TR－SLT－0066

Phoneme recognition of non-native speech

Bi Lige, Rainer Gruhn

2004/03/31

概要

This report examines phoneme recognition of non-native speakers. We perform phoneme recognition with HTK HVite. A phoneme bigram provides some phonotactic constraint. The recognition results are compared to a canonical phoneme transcription using the DP-alignment algorithm, as implemented in HTK HResults, to get the phoneme recognition accuracy and confusion matrixes. The confusion matrixes are turned into graphics to visualize confusion patterns. From the analysis of confusion matrixes and graphics, we can find which phonemes are frequently mispronounced by speakers from different nations. The influence of the type of acoustic model is examined by recognizing the same speech using monophone, biphone and triphone models. Monophone models achieved highest phoneme accuracy.

（株）国際電気通信基礎技術研究所
音声言語コミュニケーション研究所
〒619－0288「けいはんな学研都市」光台二丁目 2 番地 2 TEL：0774－95－1301

Advanced Telecommunication Research Institute International
Spoken Language Translation Research Laboratories
2-2-2 Hikaridai "Keihanna Science City" 619-0288,Japan
Telephone:+81-774-95-1301
Fax :+81-774-95-1308

1 Overview

The recognition of non-native speech is harder than. that of native speakers. The non-native speaker's pronunciation influenced by their native language, and some words are difficult to be pronounced correctly. The mispronunciation by non-native speakers from different nations are quite different. In this research, we perform phoneme recognition of non-native speakers according to different nations. We compare the results with a canonical phoneme transcription, then arrangement the phoneme which are frequently mispronounced according to nations. When performing usual recognition using monophone models, biphone models or triphone models, triphone models will normally archieve the best results. But in the case of non-native speech recognition, we do not know which models can get the best accuracy. It is necessary to experiment.

2 Approach

This report examines phoneme recognition of non–native speakers. We perform phoneme recognition with HTK HVite. A phoneme bigram provides some phonotactic constraint. The recognition results are compared to a canonical phoneme transcription using the DP–alignment algorithm, as implemented in HTK HResults, to get the phoneme recognition accuracy and confusion matrixes. The confusion matrixes are turned into graphics to visualize confusion patterns. From the analysis of confusion matrixes and graphics, we can find which phonemes are frequently mispronounced by speakers from different nations. The influence of the type of acoustic model is examined by recognizing the same speech using monophone, biphone and triphone models. Monophone models achieved highest phoneme accuracy.

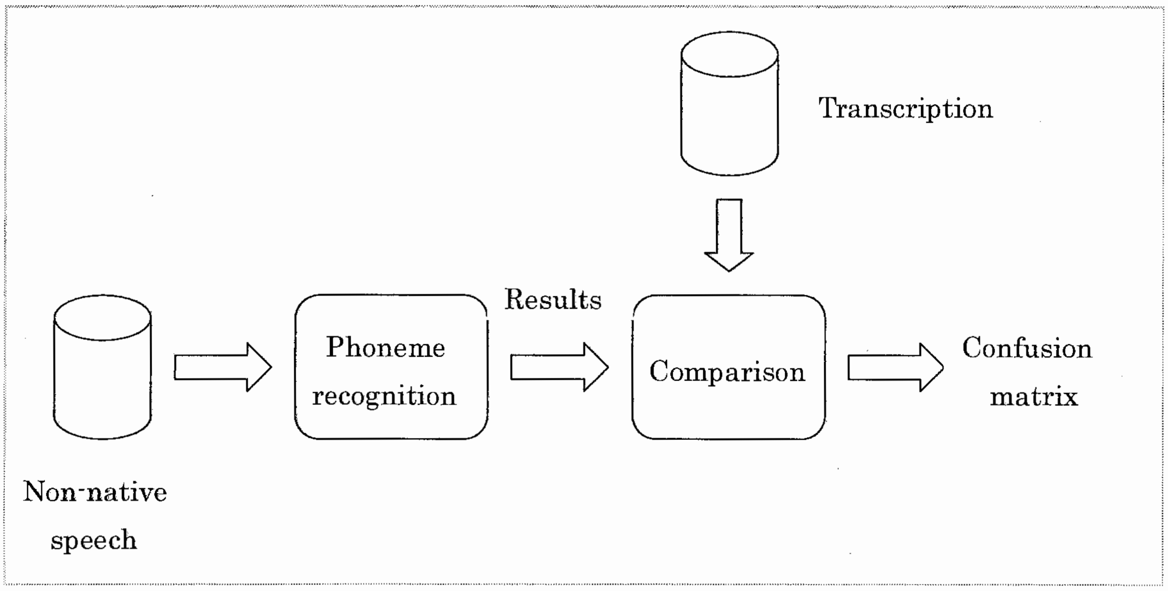

Figure 1 : Flowchart

3 Database

The data of the experiment is the ATR non−native English database. There are 118 speakers in total. Every speaker give us 12 minutes speech, there are 28 hours in total. And the contents are travel conversation.

NNS−DB	Chinese	French	German	Indonesian	Japanese	native	other	total
Number	15	16	17	15	36	9	10	118
Age	21−52	21−42	23−43	24−43	21−45	20−40	31−42	21−52

Table 1 : Number of speakers and age distribution for each language

A subset of the data has been chosen for development purpose. We use it to determine the option −s which is used in the command (1) next page. Examining only a subset of the data speeds up parameter setting experiments .

4 Experiment

4.1 Setup

This experiment performs speech recognition using HTK.

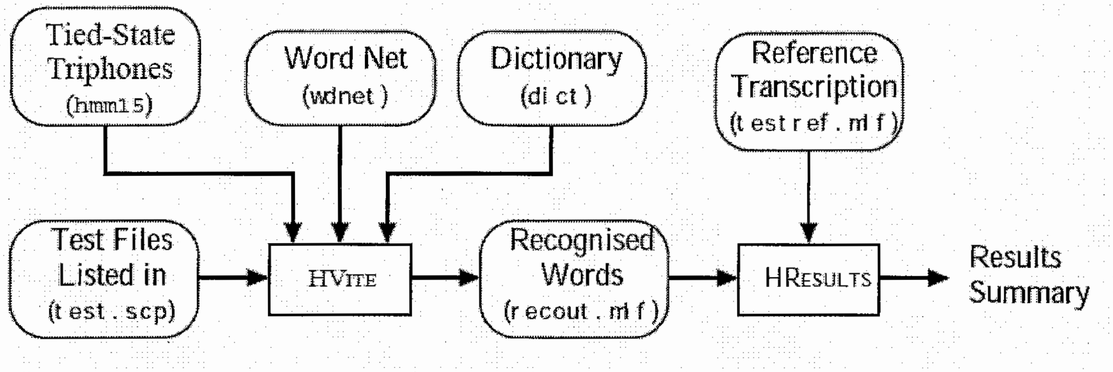
Tied-State Triphones (hmm15)
Word Net (wdnet)
Dictionary (dict)
Reference Transcription (testref.mlf)
Test Files Listed in (test.scp)
HVite
Recognised Words (recout.mlf)
HResults
Results Summary

Figure 2 : recognition and evaluation with HVite and HResults (from[1])

Figure 1 is the base principle of speech recognition. This time we perform phoneme recognition. So a phoneme bigram based "word" net and a dummy phoneme dictionary. There are two commands which are used frequently in this experiment:

```
HVite -H hmm.biph.mix10 -l "*" -S JP.scp -I JP.mlf -C
      config.phonerec.bigram -w bigfn_bi.net -p 0.0 -s
      10.0 -A rbiphone.dict rbiphone
```

------(1)

```
HResults -p -I transcription_from_align.mlf
      monophone.list results.mlf
```

------(2)

In the command (1) the options −p and −s set the word insertion penalty and the grammar scale factor, respectively. The option −p 0.0 is quite good in the experiment, but it is necessary to determine the option −s for each acoustic model.

We perform it using Development data, the numerical value of −s is chosen from 1 to 15, the best numerical value which can achieve highest accuracy is chosen for each acoustic model.

About the numerical value of ?s, at first we choose it in big distance 0, 10, 20 and 50, the accuracy between 0 and 20 is better. Then choose it in smaller distance 5, 15, and so on. The best value almost between 0 and 10, we using loop test the value from 0 to 15 cautiously.

Model	Monophone	Biphone	Triphone
Language model scale factor –s	8.0	10.0	6.0

Table 2 : option –s for each acoustic model

4.2 Comparison of three acoustic models

After find out the best value for option ?s, then we perform the phoneme recognition using each acoustic model. Monophone models achieved highest phoneme accuracy. In common, triphone models can get best accuracy. But for non-native phoneme recognition, the context relation of the non-native language is much weaker than the native language. So monophone models can get highest accuracy.

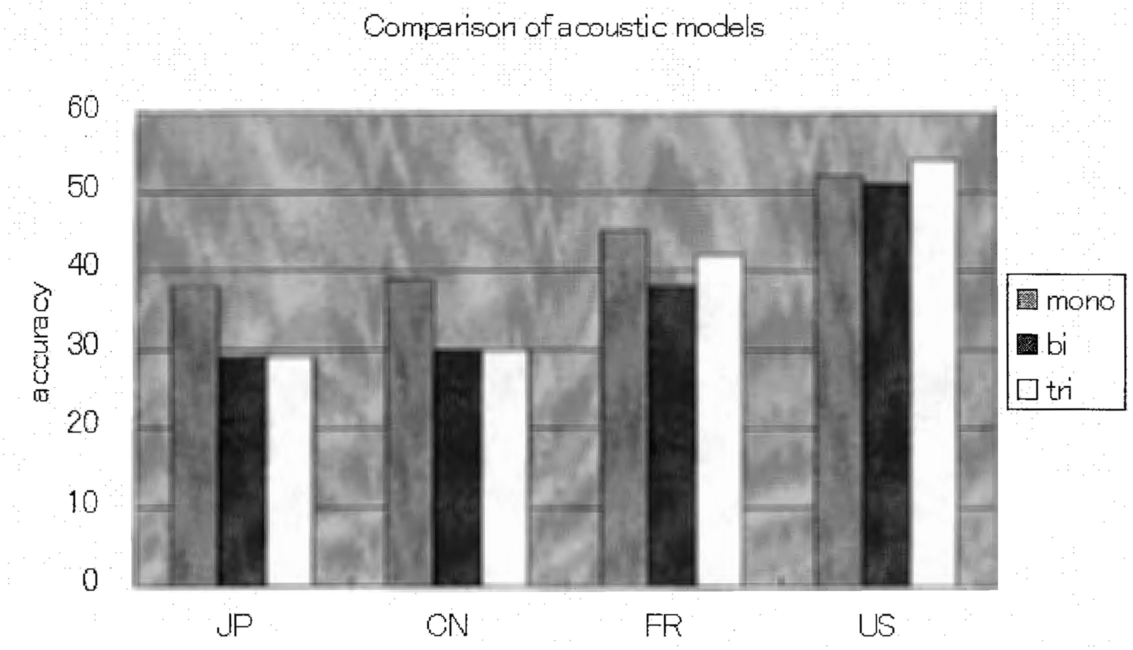

Figure 3 : comparison of acoustic models

The following list of phonemes was used:

DH, AE, T, S, IY, AA, R, K, EY, N, F, EN, DX, AXR, AX, L, IH, M, IX, Z, G, D, B, AY, AH, V, AO, Y, ER, W, TH, UW, OW, SH, UH, P, NG, HH, JH, CH, AW

The acoustic models were trained and evaluated on the Wall Street Journal Corpus (about 70 hours of read speech) with HTK and evaluated as baseline on the Wall Street Journal Hub2 Set.

Model type	#states	#mixtures	Hub2 WA
Monophone	132	16	80.8
Right-context Biphone	3000	10	86.8
Crossword Triphone	9600	12	93.6

Table 3 : word recognition rates for native speech (WSJ)

4.3 Confusion matrixes

We can get the confusion matrixes using HTK HResult, turned confusion matrixes into graphics, so we can analyze confusion patterns visually. Here is a graphic of Japanese speaker's confusion matrix(using monophone models):

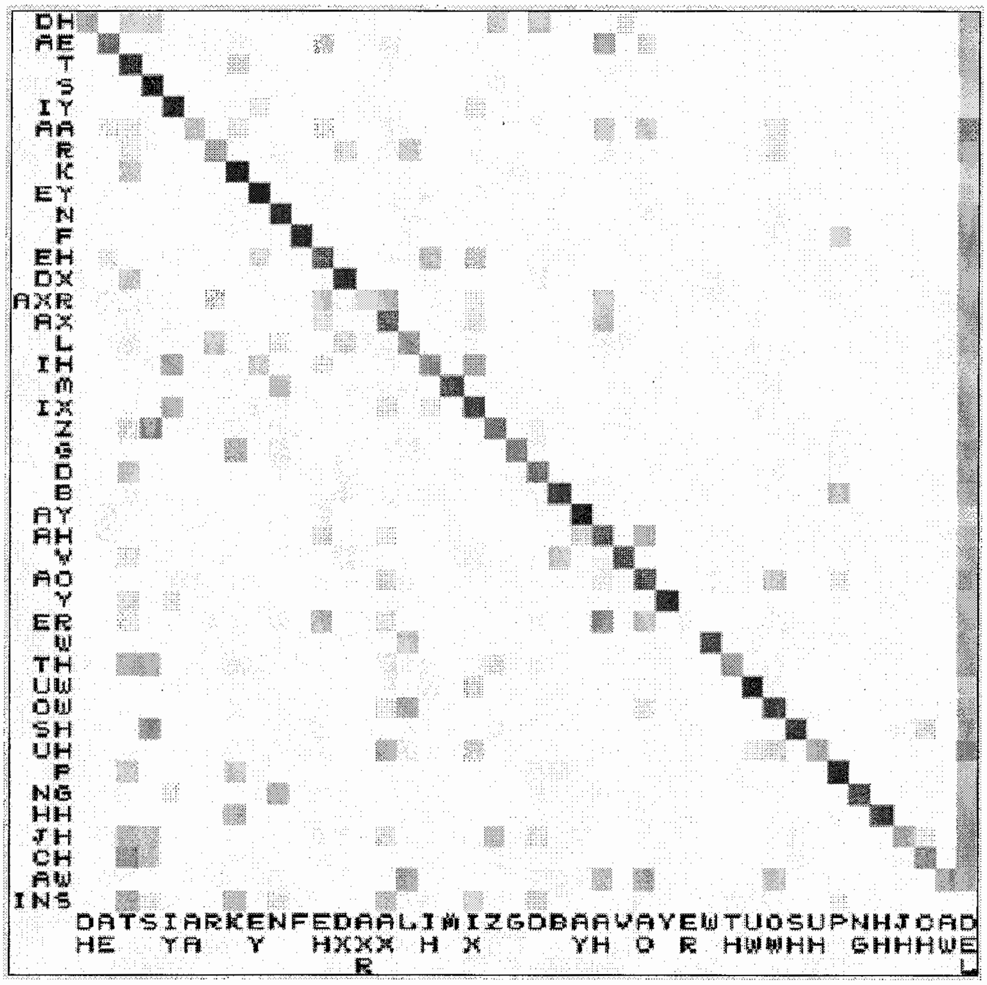

Figure 4 : Confusion matrix of Japanese speakers

From the graphic we can find out some mispronunciations easily:

/er/	>>	/eh/
/er/	>>	/ah/
/er/	>>	/ao/
/l/	>>	/r/
/z/	>>	/s/

/sh/ >> /s/

/v/ >> /b/

/ch/ >> /t/

……

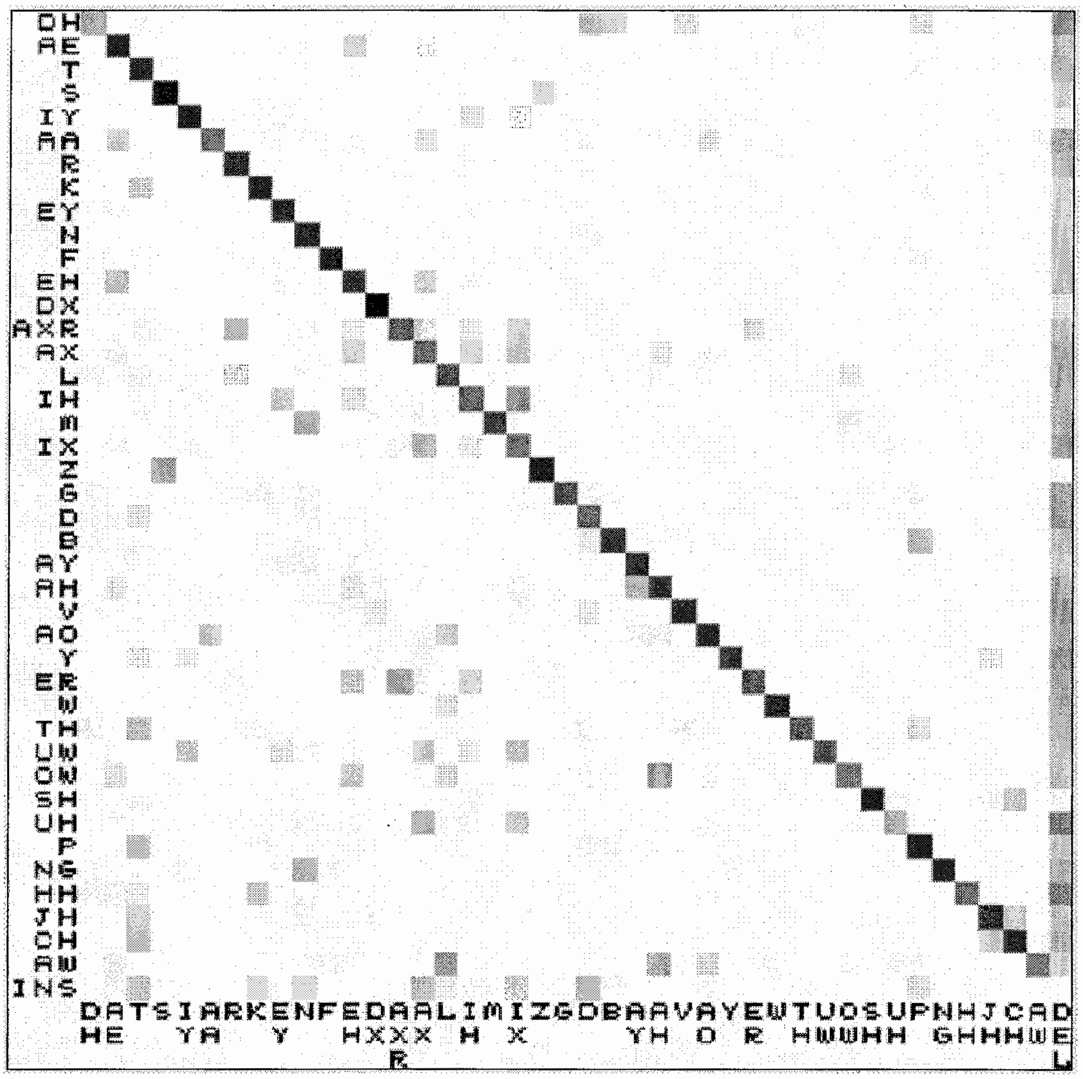

Figure 5 : Confusion matrix of USA speakers

From Figure 5 we can also find out some mispronunciations for native speakers, but much less than in Figure 4 with Japanese speakers:

/z/ >> /s/

/m/ >> /n/

/th/ >> /t/

/ow/ >> /ah/

……

The accuracy of phoneme recognition is not good enough, there are quite a few mispronunciations on the graphic even US speakers. Another way it told us that which two phoneme are very similar. Maybe the next graphic is an objective graphic which described the difference of two matrixes above.

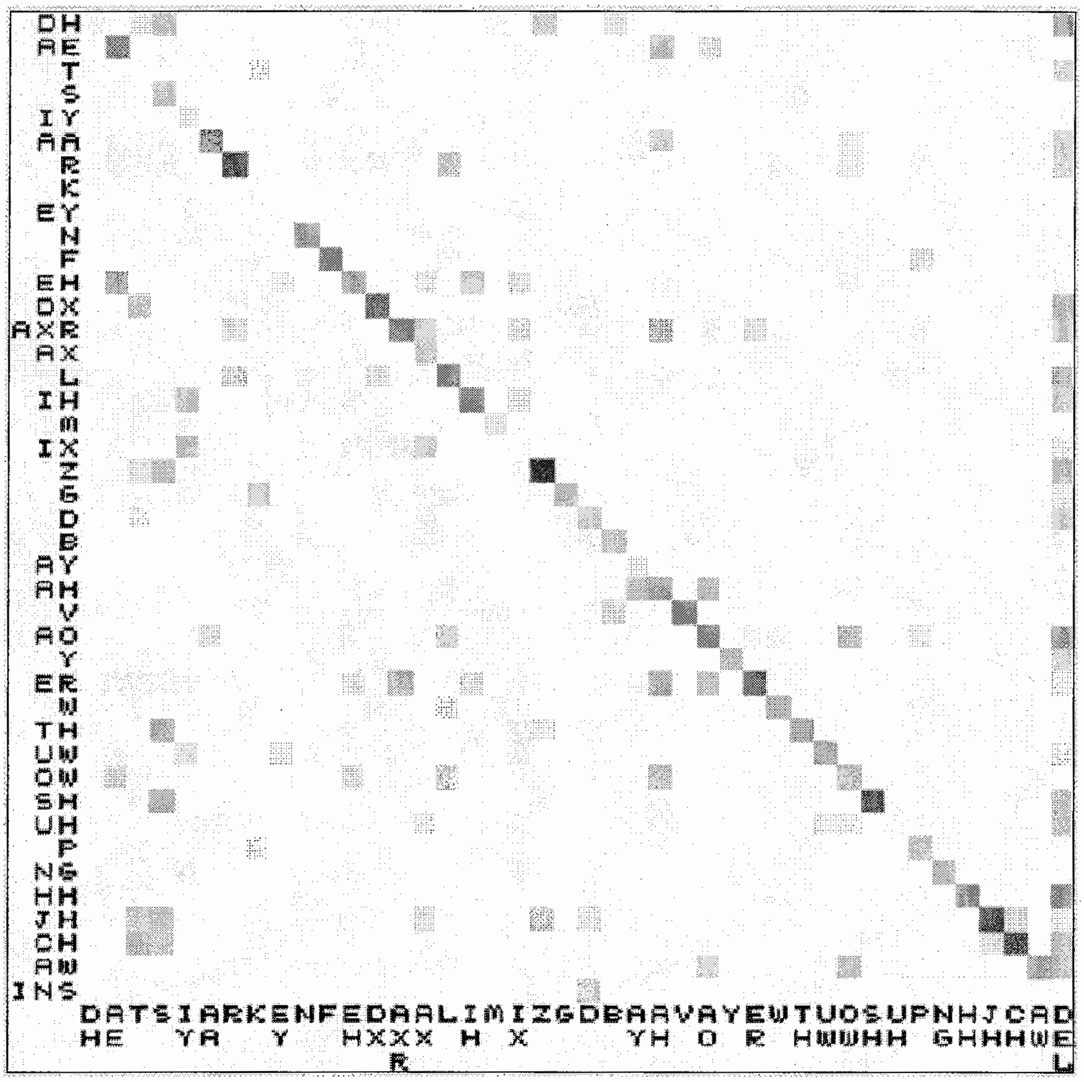

Figure 6 : Matrix difference of Japanese speakers and USA speakers phoneme confusions

From the Figure 5 we also can find out some same mispronunciations which appeared in Figure 3, and there are so many differences between Japanese speakers and USA speakers. Now we can get the graphic between two speaker groups, but we do not know concretely about the differences.

We can use Euclidean distance to compare the different speakers:

d	List2	GER	FR	INN	CN	US
List1	76	96	98	98	119	141

Table 4 : Euclidean distance

List1 and list2 are Japanese speakers which are divided two group at random. List1 is of about the same size and gender distribution as list2. Table 4 also compares the phoneme misrecognitions between the Japanese speakers of list 1 and the speakers from other countries.

5 Summary/Discussion

Non-native speech recognition is more difficult than native speech recognition, and recognition by phoneme unit is more difficult than recognition by word unit. The accuracy of phoneme recognition is quite low even for native speakers. Because of the context relation in non-native speech is much weaker than for native speech, phoneme recognition of non-native speech using monophone models can archieve higher accuracy relative to context-dependent acoustic models. From the confusion matrix graphics, mispronunciation patterns can be extracted. They are depending on the mother language of the speakers.

A Manual

1 path

There are three directories “monophone_models/”, “biphone_models/” and “triphone_models” which have the same pattern. They all have directories as follows:

“recout/”	Results after perform HVite (JP.mlf)
“results/”	Station before perform HResults (JP_ntc.mlf)
“conf_matrix/”	Confusion matrixes (JP.mat)
“graphic/”	Matrixes (JP_mat) and graphics (JP.pgm) (JP_mat is a matrix 100 minus percent)

Others :

“data/”	Non-native data
“refs/”	Standard phoneme transcription which is used in HResults
“wlist/”	Phoneme list used in HResults

2 python program in “pyth_program/”

connectMLF.py

Description:

Connect two MLF files

Usage:

connect.py <infile_M.mlf> <infile_F.mlf>

⇨ generates output file with the name *infile.mlf*

Example:

connect.py JP_M.mlf JP_F.mlf -> JP.mlf

parse_monophone.py

Description:

Parse the monophone models recognition output

Usage:

parse_monophone.py <infile.mlf>

⇨ generates output file with the name *infile_ntc.mlf*

Example:

parse_monophone.py JP.mlf -> JP_ntc.mlf

parse_biphone.py

Description:

Parse the biphone models recognition output

Usage:

parse_biphone.py <infile.hlf>

⇨ generates output file with the name *infile_ntc.mlf*

Example:

parse_biphone.py JP.mlf -> JP_ntc.mlf

parse_triphone.py

Description:

Parse the triphone models recognition output

Usage:

parse_triphone.py <infile.hlf>

⇨ generates output file with the name *infile_ntc.mlf*

Example:

parse_triphone.py JP.mlf -> JP_ntc.mlf

percentage_100.py

Description:

Arrange the confusion matrixes in order to analyze matrixes easily

Usage:

percentage_100.py <infile.mat>

⇨ generates output file with the name *infile_mlf*

Example:

percentage_100.py JP.mat -> JP_mat

mat_gra.py

Description:

Make the matrix graphics

Usage:

mat_gra.py <infile_mat>

⇨ generates output file with the name *infile.pgm*

Example:

mat_gra.py JP_mat -> JP.pgm

dif.py

Description:

Calculate the Euclidean distance and get a differ matrix

Usage:

dif.py <infile1_mat> <infile2_mat>

⇨ generates output file with the name "*dif_infile1_infile2.pgm*" and print *"Euclidean distance infile1 infile2 : d"*

Example:

dif.py JP_mat GER_mat -> dif_JP_GER.pgm

-> euclidean distance GER JP : 96.97

B Confusion matrix graphics

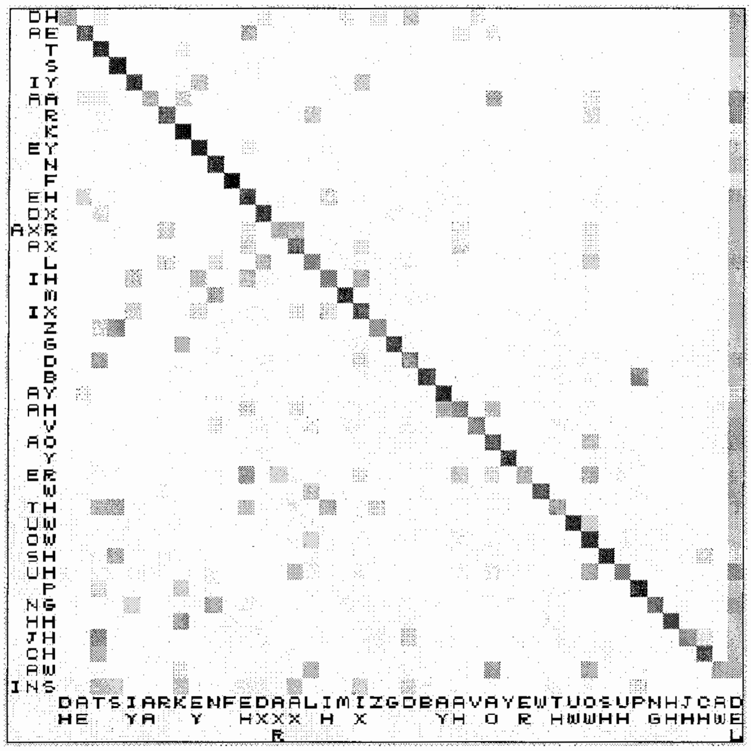

Figure 7 : Confusion matrix of Chinese speakers

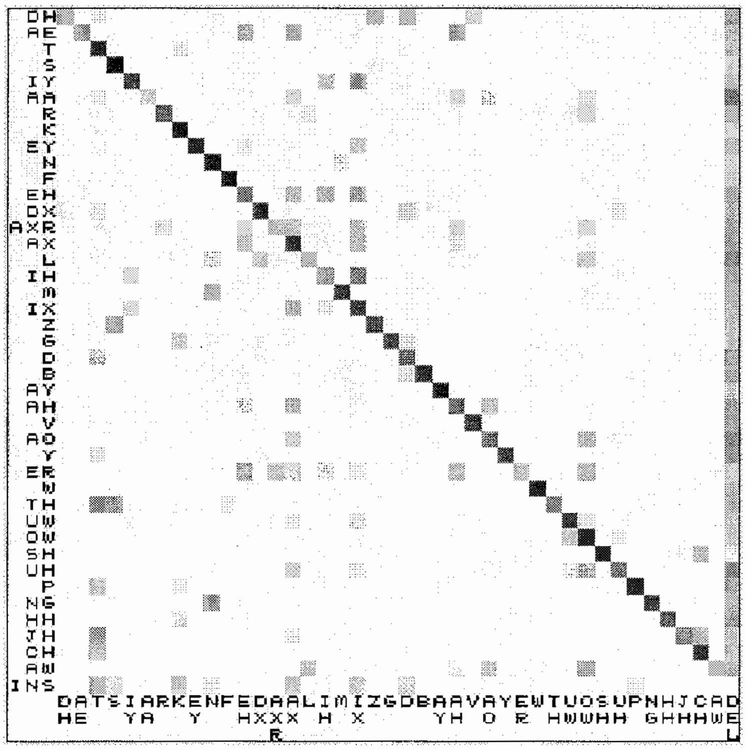

Figure 8 : Confusion matrix of France speaker

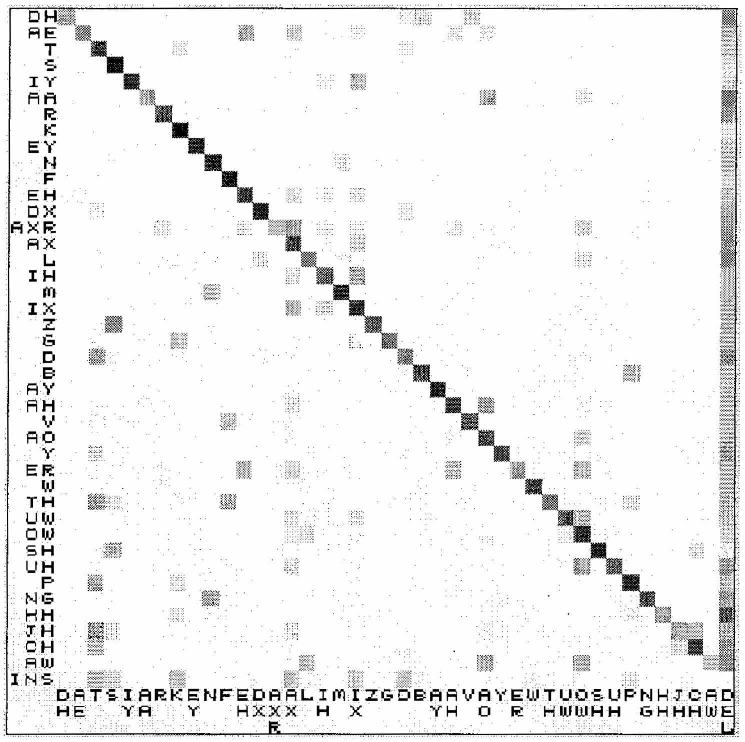

Figure 9 : Confusion matrix of German speakers

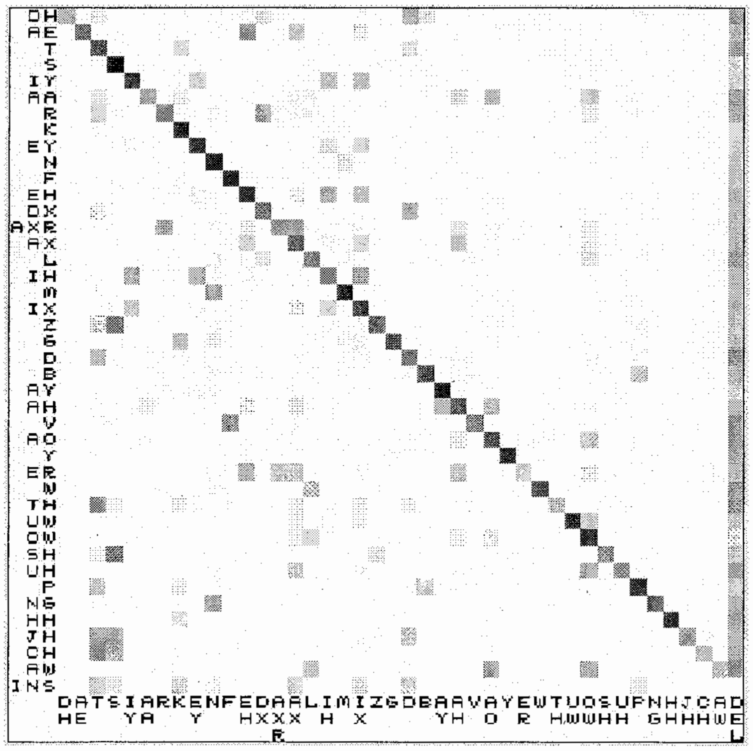

Figure 10 : Confusion matrix of Indonesian speakers

References

[1] S Young et al. The HTK Book. Entropic Ltd, 1999