Development of Web-based Vietnamese Pronunciation Training System

MINH Nguyen Tan
Tokyo Institute of Technology
tanminh79@yahoo.co.jp

JUN Murakami
Kumamoto National College of Technology
jun@cs.knct.ac.jp

Abstract

We developed a web-based Vietnamese language learning system. A learner can hear the standard pronunciation of a Vietnamese word by inputting the corresponding word to the keyboard of the system in the Internet environment. Then the system checks the learner's pronunciation with the standard one and the score is shown on the display monitor immediately. We used the Visual C++ language to describe the system in order to provide a GUI (Graphical User Interface) environment for the users. From the results of the experiments, we confirmed that the learner could be trained to improve his pronunciation of the Vietnamese words by using our system.

1. Introduction

Vietnamese is the official language of Vietnam that is spoken by over 80 million people in Vietnam and other countries. This language is a mono-syllabic tonal language, which means that all words are only one syllable long. Each syllable of the words can be pronounced in six different tones and those tones confer different meanings on the words. If the learner pronounces a word with a wrong tone by mistake, then the meaning of the word will change completely. Therefore, the learner has to pay attention to not only pronunciation but also the tones.

We developed the Vietnamese pronunciation training system for the purpose of helping a leaner of this language to enhance his skills in pronunciation. Our system gives the standard sound files by native Vietnamese speaker to the learner and compares his pronunciation to the standard one by using record-and-playback. From the results of the experiments, it is shown that the learner can be trained to improve his pronunciation of the Vietnamese words by using the system.

2. System design

We designed the system in consideration of the following conditions:

(1) For the purpose that the system can be used by many people, the system is constituted of some common elements, such as a personal computer, a microphone and an earphone.

(2) The system provides a GUI (Graphical User Interface) environment for users.

(3) The size of the system software has to be sufficiently small for the purpose that it can be distributed to the users through the Internet.

In order to satisfy above conditions, we used Visual C++ language to describe the system. The system works normally under following environment:

OS : Windows 98SE/2000/XP.
RAM : more than 64 MB.
CPU : more than 500 MHz.

2.1 Constitution of the system

Our system is constituted of two modules, that is, the input module and the recognition module. The former module acquires the pronunciation of a Vietnamese word which is pronounced by a user, and then analyzes this word to generate the standard pronunciation of the word. The latter module analyzes both the acquired pronunciation and the standard one, and computes the degree of a similarity between that two pronunciations by using a pattern matching technique. Figure.1 shows the constitution of the system.

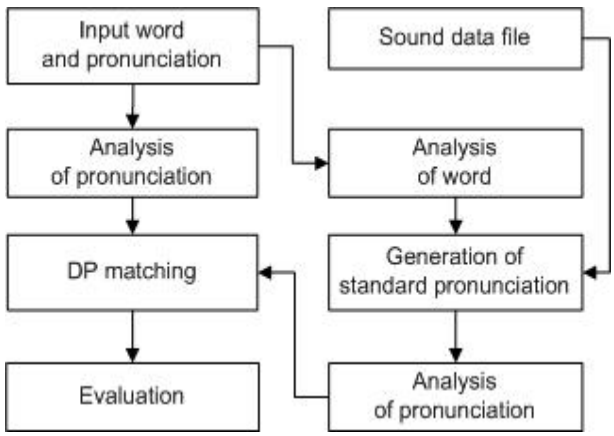

Figure 1: System construction.

2.2 Process of learning by using the system

At first, a user inputs a Vietnamese word to the system from the keyboard. Secondly, he pronounces this word toward the microphone after the standard pronunciation of the word is sounded by the speaker. Then, the system computes the correctness of user's pronunciation by comparing with the standard one,

and displays that correctness by points. The user can master the correct pronunciation of the Vietnamese word by repeating above procedure. Figure.2 shows the user interface window of the system.

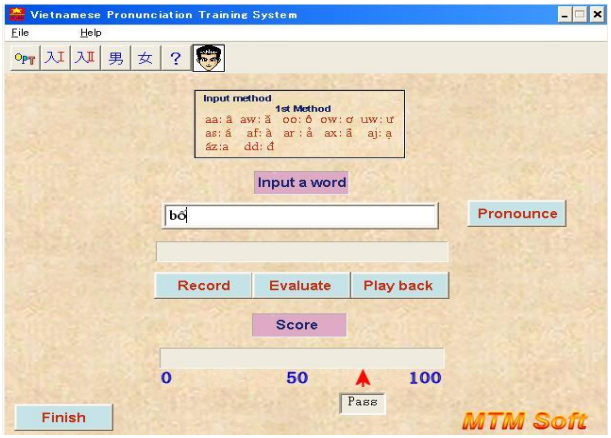

Figure 2: System interface window.

3. Making of the system

3.1 Input module

3.1.1 Speech data file

The sound data of the standard pronunciation of some Vietnamese words is stored in the speech data file. Since actual sound data size of the pronunciation is very large, that data is stored after being decomposed into vowel sounds, consonant sounds and tunes of voice.

In order to generate the pronunciation of almost all of the Vietnamese words, it is needed to store several hundred phonemes in the data file. Since the data size of a phoneme is about 10 to 15 kB when the sound data is sampled at 11025 Hz in the 16 bit monaural Wave form, the speech data file requires about the size of 10 MB.

In practice, we used two files as the speech data file, that is, wordsound.dat and wordpos.dat. The sound data of all of the phonemes are stored in the former file. In the latter file, the size and the position of each phoneme in the wordsound.dat are stored. To state how these files is used, first the size and position of a phoneme is taken out from wordpos.dat, then the sound data can be obtained from wordsound.dat by using acquired information. The construction of speech data file is shown in Figure. 3.

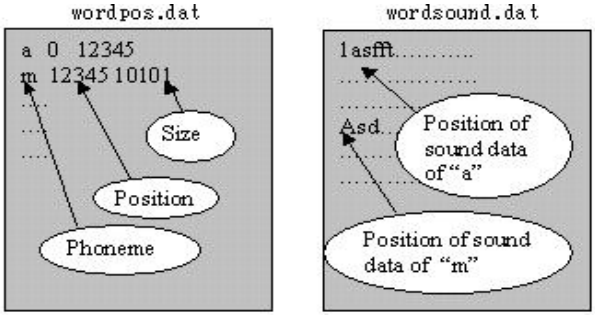

Figure 3: Speech data file.

3.1.2 Word acquisition process

As the Vietnamese words have six different tones, so we provided two sorts of ways to input the tones. The user can input the tones by using combinations of the alphabets and the numerals as a way. As another way, only the alphabets are used to denote the tones. Figure.4 and Figure. 5 show these ways. When a word is acquired from the keyboard by using those ways, then the system shows the word by Vietnamese letters on the display.

1st input \ 2nd input	1	2	3	4	5	6	7	8	9
a	á	à	ả	ã	ạ	â		ă	
e	é	è	ẻ	ẽ	ẹ	ê			
i	í	ì	ỉ	ĩ	ị				
o	ó	ò	ỏ	õ	ọ	ô	ơ		
u	ú	ù	ủ	ũ	ụ		ư		
y	ý	ỳ	ỷ	ỹ	ỵ				
d									đ

Figure 4: One of the means of inputting a Vietnamese letter.

1st input \ 2nd input	a	e	o	u	d	w	s	f	r	x	j
a	â					ă	á	à	ả	ã	ạ
e		ê					é	è	ẻ	ẽ	ẹ
i							í	ì	ỉ	ĩ	ị
o			ô			ơ	ó	ò	ỏ	õ	ọ
u				ư		ư	ú	ù	ủ	ũ	ụ
y							ý	ỳ	ỷ	ỹ	ỵ
d					đ						

Figure 5: Another means of inputting a Vietnamese letter.

3.1.3 Pronunciation acquisition process

The user of the system inputs the pronunciation of the word through the microphone after he saw that word in the display. Then, the user can hear his own pronunciation any times, and he can also pronounce the same word again.

Since it is needed about 1 to 2 seconds to pronounce a Vietnamese word for the user in general, we set a time for recording a word as 3 seconds. The recorded pronunciation data is stored after being sampled in the similar way to the case of the standard pronunciation data (11025Hz, 16 bit monaural Wave format). We used the API (Application Programming Interface) functions of the Windows to describe a program for the recording and the playing back processes.

3.2 Generation of standard pronunciation

Figure. 6 shows the procedure of a generation of the standard pronunciation for the input Vietnamese word. The obtained standard pronunciation can be repeated any times in order that the user can check his own pronunciation with that one.

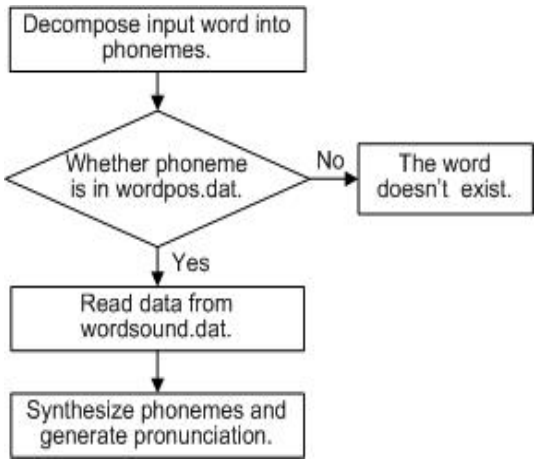

Figure 6: Overview of standard pronunciation generation process.

3.2.1 Decomposition of input word into phonemes

The Vietnamese language has following features:

(1) Every word can be decomposed into one or two phonemes. The words which begin with a consonant are divided into two phonemes, and the other words become a phoneme themselves. There are no words which have more than three phonemes.

(2) There are 26 combinations of consonants to appear in the beginning of a word.

The system decompose the input words into phonemes by exploiting those characters according to the algorithm below:

(1) The system prepares a set of the 26 combinations of consonants.

(2) In case the beginning of the word is a vowel, the word is a phoneme itself. The procedure is finished at this step.

(3) In another case the beginning of the word is a consonant, the system inspects a part of the word from the beginning to the consonant just prior to a first vowel of the word whether that part is in the set of consonants or not. If this part is in the set, then the procedure is finished with informing the user that there is no word like such a word in the Vietnamese.

Otherwise, the accorded combination of consonants in the set is a phoneme and the rest part is another phoneme respectively, so that the word is decomposed into two phonemes. The procedure is completed in this step.

3.2.2 Synthesis of standard pronunciation

After the phonemes are obtained by above procedure, the system read the size and the position of those phonemes in wordsound.dat from wordpos.dat. By using this information, the sound data of the phonemes can be read from wordsound.dat.

In case of one-phoneme-words, the sound data can be used immediately as a standard pronunciation, the other case, two corresponding sound data of phonemes are combined to form a standard pronunciation. In the latter case, obtained standard pronunciation might sound unnaturally for a difference of each pitch of the phonemes. So we adjust the pitches of two phonemes to the pitch of the second phoneme, because the pitch of a two-phonemes-word depends on that of the second phoneme. The fast algorithm in the reference [1] is adopted as a way to adjust those pitches. The following Figure. 7 shows above procedure, and Figure. 8 demonstrates an example of which the pronunciation of the Vietnamese word [ma] is synthesized from the phonemes [m] and [a].

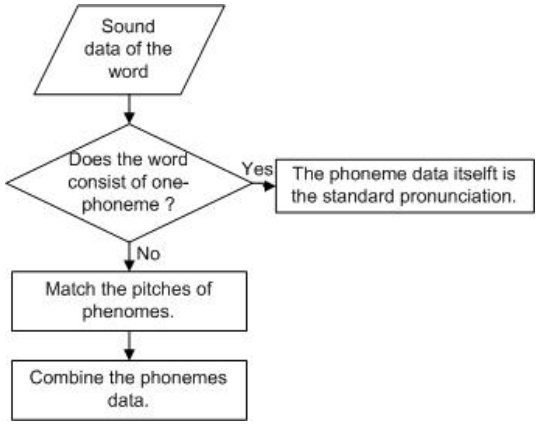

Figure 7: Segmentation and synthesis of pronunciation data.

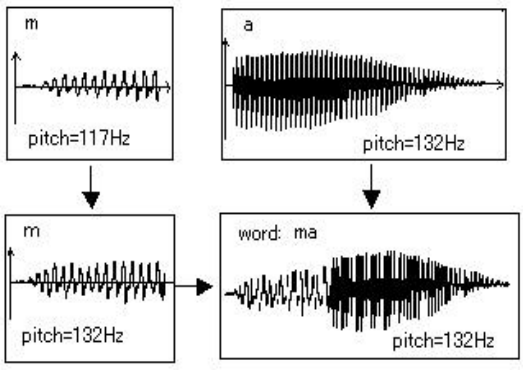

Figure 8: Example of synthesized pronunciation data.

The obtained standard pronunciation can be heard any times for the user by pressing the button from the keyboard.

3.3 Recognition module

In the recognition module, the degree how the input pronunciation of the word corrects in comparison with the standard pronunciation of that word

is computed. As the means to evaluate that degree, the cepstrum analysis [2] and the DP (Dynamic Programming) matching method [3] are exploited.

3.3.1 Detection of voice segment

At first, the section in which the voice signal of input pronunciation is exist has to be detected in order to compare with the standard pronunciation waveform. Although it is difficult to detect this voice segment exactly, we tried to do by using the energy and the number of zero cross of the voice waveform [4].

The waveform is divided into a number of frames at the period of 10 ms, and then the energy $E(n)$ and the number of zero cross $N_z(n)$ are computed for each frame, where n denotes the number of a frame. By using two threshold levels E_1 and E_2 ($E_1 > E_2$) determined from the energies, a conjectural voice segment (n_1,n_2) is detected in all of the extent as following way. That is, the beginning point of the segment n_1 is determined to the first point where the energy is greater than E_1 and besides the energy never drops below E_2 after that point. The terminal point of the segment n_2 is also determined by the same way as the case of n_1.

The conjectural segment (n_1,n_2) is extended toward the outside of this segment by using $N_z(n)$ of each frame. The beginning point n_1 can be moved to the point where the number of frames whose $N_z(n)$ is greater than some value N_0 is more than 3 in the extent of (n_1-25,n_1). By the same way, the terminal point n_2 can be moved too. By the means mentioned above, the voice segment (n_1, n_2) can be detected, provided that E_1,E_2 and N_0 are determined experimentally. Figure. 9 demonstrates an example of the detection. In this example, the beginning point n_1 is moved to $n_1^{'}$, and $n_1^{'}$ is newly denoted as n_1. On the other hand, the terminal point n_2 is not moved.

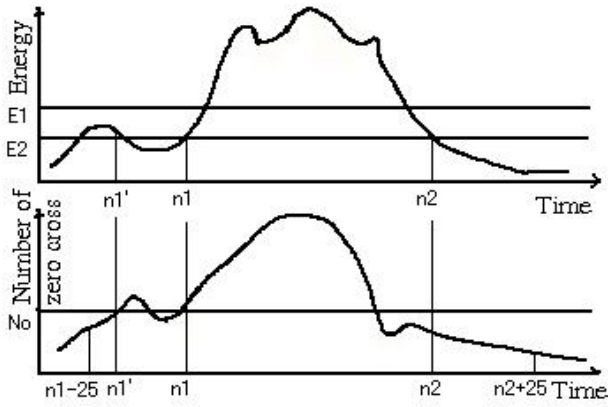

Figure 9: Detection of voice segment.

3.3.2 Cepstrum analysis

The cepstrum is obtained as a result of the inverse Fourier transformation of a logarithm of the power spectrum of a voice waveform. By using the cepstrum, we can calculate the pitch and the envelope of the spectrum of a voice waveform.

Firstly, the voice waveform is analyzed by means of the short time spectrum analysis. That is, the frame of 20 ms long is picked out from the voice waveform, analyzing by the spectrum analysis. This analysis is performed many times to the frame which is picked out from another position in the waveform by shifting the frame 14 ms successively. Since the beginning and the terminal parts of the frame exert an influence to the spectrum, the frame is multiplied by the Hamming window function [5]. In this way, we determined the length of a frame 20 ms and the interval of the shifting 14 ms experimentally. The obtained cepstrums is treated as vector numbers and stored in arrays. Figure.10 shows the procedure of cepstrum analysis.

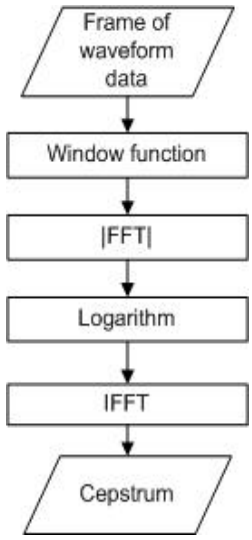

Figure 10: Procedure of cepstrum analysis.

3.3.3 Pattern recognition by DP matching method

In order to regularize some time series of cepstrums relating the time axis nonlinearly, the DP matching method is usually adopted.

By using this method, the distance between the cepstrums of the input pronunciation and the corresponding standard pronunciation is calculated.

Above two time series of the cepstrum which are obtained in 3.3.2 are denoted respectively as

$$A = a_1, a_2, a_3, ...a_i..., a_N$$
$$B = b_1, b2, b_3, ...b_i..., b_M$$

where the vector a_i and b_i is the cepstrum of a frame. The distance between the cepstrum a_i and b_i is given by addtoresetequationsubsection

$$d(i,j) = \sum_k d(a_i[k], b_j[k]) \tag{1}$$

$$d(a_i[k], b_j[k]) = \|a_i[k] - b_j[k]\|^2 \tag{2}$$

where $a_i[k]$ and $b_j[k]$ are the k-th elements of each vectors. By using the equation (1) and (2), the distance between A and B is defined as

$$D(A,B) = \frac{G(N,M)}{N+M} \tag{3}$$

$$g(i,j) = min \begin{pmatrix} g(i,j-1) + d(i,j) \\ g(i-1,j-1) + 2d(i,j) \\ g(i-1,j) + d(i,j) \end{pmatrix} \quad (4)$$

$$i{=}1..N,\ j{=}1..M$$

where $g(1,1) = 2d(1,1)$, and we use the constant r under the condition of $|i-j| \leq r$.

By using this distance, the system can evaluate the pronunciation of a user. For instance, the smaller the distance becomes, the better the learner can improve his pronunciation.

4. Experimental results

To evaluate our system, experiments are performed on the following conditions:

(1) Arrange a quiet room to avoid an influence of a noise.

(2) A personal computer (CPU : Celeron 800 MHz, RAM : 128 MB, OS : Windows 2000), a microphone and speakers are used to make up the system.

(3) The learners practice Vietnamese pronunciations time and again according to the way that is described in section 2.2.

Table 1 shows the scores of the experiments which are carried out to three subjects. Each subject pronounces a one-phoneme-word and a two-phonemes-word (shown in Figure. 11) five times respectively.

One-phoneme word	Two-phonemes word
a : letter "a"	na : custard-apple
o : aunt	ma : ghost
ô : stain	bô : father

Figure 11: Vietnamese words which is used for the experiments and their meanings.

Table 1. Experimental result.

Subject	A		B		C	
Word	o	na	a	ma	ố	bố
1st time	50	30	50	80	60	55
2nd time	70	70	40	100	70	60
3rd time	100	60	60	90	90	80
4th time	90	100	70	100	80	75
5th time	100	85	100	100	95	80
Average score	82	69	64	94	79	70

From the table, we can find that the score doesn't increase monotonously, because of both the influence of the subjects' breathing sound mixed in the pronunciations and the insufficient accuracy of the recognition process. It can be seen also that the fifth score is higher than the first one in all the experiments. This means that subjects had made progressed considerably in Vietnamese pronunciation. On the other hand, we can see that the subject could not advance his score easily for the vowels which don't exist in the Japanese language.

5. Conclusion

In this paper, we made description of our Vietnamese pronunciation training system. From the results of the experiments, we showed that the subjects' pronunciations approach to the standard pronunciations gradually by repeating a practice, while there are some room for improvement about the influence of a noise and the accuracy of the recognition process.

References

[1] Hiroshi Toda. "Sound Effect", *CMagazine* (in Japanese),Vol.8, No. 12, pp. 22-55, Dec. 1996.

[2] A. M. Noll, "Short-Time Spectrum and Cepstrum Techniques for Vocal-Pitch Detection" , J. Acoust. Soc. Amer., Vol. 36, No. 2, pp.296-302, 1964.

[3] L. R.Rabiner, R. W.Schafer. *Digital Processing of Speech Signals*, Prentice-Hall, 1978.

[4] R. W. Hamming, *Digital Filters*, Prentice-Hall, 1983.