RECOGNITION OF WORDS WITH THE HELP OF THE SERAC-IROISE EXPERT SYSTEM

Xavier Marie, Martine Gérard, Guy Mercier

Centre National d'Etudes des Télécommunications, Centre Lannion A, Route de Trégastel, 22301 LANNION Cédex, FRANCE.

ABSTRACT

In order to test the performance of the acoustic-phonetic decoding module on the Serac-Iroise expert system, we have implemented a lexical analyzer, the function of which is to match each word of the task vocabulary against the phonetic hypotheses lattice. A one-stage dynamic comparison algorithm, initially designed for global recognition of connected words, has been adapted. Our knowledge-based approach makes it possible to improve performance significantly with the help of heuristics, e.g. concerning local constraints and the measure of similarity. Introducing phonological, syllabic and prosodic information into the lexicon allows refinement of the strategy by basing on islands of reliability. Such phonological phenomena as merging, spreading, insertion, deletion and confusion are dealt with in a rather flexible way : likelihood weights, penalty factors and thresholds of reliability are determined according to the most encountered recognition errors. The object - and rule-based representation gives advanced opportunities for system extension and modification.

1. Introduction

Conclusions after ARPA SUR and subsequent projects have led to reconsidering approaches to Automatic Speech Recognition (ASR). Separate contribution of the different knowledge sources are best modelized using Artificial Intelligence (AI) knowledge representation tools such as Production Systems (PS), that supply advanced features for formalizing expertise, comparing strategies and refining parameters and heuristics.

The KEAL system developed at CNET achieves multispeaker analytical recognition and interpretation of isolated sentences taken from a few hundred words'vocabulary. The SERAC system (Système Expert pour la Reconnaissance ACoustico-phonétique) has been designed to structure the knowledge acquired with Keal and to provide a flexible tool for maintaining, improving and extending it, eventually leading to a new ASR model.

The lexical analyzer MODEM (MOdule de DEtection de Mots) is dedicated to both validating the phonetic level and evaluating heuristics for connected word verification and techniques for representing lexical and phonological knowledge.

The Iroise system is a PS using an object-oriented problem-driven rule-based language of the OPS family, it consists in three functional components : the knowledge Base, the Inference Engine and the User's Interface.

An acoustic-phonetic module : feature extraction, sentence onset detection, centisecond labelling, segmentation into pseudo-syllables, segmentation into phones, hierarchical consonant and vowel recognition, and a prosodic module, which detects the extrema of the pitch and vocalic durations, and the type and main boundary of the sentence, have now been implemented in Serac : about 600 rules total.

2. Modem : a module for word detection

The principle of this detection is to match the phonetic lattice against phonetic frames taken from the set of possible words at a given instant, and finally keep the optimal sequence of words ; our heuristic comparison approach is based on a dynamic programming (DP) sequential algorithm and a measure of possible errors and similarity driven from lexical and phonological associated knowledge.

A main feature of MODEM is the intervention of phonetic, syllabic and phonological knowledge all along the process. The phonetic lattice shows as a sequence of phonetic segments (or frames) characterized with a set of weighed properties or attributes such as phoneme type (consonant, vowel, semivowel), cues like mode of articulation (fricative, plosive, nasal, liquid), place of articulation (labial, dental, velar, palatal, pharyngal), segment duration syllable number, and the best three phonetic hypotheses with associated confidence scores. In the IROISE representation language of structured objects, each phonetic frame is an instance of the frame class with fixed attributes.

3.1. The Lexicon

Each word in the lexicon also refers to contextual or phonemic information. For manageability's sake, the whole vocabulary is represented under the form of a unique list : the Lisp basic structure is thoroughly used. The functional notation makes access to the first element much easier, so special information is entered in the beginning of sublists referring to a particular object.

Elements of a 'phonemic frame' sublist are the possible realizations in decreasing likelihood order, preceeded with special makrs such as :

- obligatory : states the frame is an accentuated syllable's vocalic nucleus, and thus absolutely must be recognized within the best three hypotheses, and may not be omitted in any event ;
- optional : states the present frame can be dropped without penalty, as it is often in oral language ;
- non-optional : states the frames is to be fully penalized whenever omitted or ill-recognized.

The marks are very useful in determining the type of treatment to be applied during the similarity calculus. This highly modular representation makes automation easy when accessing very large lexicons, and enables the expert to introduce many other features of prosodic (pitch, duration), phonological (elision, nasalization) of phonotactic (phoneme combination rules) type.

3.2. Verification of words

3.2.1. The algorithm

It is originally a one-stage DP algorithm for connected word recognition (CWR), where words are considered as sequences of acoustic frames. The search space is a finite pattern of squares. The x-axis is the pronounced sentence, divided in N temporal frames (index i), while the y-axis is composed of M word templates (index k), each divided in J(k) frames (index j). A dissimilarity measure d(i,j,k) is used and a cumulative distance D(i,j,k) at point (i,j,k) is to be minimized to obtain the optimal sequence of templates (or optimal super-template).

```
1. Initialize D(1,j,k)= d(1,1,k)+...+d(1,J(k),k) ;
2a.for i:2..N do 2b-2e;
2b.for k:1..M do 2c-2e;
2c.D(i,1,k)= d(i,1,k)+min(D(i-1,1,k), D(i-1,J(k'),
   k'); k':1..M);
2d.for j:2..J(k) do 2e;
2e.D(i,j,k)= d(i,j,k)+min(D(i-1,j,k), D(i-1,j-1,k),
   D(i,j-1,k);
3.find k* such that D(N,J(k*),k*) be minimum, and
  trace back the path leading to (N,J(k*),k*).
```

Problems of i) practical implementation and ii) adaptation to recognition from phonemes are raised :

i) in order to reduce memory size, we use two column vectors Di(j,k) and Di-1(j,k) updated after every comparison, and two backpointer vectors Bi(j,k) and Bi-1(j,k) that state the instant when last template of the current supertemplate : T(i), beginning at frame F(i), terminates. The temporal complexity is M.N.J and the spatial complexity is 2(N+M.J) if J=moy(J(k)).

ii) modifications are to be introduced at the following levels :
- similarity measure (SM) between phonemes and dealing with phonological variations in a balanced way ;
- local constraints (LC) : choice of allowed transitions, introduction of penalty factors for phonological deformations ;
- normalization for keeping the SM homogene and optimal with the LC broadening ;
- heuristics dedicated to pruning the search and taking domain expertise into consideration.

3.2.2. Local constraints

They define the transition mode between two points of the search space. With Sakoe and Chiba's symmetrical LC, the path leading to (i,j) may come from :
(i-1,j): spreading ; (i,j-1): merging;
(i-1,j-1): normal ; (i-2,j-1): deletion;
(i-1,j-2): insertion,
segmentation errors and phonological phenomena being the main causes for abnormal cases. To normalize the cumulative similarity (CS) along a path, we use the following method: the length of a path always equals the sum L(i,j)=i+j of its ending coordinates. Penalizing abnormal transitions induces a strategy based on islands of reliability. Penalty factors depend on error type and template length; deletions and insertions are much more severely penalized as more likely to come from segmentation deficiencies.

3.2.3. Similarity

Given a point (i,j), the best path leading to it is determined using CS at points (i',j') from which transition is allowed, and similarity index (SI) s(i,j) between templates and the lattice :

$$S(i,j)= S(i',j') + (1-F) (L(i,j)-L(i',j')) s(i,j)$$

where s's factors are the penalty and normalization factors. The origin is the fictive point (-1,-1,-1) Points where a template terminates or almost terminates need a special treatment : the best among them are selected before being copied as points of -1 or -2 ordinate from where they will be reached without introducing any discontinuity in the DP process.

The SI is computed as the maximization of punctual similarity on every pair (lattice phoneme, template phoneme), the value of which is the phonetic score multiplied with the similitude between the two phonemes. The latter is computed once for all using an empirical measure : the C-V similitude is generally 0, while the V-V similitude is function of additive formantic distance, and the similitude between consonants is the weigthed mean of their hierarchized cues similitude : voicing, mode and place of articulation, with weights of (resp.) 3, 5 and 2, supposedly approximating the reliability of these cues detection.

3.3. Implementation

The objects used are the template and frame objects, representing information associated to the phonemic segment in the lexicon or in the phonetic lattice, and the path object, that characterizes the current search point : its attributes are : coordinates, type of transition leading to it, path length in the supertemplate, backpointer values, special marks, CS and SI. Loop control variables are also represented as objects.

The strategy of detection holds three stages :
- process control problems for computing general data (similitude matrices, similarity thresholds, penalty factors), loading files, commanding the DP loops, instantiating objects, pruning the search, displaying results and normalizing the distance ;
- search for adequate transitions with anterior path, according to the LC and existing marks ;
- path evaluation problems that compute the SI, select the best transition and validate the pat, with the help of heuristics for improving speed or deleting ill paths.

CONCLUSION

This makes 10 problems for about 60 rules total, calling a number of Lisp functions ; this skeleton is presently being extended to introduce new knowledge and efficient heuristics. A very useful development basis is supplied for evaluating phonetic decoding and adjusting heuristics able to improve lexical search in a general frame for ASR.

REFERENCES

(MER84) MERCIER G., GILLOUX M., TARRIDEC C., VAISSIERE J. "From Keal to Serac : A New Rule-Based Expert System for Speech Recognition" NATO/ASI, Bonas, Jul.1984.
(NEY85) NEY H. "The Use of a One-Stage Dynamic Programming Algorithm for Connected Word Recognition" IEEE ASSP 32, no 2, Apr. 1982.
(SAK78) SAKOE H., CHIBA S. "Dynamic Programming Algorithm Optimization for Spoken Word Recognition" IEEE ASSP 26, n° 1, Feb. 1978.
(VIV85) VIVES R. "Mise en Correspondance Temporelle de Descriptions Phonologiques Syllabiques et Prosodiques de mots dans le Système de Reconnaissance de la Parole Keal" 14e JEP, Paris, 1985.