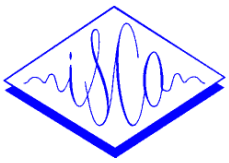

2nd International Conference on
Spoken Language Processing (ICSLP 92)
Banff, Alberta, Canada
October 12-16, 1992

A Phoneme Labelling Workbench using HMM and Spectrogram Reading Knowledge

Shingo FUJIWARA, Yasuhiro KOMORI † *and Masahide SUGIYAMA*

ATR Interpreting Telephony Research Laboratories
2-2, Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-02, Japan

ABSTRACT

This paper proposes a workbench for the phoneme labelling of speech data, that acts as a powerful tool in reducing the effort required to create phoneme labels. The proposed workbench consists of two modules: a user interface module and a phoneme segmentation engine that performs automatic phoneme segmentation. An operator can label speech data interactively by using the window and referring easily to automatic phoneme boundaries. The phoneme segmentation engine is based on the hidden Markov model (HMM) and spectrogram reading knowledge (SRK). The performance of the phoneme segmentation engine was estimated with a 5,240 Japanese word speech database. The segmentation rates of the engine were 96.1%(50ms) and 89.1%(30ms). The quality of phoneme labels produced by operators using the workbench was then estimated. The average error of the created labels was about 6ms, with the standard deviation at about 10ms. The workbench architecture, the user interface and the performance of the phoneme segmentation engine are presented. An implemented workbench is also described.

10.21437/ICSLP.1992-253

1 INTRODUCTION

In speech processing research, speech databases have played important roles. In particular, phoneme-labelled databases have contributed to the improvement of speech recognition systems[1, 2, 3]. Conventionally, phoneme labels are created by human experts who painstakingly read a sound spectrogram. Considerable experience and knowledge are required to correctly and consistently label phonemes for speech data. Consequently, various automatic phoneme labelling systems have been developed, and several speech processing tools have been implemented[4, 5, 6, 7].

Generally speaking, a tool to aid phoneme labelling requires the following features: 1) a user-friendly interface and 2) automatic phoneme segmentation. A user interface is expertly implemented by using a graphical user interface, such as SunView and X Window System, on an engineering workstation (EWS). An automatic phoneme segmentation feature should be designed to accurately detect boundaries. It should also have a simple architecture, which makes it easy to refine the system. However, most current phoneme segmentation systems use either simple acoustic features or the hidden Markov models, which can perform only rough segmentation. Other systems, e.g., rule-based, require a very complex architecture to obtain better performance.

The phoneme HMM (hidden Markov model) provides phoneme segmentation in a very simple manner. The HMM stochastic criterion limits the range of variation in estimating the phoneme boundary, although HMM segmentation is not accurate enough to determine exact boundaries.

Phoneme segmentation based on spectrogram reading knowledge (SRK) is also possible because this method allows a human expert to determine exact phoneme boundaries by carefully observing sound spectrograms[6]. However, because it uses various precise acoustic features, the system is too complex to avoid complicated rules for accurate phoneme labelling.

In this paper, a workbench to aid phoneme labelling is proposed, that acts as a powerful tool in reducing the effort needed to label speech data. The proposed workbench has a user-friendly interface and an automatic phoneme segmentation engine, in which HMM segmentation and SRK-based segmentation are integrated[10]. In the following section, the architecture of the phoneme labelling workbench and the performance of the phoneme segmentation engine are presented. An implemented workbench and its performance are also discussed.

2 PHONEME LABELLING WORKBENCH

Figure 1 shows a proposed phoneme labelling workbench for helping an operator to create and maintain phoneme label data. The workbench consists of two modules: a user interface module and a phoneme segmentation engine. The two modules have a client/server relationship, where the phoneme segmentation engine functions as the server and the user interface module is the

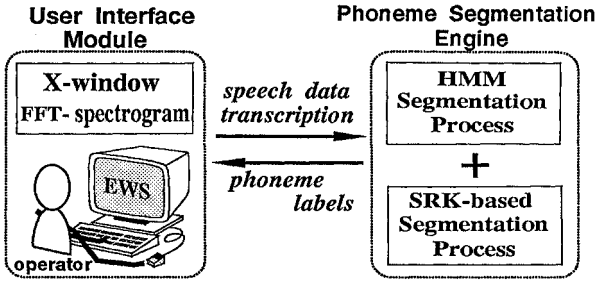

Figure 1. A phoneme labelling workbench

†CANON Information Systems Research Center, 890-12, Kashimada, Saiwai-ku, Kawasaki-shi, Kanagawa 211, Japan

client. The user interface module displays several acoustic features and automatically detected phoneme boundaries and controls the user's operation. The phoneme segmentation engine receives speech data and their phonemic transcription from the client. The engine performs phoneme segmentation and transfers boundary information back to the client. An operator can perform phoneme labelling interactively through the user interface module by referring to automatically detected phoneme boundaries. The operator can also modify the existing phoneme label database.

2.1 User Interface Module

The user interface module is implemented using the X Window System. The module is designed to customize itself through a startup file (e.g., ".lwbrc"). The window arrangement and acoustic analysis can be specified in the startup file. For instance, an operator can specify the following:

- acoustic analysis conditions, such as the number of FFT channels, LPC order, cepstrum order, frame size, frame shift width, and the sampling frequency of the input speech data;
- acoustic parameters to be displayed;
- the position and vertical size of each sub-window (panel);
- time range in a page and overlap time between pages;
- label symbols in a popup menu to input phoneme labels.

Using the startup file, the workbench window can easily be arranged as desired, and acoustic analysis can be performed under the required specification.

Figure2 shows an example of a workbench window on a display of a workstation. The window shows the first page of the input speech data, which is between 0 ms and 1200 ms. The window has a main panel, a label panel and several sub-windows (e.g., waveform sub-window). The main panel, which includes seven items (e.g, "Wave", "NextPage"), at the top of the window controls viewport, file I/O and so on. Acoustic parameters of input speech, such as waveform, power and the norm of the differential cepstrum (Δcepstrum) are displayed in each sub-window. Phoneme boundaries automatically detected by the phoneme segmentation engine and an FFT-spectrogram are also

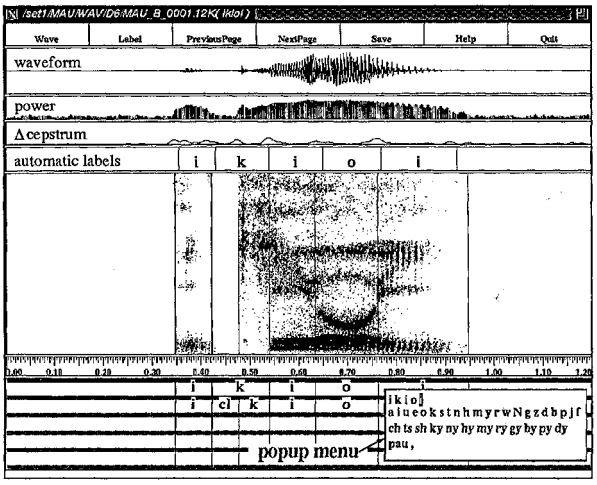

Figure 2. A window of the phoneme labelling workbench

displayed. The time scale is located below the spectrogram sub-window. The label panel to manipulate phoneme labels is located below the window, which consists of five layered sub-panels. The sub-panels are each used to manipulate a layer of phoneme label data corresponding to the structure of ATR phoneme label database[3]. For example, the first layer is for phonemic transcription where alphabetical symbols of Romanized-Japanese are expressed in the Hepburn system.

When a mouse button is clicked in the sub-window of FFT-spectrogram, a vertical line that represents the phoneme boundary is drawn in the spectrogram sub-window and the label panel. After drawing the phoneme boundary lines, phoneme label symbols are entered. The part enclosed by two boundary lines on the label panel is selected by a mouse button click. Then, a label symbol is entered by key-typing or by selecting a label symbol in a popup menu, which is displayed after another mouse button click on the selected part. In this way, an operator can easily determine phoneme boundaries by referring to the automatically detected phoneme boundaries along with a FFT-spectrogram, waveform and so on. One can also interactively manipulate phoneme label data through the label panel by using a mouse and keyboard.

2.2 Phoneme Segmentation Engine

The phoneme segmentation engine is illustrated in Figure3. The engine functions as a segmentation server, where speech data and their phonemic transcriptions are received as a client's request from a user interface module. Then, phoneme boundaries with their labels are transferred back to the client. In order to achieve good segmentation performance, the engine utilizes two processes: the HMM segmentation process (HMM process) and the SRK-based segmentation process (SRK process). These processes are integrated to exploit the strong points of each, and phoneme segmentation is performed by communicating information between the two processes. This iterative communication improves segmentation performance.

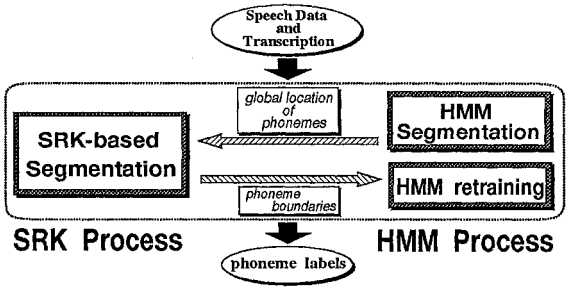

Figure 3. A phoneme segmentation engine

HMM process

The HMM process performs rough phoneme segmentation and is controlled in a simple top-down fashion. The initial phoneme HMMs are trained without phoneme labels, using all the speech data by which the HMMs are concatenated according to each phonemic transcription. Phoneme alignment is then performed using the Viterbi decoding algorithm. In this way, the HMM process determines the rough locations of phoneme boundaries

according to a stochastic criterion. The boundaries are then passed to the SRK process.

The HMMs are retrained by using the SRK-process outputs as HMM training constraints. The retrained HMMs can generate more accurate boundaries, which are passed back to the SRK process, than those generated by the initial HMMs because label-trained HMMs can perform accurate phoneme segmentation[8, 9].

SRK process

The SRK process detects exact phoneme boundaries with *a priori* described rules based on spectrogram reading knowledge (SRK). The HMM-process outputs are used as initial positions to search for boundaries, which make it easier to focus on a relevant phoneme and its acoustic features for boundary detection. This in turn simplifies describing the segmentation rules and facilitates control of the rules.

The detected boundaries are transferred back to the HMM process to retrain the HMMs. Retraining the HMMs improves the segmentation engine's overall performance as well as that of the HMM process.

Segmentation rules are described with reference to FFT-spectrograms of various speech data. Power, differential power and Δspectrum over several FFT channels in the spectrogram are used in the rules as acoustic features, where Δspectrum indicates the norm of the differential FFT vector. Using the rules, the SRK process searches for boundaries in proximity to the positions determined by the HMM process. If acoustic evidence, i.e., an acoustic feature related to a relevant phoneme, is found, the SRK process is able to detect the exact boundaries of the phoneme. If acoustic evidence is not found, the SRK process outputs the boundary initially determined by the HMM process.

3 EXPERIMENTS ON THE PHONEME SEGMENTATION ENGINE

Experiments were performed with the ATR speech database[1] to estimate the performance of the phoneme segmentation engine (PSE). The experiments evaluated PSE performance with initial HMMs and retrained HMMs. Table 1 shows the acoustic analysis conditions used in the experiments. The continuous density single Gaussian HMMs of 25 Japanese phonemes were adopted in the HMM process. The SRK process drove approximately 300 rules, which were described with reference to only one male speaker's training data (MAU).

The experimental speech data were three sets of a 5,240 word database uttered by male speakers (MAU, MHT and MSH). Half of each speaker's data was used to train the initial HMMs. The other half was used for testing. Phoneme segmentation of the test data was performed in the following sequence:

step 1) Training the initial HMMs (*concatenation training*);
step 2) Phoneme segmentation in the HMM process;
step 3) Boundary detection using the rules of the SRK process.

After step 3, phoneme segmentation using retrained HMMs was also performed in the following sequence:

step 4) HMMs were trained again using the SRK process outputs;
step 5) Phoneme segmentation was completed by the HMM process using retrained HMMs;
step 6) The SRK process detected boundaries by using the new outputs of the HMM process.

Table 1. Acoustic analysis conditions

	HMM-process	SRK-process
Sampling freq.	12 kHz	
Frame width	21.3 ms	5 ms
Frame shift	5 ms	2.5 ms
Window	Hamming window	
Pre-emphasis	$1 - 0.97z^{-1}$	
Method	LPC (14)	FFT (128)
Parameter (Δ= 50ms)	Cepstrum(16)	
	ΔCepstrum(16)	
	Power(1)	
	ΔPower(1)	

Table 2 shows the results of the phoneme segmentation experiments using the initial HMMs and retrained HMMs, in which about 16,000 boundaries were determined. HMM_i + SRK and HMM_r + SRK denote PSE with the initial HMMs and PSE with the retrained HMMs, respectively. Error is calculated as the difference between the determined boundary and the boundary hand-labelled by human-experts. "Ave." and "S.D." denote the average and the standard deviations of the error. The row marked "average" indicates the average performance for the three speakers. The segmentation rate is defined as the percentage of boundaries within *50 ms* or *30 ms* of the hand-labelled boundaries. PSE performance with the initial HMMs resulted in an average error of 19.8 ms and segmentation rates within *30 ms* (*50 ms*) of 83.5% (92.5%). By retraining the HMMs, the average error of the PSE outputs decreased to less than 15 ms, and segmentation rates within *30 ms* increased to 89.1%.

Table 2. Phoneme segmentation results of PSE

	Speaker	Error (ms)		Seg. Rate (%)		
		Ave.	S.D.	50ms	30ms	10ms
HMM_i + SRK	MAU	11.8	20.4	96.8	87.1	59.4
	MHT	10.7	18.1	96.3	89.3	60.1
	MSH	36.8	81.1	84.4	74.1	43.6
	average	19.8	—	92.5	83.5	54.4
HMM_r + SRK	MAU	10.2	18.5	97.7	91.1	60.6
	MHT	9.6	16.5	96.9	91.0	63.1
	MSH	14.7	26.9	93.7	85.1	51.6
	average	11.5	—	96.1	89.1	58.4

The experimental results indicate that the rules of the SRK process are speaker-independent, although they were described with reference to only one speaker's data (MAU). The results also indicate that retraining HMMs improves the overall performance of the phoneme segmentation engine.

4 IMPLEMENTATION OF PHONEME LABELLING WORKBENCH

4.1 Implemented Phoneme Labelling Workbench

A phoneme labelling workbench (LWB) was implemented on UNIX workstations as a type of client/server system. Figure 4 illustrates the system configuration of the implemented LWB. A user interface module, which is the main body of the LWB, is driven on each local workstation ($WS_{1\sim3}$). The user interface

modules are connected to a phoneme segmentation engine that is driven as a segmentation server on a host workstation (WS_A), via TCP/IP network communication. The portability of module is such that it works on various platforms, such as DECstation, SPARCstation, HP, because the module is described in programming language C using the lowest-level X interface library (Xlib) of the X Window System (X v11 release 4). In the phoneme segmentation engine, the SRK process is implemented on a DECstation 5000 using ART-IM, which is a commercial expert system package. Rules for the SRK process are described as speaker-independent rules. The HMM process runs on an HP9000/s750 (WS_B), which also functions as an HMM segmentation server. The HMM process has phoneme HMMs trained by using eight speakers' speech data as initial HMMs. The HMM retraining feature is not yet implemented.

An operator uses one of the local workstations and operates the LWB for a given set of target speech data and its transcription.

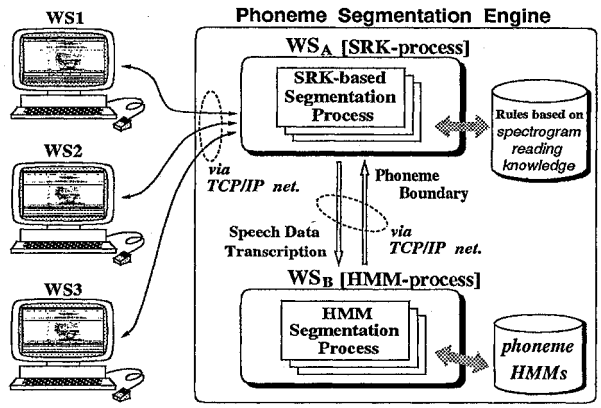

Figure 4. The implemented phoneme labelling workbench

4.2 Quality of Labels Created with the LWB

An experiment was carried out to evaluate the quality of phoneme labels created by operators using the implemented labelling workbench. Two operators used LWB and produced phoneme label data, where the target speech data were 216 isolated words. Table 3 shows the result for the labels produced by LWB. The average error and standard deviation were approximately 6 ms and 10 ms, respectively, where the references were boundaries hand-labelled by experts[1]. This result indicates that phoneme labels created with LWB have almost the same quality as those created by labelling experts.

Table 3. Phoneme Labelling Result with LWB

	Error(ms)		Seg. Rate(%)		
	Ave.	S.D.	50ms	30ms	10ms
LWB	6.1	10.2	99.2	96.7	73.4

Conventionally, a labelling-operator determines phoneme boundaries by observing a printed sound spectrogram of the target speech data, and maintains a phoneme label data file on an EWS. Using the LWB, it takes $10 \sim 20$ sec to display one page of target data, in which acoustic features and automatically detected phoneme boundaries are displayed. Most time is spent in waiting for a reply from the phoneme segmentation server. Comparing the flexibility, performance and processing speed of the conventional procedure versus the LWB, the implemented LWB should replace the conventional procedure in phoneme labelling.

5 SUMMARY

A phoneme labelling workbench, which works as an aid for labelling-operators to create and maintain phoneme label data, has been proposed. The user interface and phoneme segmentation engine of the workbench were presented. Several experiments on the phoneme segmentation engine were performed. The experimental results indicate that HMM retraining in the phoneme segmentation engine is an effective method for achieving good segmentation performance. An experiment using the workbench proved that the quality of the phoneme labels created with the workbench is almost equal to that of phoneme labels created in the conventional way by labelling-experts. The workbench can easily be adapted to other languages than Japanese by tuning the rules and the phoneme HMMs. Further research should be concentrated on expanding the phoneme segmentation engine to include continuous speech labelling and on improving the user interface.

ACKNOWLEDGEMENTS

The authors wish to thank Mr. Ken Shimomura (Toyo Information Systems co.,ltd.), Miss Hiroko Kida and Miss Yayoi Kitano for their contributions to implementing the user interface module and the trial experiments with the workbench. They also acknowledge Dr. Akira Kurematsu, President of ATR Interpreting Telephony Research Laboratories, and Mr. Shigeki Sagayama, Head of the Speech Processing Department, for their continuous support of this project.

REFERENCES

[1] K. Takeda, Y. Sagisaka, S. Katagiri and H. Kuwabara, "*Construction of an Acoustically-phonetically Transcribed Japanese Speech Database*," IEICE Technical Report, **SP87**(19), pp.25-32 (June 1987).

[2] W. Fisher, V. Zue, J. Bernstein and D. Pallett, "*An Acoustic-phonetic Data Base*," J. Acoust. Soc. Amer. Suppl.(A), **81**(S92) (1987).

[3] Y. Sagisaka, K. Takeda, S. Katagiri, T. Umeda and H. Kuwabara, "*A Large-Scale Japanese Speech Database*," Proc. of ICSLP 90, pp.1089-1092 (Oct. 1990).

[4] K. Maruyama and T. Kawabata, "*Speech Processing Workbench on X-window*," Proc. of the ASJ 88 Spring, pp.91-92 (Mar. 1988).

[5] S. Nakagawa and Y. Hashimoto, "*Segmentation of Continuous Speech by HMM And Bayesian Probability*," Trans. IEICE, **J72-D-2**(1), pp.1-10 (Jan. 1989).

[6] K. Hatazaki, Y. Komori, T. Kawabata and K. Shikano, "*Phoneme Segmentation Using Spectrogram Reading Knowledge*," Proc. of ICASSP 89, pp.393-396 (May 1989).

[7] K. Arai, et al., "*A Speech Labeling System Based on Knowledge Processing*," Trans. IEICE, **J74-D-2**(2), pp.130-141 (Feb. 1991).

[8] S. Fujiwara, Y. Komori and M. Sugiyama, "*A Hybrid System For Phoneme Labelling Based On HMM And Spectrogram Reading Knowledge*," Korea-Japan Joint Workshop on Advanced Technology of Speech Recognition and Synthesis, pp.121-126 (July 1991).

[9] N. Iwahashi, S. Fujiwara, Y. Komori, M. Sugiyama and Y. Sagisaka, "*Generation of Speech Synthesis Units By Automatic Labelling*," Proc. of the ASJ 91 Fall, pp.231-232 (Oct. 1991).

[10] S. Fujiwara, Y. Komori and M. Sugiyama, "*An Integrated System for Automatic Labelling Base on HMM and Spectrogram Reading Knowledge*," to appear in Proc. of ISSPA92 (Aug. 1992).