知識工学と人工知能 44－3
（1986. 1. 20）

A SUPPORTING ENVIRONMENT FOR CONTINUOUS SPEECH RECOGNITION SYSTEM
—— Implementation of Rule Database and Rule Translator ——

Katsuhiko TSUJINO, Riichiro MIZOGUCHI and Osamu KAKUSHO

The Institute of Scientific and Industrial Research, Osaka University,
8-1, Mihogaoka, Ibaraki, Osaka, 567 Japan.

abstract: The authors have developed a knowledge-based continuous speech recognition system which simulates the behavior of a human expert who can recognize speech by inspecting the trajectories of feature parameters. This paper describes an environment for building knowledge-based systems. The environment is designed based on several issues uncovered during the development of the above system. It helps an expert to explore recognition rules and to verify them by speech-specific techniques and statistical analyses. Detailed descriptions of the environment are presented together with a simple example.

1. INTRODUCTION

Knowledge-based systems have been investigated in these several years. They are divided into the following two categories :

1) Expert system [1][2][3].
2) Knowledge base management system (KBMS) [4][5][6].

The former has domain-specific knowledge in the knowledge base in order to realize the performance comparable to an expert in the domain. On the other hand, the latter is mainly concerned with theoretical discussions on truth maintenance of domain independent knowledge. In these researches, several frameworks and tools for building expert systems are obtained. But all of them are not so powerful and do not have a good environment to manage the knowledge of any specific domains, since they are too general purpose ones.

The authors have been constructing a continuous speech recognition system using knowledge engineering techniques. The system simulates the behavior of a human expert who can recognize speech by inspecting the trajectories of feature parameters such as formant frequencies. The expertise is encoded in the form of production rules. In the current implementation, it has 112 rules, which obtain 93% segmentation rate and 85% recognition rate for continuous speech of about 30 seconds long uttered by 3 male adults [7]. The Knowledge engineering approach to continuous speech recognition is rather revolutionary in speech community. It is well-known that a lot of heuristic knowledge is necessary for constructing a continuous speech recognition system. However, conventional systems are constructed by encoding these heuristics in procedural languages such as FORTRAN, which makes the systems inflexible. Knowledge engineering techniques are superior to procedural ones in the treatment of a lot of heuristic knowledge. This is one of the main reasons why our system can attain the high recognition rates mentioned above.

In the early stages of the development, the only tool available had been a production system called HIPS [8], and it had been enough for our initial size of the system. Although the first implementation of the system was satisfactory, some significant issues came to appear during development, which are summarized as follows :

1) Organization of rules.
2) Management of the information about rules and speech data.
3) Interface.

As described above, the key technology of our speech recognition system is knowledge base construction. In contrast to other expert systems based on the established knowledge of a human expert, our system must deal rather uncertain knowledge for speech recognition. Therefore, it's very important and valuable to construct a supporting environment which helps to find out new knowledge and verify their validity by domain-specific, i.e, speech-specific techniques. The current effort of our research is focussed on development of a powerful environment for supporting knowledge base construction in order to resolve the above three problems.

This paper describes a supporting environment of knowledge base construction for speech recognition. A brief overview of the recognition system is shown in Section 2. Section 3 summarizes the design issues of the environment, and Section 4 gives detailed descriptions of the supporting environment. Finally, some examples of the system are shown in Section 5.

2. OVERVIEW OF THE RECOGNITION SYSTEM

Figure 1 shows the trajectories of feature parameters of continuous speech uttered by a male adult (/GINSEKAINO/, in Japanese). Parameters shown in this figure are energy, pitch

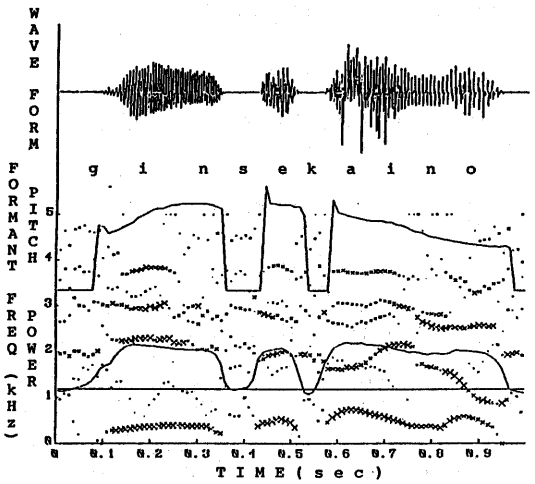

Fig.1 Trajectories of feature parameters.

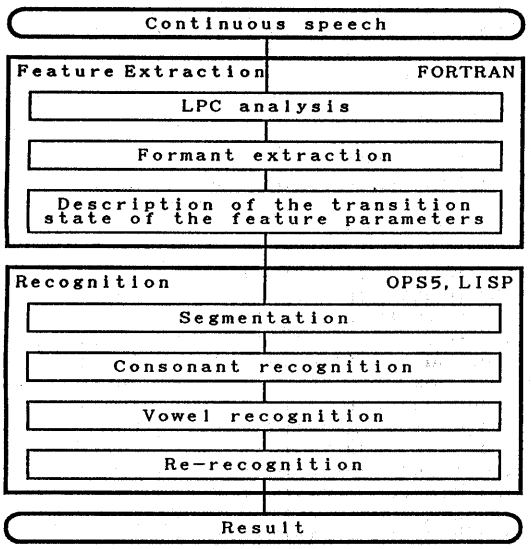

Fig.2 Main flow of the recognition system.

and formant frequencies which are the resonance frequencies of vocal tract. The main flow of the recognition system is shown in Fig.2. First, the transition of each parameter is described in term of eight level descriptions such as rapid increase, moderate decrease, and so on. After obtaining the descriptions, they are sent to working memory of production system. Then, segmentation, consonant recognition and vowel recognition are made in this order by interpreting production rules.

The state descriptions of the trajectories shown in Fig.1 are depicted in Fig.3, in which the description is made by using the following labels:

N(Noise): Energy is at noise level.
I(Increase): The parameter is increasing.

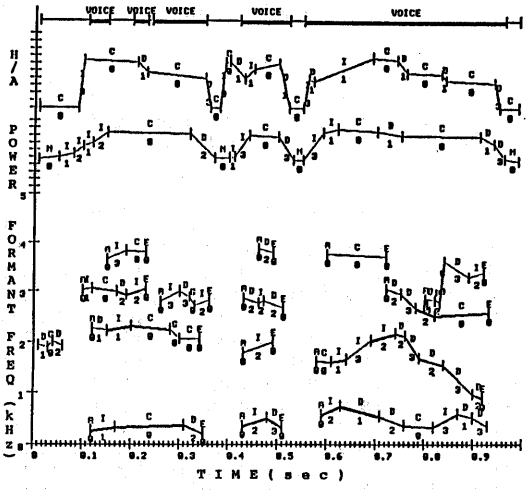

Fig.3 Descriptions of feature parameters.

D(Decrease): The parameter is decreasing.
C(Constant): The parameter is stable.
A(Appearance): The formant appears here.
E(Extinction): The formant disappears here.
G(Gap): The formant has a discontinuity here.

I and D have three levels, i.e., D1,D2 and D3 where D3 means the most radical change. Typical rules are shown below.

* Rules for segmentation

If CDC or CIC is in energy transition
then the changing part is a candidate of segment boundary.

If A, E or G is in formants
then the place is a candidate of segment boundary.

* Rules for consonant recognition

If energy has a dip(DI), first formant has a discontinuity or dip and duration is shorter than 30msec.
then /R/.

If silent term of length is less than 100msec. and rapid increase of energy
then /P,T,K/.

* Rules for determining a representative formant frequencies for vowel recognition

If there is a maximum or minimum point in formant
then use the extremum point.

If IxIyIz or DxDyDz, where y<x,z

then use the average frequency of Iy or Dy.

* Rules for vowel recognition

If first formant frequency is higher than 550Hz and second formant frequency is between 1100Hz and 1600Hz
then /A/.

If first formant frequency is lower than 420Hz and second formant frequency is higher than 1600Hz
then /I/.

* Rules for re-recognition

If result is /P,K/-/U,O/ and first formant is I1, second formant is D3 around the segment boundary between them
then /P,K/-/Y,I/-/U,O/.

If result is /I,E/-/U,O/-/A/ and transition of first formant is C I2 C, transition of second formant is D2,3 C C
then /I,E/-/W/-/A/.

3. DESIGN ISSUES OF THE ENVIRONMENT

3.1. Organization of Rules

In HIPS version system, the priorities of rules and hierarchical relationship among them are embedded in the textual order of the rule memory. In other words, the former rules in rule memory have higher priority and slave rules must be located under its master rule. Therefore, improper insertion of a new rule or careless relocation of the existing rules causes some unexpected bugs. To avoid this, all of the rules were re-organized into several knowledge sources shown in Fig.4, which makes explicit representation of rule dependencies.

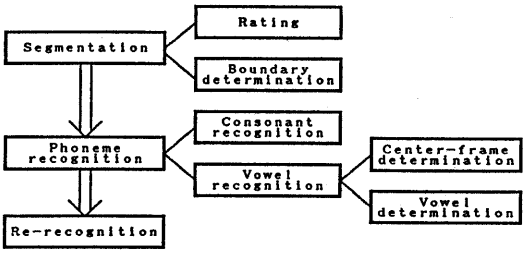

Fig.4 Knowledge sources.

3.2. Management of Rules

When an expert debugs a rule, he has to know the role of the rule, from what speech data the rule is made, and how the rule has been applied. If he can't get these information, the new revised rule may lose its original responsibility. Furthermore, if he does not notice the relations among rules, such as partner or master-slave relation, he may leave some rules unchanged when he modifies one of the related rules. These bugs, which are difficult to find out, are caused by lack of management facilities of rule documentation. This observation suggests the necessity of a rule databases storing useful rule documentation.

3.3. Interface

When we construct an expert system, it's very important to be able to modify and check the rules as quickly as possible. But in HIPS version system, it takes a lot of time to prepare associated rules and speech data when new rules are added and examined. This time-consuming job makes it difficult to debug the rules and to improve the performance. An efficient interface must be developed in order to remove these difficulties.

4. SUPPORTING ENVIRONMENTS

To solve the problems described in the last section, a new recognition system with supporting environment was developed as shown in Fig.5. The new system is composed of five subsystems described below.

4.1. Recognition Subsystem

This subsystem recognizes continuous speech, and returns the recognition results with some additional information such as fired rules and some data at an important point in the recognition. High speed production system OPS5 [9] was adopted as an inference engine. Figure 6 shows an example of new rules reorganized into several knowledge sources, where three control elements; 'Context', 'Req' and 'Data' are used, because OPS5 does not allow to write any function in its condition part (left hand side; LHS). If we want to check A+B<X in LHS, we must prepare two rules as shown in Fig.7. The first rule makes a working memory element with the result of A+B and the second one checks the result whether it is smaller than X. The three control element used in recognition rules are described below.

1) Context —— This element appears in LHS of a master rule. It denotes the name of knowledge source which the rule belongs to and represents the relevant context of the rule.
2) Req —— A master rule makes this element in its action part (right hand side; RHS) to indicate an explicit request given to its slave rules.
3) Data —— A master rule makes this element to pass pre-calculated

data to its slave rules. The slaves rules catch this data when a request comes into effect.

4.2. Rule Translator

As described above, one conceptual rule must be divided into at least two rules in OPS5 syntax when the rule has some functions in its LHS. It is rather difficult for us to read and modify such rules, since they have many temporary variables which do not appear in the conceptual rules. Therefore, a rule language for speech recognition (RL/SR, shown in Fig.8) and its translator into OPS5 syntax have been developed. This rules language enables a human expert to make straightforward description of his expertise, because it provides high level descriptors corresponding to his concepts in speech recognition. The rule translator is implemented so that it can be easily updated when the descriptors are modified, since the rules are not established yet.

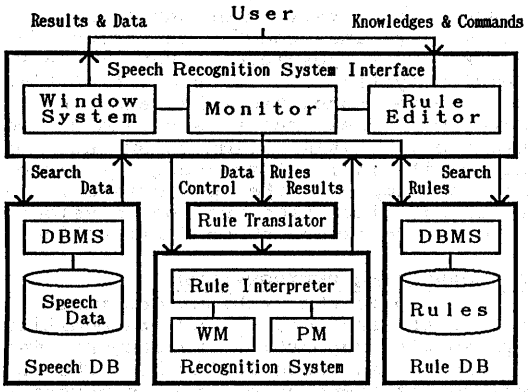

Fig.5 Overview of the environment of the recognition system.

```
(defp dip-3                    ;A master rule
  (context rating)
  (pwr  ^s <x1>  ^e <x2>  ^state D  ^class <c1>
  (pwr  ^s <x2>  ^e <x3>  ^state << D I C >>      ...
  (pwr  ^s <x3>  ^e <x4>  ^state I  ^class > <c2>
  -->
  (make data ^rule dip-3 ^at <at> ^atr dxx
             ^val (compute <x4> - <x1>))
  (make data ^rule dip-3 ^at <at> ^atr dpp
             ^val (u-diff <p1> <p4>))
  (make data ^rule dip-3 ^at <at> ^atr c13
             ^val (u-aux1 <c1> <c2>))
  (make data ^rule dip-3 ^at <at> ^atr r13
             ^val (compute (<p1> - <p2>) // (<p4> - <p3>)))
  (make req  ^rule dip-3 ^at <at> ^state active))

(defp dip-3-1                  ;A slave rule
  (req  ^rule dip-3 ^at <at> ^state active)
  (data ^rule dip-3 ^at <at> ^atr dxx ^val <= 16)
  (data ^rule dip-3 ^at <at> ^atr dpp ^val <= 100)
  (data ^rule dip-3 ^at <at> ^atr c13 ^val OK)
  (data ^rule dip-3 ^at <at> ^atr r13 ^val { > 0.33 < 3.0 })
  (pwr ^no <at> ^s <s1> ^e <e1> ^sv <sp1> ^ev <ep1>)
  (pwr ^s <e1> ^e <s2>)
  (pwr ^s <s2> ^e <e2> ^sv <sp2> ^ev <ep2>)
  -->
```

Fig.6 An example of recognition rule.

```
   --- Bad syntax ; Virtual rule ---
(defp virtual-rule
  (element1 ^value <A>)
  (element2 ^value <B>)
  (threshold ^value {<X> > <A> + <B>})
  -->
  *** Action part ***)

   --- Good syntax ; Actual rules ---
(defp actual-rule-1
  (element1 <A>)
  (element2 <B>)
  -->
  (make result ^value (compute <A> + <B>)))

(defp actual-rule-2
  (result ^value <Sum>)
  (threshold ^value {<X> > <Sum>})
  -->
  *** Action part ***)
```

Fig.7 A difficulty in using a function in LHS.

```
(DIP-2 Rating          ;Definition of Rule DIP-2 for Rating
   (PWR (D =N =N I)) ;If power is Decrease, Not Noise, Not Noise, Increase,
   (IF  @pv1≥40      ;    the amount of power change is more than 4dB
        @pv4≥40      ;      in the first and fourth areas,
        @pt2≤4       ;    time lengths of second and third area are
        @pt3≤4       ;      less than 40msec.,
        @ptx≤16      ;    total time length of four areas is less than 160msec.,
        @pc1>@pc2    ;    the amount of power change in the first area is
                     ;      more than that in the second,
        @pc4>@pc3    ;    its change in the fourth area is more than that in the third,
        @pax≤100     ;    it changes less than 10dB between the start point of
                     ;      the first area and the end point of the fourth area,
        @prv41<3.0   ;    ratio of the amount of power change in the first area to that in the fourth and
        @prv14<3.0   ;      ratio of that in the fourth to that in the first are less than 3,
        @prv12>3.0   ;    in the first area, its change is more than three times in the second area,
        @prv43>3.0 ) ;    and in the fourth area, its change is more than three time in the third area
   (THEN             ;Then
        @pd1=DIPD    ;    the first area is DIPD ( Dip, Decreasing ),
        @pd2=DIPN    ;    the second and
        @pd3=DIPN    ;      third area are DIPN,
        @pd4=DIPI    ;    the fourth area is DIPI ( Dip, Increasing )
        (PNT1 1 4))) ;   and put certain scores at the centers of first and fourth areas
                     ;      because they are candidates of segment boundaries.
```

Fig.8 Rule language for speech recognition.

6) Historical information of how the rule has been applied.
7) Relation between other rules.

When rules are stored into database, most of the above information is loaded into database automatically, since it is identified by the descriptors of RL/SR. The historical information, which is very important because it guarantees that the new rule does not lose its original responsibility, is also updated automatically by the recognition subsystem whenever the rule is fired. Rule database was implemented so that any LISP function can be used in the retrieval condition. Such implementation may decrease the retrieval speed but it enables very flexible retrieval. For example, a user defined function can be used as a virtual attribute, which has no explicit expression in the rule database but is specified by the function, for retrieval. If it turns out that the attribute is useful and fast retrieval is required, we can define a new real attribute corresponding to the virtual one and set its value of all contents in database automatically by applying the function used to define the virtual attribute.

4.4. Speech Database

Speech database manages and provides speech information, such as speech signals, feature parameters, contents of speech, and speaker's attributes such as age, sex, and dialect. This database stores the speech signals by indexing them with phonemes [10]. For example, one can get speech by specifying contained syllables. This database makes it easy for us to prepare speech data when we debug rules.

4.5. System Interface

The quality of the supporting environment depends on the way how to use the powerful facilities described above and how to combine them together. System interface plays a central role for this control. In addition to the three subsystem, system interface has many useful functions such as graphic interface, window system, built-in smart text editor and statistical functions based on the data in the two database. They also improve the interface between users and the three subsystems. For example the graphic interface helps to fix threshold levels which discriminate between vowels /I/ and /E/ by plotting feature parameters such as formant frequencies on another window.

There are two kinds of supports in rule development; support for discovery and support for verification. The former analyzes speech data and rules and helps us to find out new features. The latter analyzes recognition results and helps us to find out bugs. In both supports, the statistical functions of system interface and the data in rule database are very useful and important. For example, the following supports is provided.

1) When a rule which is seldom used is fired, system interface reports us and stores the information into the rule database. This information is useful when we check whether the rule is worthless or not.
2) When system interface find a data which is close to a threshold, it warns that the threshold may be improper. This warning makes it easy to find out uncertain conditions in a rule.

5. IMPLEMENTATION AND A SIMPLE EXAMPLE

The recognition system and its supporting environment are being developed mainly on Symbolics 3600. The recognition subsystem adopts OPS5e as its inference engine. Speech database is constructed on MV/8000II. These two computers are connected through Ethernet. An overview of software configuration is shown in Fig.9. Figure 10 shows results when a user accessed the rule database and the speech database. In this example, the user retrievals a rule which is for segmentation and refers to power parameter from the rule database. And next, he obtains a figure of state description of feature parameters related to the rule retrieved.

System	Hardware	Software
Recognition System	Symbolics Lisp Machine 3600	OPS5e+LISP
Rule Database		LISP
Rule Translator		
Interface		
Speech Database	(Ethernet) DG MV/8000II	DG/SQL

Fig.9 Software configuration.

6. CONCLUSIONS

We have discussed a supporting environment for continuous speech recognition expert system. One of main objectives of our research is to construct a knowledge base for continuous speech recognition. This knowledge engineering approach has been shown to be very promising. We have discussed several issues uncovered during the development of the knowledge base. These issues reveal real and essential characteristics of knowledge

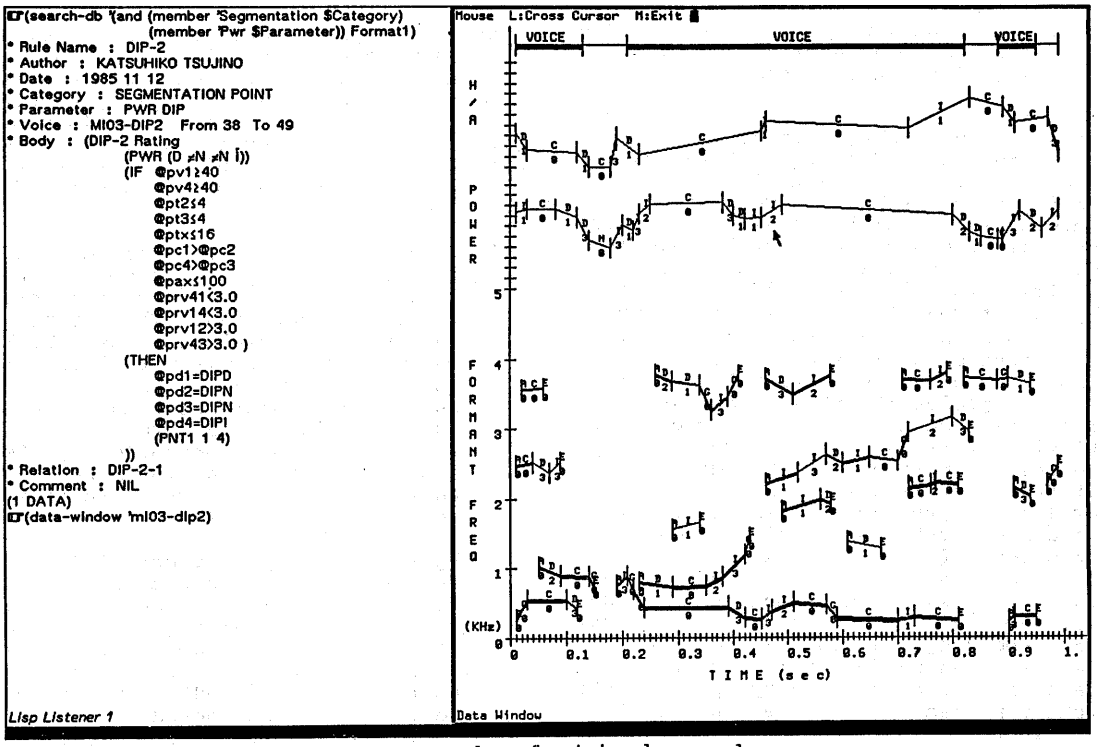

Fig.10 Examples of retrieval commands.

base construction and specify the architecture of the supporting environment presented above. The authors believe that the environment will make a valuable contribution to the research of knowledge base construction as well as that of speech recognition. The major facilities of the environment are summarized below.

1) The rule database in the environment stores various information about rules useful for understanding and managing them.
2) It helps us to find out new features of speech according to statistical analyses of the data stored in the speech database.
3) It presents an intelligent help for debugging rules by analyzing the recognition results.
4) Sophisticated interface accelerates rule development by utilizing the facilities.

[REFERENCES]

[1] Davis, Randall, "Teiresias: Applications of meta-level knowledge," pp.227-490 in knowledge-Based Systems in Artificial Intelligence, ed. Randall Davis and Douglas B. Lenat, McGraw-Hill, New York (1982).
[2] McDermott, John, "R1:The formative years," AI Magazine 2(2) pp. 21-29 (1981).
[3] Hayes-Roth, "Knowledge-Based Expert Systems," IEEE COMPUTER 17(10) pp.263-273 (1984).
[4] Furukawa, R. et al. "MANDARA:A logic based knowledge programming system," pp.613-622 in Proceedings of Fifth Generation Computer Systems, ICOT (1984).
[5] Doyle, Jon, "A truth maintenance system," Artificial Intelligence 12(3) pp.231-272 (1979).
[6] McDermott, Drew V. and Doyle, Jon, "Non-monotonic logic I," Artificial Intelligence 13(1,2) pp. 41-72 (1980).
[7] Mizoguchi, R. et al. "Continuous speech recognition based on knowledge engineering techniques," pp. 638-640 in Proceedings of the International Conference on Pattern Recognition, IEEE (1984).
[8] Mizoguchi, R. et al. "Hierarchical production system," pp. 586-589 in Proceedings of IJCAI-79 (1979).
[9] C.L.Forgy, "OPS5 User's Manual," technical report CMU-CS-81-135, Carnegie-Mellon University, Pittsburgh, Pennsylvania (1981).
[10] Mizoguchi, R. et al. "Speech database with an intelligent access mechanism —— SPEECH-DB," Transactions of Information Processing Society of Japan 24(3), pp.271-280 (1983) (in Japanese).