context of the pronunciation.

The interest of larger units like sub-words or words is that 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.

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 5,200 different phonetic syllables are required for a complete description of our 200,000 word dictionary.

Another advantage of the syllable is that the problems of liaisons and apostrophes in sentences can be handled easily, whereas word templates would mean having references for all possible liaisons and apostrophes. If we take, for example, the word artiste, using word templates would require the addition of the following references:

- lartist found in l’artiste
- dartist found in d’artiste
- narist found in un artiste
- zarist found in les artistes
- and a few more...

Since approximately one fourth of all words begin with a phonetic vowel, this would greatly multiply the number of word templates.

With the syllabic approach, liaisons and apostrophes require the introduction of only 1,200 new syllables. We now have a total of 6,400 phonetic syllables to cover isolated syllable speech, when using a 200,000 words dictionary and taking into account liaisons and apostrophes.

ACOUSTIC PHONETIC MODELING

From an acoustic observation \( A \), the recognition process aims to find a word sequence \( W \) so as to maximize \( p(W|A) \). The observation \( A \) consists of a sequence of centisecond labels from 200 elements codebook corresponding to 20 coefficients spectra (IBM Yorktown acoustic processor).
Phonetic sources.
Our group already designed an acoustic decoder based on 40 phonetic Markov machines. Since there is one machine per phoneme, these models are an average of the phoneme behavior in different coarticulation situations. All these phonetic machines have the same number of states (7), except the machine for silence, which has only 2 states.

Phonetic recognition.
Although the goal of our system is to recognize words, we can use the phonetic models alone, without a language model, to recognize phonemes. This is a way to test the acoustic modeling. 80 test sentences are pronounced in isolated syllable mode, and a Viterbi decoding chooses the most likely sequence of phone machines. A first order model is used at the phone sequence level.

Error analysis
The analysis of the phonetic errors with the classical context-independent models shows the different contexts that are most frequent. The vowels are generally well recognized despite the coarticulation, because of the stable part. On the contrary, some classes of consonants, especially voiced plosives, unvoiced plosives, affricates, give rise to many intra-class confusions. A systematic analysis [2] suggests that the right vocalic contexts influence the consonant according to the place of the vowel articulation. For example, when a p (labial articulation), is misrecognized as a k (palatal), the following vowel was always a back vowel, conversely k is taken for a p by the system when the following vowel is a front vowel, and so on. This agrees with what is known as the anticipatory vocalic influence in linguistic and phonological studies [3].

Choice of a context-dependent system
Definition
Early experiments showed the interest of a classification of the right contexts of voiceless plosives into 5 types: front, middle and back vowels, liquids, and silence. The introduction of context dependent models for these phonemes allowed to increase the phoneme recognition performance significantly [2].

The method has been generalized to other consonants, and all types of right contexts. Contextual machines are defined for voiced and unvoiced plosives, fricatives, semi-vowels, nasals, liquids. Since consonant clusters are rather frequent in French, consonant contexts have to be accounted for. Ten types of contexts are defined for the consonants. To each phoneme, correspond as many machines as relevant contexts. The new set has 130 markovian machines, which have the same structure as previously.

Results
Our phonetic recognition rate is 86.9% with the classical phonetic system, with a 3.6% insertion rate, compared to 89.8% and 3.8% with the contextual one. A comprehensive analysis of the results shows a significant improvement for most of the consonants: at least 10% better recognition performance on plosives, 5% better on fricatives, 8% on liquids.

SYLLABIC FAST MATCH
In order to reduce the number of possible candidates tested by the recognizer at the decoding stage, we introduce a syllable fast-match. This pre-filter aims to provide the detailed match with a list of the most likely syllables
- short enough to reduce the amount of computation
- long enough to include the right syllable with a confidence of 99%

Two kind of procedures have been developed and tested.

Classification of syllables
The first one is based on a classification of syllables around their vocalic part [4], and the matching of coarse machines [5] corresponding to syllabic classes.

In the following table, we present the results obtained with 1150 syllables uttered by one speaker. On average, the correct choice is at rank 22. If we accept to miss 1 syllable out of 100, the list size is limited to 274. The fifth column shows how often the correct syllable is at the top of the list.

| Table 1. 1150 syllables tested in a text of 79 sentences. |
|---|---|---|---|---|
| speaker | corpus | average rank | 99% | top |
| SVS | TSI 1150 | 21.54 | 274 | 27.0% |

Poisson Polling Fast-Match
The second one uses a voting function:

\[
F_s = \sum_{a=1}^{T} V(a, s) + I_s \quad (s = 1 \ldots 6,400 )
\]

This second approach is named Poisson Polling Fast-match [6], when the label frequencies are assumed to have Poisson distributions. For each syllable (s) of the dictionary, each label (a) casts a varying vote \( V(a, s) = \log L_a s \), where \( L_a s \) denotes the expected frequency of label (a) in an utterance of a given syllable (s). After deriving some equations, we may express the initial value of the function as follows:

\[
I_s = \log \text{pr}(s) - L_s
\]

where \( L_s \) is the expected length of (s) and \( \text{pr}(s) \) is a prior probability of finding (s) in a text. The final score of a syllable is obtained by summing votes from the beginning to the end of the utterance \( (a_1, \ldots, a_T) \).

Results:
(Training on 400 sentences)

| Table 2. Performance of Poisson Polling Fast-Match |
|---|---|---|---|
| speaker | corpus | average rank | 99% |
| MOY | TSI 1130 | 4.26 | 49 | 68.5% |
| BYM | TSI 1120 | 6.43 | 69 | 70.9% |
| MEB | TSI 1150 | 6.3 | 69 | 65.3% |
| CRI | TSI 1150 | 4.85 | 72 | 62.6% |
| SVS | TSI 1150 | 5.33 | 73 | 66.2% |
| FAB | TSI 1150 | 8.75 | 95 | 59.4% |
| JCM | TSI 1150 | 8.01 | 137 | 60.4% |
| UBU | TSI 1137 | 9.21 | 185 | 60.0% |
| FAR | TSI 1150 | 15.08 | 197 | 59.6% |

Comparing a Polling Fast-Match on words and syllables
First, the language model used to estimate \( \text{pr}(s) \) is only based on syllable frequencies. As reported in [6], the polling fast-match leads to good results, when using a 3-gram language model in order to estimate the prior probabilities \( \text{pr}(w) \). This is not adequate for syllabic units.

Secondly, syllables are shorter-timed than words so that scores are less contrasted.
Last, the Polling Fast Match doesn't take into account the time sequence information. Identical syllable scores are obtained if permuting the labels. For example, phonetic anagrams as reste, certes, or tresse lead to similar syllable scores. Unfortunately, there are many anagrams in French: On average, in a French text one syllable out of 2 has, at least, one phonetic anagram.

**Discussion**

Although speakers are asked to manage pauses between syllables, it is not easy to determine syllable end-points for some "bad" speakers. However, polling fast match requires very little computation to obtain short lists of 128 syllables and reduces 50 times the computational cost of the detailed match, introducing an average loss not greater than 1%.

**STATISTICAL LANGUAGE MODELING**

As stated in the introduction, the goal of the language model is to compute a probability for a word string. Of course, the notion of probability for a sentence is not well defined in an abstract way, and it can only be justified when referring to a given mathematical model.

Rigorously, the probability for a word to be produced, should depend conditionally upon the whole past string. The probability of a word sequence \( W_l \) would be written as the product of conditional probabilities:

\[
p(W_l) = p(W_1) \prod_{i=2}^{n} p(W_i | W_{i-1})
\]

In practice, these probabilities would be difficult to estimate (from a huge corpus), and would require much too much storage. As first proposed by F. Jelinek [7], a possible solution is to reduce the number of events, by defining an equivalence relation on the word strings.

For our dictation experiment, we use the "tri-pos" language model which was elaborated to transcribe phonetic to French [8]. We will denote by "tri-POS" (respectively "bi-POS") a triplet (respectively a pair) of parts of speech. The Markov source modeling the production of a word is precisely defined as follows:

- The states are the pairs of parts of speech.
- The alphabet of symbols is the set of words in the dictionary.
- A transition between the states \((p_1, p_2)\) and \((p_3, p_4)\) is given by a word \( w \) of part of speech \( p_0 \).

We first estimate the probability of such a transition by the product of the relative frequency of \( p_1 \) after \((p_0, p_2)\), and the probability of the word \( w \) given the POS \( p_0 \). This last distribution is stored in the dictionary for each word.

The corpus where the frequencies are collected contains 1.2 million words. We take for estimate of the probability to produce the part of speech \( p_0 \) after \((p_0, p_2)\), a linear interpolation of the two frequency distributions. The final expression of the probability to produce the word \( w \) from the state \((p_1, p_2)\) is as follows:

\[
p(w_j | p_1, p_2) = (\lambda \cdot f(p_j | p_1, p_2) + \lambda \cdot f(p_j | p_2)) \times k(w_j | p_2) + \varepsilon
\]

The weights \( \lambda \) are automatically adjusted [9].

**DECODING STRATEGY**

The Multi-Level Decoding strategy allows to support a Very Large Size Dictionary for speech recognition:

- A Syllable recognizer uses the acoustic models to build a list of the most probable syllables that match the acoustic observation from a given time frame.
- From this list, a Word recognizer uses the dictionary to build partial word hypothesis.
- A Sentence recognizer uses the language model to build partial sentence hypothesis until complete sentences are found.

**The Syllable recognizer**

Given an acoustic observation \( A_t \), the Syllable recognizer computes the list of the most probable syllables (candidates) matching the acoustic observation from a given time frame \( t \). Also, for each candidate, it provides 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 \( t = 1 \)).

We consider the Markov model composed of the concatenation of:

- the Syllabic Tree (ST, which describes syllables in terms of phonetic machines),
- a silence machine (used to model the pause after the first syllable),
- and the Looped Phonetic Model (LPM, which connects a copy of each phonetic machine to each other one with a diphone probability) [10].

This model is intended to produce the utterance of any syllable, a pause, and then any sequence of phonemes.

**The Word recognizer**

The Word recognizer uses a tree (called W/S tree) that describes words in terms of phonetic syllables. Nodes correspond to partial word. Arcs correspond to phonetic syllables. The arcs leaving a given node indicate which syllables can extend a given partial word.

At each time, the Word recognizer maintains a list of nodes with high score. When syllable candidates are provided by the Syllable recognizer, it looks which nodes can be extended by these syllables, and constructs the new list of best nodes. Leaves with high score are kept as word candidates.

To take liaisons and apostrophes into account, the W/S tree is augmented with auxiliary roots (called "heads"). Each head corresponds to a consonant or consonant cluster that may occur because of a liaison or apostrophe (like I m z... ). Each syllable S that starts at the root of the tree and begins with a vowel creates new syllables starting from each head (for example, the phonetic syllable ar will create the syllable iar from head T, zar from head Z, etc...).

For liaisons, 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 liaison z ).

For apostrophes, 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 .

Our current dictionary contains 200,000 words. There are about 120,000 nodes (partial words), 340,000 arcs (syllables starting from a node) and 16 different heads.
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 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 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 equal to the beginning of the word. This leads to a longer partial sentence hypothesis, for which a new score is computed.

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 non-extended hypothesis so far. In our case, we use a time-synchronous strategy which extends the shortest non-extended 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 shorter hypotheses have been processed, so that the list of hypotheses ending at time $t$ is complete.

RESULTS AND PROTOTYPE.

Results.

Our system has been implemented on an IBM 4381. It runs in batch mode. Our standard test text is composed of 79 short sentences (corresponding to 722 words and 1150 syllables). These sentences come from a set of letters that do not belong to the corpus used to select frequent words. This text is recorded by every speaker in Isolated Syllable mode, together with a set of training texts used to build the acoustic model for this speaker. Using the 130 phonetic machines, the VLSD (200,000 words) and the tri-POS language model, the results across 9 speakers show an average of 10.97% words not correctly recognized.

Present prototype.

We also designed a demonstration version of our prototype that allows interactive recognition. It has the following characteristics:

- a dictionary composed of the 118,000 words which are formed by the 957 more frequent phonetic syllables. This dictionary contains 96.75% of the words in a representative corpus assuming that one counts a word as “missing” only on its first occurrence in the corpus.
- the Yorktown acoustic front end, composed of a microphone attached to an IBM PC which contains a signal processor and runs a software that performs the acoustic processing of the speech signal.
- the 4381 mainframe running VM where all the recognition takes place. The PC is connected to this mainframe with a standard emulation program.
- this reduced version of the system runs “close to” real-time.

Bibliography.


