Fluency estimation and prosodic error analysis for Japanese English using multi-resolution posteriorgram

(多重分解能のポステリオグラムを用いた日本人英語を対象とした流暢性推定と韻律誤り分析)

瀋 陽

SHEN Yang

ID Number: 37-186992

Supervisor: Prof. Nobuaki Minematsu

Department of Electrical Engineering and Information Systems,
Graduate School of Engineering,
The University of Tokyo

Master Thesis
August 2020

Declaration

I hereby declare that except where specific reference is made by the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and acknowledgements.

SHEN Yang
August 2020

Acknowledgements

I would like to dedicate this thesis to my family. Without their support, I could not come to Japan and finish my study and research. I would like to thank my supervisor Prof. Nobuaki Minematsu, Lecturer Daisuke Saito, and all lab members who have helped me in both lives and research. I would also like to thank my friends and teachers who helped me during my application for UTokyo. It is my honor to meet you all in my life.

Abstract

Computer-Assisted Language Learning (CALL) is a field aimed to help learners learn different skills of foreign language using computer technology, such as reading, writing, speaking and listening skills. In this thesis, we mainly investigate how to assess and analyze non-native Japanese English utterances using CALL. Specifically, fluency estimation and prosody error analysis is the focus in this thesis. Multi-resolution phoneme-based posteriorgram is used to estimate fluency and to analyze prosody errors for Japanese English. The original posteriorgram has a few thousand dimensions and different clustering strategies are investigated to get more compact representation of the posteriorgram that can show a better performance. For fluency estimation, a bottom-up strategy is used and it shows a better performance than human raters. For prosody error analysis, a top-down strategy is used and it shows good possibility to detect prosody errors adequately in Japanese English.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Mastering English is very important for those who want to interact with foreigners. To support people learn a new language, various types of technical aids have been examined [10, 23, 7, 13] and realized as commercial products or services [6, 18]. These days, computer-based scoring systems are available by companies providing TOEFL, TOEIC, IELTS, EIKEN, GTEC, etc. Basically, proficiency of English can be assessed from aspects including listening, reading, speaking and writing. Among these, speaking is the most important skill for learners to master when they want to interact with native speakers in person. If there are pronunciation errors in one's utterances, it will be difficult for native speakers to understand. Unintelligible utterances by English learners can be produced by several factors. For example, phoneme errors, including phoneme insertion, deletion and substitution, sometimes make it difficult for native speakers to understand. Specifically, for Japanese leaners, their language has a limited set of vowels and cannot pronounce some vowels specific to English. As a result, they may substitute English-specific vowels with Japanese vowels or even delete these vowels when they encounter them. Besides phoneme pronunciation, inadequate prosody control is another kind of pronunciation errors for English learners. In English, there are many words that are spelled the same but pronounced differently. Depending on where the stress is placed, the meaning can be totally different. If learners' prosody control is inadequate, native speakers may misinterpret their meaning. Moreover, fluency sometimes also has effect on intelligibility of learners' utterances.

There have been works [17, 20, 27] about how to detect unintelligible segments in English learners' utterances. It was demonstrated that places of unintelligible segments can be detected. However, the causes for unintelligibility is still unknown. In order to help learners correct and improve their pronunciation, it is important to know what leads to unintelligibility in utterances. Therefore, this thesis mainly investigate how to detect prosody error in learners' utterances which is a kind of pronunciation error as introduced before. For fluency estimation, there have been previous work on it[1, 38] . In [38], fluency assessment was mainly based on manual acoustic feature extraction which is not suitable for building a system evaluating fluency of learners' utterances automatically. In [1], automatic feature extraction was investigated for fluency evaluation. But the accuracy of the model was not comparable to human raters, which also make it difficult to build a reliable system for fluency evaluation. In this thesis, we designed new features closely related with fluency to improve the performance of existing models.

This thesis is organized in two parts. The first part is mainly about designing new features and building models for fluency estimation. The second part is mainly about how to automatically locate and detect prosody errors in English learners' utterances.

In Chapter 2, I will introduce the basics of deep neural network (DNN) acoustic model, especially the input features and output classes of it. Besides, prosody and features represent it will also be introduced. These features can be used in CALL systems.

In Chapter 3, I will introduce previous work on prosody error detection and fluency estimation for English learners' utterances. For fluency estimation, it is mainly about definition for English fluency [5, 26, 40], investigation on acoustic features related with fluency [38] and automatic feature extraction for fluency estimation [1]. For prosody error detection, it is mainly about how to locate and detect unintelligible segments in English leaners' utterances [17, 20, 27].

In Chapter 4, newly designed features related with fluency and linear regression model will be introduced. Features used in the model is mainly extracted from posteriorgram which is output of DNN acoustic model. Besides, two methods are investigated to cluster posteriorgram and find optimal resolution of posteriorgram.

In Chapter 5, improvement on clustering posteriorgram is introduced. Instead of fixing clustering matrix, it is incorporated into a network model and optimized.

In Chapter 6, experiment details on prosody error detection will be introduced. Multiple DNN acoustic models, which mainly differ in input features and output classes, are trained to detect prosody error. Some models have input features and output classes related with prosody and others do not.

In Chapter 7, conclusion and future development for fluency estimation and prosody error detection will be introduced.

Chapter 2

Basics of DNN acoustic model and speech prosody

This chapter gives introduction to DNN acoustic model and prosody of speech which lays the foundation for the following chapters.

2.1 DNN acoustic model

The task of speech recognition is that given observed speech features O, recognize the intended word sequence W. This can be formulated as following:

$$W^* = \arg\max_W P(W|O) \tag{2.1}$$

After applying Bayes' theorem:

$$\begin{aligned} P(W|O) &= \frac{P(O|W)P(W)}{P(O)} \\ &\propto= P(O|W)P(W) \end{aligned} \tag{2.2}$$

In Equation 2.2, $P(O|W)$ is usually computed by acoustic model and $P(W)$ is computed by language model. The acoustic model can be modeled by Hidden Markov Models (HMM) and DNN and language model can be modeled by N-gram model. In this thesis, we mainly use DNN acoustic model instead of HMM. DNN acoustic model is different from HMM. In fact, HMM approximates the distribution $p(x|y)$ which is the probability of observing a given short span of acoustic features, x, conditoned on an HMM state label y. The acoustic features represent about 25ms of speech. The HMM state label y are senones, which is clustered, context-dependent sub-phonetic state [29]. On the other hand, DNN does not explicitly model the distribution $p(x|y)$. Instead, it estimates $p(y|x)$, which is essentially a classifier of phoneme given speech features. We can use Bayes' theorem again to obtain $p(x|y)$ given the DNN output distribution $p(y|x)$.

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \tag{2.3}$$

As a result, we can replace HMM with DNN and build DNN-based speech recognizer. In the following, we will introduce input and output of DNN acoustic model.

2.1.1 Input to DNN acoustic model

In order to recognize the speech, we need to extract compact and efficient features from the speech. The most commonly used speech feature is Mel-Frequency Cepstral Coefficients (MFCC). Following are typical steps to compute MFCC features from speech.

Cut speech signal into short frames

The speech signal is a slowly time-varing signal. For stable acoustic characteristics, speech needs to be examined over a sufficiently short period of time. Short-term spectral measurements are typically carried out over 25ms windows and advanced every 10ms [9]. These windows are also referred as frames. Figure 2.1 is an illustration of frames and frame shift for speech signal. On each frame, a window function such as Hamming window is applied to smooth edges of frames and reduce the edge effect while converting speech signal to spectrum.

Compute power spectrum

The next step is to calculate the power spectrum of each frame [28]. This is motivated by the human cochlea (an organ in the ear) which vibrates at different spots depending on the frequency of the incoming sounds. Depending on the location in the cochlea that vibrates (which wobbles small hairs), different nerves fire informing the brain that certain frequencies are present. This is similarly achieved by applying short-time Fourier transform (STFT) to signal frame, and then transforming it from complex frequency domain into power frequency domain.

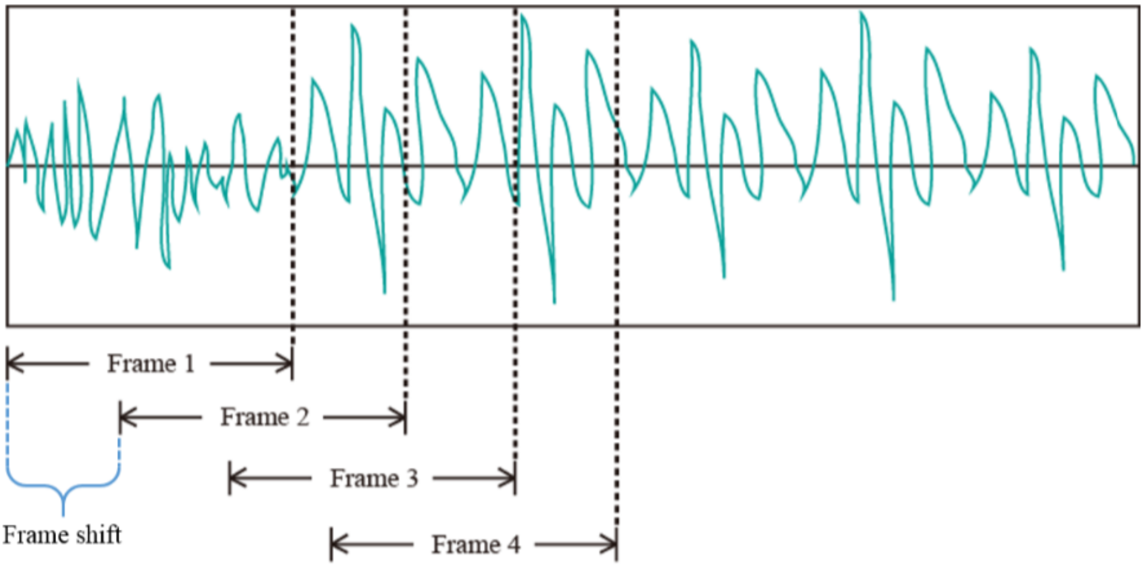


Fig. 2.1 Illustration of frames and frame shift

Apply the mel filterbank

According to [28], the power spectrum still contains a lot of information not required for Automatic Speech Recognition (ASR). In particular the cochlea can not discern the difference between two closely spaced frequencies. This effect becomes more pronounced as the frequencies increase. For this reason we take clumps of periodogram bins and sum them up to get an idea of how much energy exists in various frequency regions. This is performed by Mel filterbank: the first filter is very narrow and gives an indication of how much energy exists near 0 Hertz. As the frequencies get higher our filters get wider as we become less concerned about variations. We are only interested in roughly how much energy occurs at each spot. The following gives a formula from frequency to Mel scale:

$$M(f) = 1125 \ln(1 + \frac{f}{700}) \tag{2.4}$$

Figure 2.2 gives an illustration of mel filterbank. Once we have the filterbank energies, we take the logarithm of them. This is also motivated by human hearing: we don't hear loudness on a linear scale [28].

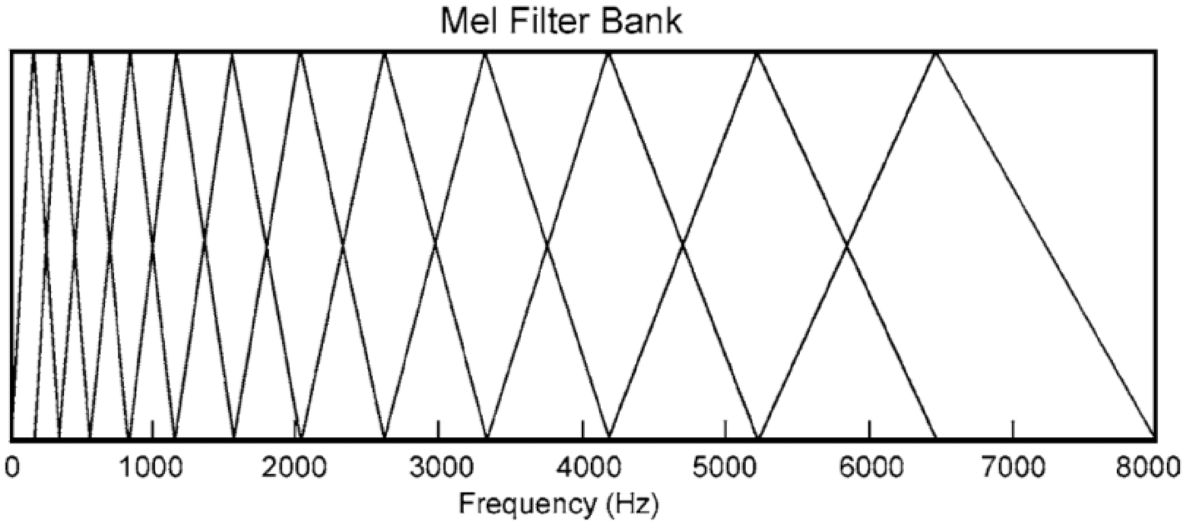


Fig. 2.2 Illustration of mel filterbank

Discrete cosine transform (DCT)

Since the vocal tract is smooth, the energy levels in adjacent bands tend to be correlated. The DCT is applied to the transformed Mel frequency coefficients and produces a set of cepstral coefficients.

After taking DCT, keep the first 12 coefficients (expect c_0) instead of all of them. This is because the higher DCT coefficients represent fast changes in the filterbank energies and it turns out that these fast changes actually degrade ASR performance, so we get a small improvement by dropping them [28]. Finally, those taken out coefficients are called MFCC.

2.1.2 Output of DNN acoustic model

As introduced in Section 2.1, DNN acoustic model estimates distribution $p(y|x)$ where x are acoustic features, namely MFCC and y is phoneme class. In fact, the output class is not a standalone phoneme. Instead, each output class is called triphone which is essentially context-dependent phoneme in the form of $a-b+c$. a,b,c are all phonemes and a,c are context phonemes for the center phoneme b. If two triphones have the same center phoneme but different context phonemes, they are treated as different output classes. As a result, there are several thousand output classes for DNN acoustic model.

In brief, DNN acoustic model takes MFCC as input and outputs a probability vector representing the posterior probabilities of a set of pre-defined phonetic classes for speech frames. And the probability vector is also called posteriorgram.

2.2 Prosody of speech

According to [19], prosody conveys various types of information. For example, prosody can emphasize words that the speaker think are important. Besides, it can also convey speakers' attitude and emotional state. Prosody is a suprasegmental information. Specifically, it is defined on segments larger than the phones [19]. Several factors influence the production of prosody in speech such as fundamental frequency, the duration of the sounds and the energy of the sounds. Following is introduction to these factors.

Fundamental frequency

Fundamental frequency (F0) is an important prosody feature. It corresponds to the frequency of vibration of the vocal folds [19]. Many algorithms have been developed in the past to compute the fundamental frequency of speech signals. Some algorithms operate in the time domain [2, 8] and some algorithms operate in the frequency domain [4].

Phone duration

The phone duration is determined from a phonetic segmentation of the speech signal. Such segmentation can be done manually using some speech visualization tools such as Praat [3] or automatically using forced speech-text alignment procedures.

Energy of the sounds

The raw local energy of speech signal is easy to compute and is part of many acoustic features. However getting the phone energy implies some choices: should it be an average value over the whole phone segment, or and estimation in the middle of the phone segment. What is the impact when applied to non stationary sounds such as plosives and diphthongs. Errors on the phone boundaries will also affect the estimation [3].

2.2.1 Prosodic features in CALL

In CALL, pronunciation of L2 language is always the focus and the main goal is to automatically detect mispronunciations [10, 23]. This can be achieved with the help of ASR technology. One common approach is computing goodness of pronunciation (GOP) scores [45] which amounts to computing log likelihood ratio between a forced alignment corresponding to the expected pronunciation and another alignment over an unconstrained phonetic loop. Another typical approach models frequent mispronunciation patterns for a specific group of leaners and detects mispronunciations using the built model as language model [36].

When it comes to correct pronunciation, not only correct pronunciation of phonemes should be considered, but also correct lexical stress should be focused. For English, lexical stress is very important for understanding the meaning of words. It should be noted that some English words have the same phoneme sequence but different lexical stress which lead to completely different meaning. In [11], prosodic features are used to build a system to detect lexical stress in English words. It was shown that prosodic features can be used in detecting inadequate lexical stress for English words.

Chapter 3

Previous work

This chapter is an overview about previous work on fluency estimation and prosody error detection.

3.1 Previous work on fluency estimation

3.1.1 Corpus [38]

Our work is mainly based on [38]. In [38], the task description is as following: 90 native Japanese students, who learned English only in public education for six years, three in middle school and three in high school, as well as 10 native speakers participated in data collection. The task was picture description, where three independent photos were presented with three keywords to the participants, as shown in Figure 3.1. They were asked to describe the pictures orally using the keywords. Their utterances were recorded with 16 bits and 44.1 kHz as sampling frequency. From each of the three recordings, the first 10-sec segment was extracted and they were connected to form an about 30-sec long utterance.

10 native raters, who did not participate in data collection, were recruited for manual assessment of the 100 utterances. They are native speakers, but not teachers or researchers of language education. The raters were asked to listen to each utterance and assign a score with respect to fluency of the utterance. The score varied from 1 (=least fluent) to 9 (=extremely fluent). Before rating, the definition of fluency in [38] was explained to the raters, who showed a high consensus on that definition.

Each rater assigned 100 scores to the 100 utterances. In this thesis, correlations are calculated between every pair of the raters. The minimum, average, and maximum of one-to-one correlations are 0.677, 0.786, and 0.897. Correlations are also calculated from each rater to the averaged scores of the other nine raters. The minimum, average, and maximum

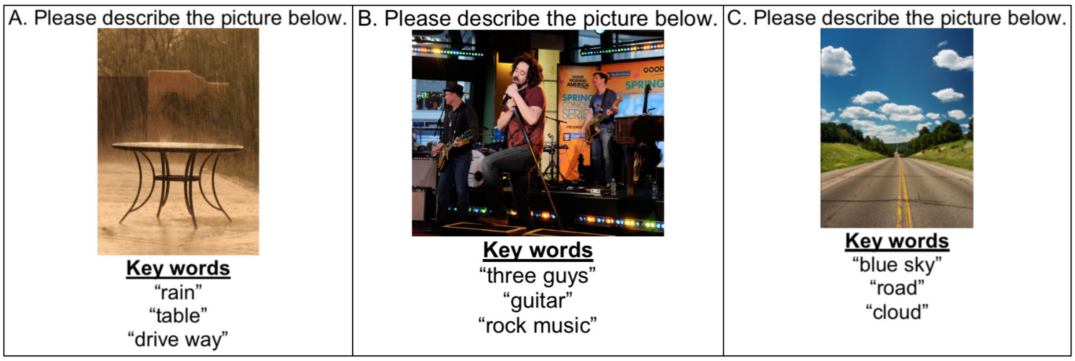

A. Please describe the picture below.
Key words
"rain"
"table"
"drive way"
B. Please describe the picture below.
Key words
"three guys"
"guitar"
"rock music"
C. Please describe the picture below.
Key words
"blue sky"
"road"
"cloud"

Fig. 3.1 Three photos for data collection.

of one-to-others correlations were 0.798, 0.873, and 0.910. These are used later as reference when assessing the performance of automatic prediction of fluency.

3.1.2 Manual extraction of features related to fluency [38]

The collected data were transcribed by graduate students who majored in foreign language education and the following features were manually extracted with Praat [3].

1. The number of breakdowns, (un)filled pauses, per unit time
2. Speaking rate, the number of syllables per unit time
3. The number of repairs per unit time

The number of breakdowns were counted separately for two cases, within and between clauses. Repairs can also be divided into repetitions and self-corrections. These five features were selected as primary features through an extensive review of the related literature, and feature correlations were calculated.

We regard the above features as strongly related to quantity of phonation, especially quantity of meaningful phonation, per unit time. In this thesis, at first, we applied Elastic Net regression to the above manual features to predict the fluency scores, averaged over the 10 raters. 5-fold cross-validation showed that the predicted scores had a correlation of 0.788 to the human scores, which is comparable to the average of one-to-one correlations. This value can also be used as reference when assessing the performance of automatic prediction of fluency.

3.1.3 Automatic feature extraction related to fluency [1]

Manual feature extraction combined with linear regression model [38] achieved correlation of 0.788 which is lower than average one-to-others correlation. Besides, it is impossible to apply this model to practical use since it requires manual feature extraction. In order to come around this problem, automatic feature extraction was investigated in [1].

In [1], three types of features for automatic prediction were introduced.

1. those derived only from speech acoustics with signal processing techniques
2. those derived from ASR results of the utterances
3. those derived from posteriorgrams of utterances

Since the utterances in the corpus [38] are with unignorable noises, two versions of WSJ-KALDI-based English speech recognizers [35] were trained, one with the WSJ corpus only and the other with WSJ and its noisy versions, where three levels of noises (SNR=10, 30, 40[dB]) were added and all the clean and noisy utterances were used together to train a noise-robust speech recognizer.

Features derived with signal processing

Following a previous study [12], envelope-based syllable detection was used, which is provided as Praat script [3]. Then, speaking rate was calculated as

$$\text{speaking rate} = \frac{\text{\#syllables}}{\text{total duration of phonation}}$$

The denominator is defined as the utterance length minus its entire duration of pauses. Speaking rate dose not tell anything on how many silent frames are found in the utterance. Besides speaking rate, a similar but different feature was introduced which is phonation ratio

$$\text{phonation ratio} = \frac{\text{total duration of phonation}}{\text{utterannce length}}$$

Figure 3.2 and Figure 3.3 show how to detect syllables and duration of phonation with Praat.

Features derived from ASR results

In [1], two versions of WSJ-KALDI-based speech recognizers were tested, i.e. clean model and noise-robust model on all the 100 utterances. After recognition, total number of words and size of vocabulary in ASR results were used as feature for fluency estimation.

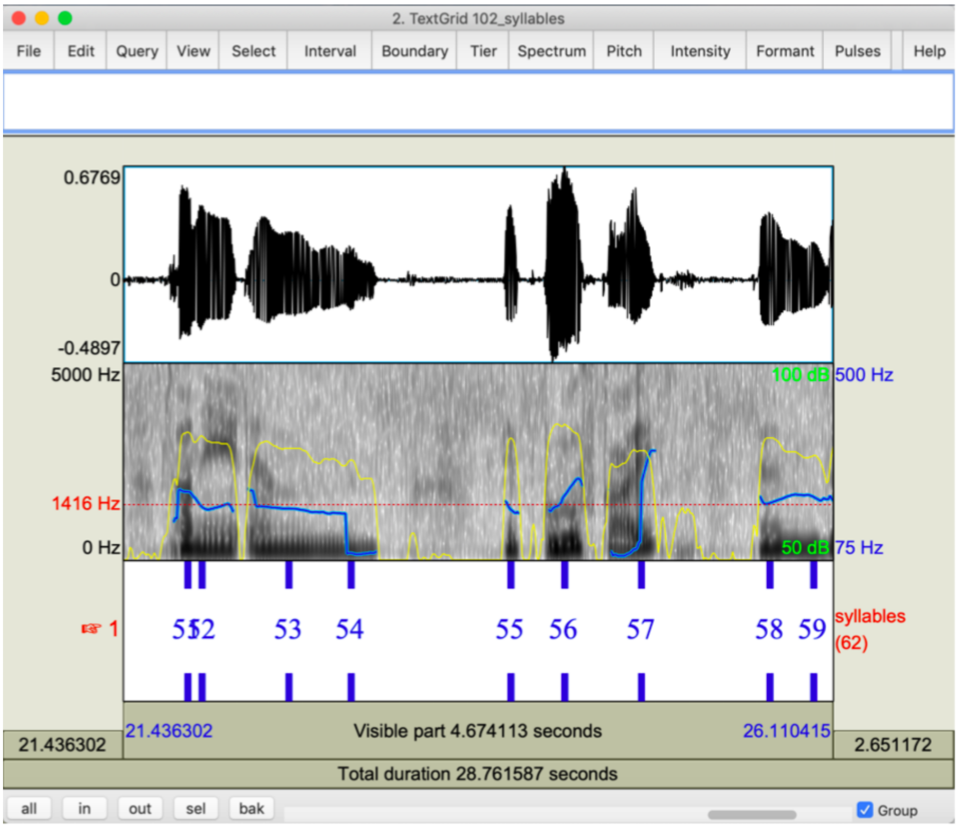


Fig. 3.2 Illustration for how to detect syllables with Praat

Features derived from posteriorgrams

When native posteriorgrams are visualized with the number of classes being set to the number of phonemes, a posterior vector at each time often looks like a one-hot vector. This means that the phoneme intended has a probability close to 1.0 and the others have almost 0.0. In [1], from a given posteriorgram, the maximum posterior probability for each time was computed and it is averaged over time. The higher the average is, the more distinct pronunciation the utterance is made with. However, this feature is not sufficient for evaluating pronunciation. If a learner keeps pronouncing the same vowel $/AA/$, which is easy to pronounce, then posterior vectors for the utterance will all look like a one-hot vector and the maximum posterior probability for each time will be equal to 1. In this case, even though the averaged maximum posterior probability is large, we cannot say that the learner has good pronunciation.

With these features and Elastic Net regression model, it was shown that the correlation achieved 0.87 which is comparable to the average of one-to-others correlation. But the correlation is lower than the maximum of one-to-others correlation. In order to improve the performance of the model, we designed new features and tested with new models.

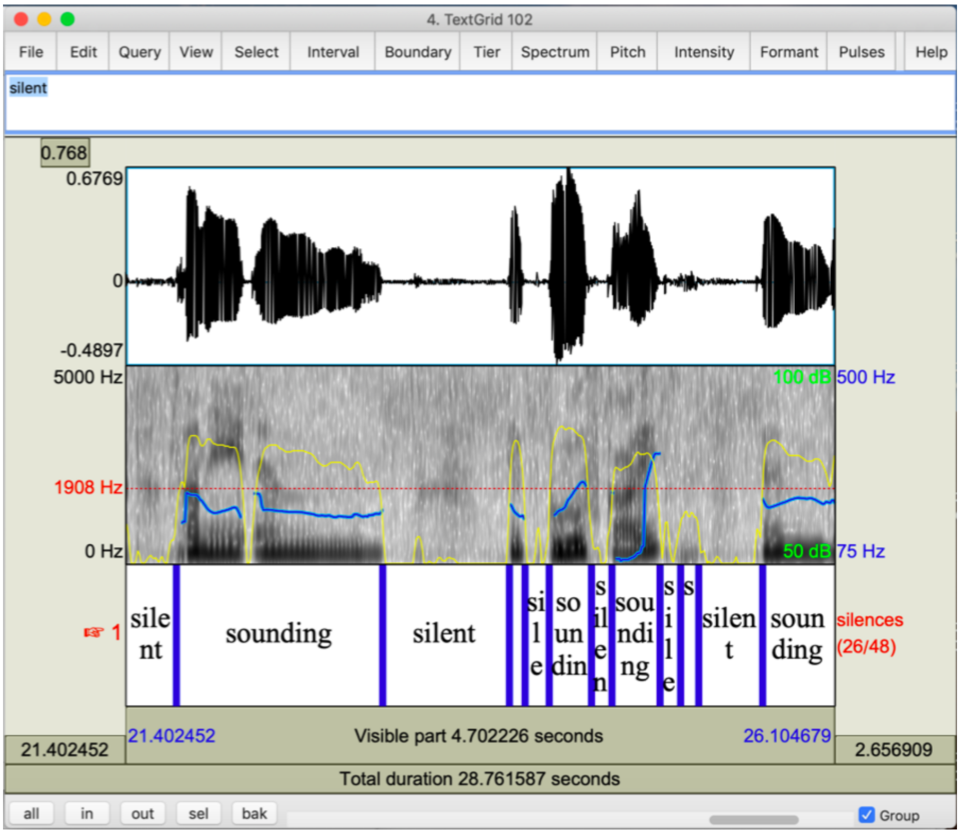


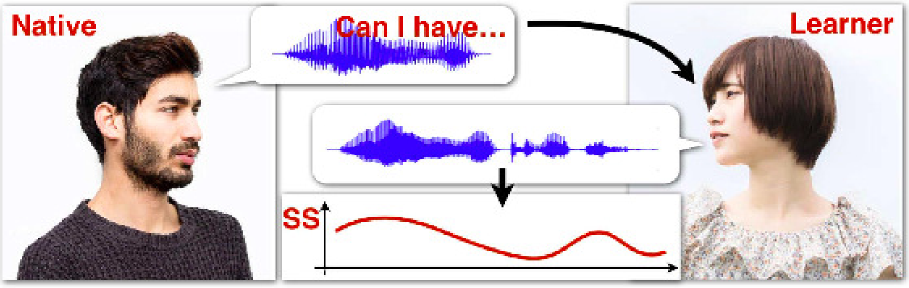
Fig. 3.3 llustration for how to detect duration of phonation with Praat

3.2 Previous work on prosody error detection

Shadowing is a technique used in L2 language learning where L2 learners are asked to repeat a given native utterance while hearing it [14, 16, 31]. Figure 3.4 shows the process of conventional shadowing, where learners shadow natives.

In previous works [17, 20, 27], reverse shadowing was proposed to manually detect incomprehensible segments in L2 speech. In reverse shadowing, native listeners are not asked to imitate L2 pronunciations but to repeat in their own native pronunciation what they heard. The illustration for reverse shadowing is shown in Figure 3.5, where natives shadow learners.

In [27], in addition to reverse shadowing, reading was also asked for native listeners to do. L2 utterances presented for reverse shadowing were often obtained as L2 readings. After reverse shadowing, the text which had been read by learners was presented to shadowers, who were asked to read that text aloud. Shadowing can be viewed as least prepared speech and reading as most prepared speech. If shadowing is found to be close to reading, the comprehensibility level for the L2 utterance is high, because good shadowing indicates easiness and quickness of understanding.



SS means smoothness of shadowing

Fig. 3.4 Conventional shadowing, where learners shadow natives.

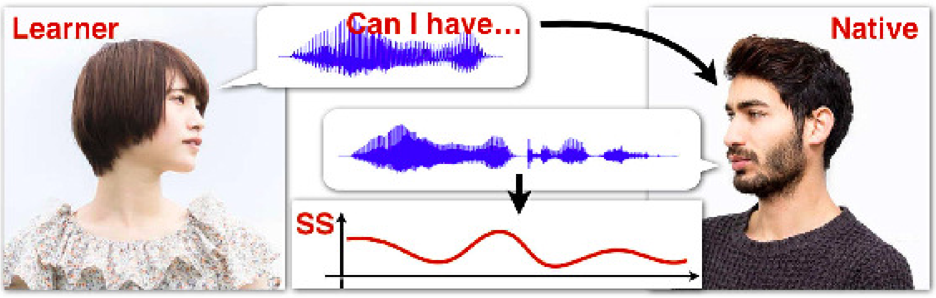


Fig. 3.5 Reverse shadowing, where natives shadow learners.

In order to detect unintelligible segments in learners' utterances, Dynamic Time Warping (DTW) between native speaker's shadowing and reading was applied [27]. Specifically, DTW could present optimal path for shadowing and reading, on which a sequence of local distance can be viewed as a sequence of comprehensibility. If the distance between corresponding segments in shadowing and reading is large, it can be deduced that the native speaker did not understand these segments and the learner pronounced inadequately for these segments.

As introduced before, several factors can lead to unintelligible segments in learners' utterances such as inadequate phoneme pronunciation and prosody control. In order to help learners improve their speaking skills, only detecting unintelligible segments is not enough. We have to identify causes for these unintelligible segments to help learners correct errors. Based on [27], we proposed methods to investigate whether prosody errors, especially stress assignment errors, exist in Japanese English speech.

Chapter 4

Designing new features for fluency estimation

In this chapter, based on [1], we proposed new features extracted from posteriorgrams which is output of DNN acoustic models. First, we introduce the basic idea behind these newly designed features. The relation between these features and fluency, or more precisely, pronunciation quality is intuitive. Then we proposed two methods to optimize the resolution of posteriorgrams. The original resolution (dimension) of posteriorgrams vary from 2 to 2,000. In order to reduce the dimension, top-down approach or bottom-up approach can be used.

4.1 Phonotactic modeling of languages [30]

A classical approach of language identification is applied to quantify native-likeness. In the classical approach, a continuous phoneme recognizer of a specific language, e.g. English, was applied to a given utterance of any language. Then, the utterance was represented in a forced way as a sequence of English phonemes. Languages of interest were modeled separately as phoneme N-gram using the forced English phonemes. If we consider a special case of $N=1$, the model becomes phoneme distribution. After converting the 30-sec long utterance of each participant into its posteriorgram, we can calculate the averaged posterior probability of the n classes ($2 \leq n \leq 2{,}000$), which directly corresponds to distribution of the n classes.

4.2 Elastic Net regression [47]

In this thesis, for feature selection and for prediction, Elastic Net regression is used. Mathematically speaking, Elastic Net regression is a combination of Ridge regression [15] and

Lasso regression [42], i.e. a combined use of L1 norm and L2 norm as regularization terms for weights. Value normalization is also done for each feature. Because of these, weight coefficients attached to features of less predictability become zero and this is why the function of Elastic Net is said to be prediction based on feature selection. Further, if we take a set of weights as weight vector, it tends to be sparse. This is preferable to many machine learning algorithms, when the resulting weights are applied to those algorithms as initial weight vector. In this thesis, we use scikit-learn, which is a machine learning library in Python, to build the Elastic Net regression model.

4.3 New features derived from posteriorgrams

If the task adopted in [38] was reading-aloud, we can use the sentences intended by participants and the GOP [45] scores can be calculated. The task was, however, picture description, which gave us only spontaneous speech and its correct transcript is generally unavailable. From the posteriorgram of each utterance, there have been one kind of feature extracted [1], which is average of maximum posterior probabilities. As introduced in Section 3.1.3, this feature is not enough for evaluating pronunciation. Besides, the feature correlation between this feature with fluency score is 0.57 which is not very desirable. In this thesis, besides this feature, another two kinds of feature are proposed and extracted from posteriorgram.

4.3.1 Averaged posterior distribution as fine phoneme distribution

As discussed in Section 4.1, the averaged posterior vector can be viewed as the distribution of phonemes. Since the utterance from each participant is so long as 30 sec, a variety enough of phonemes are supposed to exist and the averaged posterior vector can characterize native-likeness of each participant. The average posterior vector is directly used for prediction.

4.3.2 Posterior gap between a participant and native speakers

For each participant, we calculate his/her averaged posterior vector. Since we have 10 native speakers in the participants, we calculate distance from a participant to each native speaker, 10 gaps in total. The Bhattacharyya distance [24] is used as metric. These gaps quantify native-likeness of each participant more directly and the averaged gap is used for prediction of fluency. Figure 4.1 visualizes the averaged posterior and the posterior gap. The former characterizes quality of pronunciation, location in the feature space, and the latter characterizes relative distances to the 10 native speakers.

4.3.3 Clustering of phonemic classes using posteriors [22]

In this thesis, newly proposed features are all extracted from posteriorgrams. For this end, all the utterances are converted to poseriorgrams. Posteriorgrams generally use a set of phoneme classes, the number of which is several thousands. They can be viewed as finely-defined context-dependent phonemes, but they may be too fine to be used for assessment. In this study, we examine a smaller number of classes introduced by top-down clustering and bottom-up clustering with Ward's method [43].

Top-down clustering

For top-down clustering, the basic idea is as following. As introduced in Section 2.1, triphone is used as output class of DNN acoustic model. A triphone is simply a group of 3 phonemes in the form $a-b+c$, where a,b,c are all English phonemes. For convenience, we refer b as center phoneme while a and c as context phonemes. For two triphones with the same center phoneme but with different context phonemes, they are treated as different output classes of DNN acoustic model. For example, triphones $t-ih+n$ and $t-ih+ng$ are different output classes. As a result, we can cluster output classes with the same center phoneme. Since this kind of clustering is based on the structure of triphone, it can be treated as a top-down clustering approach. After top-down clustering, the number of classes is reduced from 2,000 to 53, which means there are 53 different center phonemes.

Bottom-up clustering

For bottom-up clustering with Ward's method [43], it requires the distance matrix between any two classes (dimensions). The Bhattacharyya distance [24] between two classes a and b is rewritten using class posteriors through Bayes' theorem [22] as

$$\begin{aligned}
BD(a,b) &= -\ln \int \sqrt{p(\boldsymbol{x}|a)p(\boldsymbol{x}|b)}d\boldsymbol{x} \\
&= -\ln \int \sqrt{\frac{p(a|\boldsymbol{x})p(\boldsymbol{x})}{p(a)} \frac{p(b|\boldsymbol{x})p(\boldsymbol{x})}{p(b)}} d\boldsymbol{x} \\
&= -\ln \int p(\boldsymbol{x})\sqrt{p(a|\boldsymbol{x})p(b|\boldsymbol{x})}d\boldsymbol{x} + \frac{1}{2}\ln p(a) + \frac{1}{2}\ln p(b).
\end{aligned} \tag{4.1}$$

$p(\mathbf{x})$ is a prior probability for $\mathbf{x}$, which can be calculated using the universal background model. $p(a|\mathbf{x})$ and $p(b|\mathbf{x})$ are class posteriors, which are outputs from DNN-based acoustic models of Automatic Speech Recognition (ASR) to input vector $\mathbf{x}$. $p(a)$ and $p(b)$ are prior

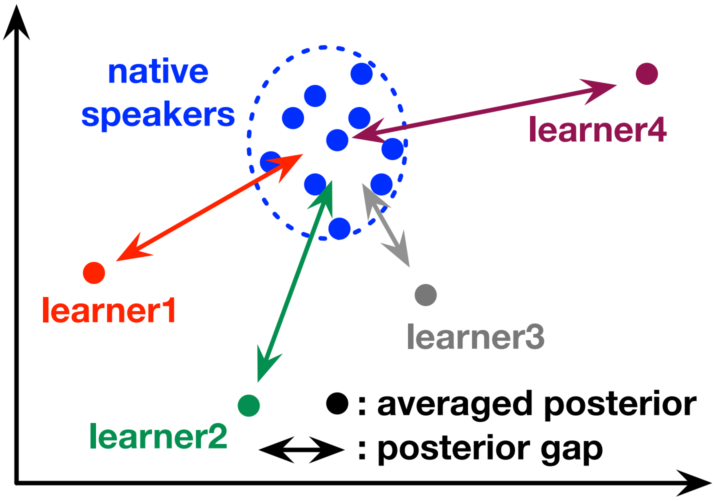


Fig. 4.1 Averaged posterior and posterior gap

probabilities for the two classes, which can be obtained as normalized frequency from the training corpus. Once DNN models are trained for ASR, any speech sample can be converted to its posteriorgram, which is a sequence of vectors comprised of probabilities of several thousand classes. With the above formulation, however, a given posteriorgram can be reduced into a smaller dimension of classes. In the current study, the baseline number of classes is 2,000 and n-class posteriorgrams can be calculated for any n ($2 \leq n \leq 2{,}000$).

Once the distance matrix is calculated from the original posteriorgram with n=2,000, dimension reduction is possible by clustering and the m-dimensional ($2 \leq m < 2{,}000$) posterior vector can be calculated from its original vector. Since each element of the m-dimensional vector is obtained as summation of one or more element(s) of the original vector, the m-dimensional posterior vector is formulated as

$$v_m = Av_n,$$

v_n and v_m are the original vector and the m-dimensional vector, respectively. A=$\{a_{ij}\}$ is an $m \times n$ ($m<n$) binary matrix, where the i-th row vector $\{a_{i*}\}$ shows which elements in v_n are selected and summed to calculate the i-th element in v_m, and column vector $\{a_{*j}\}$ is a one-hot vector, indicating to which element in v_m, the j-th element in v_n is added. If we can view v_m as result of hard clustering from v_n, soft clustering from v_n will be realized with another matrix B, which is not a binary matrix but is expected to improve the prediction performance. Functionally speaking, linear transform between the k-th layer and the $k+1$-th layer in DNN works in a similar way to matrix B and learning of matrix B will be discussed in the Section 5.

4.4 Experiment

For features we have newly designed and those from previous work [1, 38], they can be classified into two classes. First is related with quantity of phonation which is as following.

1) speaking rate

2) phonation ratio

3) size of vocabulary

Second is related with quality of pronunciation which is as following.

a) average of maximum posteriors

b) averaged distribution of posteriors

c) averaged posterior gap to natives

d) correct recognition rate

As introduced before, correct transcript is not available in the task. As a result, correct recognition rate is also unavailable but it is tentatively considered in the experiment. In order to compare the performance of the two sets of features, they are taken as input to Elastic Net regression model separately. For specific experiment setting, feature standardization is applied to features as preprocessing and then the model is trained with 5-fold cross validation.

4.4.1 Prediction with quantity-of-phonation features

Table 4.1 describes results of Elastic Net regression with quantity features for fluency prediction, where correlations between the averaged fluency scores over the 10 native raters and the machine scores are calculated. In the table, clean and noise mean the two types of ASR models, and the three values assigned to each kind of feature is the weight coefficients calculated for that feature. Clearly shown, phonation ratio and speaking rate are very effective for prediction. The performance is higher than the average of one-to-one inter-rater correlations but much lower than the average of one-to-others correlations.

4.4.2 Prediction with quality-of-pronunciation features

Since we can obtain posteriorgrams and cluster them with two different methods, that is top-down clustering and bottom-up clustering, we will compare the performance of the two methods.

Table 4.1 Prediction of fluency with quantity features

ASR	DNN	1)	2)	3)	4)	corr.
w/o	—	0.768	1.333	—	—	0.819
with	clean	0.744	1.245	0.000	0.182	0.821
with	noise	0.765	1.281	0.000	0.097	0.817

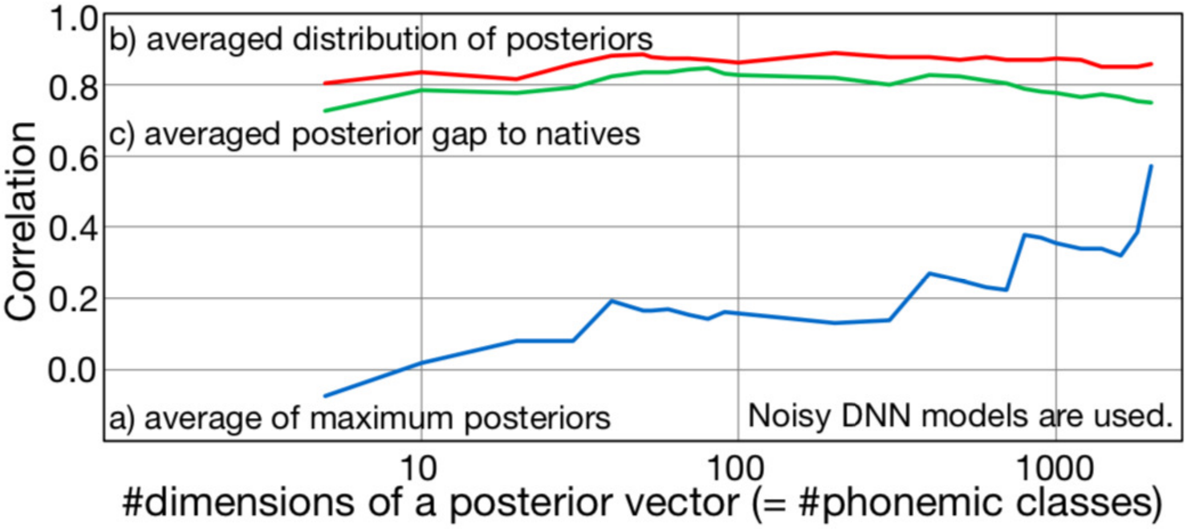


Fig. 4.2 Correlations as a function of the size of posteriors

Prediction with bottom-up clustering

Figure 4.2 shows correlations as functions of the dimension n of posterior probabilities calculated with noisy DNN models. For a) and c), feature correlations are plotted while, for b), model correlations (prediction correlations) are shown with Elastic Net regression. Correlations with b) and c) are maximized around n=50, while those with a) seem to be higher with larger n, but still lower than those with b) and c). From these results, we select 50 as n and use it for testing all the quality features. Table 4.2 describes results of Elastic Net regression with the quality features for fluency prediction. As b) is a multivariate feature, its weight means the largest weight among the n dimensions. Clearly shown, c) and b) are very effective for prediction. It is very surprising to us that the correlation with the quality features only even without ASR overcomes the average of one-to-others correlations (0.873), and is comparable to the maximum (0.910). This claims that the trained model to comparable to the most stable and reliable human rater.

Table 4.2 Prediction of fluency with quality features (bottom-up clustering)

ASR	DNN	a)	b)	c)	d)	corr.
w/o	clean	0.124	0.365	-0.624	—	0.903
w/o	noise	0.233	0.272	-0.753	—	**0.917**
with	clean	0.000	0.254	-0.491	0.628	0.922
with	noise	0.045	0.214	-0.549	0.537	0.921

Table 4.3 Prediction of fluency with quality features (top-down clustering)

ASR	DNN	a)	b)	c)	d)	corr.
w/o	clean	0.256	0.245	-0.541	—	0.850
w/o	noise	0.351	0.222	-0.646	—	**0.872**
with	clean	0.009	0.196	-0.402	0.870	0.906
with	noise	0.015	0.191	-0.545	0.808	0.905

Prediction with top-down clustering

There are 53 different center phonemes, we can reduce the dimension of posteriorgrams from $2,000$ to 53. Then we extract features from clustered posteriorgrams and apply Elastic Net regression. Table 4.3 shows the result of Elastic Net regression with quality features. From the table, we can see that the correlation is lower than that obtained in the case of bottom-up clustering. Without d) correct recognition rate, the gap is even bigger. This shows that bottom-up clustering is more effective than top-down clustering.

4.4.3 Prediction with all the features

Table 4.4 describes results of Elastic Net regression with all the features obtained with bottom-up clustering. Only the top four features in the case of noisy DNN but without ASR are shown also for other cases with or without d) correct recognition rates. In the table, the top four features are c) averaged posterior gap to natives, 1) speaking rate, a) average of maximum posteriors, and 2) phonation ratio. i.e. two quality features and two quantity features. In the table, very high usability of the quality features is shown again and, even without ASR, the trained model gives a higher correlation of 0.925 than the maximum of one-to-others correlations (0.910).

Table 4.4 Prediction of fluency with all the features

ASR	DNN	c)	1)	a)	2)	d)	corr.
w/o	clean	-0.589	0.224	0.112	0.364	—	0.906
w/o	noise	-0.748	0.264	0.255	0.231	—	**0.925**
with	clean	-0.580	0.194	0.131	0.334	—	0.906
with	noise	-0.715	0.242	0.233	0.200	—	0.923
with	clean	-0.476	0.192	0.000	0.311	0.602	0.923
with	noise	-0.543	0.276	0.033	0.239	0.561	0.928

4.5 Discussion

What is *so-called fluency*? It is often explained as degree of smoothness and fluidity in utterances [38]. In this section, we tried to predict subjective scores of fluency only with acoustic facts, calculated with speech technologies. What we found is that the fluency scores can be much more highly predicted with quality features than with quantity features. This result implies that 1) judgments of the 10 native raters were rather biased to the quality of pronunciation, which is logically independent of smoothness and fluidity in utterances, or 2) quantity features and quality features are highly correlated and the latter were extracted with higher accuracy. Further, as explained in Section 4.2 and Section 4.3, the trained models in this paper can be network-based sequential prediction of perceived fluency, where the hard clustering binary matrix A will be used as initial weight and optimized as soft clustering matrix B.

Besides, it is shown that bottom-up clustering can produce more effective features for fluency prediction when compared with top-down clustering. In oder to get insight into the effectiveness of bottom-up clustering, we decide to visualize the bottom-up clustering. Since clustered dimensions are 50, we choose to visualize only part of the clustering. Figure 4.3 shows which triphones are clustered into the one of the 50 target dimensions. In [41, 46, 21], common pronunciation error patterns for Japanese speaking English is introduced which is as following.

1. /AO/ → /OW/
2. if /AX/ is followed by /UH/: /AX/ → /OW/, else: /AX/ → /AA/
3. /AE/ → /AA/
4. /AH/ → /AA/

5. /ER/ → /AA/
6. /IH/ → _
7. /UH/ → _
8. /R/ → /L/
9. /L/ → /R/
10. if /HH/ is followed by /UW/: /HH/ → /F/, else if /HH/ is followed by /IY/: /HH/ → /SH/
11. if /F/ is followed by /AO/: /F/ → /HH/
12. /TH/ → /S/ / /AA/
13. /DH/ → /Z/ / /ZH/
14. /V/ → /B/
15. /N/ → /M/ / /NG/ / _
16. if /T/ is followed by /IH/ / /IY/: /T/ → /CH/, else if /T/ is followed by /UH/ / /UW/: /T/ → /TS/
17. if /D/ is followed by /IH/: /D/ → /DH/, else if /D/ is followed by /IY/: /D/ → /CH/, else if /D/ is followed by /UH/ / /UW/: /D/ → /DH/ / /Z/
18. if /S/ is followed by /IH/ / /IY/: /S/ → /SH/
19. if /Z/ is followed by /IH/ / /IY/: /Z/ → /ZH/
20. /JH/ → /JH IH/
21. /CH/ → /CH IH/
22. /D/ → /D OW/
23. /T/ → /T OW/
24. /NG/ → /NG UW/

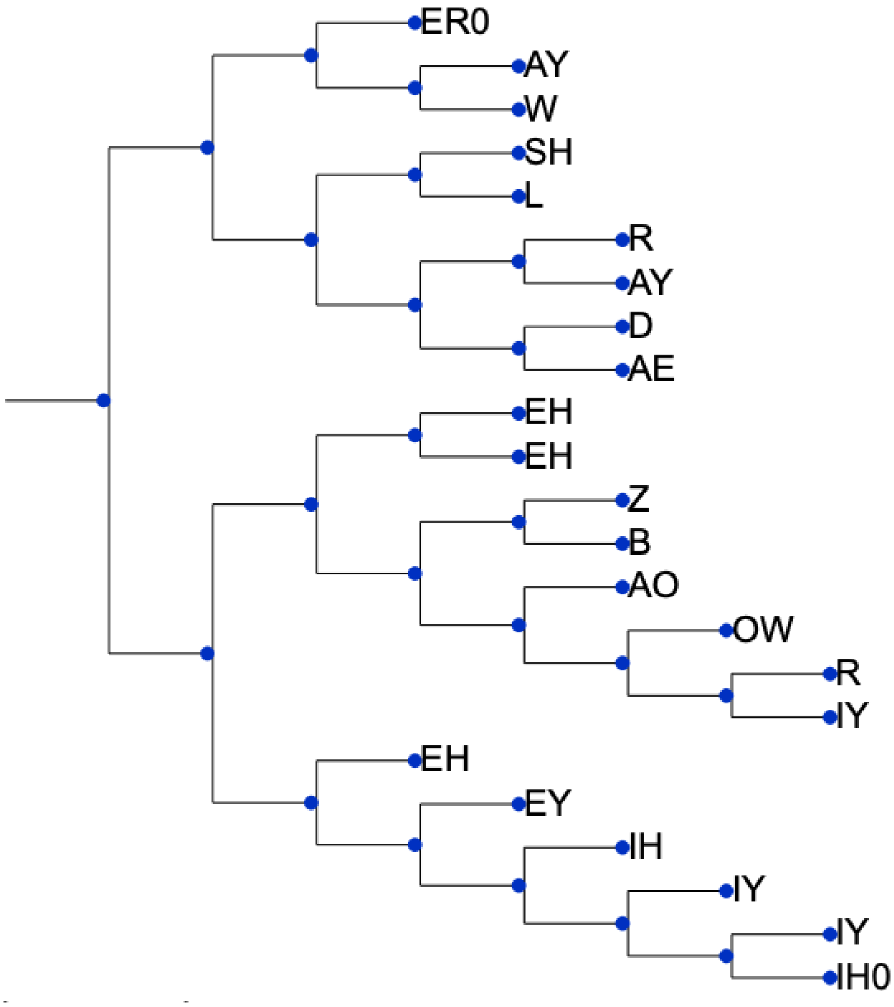


Fig. 4.3 Visualization of bottom-up clustering

From the figure and these patterns, only part of the patterns above can be seen. Besides, vowels and consonants are clustered together. For example, vowels like /ER0/, /AE/ and /AY/ are clustered with consonants like /Z/ and /L/. Pronunciation of these vowels and consonants are very distinct. Typically, Japanese do not confuse these vowels with these consonants. On the other hand, only center phoneme of each triphone is shown in this figure. As a result, It is possible this kind of clustering is influenced by context phonemes of triphones. If context phonemes surrounding consonants are vowels, they may pronounce these consonants as surrounding vowels or vice versa.

Chapter 5

Network-based optimization of clustering matrix

In this chapter, a simple network model is used to learn soft clustering matrix B and estimate fluency score. As discussed in Section 4.3, the clustering matrix A is binary matrix whose shape is $50 * 2,000$. When applying the clustering matrix A to posterior vector to reduce dimensions from $2,000$ to 50, it is essentially a kind of hard clustering. Apart from hard clustering, there is also soft clustering. In this case, the summation of each column for matrix A will be 1 and each element have to be $0 \leq A_{i,j} \leq 1$. In fact, hard clustering and soft clustering are suitable for different situations [25]. We have proved that hard clustering is suitable for our task in previous section. In order to investigate the effectiveness of soft clustering for the problem, we can incorporate the clustering matrix into model parameters and optimize it. As a result, we propose to use a network model for fluency estimation.

5.1 The structure of the Network model

Figure 5.1 and Figure 5.2 show the structure of the network layer and notation for it. One input to the layer is the posterior vector at current timestamp which is going to multiply soft clustering matrix B to reduce dimensions. Besides, clustered posterior vector and maximum of clustered posterior vector summed until previous timestamp are also input to the layer. With a sequence of posterior vectors input to the network layer, clustered posterior vector and maximum of clustered posterior vector are summed over timestamps. By using this structure, it is possible to incorporate the soft clustering matrix B into trainable parameters of the model.

The other part of the model is to compute pronunciation gap to 10 native speakers. Figure 5.3 and Figure 5.4 show the structure and notation for this part of model. As you can see, posterior

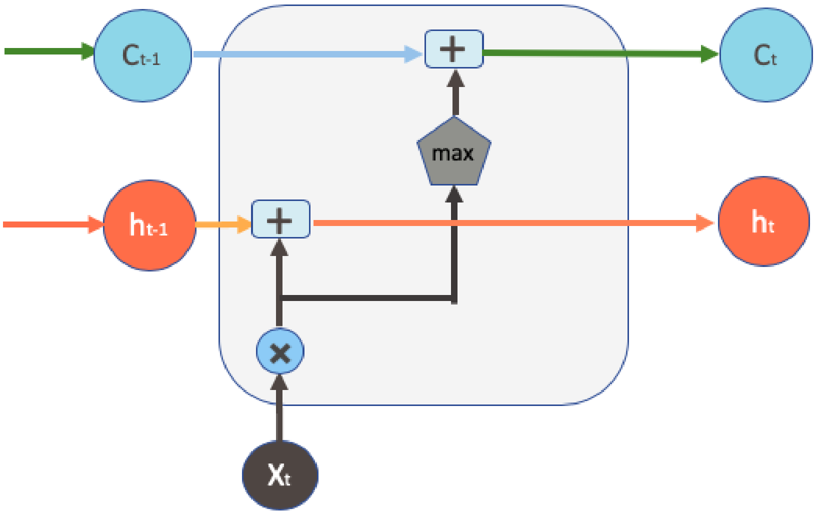


Fig. 5.1 The structure of the network layer

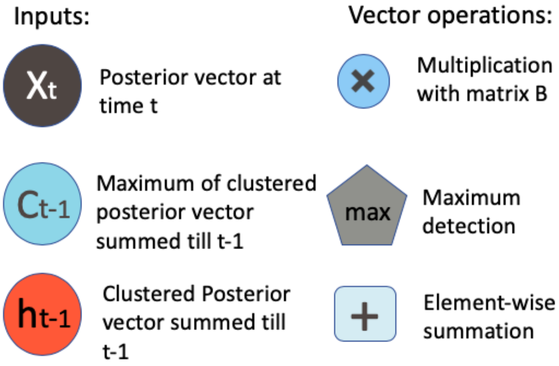

Inputs:
Xt
Posterior vector at time t
Ct-1
Maximum of clustered posterior vector summed till t-1
ht-1
Clustered Posterior vector summed till t-1
Vector operations:
×
Multiplication with matrix B
max
Maximum detection
+
Element-wise summation

Fig. 5.2 Notation for the network layer

vectors of 10 native speakers averaged over timestamps are clustered with clustering matrix B. Then Bhattacharyya distances [24] between clustered posterior vector h_T and those from native speakers are computed. These distances are summed over. The function of this part is equivalent to posterior gap feature extraction in Section 4.

The overall structure of the network model and notation for it are shown in Figure 5.5 and Figure 5.6. As seen in the figure, after getting clustered posterior vector and maximum of clustered posterior vector summed over timestamps, we average them over the timestamps and take them as input to the other part of the model. Besides, speaking rate and phonation ratio, which is introduced in Section 3, for each utterance are also input to the output layer. At last, we just use the output layer of the model to predict fluency score.

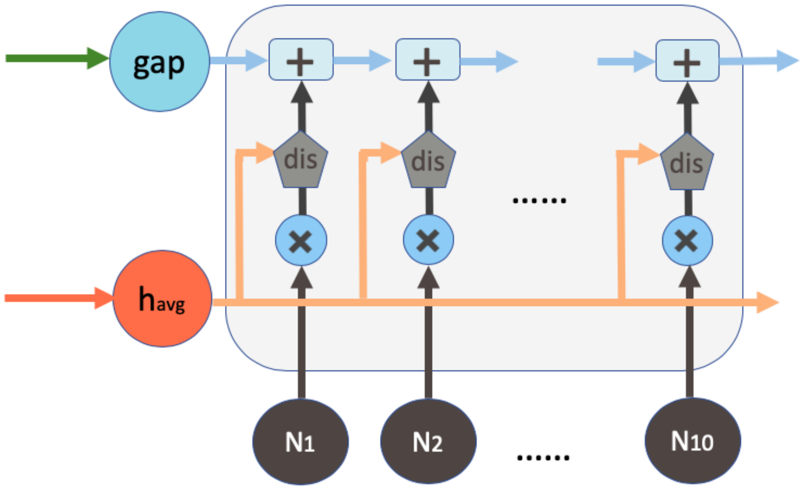


Fig. 5.3 The structure of the other part of the model

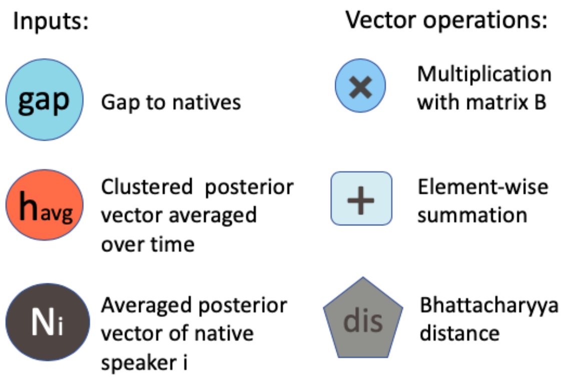

Inputs:
Vector operations:
gap
Gap to natives
Multiplication with matrix B
havg
Clustered posterior vector averaged over time
Element-wise summation
Ni
Averaged posterior vector of native speaker i
dis
Bhattacharyya distance

Fig. 5.4 Notation for the other part of the model

5.2 Experiment

The following is introduction to experiment. As seen in Section 4, there are four kinds of DNN acoustic models used to recognize utterances. And the DNN acoustic model trained with noisy data achieved the best performance when taking input from clean corpus without any noise subtraction processing. As a result, we decide to use noisy DNN acoustic model to compute posteriorgrams for utterances from the corpus. After getting posteriorgrams, toolkit Pytorch is used to build the network model. Hard clustering matrix A and coefficients of Elastic Net regression model obtained in Section 4 are used as initial weight for parameters in the built model. Then 5-fold cross validation is used to train and test the built model. Besides, the epoch number for each set of training data is 200. The experiment result is shown in Table 5.1.

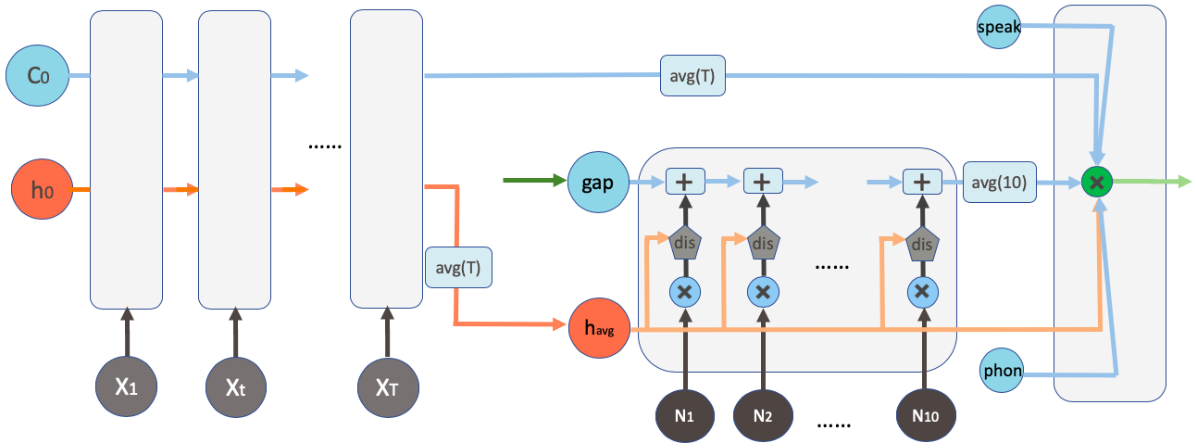


Fig. 5.5 The structure of the overall model

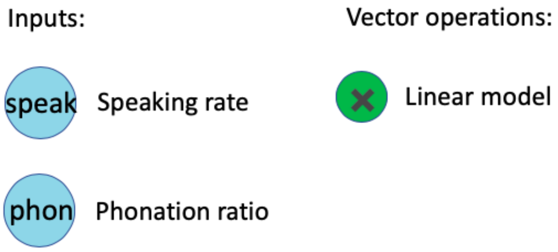


Fig. 5.6 Notation for the overall model

From the table, we can see that the correlation is 0.905. Compared with the correlation, 0.917, which is shown in Table 4.2, it does not increase. For the network model used in this section, one difference with Elastic Net regression model is that standardization is not applied to training data as preprocessing. For Elastic Net regression model, the input consists of 5 different features, each of which is on different scale. As a result, it is necessary to standardize these features before input them into the model. For the network model, the input is pure posterior vectors which do not require standardization. For output layer of the network model, we can get the same set of features used in Elastic Net regression model to predict fluency score. However, we cannot standardize these features in the network model since we cannot obtain them before the training. It has been shown that in machine learning, feature scaling and normalization have impacts on model performance [37, 39].

In order to combine soft clustering with standardization, we decide to use clustering matrix B optimized with the network model again to extract features from posteriorgrams but this time the matrix B is fixed. Then we apply standardization to obtained features and use Elastic Net regression model for fluency estimation. Specific experiment setting is the same as the original

Table 5.1 Prediction of fluency with the network model

Model	R^2	corr.
Network model	0.76	0.905

Table 5.2 Prediction of fluency with Elastic Net regression model

Model	R^2	corr.
Elastic Net	0.902	0.956

Elastic Net regression model. 5-fold cross validation is used to train the model. The result is shown in Table 5.2. From the table, we can see that the correlation, 0.956, increases compared with the best result we get in Section 4.

The result may show that soft clustering is more reasonable than hard clustering. In order to get more insight into it, we decide to visualize part of the hard clustering matrix A and soft clustering matrix B which are shown in Figure 5.7 and Figure 5.8. In the figure, vertical axis represents center phonemes of original $2,000$ triphone classes of posteriorgram and horizontal axis represents 50 clustered dimensions. Due to the page size restriction, the vertical axis only represent part of the 2,000 dimensions. It can be seen that compared with hard clustering, posteriorgram mainly spreads to a restricted set of dimensions after soft clustering. As introduced in Section 4.1, posterior vector can be seen as pronounced phoneme distribution for specific groups, like Japanese and native English speakers. It is obvious that Japanese and native English speakers have different phoneme distribution because Japanese tend to substitute phonemes when speaking English. For example, they may replace vowels in English with a restricted set of vowels in Japanese since they have never encountered these vowels in Japanese. In Section 4, we used distance matrix to compute the binary clustering matrix A and clustered posteriorgrams using this matrix. However, this kind of hard clustering pattern may not be reasonable for maximizing phoneme distribution gap between Japanese and native speakers. Suppose there are two sets of English phonemes. One set is M and the other is N. And Japanese tend to replace phonemes in M with those in N. In order to represent the phoneme distribution gap between Japanese and native speakers using binary matrix, we can cluster phonemes in M as one phoneme and phonemes in N as another phoneme. However, if Japanese want to pronounce words containing phonemes in N and they can pronounce it correctly, then hard clustering cannot differentiate the two cases: 1) wrongly pronouncing phonemes in M with those in N, 2) correctly pronouncing words containing phonemes in N. However, soft clustering can distribute the posterior of phonemes in N among clustered dimensions according to occurrence frequency of the two cases. As a result, we can use soft clustering to represent and maximize phoneme distribution gap between Japanese and native speakers. This may account for the effectiveness of soft clustering.

5.3 Discussion

In this experiment, we only have utterances from 100 people. At first, we use the data to train the network model and optimize soft clustering matrix B using 5-fold cross validation. Then we fix the obtained matrix B and use it to cluster posteriorgrams from which features are extracted. Though combing soft clustering with standardization achieve better result than that in Section 4, it may result from overfitting. In order to validate the effectiveness of soft clustering, it is necessary to collect more data and test soft clustering matrix B on unseen data.

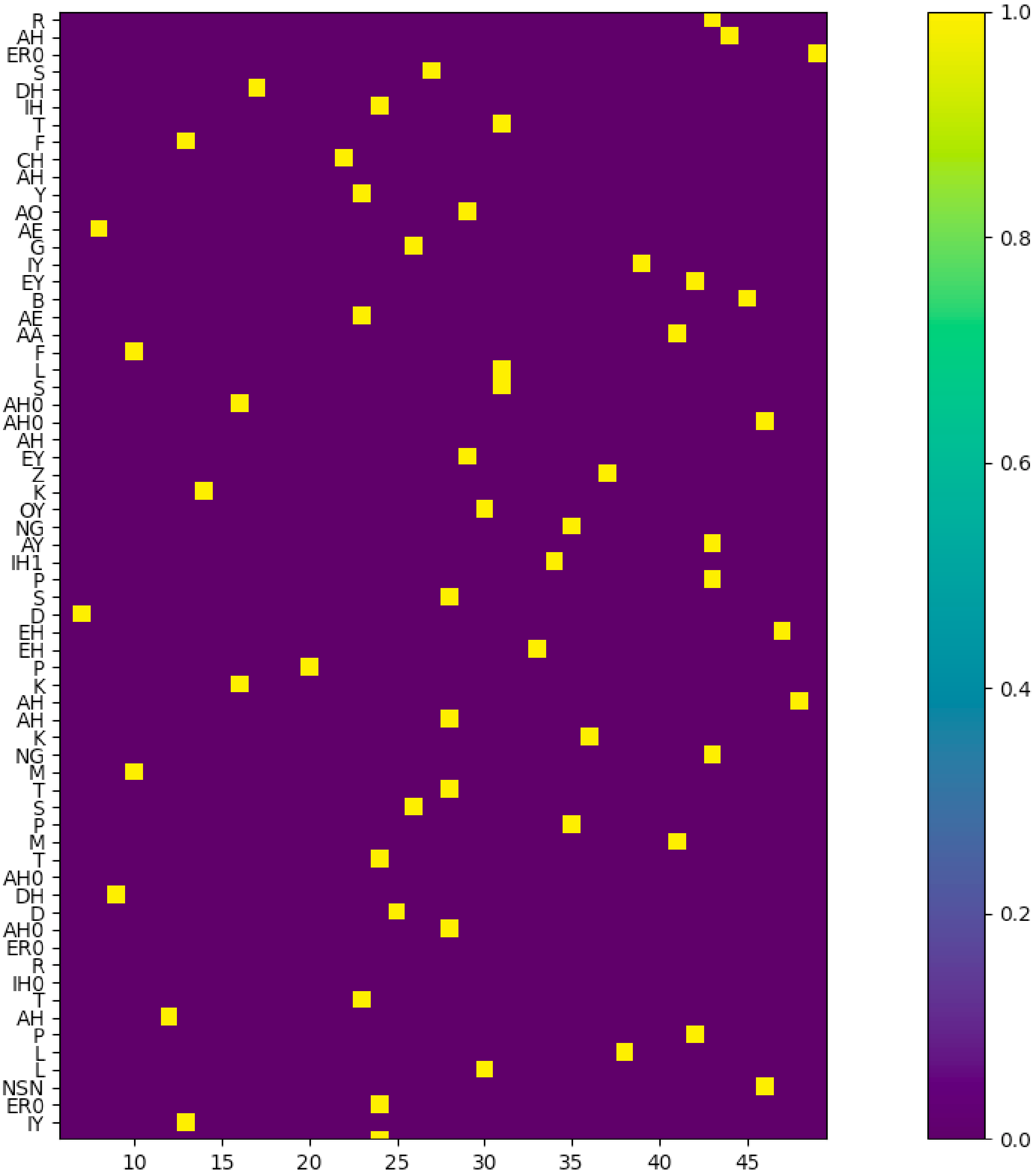


Fig. 5.7 Visualization of hard clustering matrix A

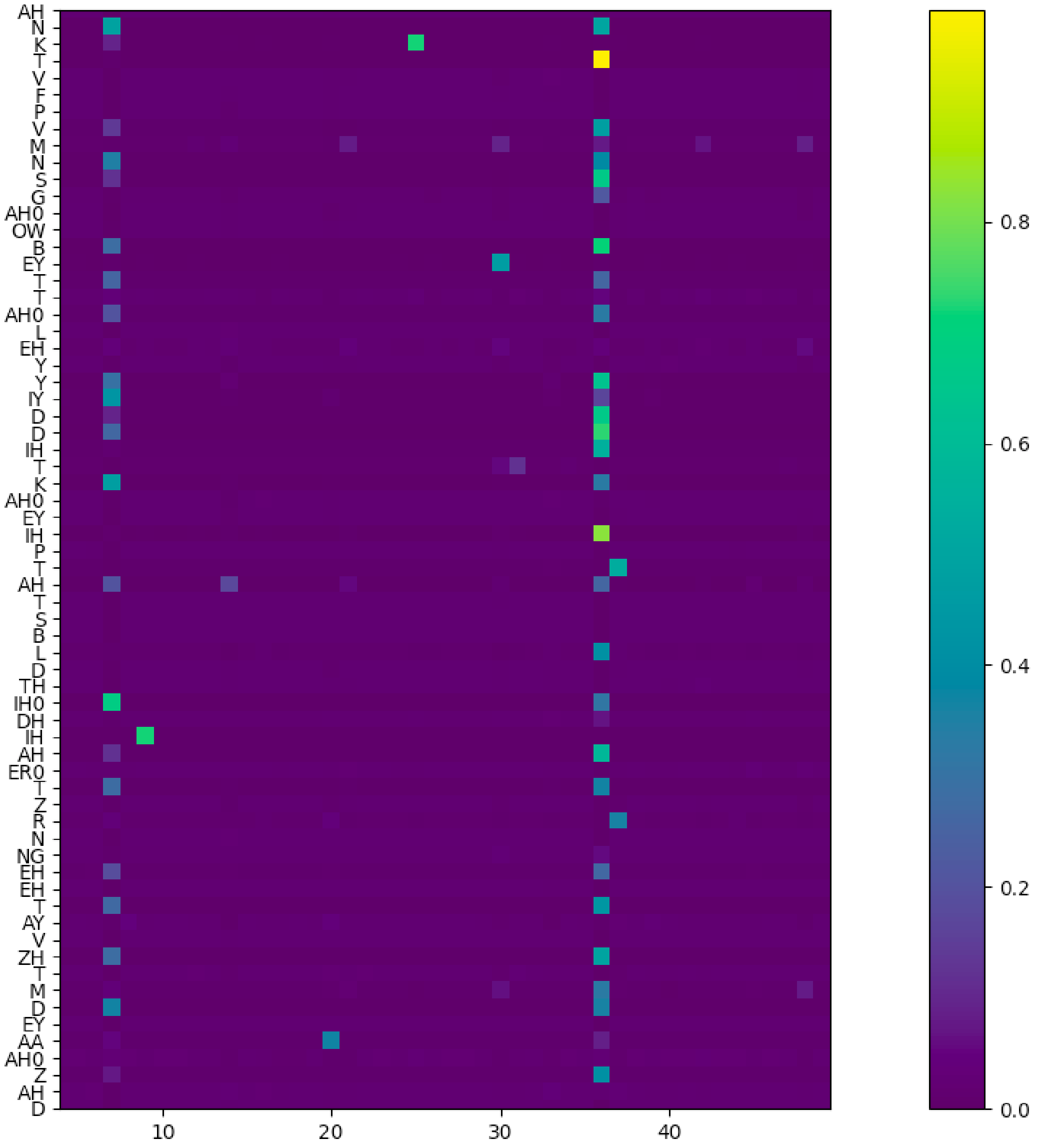


Fig. 5.8 Visualization of soft clustering matrix *B*

Chapter 6

Prosody error analysis based on DNN

In this chapter, we introduce how to use DNN models to detect prosody errors in learners' English utterances. In fact, we build four different DNN models with different input features and output classes. These input features and output classes mainly differ in whether they are related with prosody or not. For a specific learner's English utterances with text known, we have multiple native speakers' utterances for the same text. The basic idea is to compute posteriorgram sequences for learner's and native speakers' utterances. Then DTW is applied to quantify difference in posteriorgram sequences between learner and native speakers. At last, differences from four DNN models is compared to determine whether prosody error exists or not. Thus we will first introduce input and output of DNN models and ways to compute difference in posteriorgrams. Then corpus and specific experiment settings will be introduced.

6.1 DNN models trained with prosody

ASR generally uses spectrum features only, where prosodic features were often removed from speech signals in the front-end of ASR. In this experiment, however, the ultimate goal is to identify causes of comprehensibility reduction of L2 speech, i.e. phoneme pronunciation or prosody control. In this thesis, we consider prosodic errors but we put a special focus on word-level prosody, that is assignment of the stress level for individual vowels in a word utterance. Like the CMU pronunciation dictionary [44], we consider three levels of stress, unstress (0), primary stress (1), and secondary stress (2). For example, vowel ER has three variations of ER_0,ER_1 and ER_2.

To detect prosodic errors at word level, it is necessary to extract prosodic features of speech. [19] shows that speech energy, fundamental frequency (pitch) and duration are related to prosody of speech. Since DNN-based acoustic modeling requires frame-based features, we used energy and pitch as prosodic features.

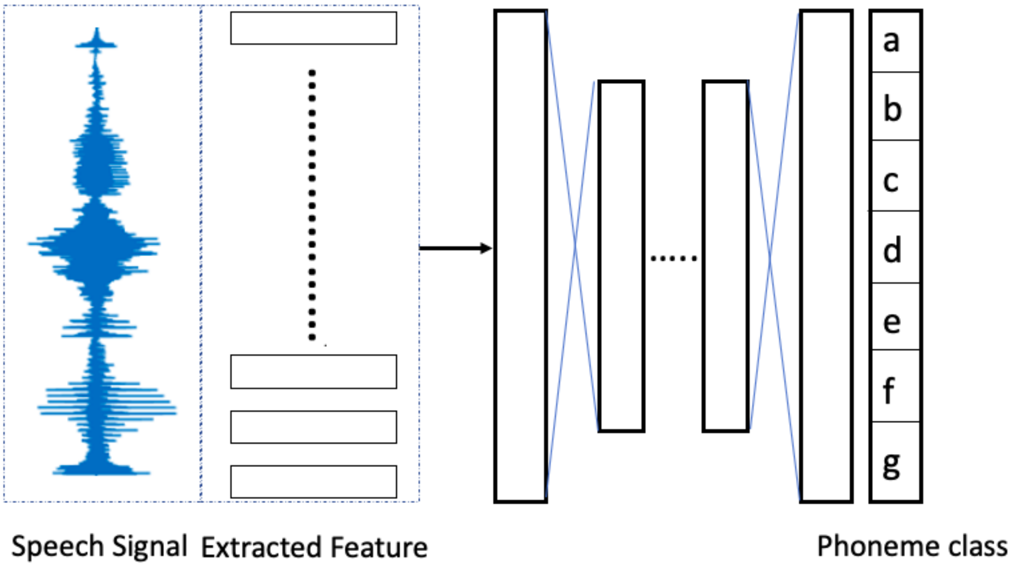


Fig. 6.1 Structure of DNN models trained with different input features and output phoneme classes

Table 6.1 Four different kinds of DNN models

A	MFCC only without stress labels
B	MFCC+prosody without stress labels
C	MFCC only with stress labels
D	MFCC+prosody with stress labels

Four different DNN models are trained with the WSJ corpus [34], which mainly differ in input features and output phoneme labels. We use two different feature sets and two different label sets, leading to four different DNN models. The two features are MFCC+Δ+ΔΔ of 12+12+12 dimensions and MFCC+energy+pitch+Δ+ΔΔ of 14+14+14 dimensions. Here, pitch values for unvoiced segments are obtained by interpolation. The two phoneme class sets differ only in vowel classes, where the first set uses stress labels (0, 1, and 2) for individual vowels and the second set does not. Illustration for the structure of DNN acoustic models is shown in Figure 6.1. As you can see, we can change extracted features and output phoneme classes. That's how we build four different DNN acoustic models.

Even without prosodic features, DNN can be trained using vowels with stress labels. Table 6.1 shows the four models used in this study.

Two utterances are compared by DTW using their posteriorgrams. As we have four models, DTW comparison gives us four different results or DTW paths based on the four different DNN models. The averaged DTW-difference measured from “black b*ir*d" and “bl*a*ckbird", for example, is expected to be emphasized and larger with DNN models with prosody compared to the baseline model **A**.

6.2 Dynamic Time Warping (DTW)

In this thesis, DTW is used to compute posteriorgram-based difference. DTW is an algorithm used to quantify similarity between two time series which may have different length. This method has been applied to many different fields, such as temporal sequences of video, audio and graphics data. Technically speaking, DTW calculates an optimal match between two given sequences with following restrictions and rules [33].

1. Every index from the first sequence must be matched with one or more indices from the other sequence, and vice versa.
2. The first index from the first sequence must be matched with the first index from the other sequence (but it does not have to be its only match).
3. The last index from the first sequence must be matched with the last index from the other sequence (but it does not have to be its only match).
4. The mapping of the indices from the first sequence to indices from the other sequence must be monotonically increasing, and vice versa, i.e. if $j > i$ are indices from the first sequence, then there must not be two indices $l > k$ in the other sequence, such that index i is matched with index l and index j is matched with index k, and vice versa.

The optimal match is denoted by the match that satisfies all the restrictions and the rules above and that has the minimal cost, where the cost is computed as the sum of absolute differences, for each matched pair of indices, between their values.

In this paper, each pair of indices is a pair of vectors, namely phoneme posteriorgrams. And we compute the difference between posteriorgrams by using Bhattacharyya distance [24]. The formular for computing this kind of distance is as following.

$$BC(p,q) = \sum_{\chi \in X} \sqrt{p(x)q(x)} \tag{6.1}$$

$$D_B(p,q) = -\ln BC(p,q) \tag{6.2}$$

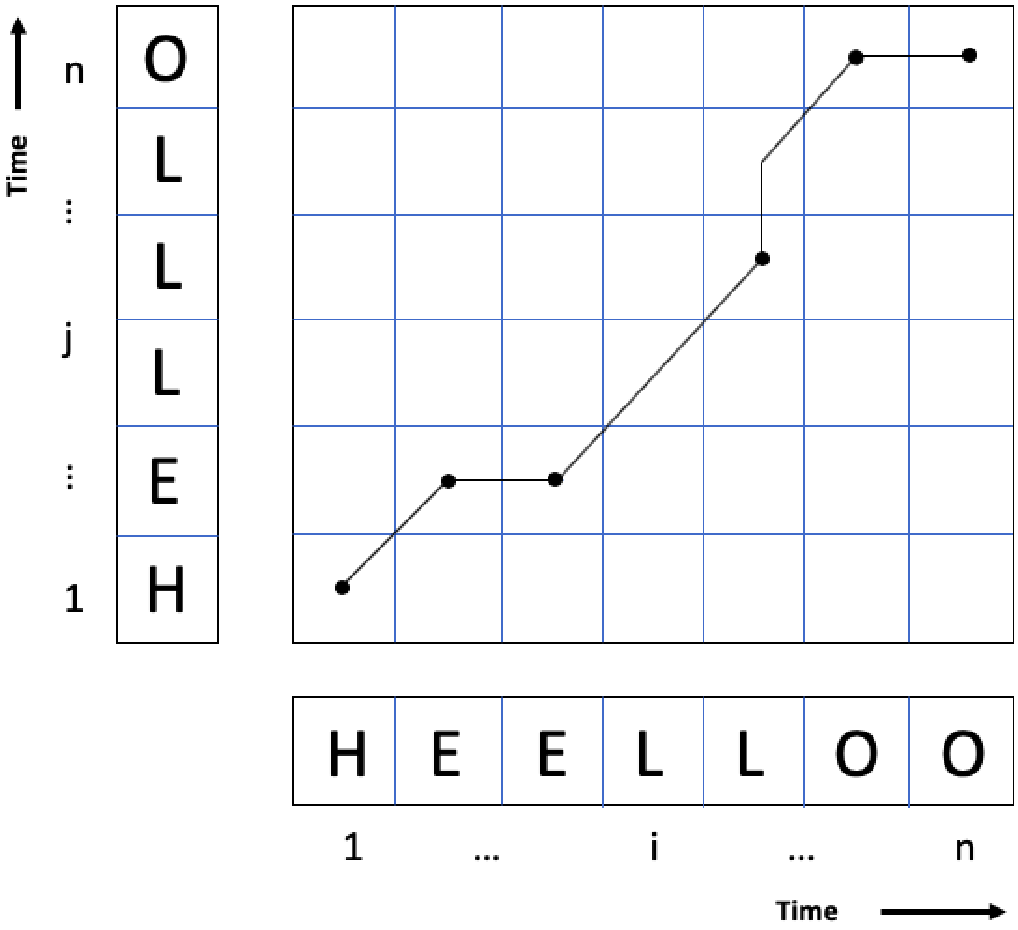


Fig. 6.2 Illustration of posterior-based DTW

In the above equation, p and q are possibility distributions and are in vector format. In fact, Bhattacharyya distance is used to measure the similarity between probability distributions which is suitable to be used on posteriorgrams. What's more, since different DNN models may have different output dimension, by using Bhattacharyya distance, posteriorgram-based difference can be normalized.

Figure 6.2 is an illustration of DTW in posteriorgrams between two utterances. As you can see, the vertical and horizontal axis are posterior vector sequences of two utterances for the word "HELLO". Each utterance is comprised of different phoneme segments whose phonation duration is different. DTW just finds and matches segments for the same phoneme in two utterances. And then distance between posterior vectors of matched segments in two utterances are computed using Bhattacharyya distance [24].

6.3 Corpuses used in this paper

In [17, 20, 27], we collected L2 utterances and native shadowings, and from them, we detected incomprehensible segments. However, with these segments, we do not have any labels or annotations related to causes of comprehensibility reduction (phoneme pronunciation or prosody). Therefore in this study, we used word utterances from the ERJ (English Read by Japanese) corpus [32] because they had segmental and/or prosodic manual annotations. From ERJ, we extracted two sets of word utterances. The first set are spoken by Japanese learners and the second set are spoken by native (model) speakers.

In ERJ, word utterances from 100 male and 100 female Japanese college students are found. To a part of them, manual scores of goodness of stress assignment are given, but it should be noted that, to those words, no manual scores of goodness of articulation are given. Score 5 is best and score 1 is poorest. Those words are divided into two groups, words with lower scores (1, 2, and 3) and those with higher scores (4 and 5). These L2 word utterances are compared to model utterances based on prosody-less posteriorgrams and with-prosody posteriorgrams.

In the second set of words, only native (model) speakers' utterances are contained. In English, some compound nouns and their original word sequences can have different stress assignments, leading to different meanings. One example is “black b*ir*d" and “bl*a*ckbird". They share the same phoneme sequence but stress assignment is different. By using these native utterance samples, compound expressions and their original word sequences are compared based on prosody-less posteriorgrams and with-prosody posteriorgrams. In this case, learners' utterances are not used.

6.4 Experiments

6.4.1 Word set from ERJ used for analysis

There are 829 English word utterances produced by Japanese students, to which scores of goodness of stress assignment are given. The scores vary from 1 to 5. The number of words with higher scores (4 and 5) is 480 and that with lower scores (1, 2, and 3) is 349. In ERJ, each word has native (model) samples and the number of native speakers is three or four, depending on the word. This means that any Japanese-English word utterance can be compared to three to four native samples, resulting in multiple posteriorgram comparisons.

Table 6.2 Average DTW differences obtained from L2 word utterances with lower scores

A	B	C	D
0.353	0.347	0.383	0.378

Table 6.3 p-values obtained between two models with L2 word utterances with lower scores

	A	B	C	D
A		0.899	0.003	0.016
B			0.000	0.001
C				0.956

6.4.2 DTW-differences obtained with the four models

We have two groups of L2 word utterances: with higher scores and with lower scores. For each group, the average posteriorgram-based DTW difference is computed separately for each model, shown in Tables 6.2 and 6.4. Besides, Analysis of variance (ANOVA) is done between any two models of the four ones, and p-values are calculated and shown in Tables 6.3 and 6.5.

From Tables 6.2 and 6.3, we can say that adding prosodic features did not increase DTW differences with lower scores. In contrast, adding stress labels increased them significantly. In English, stressed vowels and unstressed vowels of the same vowel class differ not only in prosodic features but also in spectrum features. This will be why DNN trained with MFCC only but with stress labels can emphasize DTW-differences between L2 poor utterances and native utterances.

From Tables 6.4 and 6.5, for good L2 utterances with higher scores, it is clearly shown that addition of prosodic features or labels did not influence differences effectively.

These results may indicate that, for a given L2 speech segment whose pronunciation is inadequate, the following method may be able to identify causes of comprehensibility reduction. The L2 segment and its corresponding native segment are compared based on two kinds of posteriorgrams, with and without prosody. When the ratio of two DTW-differences, i.e. with-prosody to prosody-less, is large, the reduction can be attributed to inadequate control of prosody. In this experiment, however, the L2 utterances with lower stress scores may also have inadequate articulation. We have to admit that this fact makes it difficult to lead to clear conclusion.

Table 6.4 Average DTW differences obtained from L2 word utterances with higher scores

A	B	C	D
0.250	0.255	0.261	0.261

Table 6.5 p-values obtained between two models with L2 word utterances with higher scores

	A	B	C	D
A		0.883	0.374	0.336
B			0.819	0.782
C				1.000

6.4.3 DTW-differences obtained from native utterances of compounds

In this section, we focus on utterances always with perfect articulation but maybe with inadequate control of prosody. Here, we use only native samples where different stress assignment is possible and it causes different meanings. In ERJ, we have utterances of four compound expressions paired with those of their original word sequences:

1) Pair1(/ER/): black b*ir*d - bl*a*ckbird

2) Pair2(/UW/): dark r*oo*m - d*ar*kroom

3) Pair3(/AW/): light h*ou*sekeeper - lighthouse k*ee*per

4) Pair4(/EY/): brief c*a*se - br*ie*fcase

Different from the previous section, only native samples for these compounds are used for the following analysis. We firstly select two native speakers out of the eight native speakers in ERJ, the first of whom reads the first expression and the other of whom reads the second expression in the above four pairs. Posterior-based DTW is done between the two utterances. By selecting other speakers, this process is repeated. It should be noted that compound expressions are longer than the words used in the previous section, here we focused only on the vowel segments shown as *italic*. Finally, the averaged DTW-differences in the target four vowels are calculated using the four DNN models.

In ERJ, for each compound word, there are eight native speakers who read it aloud. As a result, there are 56 DTW-differences for each compound word pair from each model. The number of samples is much smaller than that in the previous section. The averaged vowel-based DTW-differences are shown in Table 6.6 for all the four models. Their p-values between

Table 6.6 Averaged vowel-based DTW differences for compounds

	A	**B**	**C**	**D**
Pair1/ER	0.082	0.090	0.245	0.195
Pair2/UW	0.042	0.223	0.396	0.284
Pair3/AW	0.745	0.793	0.781	0.877
Pair4/EY	0.011	0.012	0.009	0.015

different pairs of the four models are shown in Tables 6.7 to 6.10, each corresponding to each vowel.

In comparison between the baseline model (**A**) and the models with stress labels (**C** and **D**), we can say that /ER/ and /UW/ have rather clear differences between a compound expression and its original form. In contrast, in the case of /AW/ and /EY/, posteriorgrams with/without stress labels seem to generate no clear differences. After obtaining the result, we let three native speakers listen to utterances of third and fourth compound pairs to examine their prosodic differences. It was found that, in the third and forth pairs, it is difficult to distinguish in each pair.

6.5 Discussion

Four DNN models that differ in input features and output phoneme classes were trained to realize posteriorgram-based DTW with/without prosodic features or classes. It was shown that by adding stress levels to vowels, the DTW difference between L2 word utterances with low scores and native readings tends to increase. However, adding prosodic features does not influece DTW difference significantly. This demonstrates that it is possible to detect inadequate stress assignment in L2 speech.

Posteriorgram with stress labels can be viewed as that with higher phonemic resolution compared to that without stress labels. With stress labels, words with inadequate stress assignment can be detected.

Table 6.7 *p*-values obtained between a pair of models with vowel /ER/

	A	**B**	**C**	**D**
A		0.996	0.000	0.014
B			0.000	0.026
C				0.518

Table 6.8 *p*-values obtained between a pair of models with vowel /UW/

	A	**B**	**C**	**D**
A		0.004	0.000	0.000
B			0.006	0.643
C				0.148

Table 6.9 *p*-values obtained between a pair of models with vowel /AW/

	A	**B**	**C**	**D**
A		0.999	0.999	0.976
B			1.000	0.993
C				0.991

Table 6.10 *p*-values obtained between a pair of models with vowel /EY/

	A	**B**	**C**	**D**
A		0.872	0.903	0.302
B			0.473	0.756
C				0.074

Chapter 7

Conclusions and Future Works

7.1 Conclusions

In this thesis, we mainly present the how to efficiently use multi-resolution posteriorgrams to estimate fluency and detect prosody errors for Japanese English.

First, we discuss how to design and extract features from posteriorgrams for fluency estimation with Elastic Net regression model. We show that quality features like posterior distribution and distribution gap to natives are effective for fluency estimation. Besides, we investigate two methods to reduce the resolution of posteriorgrams. It is found that bottom-up clustering is more effective than top-down clustering and the optimal resolution of posteriorgrams is 50 in the case of bottom-up clustering. At last, with newly designed quality features and quantity features from previous work, we achieve higher correlation of fluency score when compared with the maximum of one-to-others human raters.

Second, we discuss how to learn soft clustering matrix B in bottom-up clustering. Network-based model is introduced to optimize the matrix and estimate fluency score. The correlation obtained with the network model is lower than that of Elastic Net regression model, possibly due to lack of standardization. In order to utilize standardization, we fix the optimized clustering matrix and combine it with Elastic Net regression model again to estimate fluency score. It is shown that the final correlation exceeds that of Elastic Net regression model.

Finally, we discuss how to detect prosody error, especially inadequate stress assignment, in Japanese English. Posterior-based DTW difference between learners' and native speakers' utterances can be computed with DNN acoustic models. Since four DNN models with or without input features and output classes related with prosody are trained, we can compare posterior-DTW differences from these DNN models. It is found that adding vowels labeled with stress to output classes of DNN acoustic model can help increase posterior-based DTW difference for learners' utterances with low scores. As a result, it is possible to detect prosody errors in learners' utterances with high resolution posteriorgrams.

7.2 Future works

In future studies, we are going to the following.

- Newly designed quality features extracted from posteriorgrams are effective for Japanese English fluency estimation. These features can be tested in different tasks. In the future, we will obtain different corpuses. For example, the corpus introduced in Section 3 will be expanded from 100 utterances to 128 utterances. Besides, Polish English and English spoken by various learners (learners speaking different L1 languages) will also be collected. In the case of Japanese as L2 language, Japanese spoken by various learners will also be collected. The method proposed in this thesis for fluency estimation can be tested on these corpuses to see whether it is generally effective.
- High correlation obtained by combing optimized soft clustering matrix and Elastic Net regression model may due to overfitting. In order to validate the effectiveness of soft clustering, we need to collect more data and do the same experiment again on unseen data.
- It is well-known that a stressed vowel and its unstressed version differ in spectrum, pitch, power, and duration. Direct comparison is made often with only one type of feature, but stress is a result of controlling multiple factors. In this case, two utterances should be compared not based on individual speech features but based on stress-based posteriors, which are calculated using segmental features and prosodic features as input. This theoretical claim should be investigated experimentally in future work.

References

[1] ayano yasukagawa, Shintaro Ando, Eisuke Konno, Zhenchao Lin, Yusuke Inoue, Daisuke Saito, Nobuaki Minematsu, and Kazuya Saito. An experimental study of automatic scoring of fluency of spontaneous english utterances by japanese learners. In *Proc. Annual Meeting of the Japan Association for Language Education and Technology*, 2020.

[2] Paul Boersma. Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound. In *Proceedings of the institute of phonetic sciences*, volume 17, pages 97–110. Amsterdam, 1993.

[3] Paul Boersma. Praat: doing phonetics by computer. *http://www.praat.org/*, 2006.

[4] Arturo Camacho and John G Harris. A sawtooth waveform inspired pitch estimator for speech and music. *The Journal of the Acoustical Society of America*, 124(3):1638–1652, 2008.

[5] Francine Chambers. What do we mean by fluency? *System*, 25(4):535–544, 1997.

[6] Lei Chen, Klaus Zechner, Su-Youn Yoon, Keelan Evanini, Xinhao Wang, Anastassia Loukina, Jidong Tao, Lawrence Davis, Chong Min Lee, Min Ma, et al. Automated scoring of nonnative speech using the speechrater sm v. 5.0 engine. *ETS Research Report Series*, 2018(1):1–31, 2018.

[7] Catia Cucchiarini, Helmer Strik, and Lou Boves. Quantitative assessment of second language learners' fluency by means of automatic speech recognition technology. *The Journal of the Acoustical Society of America*, 107(2):989–999, 2000.

[8] Alain De Cheveigné and Hideki Kawahara. Yin, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, 111(4):1917–1930, 2002.

[9] John R Deller, John G Proakis, and John HL Hansen. Discrete-time processing of speech signals. Institute of Electrical and Electronics Engineers, 2000.

[10] Maxine Eskenazi. An overview of spoken language technology for education. *Speech Communication*, 51(10):832–844, 2009.

[11] Luciana Ferrer, Harry Bratt, Colleen Richey, Horacio Franco, Victor Abrash, and Kristin Precoda. Classification of lexical stress using spectral and prosodic features for computer-assisted language learning systems. *Speech Communication*, 69:31–45, 2015.

[12] Lionel Fontan, Maxime Le Coz, and Sylvain Detey. Automatically measuring l2 speech fluency without the need of asr: A proof-of-concept study with japanese learners of french. In *INTERSPEECH*, pages 2544–2548, 2018.

[13] Jiang Fu, Yuya Chiba, Takashi Nose, and Akinori Ito. Automatic assessment of english proficiency for japanese learners without reference sentences based on deep neural network acoustic models. *Speech Communication*, 116:86–97, 2020.

[14] Yo Hamada. The effectiveness of pre-and post-shadowing in improving listening comprehension skills. *The Language Teacher*, 38(1):3–10, 2014.

[15] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[16] Kun-Ting Hsieh, Da-Hui Dong, and Li-Yi Wang. A preliminary study of applying shadowing technique to english intonation instruction. *Taiwan Journal of Linguistics*, 11 (2):43–65, 2013.

[17] Yusuke Inoue, Suguru Kabashima, Daisuke Saito, Nobuaki Minematsu, Kumi Kanamura, and Yutaka Yamauchi. A study of objective measurement of comprehensibility through native speakers' shadowing of learners' utterances. In *INTERSPEECH*, pages 1651–1655, 2018.

[18] Talia Isaacs. Fully automated speaking assessments: Changes to proficiency testing and the role of pronunciation. Routledge, 2017.

[19] Denis Jouvet. Speech processing and prosody. In *International Conference on Text, Speech, and Dialogue*, pages 3–15. Springer, 2019.

[20] Suguru Kabashima, Yuusuke Inoue, Daisuke Saito, and Nobuaki Minematsu. Dnn-based scoring of language learners' proficiency using learners' shadowings and native listeners' responsive shadowings. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 971–978. IEEE, 2018.

[21] Emiko Kaneko. Vowel selection in japanese loanwords from english. *Proceeding LSO Working Papers in Linguistics*, 6:49–62, 2006.

[22] Yosuke Kashiwagi, Congying Zhang, Daisuke Saito, and Nobuaki Minematsu. Divergence estimation based on deep neural networks and its use for language identification. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5435–5439. IEEE, 2016.

[23] T Kawahara and N Minematsu. Computer-assisted language learning (call) based on speech technologies. *IEICE Tans. Inf. & Syst.(Japanese Edition)*, 96:1549–1565.

[24] Dimitri Kazakos. The bhattacharyya distance and detection between markov chains. *IEEE Transactions on Information Theory*, 24(6):747–754, 1978.

[25] Michael Kearns, Yishay Mansour, and Andrew Y Ng. An information-theoretic analysis of hard and soft assignment methods for clustering. In *Learning in graphical models*, pages 495–520. Springer, 1998.

[26] Paul Lennon. Investigating fluency in efl: A quantitative approach. *Language learning*, 40(3):387–417, 1990.

[27] Zhenchao Lin, Yusuke Inoue, Tasavat Trisitichoke, Shintaro Ando, Daisuke Saito, and Nobuaki Minematsu. Native listeners' shadowing of non-native utterances as spoken annotation representing comprehensibility of the utterances. In *SLaTE*, pages 43–47, 2019.

[28] James Lyons. Mel frequency cepstral coefficient (mfcc) tutorial. *Practical Cryptography*, 2015.

[29] Andrew L Maas, Peng Qi, Ziang Xie, Awni Y Hannun, Christopher T Lengerich, Daniel Jurafsky, and Andrew Y Ng. Building dnn acoustic models for large vocabulary speech recognition. *Computer Speech & Language*, 41:195–213, 2017.

[30] Pavel Matejka, Petr Schwarz, Jan Cernockỳ, and Pavel Chytil. Phonotactic language identification using high quality phoneme recognition. In *Ninth European Conference on Speech Communication and Technology*, 2005.

[31] TA Matthew. Language learning theories and cooperative learning techniques in the efl classroom. *Doshisha studies in Language and Culture*, 9(2):277–301, 2006.

[32] Nobuaki Minematsu, Yoshihiro Tomiyama, Kei Yoshimoto, Katsumasa Shimizu, Seiichi Nakagawa, Masatake Dantsuji, and Shozo Makino. English speech database read by japanese learners for call system development. In *LREC*, 2002.

[33] Meinard Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007.

[34] Douglas B Paul and Janet Baker. The design for the wall street journal-based csr corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*, 1992.

[35] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, number CONF. IEEE Signal Processing Society, 2011.

[36] Xiaojun Qian, Helen Meng, and Frank K Soong. On mispronunciation lexicon generation using joint-sequence multigrams in computer-aided pronunciation training (capt). In

Twelfth Annual Conference of the International Speech Communication Association, 2011.

[37] Sebastian Raschka. About feature scaling and normalization and the effect of standardization for machine learning algorithms. *Polar Political Legal Anthropology Rev*, 30(1): 67–89, 2014.

[38] Kazuya Saito, Meltem Ilkan, Viktoria Magne, Mai Tran, and Shungo Suzuki. Acoustic characteristics and learner profiles of low, mid and high-level second language fluency. *Applied psycholinguistics*, 39(3):593–167, 2018.

[39] Bikesh Kumar Singh, Kesari Verma, and AS Thoke. Investigations on impact of feature normalization techniques on classifier's performance in breast tumor classification. *International Journal of Computer Applications*, 116(19), 2015.

[40] Shungo Suzuki and Judit Kormos. Linguistic dimensions of comprehensibility and perceived fluency: An investigation of complexity, accuracy, and fluency in second language argumentative speech. *Studies in Second Language Acquisition*, 42(1):143–167, 2020.

[41] Michael Swan and Bernard Smith. A teacher's guide to interference and other problems, 2001.

[42] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[43] Joe H Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

[44] Robert Weide. The cmu pronunciation dictionary, release 0.6, 1998.

[45] Silke M Witt and Steve J Young. Phone-level pronunciation scoring and assessment for interactive language learning. *Speech communication*, 30(2-3):95–108, 2000.

[46] Kakeru Yazawa, Takayuki Konishi, Keiko Hanzawa, Greg Short, and Mariko Kondo. Vowel epenthesis in japanese speakers' l2 english. In *ICPhS*, 2015.

[47] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320, 2005.

Appendix A

Publications

Domestic conferences and meetings

- Yang Shen, Shintaro Ando, Nobuaki Minematsu, Daisuke Saito and Satoshi Kobashikawa, “Automatic estimation of prosodic control made in English utterances using DNN-based acoustic models trained with prosodic features and labels”. In 電子情報通信学会音声研究会資料, pp. 201-206, 2020-3.

- Yang Shen, Shintaro Ando, Nobuaki Minematsu, Daisuke Saito and Satoshi Kobashikawa, “Experimental study on prosodic error detection in English utterances using DNN-based acoustic models trained with prosodic features and labels”. In 日本音響学会講演論文集, pp. ??-??, 2020-3.

- Yang Shen, Ayano Yasukagawa, Daisuke Saito, Nobuaki Minematsu, Kazuya Saito, “Improved Prediction of Perceived Fluency of Japanese English using Quantity of Phonation and Quality of Pronunciation”. In 日本音響学会講演論文集, pp. ??-??, 2020-9.

National conferences and meetings

- Yang Shen, Ayano Yasukagawa, Daisuke Saito, Nobuaki Minematsu, Kazuya Saito, “Improved Prediction of Perceived Fluency of Japanese English using Quantity of Phonation and Quality of Pronunciation". IEEE Spoken Language Technology Workshop, 2020.