Paralinguistic and Nonverbal Information Extraction from Speech Signal towards Empathetic Dialogue Systems

FUJIMURA, Hiroshi / 藤村, 浩司

(開始ページ / Start Page)
1
(終了ページ / End Page)
84
(発行年 / Year)
2022-03-24
(学位授与番号 / Degree Number)
32675甲第546号
(学位授与年月日 / Date of Granted)
2022-03-24
(学位名 / Degree Name)
博士(理学)
(学位授与機関 / Degree Grantor)
法政大学 (Hosei University)
(URL)
https://doi.org/10.15002/00025229

法政大学審査学位論文

Hosei University Doctoral Dissertation

Paralinguistic and Nonverbal Information Extraction from Speech Signal towards Empathetic Dialogue Systems

Hiroshi FUJIMURA

Graduate School of Computer and Information Sciences, HOSEI University

Supervisor: Professor Katsunobu ITOU

March 2022

Contents

List of Figures

List of Tables

Abstract

In this research, we aim to extract paralinguistic and nonverbal information such as emotions, speaking style, and speaker attributes towards a human-like empathetic dialogue system. Empathy is the ability to project the other person's feelings and thoughts onto the other person's knowledge. It plays an important role in human communication. In particular, personalization and understanding emotion are essential for an advanced dialogue system. This research focuses on methods for estimating speaker attributes, personal speaking-style and emotion category that are related to personalization and emotion in real-time from a small amount of speech information like a human agent. By integrating the methods proposed in this paper, it is possible to realize more human-like recognition of paralinguistic and nonverbal information for automatic dialogue systems using speech. This doctoral dissertation consists of five chapters. In chapter 1, the introduction is described.

In chapter 2, we propose a method for identifying speaker attributes, which are nonverbal information in speech. We specially focus on the identification of male and female speeches as speaker attributes in this chapter. In order to extract speaker attributes, it is necessary to first detect a speech segment from a sound signal sequence, which is a mixture of speech and non-speech segments, and then to identify them in the speech segment. In conventional speaker attribute identification, the endpoint of speech with a certain length of continuous speech is detected, and then the features to identify speaker attributes are extracted, and an identification process is performed for the segment. However, a delay time occurs to identify speaker attributes since the process starts after the end of speech is detected. In our method, the speaker attributes and the probabilities for the speech and non-speech segments are calculated simultaneously for each time frame using a single neural network. The framework can identify speaker attributes sequentially based on their accumulated probabilities. This method made it possible to classify male and female speech with high accuracy while maintaining the accuracy of speech segment detection.

In Chapter 3, we propose a phoneme identification method that leads to the extraction of low-intelligibility speech. When low-intelligibility speech occurs, phonemes in a relevant part of a speech are unclear and differ significantly from the nature of phonemes in ordinary speech. Since features of the phonemes depend on a relative phoneme position, it is necessary to cluster them depending on the phoneme position, and a discriminative model for each cluster is trained to determine whether the phoneme is clearly uttered or not. Therefore, we propose a discriminator that contains phoneme environment-dependent clusters inside, which enables to discriminate phonemes without pre-clustering and to calculate a score for

the intelligibility.

In chapter 4, we propose a method for extracting paralinguistic and nonverbal information, such as fillers and word fragments. There are many variations of fillers and word fragments, and it is not easy to keep all patterns as a dictionary of language in advance. Therefore, existing methods use two-pass decoding to detect fillers and word fragments based on a confusion network output from first-pass recognition and sub-word language model to deal with various fillers and word fragments. However, this method is unsuitable for real-time applications because it can only start processing after decoding the end of the utterance. To solve this problem, we propose a method of learning filler and word fragment acoustic patterns as filler symbols and word fragment symbols, respectively, and incorporating a detection process using filler symbols and word fragment symbols into a WFST decoder for speech recognition, thereby processing them in a single pass of the decoder.

There is no need to register all speech patterns of filler, and word fragment in a language dictionary since the proposed method treats filler and word fragment as a single acoustic symbol. By this method, fillers and word fragments can be detected in real-time. Simultaneously, the speech is recognized in one pass without degrading the accuracy. As for fillers, the number of occurrences can be controlled by using a confidence score based on the number of occurrences of filler symbols.

In chapter 5, we propose a method for recognizing emotions, which are paralinguistic and nonverbal information. At present, the accuracy of emotion classification for 7 or 8 emotions is only 70 or 80%, even when the emotions are uttered intentionally. Therefore, performance improvement is desired. Emotional features in speech are contained in both a short speech signal and a long speech signal. Therefore, many efforts have been made to improve the performance of emotion classification by incorporating features of various temporal resolutions. Conventional emotion recognition methods tried to improve the performance by using a single neural network encompassing multiple temporal resolutions. However, they have not been able to significantly improve the performance due to the small emotional speech database.

We consider that the performance of emotion classification methods using high-level statistical functions (HSFs), which show high accuracy in emotion classification, can be improved by extracting and combining HSFs from windows with multiple temporal resolutions instead of a single fixed window length. In this paper, we aim to improve the accuracy by extending the HSFs extracted from a single fixed window in the existing methods to HSFs generated from multiple windows with temporal resolutions of 30 or more. In addition, to reduce the number of parameters to be learned simultaneously for a small amount of data, stacking with Gradient Boosting Decision Trees (GBDT) is applied when combining features of multiple temporal resolutions. As a result, we obtained the highest emotion classification performance for the American emotional speech database. In addition, although the method initially uses multiple temporal resolutions of more than 30, it is found that the same classification performance can be obtained with only 15 temporal resolution features based on analyzing by GBDT.

Chapter 1

Introduction

1.1 Background and Motivation

When people communicate face-to-face, they integrate audio information emitted by the other person, visual information obtained by seeing with their own eyes, and their own past experiences. Finally they give an appropriate response for the situation. Research on analyzing and utilizing human communication from audio and vision has been conducted extensively for many years. For example, [3][4], which combine human images and speech information to improve speech segment detection and speech recognition performance. In addition, hand gestures are analyzed as to the form, use, and meaning in human communication [5]. In [6], turn-taking is predicted between listeners and presenters in a poster presentation using linguistic information, head nodding, and eye movement information. As derived from these research results, audio and visual information is the key to human communication.

Visual information includes various essential communication information, such as the other person's facial expressions and gestures. However, in call centers, intelligent speakers, teleconferences, and other forms of communication where image information is unavailable, the communication is limited to speech only. In these cases, it is necessary to estimate the other status based only on the speech information, and return an appropriate response. There is a strong need for automatic spoken dialogue systems, especially from labor-saving for call centers. In recent years, some call centers have been using spoken-dialogue system support. In addition, intelligent speakers allow users to talk to, and order from machine agents. However, it is still a difficult task to achieve the same level of response as a human agent in an automated spoken dialogue. There has been much research on speech recognition and spoken dialogue in the world, with the advent of the deep neural network [7] and the application of BERT [8] to spoken language processing. They have dramatically improved the performance.

On the other hand, it is said that in order to achieve more human-like conversations, models with more empathetic responses are needed in [9]. Empathy is the ability to project the other person's feelings and thoughts onto the other person's knowledge. It plays an essential role in human communication. Current spoken dialogue systems cannot capture and use personalization and emotional information to build a more comprehensive empathy system well. Information such as speaker attributes, personal speaking-style, is used to incorporate

personalization. These may be given as information in advance or may be inferred from the conversation. However, a system cannot obtain the information in advance for many cases from the viewpoint of privacy protection, so human agents and others respond by guessing to some extent from the audio information of the conversation. As for emotions, it is necessary to infer them from the audio information of the conversation in the case without image information since users do not tell their emotions in advance. However, research is not enough for estimating emotion, personal speaking-style and speaker attributes related to dialogue response that is useful for personalization in real-time from a small amount of speech information like a human agent.

Human speech information includes linguistic information that can be converted into text, paralinguistic and nonverbal information such as emotion, speaking style, and speaker attributes [10][11]. Current speech communication between humans and machines is often designed to make the machine understand the intent by the textual information of the human utterance [12][13][14]. In order to make machines respond humanely, incorporating empathy such as emotion, it is essential to make them understand not only verbal information but also paralinguistic and nonverbal information from speech as mentioned above. However, it is challenging to make use of it in the current state of machine learning.

In order to build a spoken dialogue system that integrates these elements, a massive amount of data is required for each spoken dialogue task. Therefore, it is essential to learn and build each element as a module to be used in combination.

Another reason why the use of paralinguistic and nonverbal information has not progressed is that it is difficult to estimate the state of such information since it is not directly observed in many cases [15][16]. In addition, a factor is the lack of a real-time method without delay. In order to respond like a human agent, we need a method that can listen to human words and guess at the same time. In this paper, we aim to improve the accuracy of extracting paralinguistic and nonverbal information. In addition, real-time performance, immediacy, and the elimination of language dependency is improved towards a spoken dialogue system that can respond like a person.

Typical examples of nonverbal information are gestures, facial expressions [17][18], and the appearance of the speaker, as well as speaker attributes, which are derived from voice characteristics [19]. The speaker attributes include age and gender, for example. In order to extract this information from speech alone, it is necessary to identify the attributes using the speech signal. In this paper, we take the identification of the gender of a speaker by speech. In particular, we describe a method to improve the real-time and immediacy of identifying speech attributes, which has not been studied so far.

There are two types of paralinguistic and nonverbal information: those that appear acoustically independently, such as filler, word fragments, and low-intelligibility speech, and those that do not appear as independent acoustic phenomena, such as emotions are caused by various emotions in the background of the speech. Fillers, word fragments, and low-intelligibility speech are highly dependent on a personal speaking style.

In previous research, the recognition of fillers, word fragments, and low-intelligibility speech is a highly language-dependent method that required processing after the utterance. Furthermore, it requires the preparation of separate text and acoustic dictionaries. In this

paper, we describe a method to improve the real-time and immediacy of recognition of fillers, word fragments, and low-intelligibility speech without language dependency. For the recognition of emotions in the background of speech, the most recent challenge is to improve emotion recognition performance [20][21][22] and [23][24][25][26]. In this paper, we focus on improving the performance of emotion recognition.

1.2 Speaker Attribute Recognition

In chapter 2, we propose a method for identifying speaker attributes, which are nonverbal information in speech. Speaker attributes include gender, age and native language, for example. We consider the problem of estimating speaker attributes from an input voice. Gender classification using speech can be used for a wide variety of applications, such as recommendation systems and trend analysis. For some applications, such as trend analysis, it is not necessary to execute gender classification in real-time. However, for other applications such as recommendation systems, real-time gender classification is required. First, we focus on gender classification for real-time applications. We specially focus on the identification of male and female speeches as speaker attributes in this chapter.

In order to extract speaker attributes, it is necessary to first detect a speech segment from a sound signal sequence, which is a mixture of speech and non-speech segments, and then to identify them in the speech segment. In conventional speaker attribute identification, the end point of speech with a certain length of continuous speech is detected, and then the features to identify speaker attributes are extracted and an identification process is performed for the segment [27]. However, a delay time occurs to identify speaker attributes since the process starts after the end of speech is detected. In our method, the speaker attributes and the probabilities for the speech and non-speech segments are calculated simultaneously for each time frame using a single neural network. The framework can identify speaker attributes sequentially based on their accumulated probabilities. This method make it possible to classify male and female speech with high accuracy while maintaining the accuracy of speech segment detection.

1.3 Phoneme Recognition

In Chapter 3, we propose a phoneme identification method that leads to the extraction of paralinguistic and nonverbal information, i.e., low-intelligibility speech. When low-intelligibility speech occurs, phonemes in a relevant part of the speech feature differ from the nature of phonemes in ordinary speech. In general, it makes discrimination difficult because of spectral reduction [28]. Therefore, the spectral-reduction degree of a phoneme is related to the discrimination score to the other phonemes. We focus on calculating a discrimination score for a phoneme in speech. The spectral reduction is lower when the discrimination score is higher. Ganapathiraju et al. [29] combined HMM and the support vector machine (SVM), which is one of the effective discriminative models. Their technique applies SVMs to phoneme segments obtained from the alignment of HMM, and N-best hypotheses are rescored by scores of

SVMs. Padrell-Sendra et al. [30] applied SVM to the frame-level discrimination, and obtained better performance than a standard GMM-HMM. However, since properties of the phonemes depend on a relative phoneme position, it has been necessary to cluster them depending on the phoneme position and train a discriminative model for each cluster to determine whether the phoneme is clearly uttered or not. Therefore, we propose a adaboost-based discriminator that contains phoneme environment-dependent clusters inside, which enables to discriminate phonemes without pre-clustering and to calculate a score for the degree of spectral reduction.

1.4 Acoustic Event Detection and Recognition

In chapter 4, we propose a method for extracting paralinguistic and nonverbal information, such as fillers and word fragments. Spontaneous speech often includes a lot of fillers and word fragments such as "um," "ah," and "thi-this" [31]. These are deeply related to a human state [32]. They also lead to significant performance deterioration of speech recognition but also decrease the readability of real-time captioning for humans. Speech recognition systems need to detect and selectively remove fillers and word fragments to prevent these problems. Since many applications of spontaneous speech recognition, such as speech-to-speech translation and speech dialogue systems, require real-time computation, it is necessary to detect fillers and word fragments with very low latency. Therefore, we focus on speech recognition and speech event recognition simultaneously.

There are many variations of fillers and word fragments, and it is challenging to keep all patterns as a dictionary of language in advance. Therefore, existing methods use two-pass decoding to detect fillers and word fragments using outputs from first-pass recognition [33][34]. However, these methods are unsuitable for real-time applications because they can only start processing after decoding the end of the utterance. To solve this problem, we propose a method of learning filler and word fragment acoustic patterns as filler symbols and word fragment symbols, respectively, and incorporating a detection process using filler symbols and word fragment symbols into a WFST decoder for speech recognition, thereby processing them in a single pass of the decoder. There is no need to register all speech patterns of filler and word fragment in a language dictionary since the proposed method treats filler and word fragment as a single acoustic symbol. Using this method, we can detect fillers and word fragments in real-time and decode them in one pass without degrading the recognition accuracy. As for fillers, the number of occurrences can be controlled by using a confidence score based on the number of occurrences of filler symbols.

1.5 Acoustic Emotion Recognition

In chapter 5, we propose a method for recognizing emotions, which are paralinguistic and nonverbal information. Speech emotion recognition also has an essential role in speech applications. If a speech application using a spoken dialogue system can recognize various human emotions, a machine navigator can provide more appropriate replies for a given

situation. However, it is challenging to realize speech applications capable of recognizing emotions because it is difficult to prepare a speech database that contains a variety of emotions for various situations. Therefore, we focus on acoustic speech emotion recognition using small databases.

There are two major types of speech emotion recognition: the "categorical emotion task," which classifies speech into a single emotion, and the "dimensional emotion task," which continuously converts speech into three-dimensional numerical expressions such as Valence, Arousal, and Dominance. In this paper, we focus on emotion recognition using small amounts of speech data, especially the "categorical emotion task" that classifies emotions. At present, the accuracy of emotion classification for 7 or 8 emotions is only 70 or 80%, even when the emotions are uttered intentionally [20][21][35][36][37][38]. Therefore, the performance improvement is desired.

For emotion recognition, both short and long-term features are important. Therefore, many efforts have been made to improve the performance of emotion identification by incorporating features of various temporal resolutions. Conventional emotion recognition methods tried to improve the performance by using a single neural network encompassing multiple temporal resolutions using small speech database [39][40][41].

We consider that the performance of emotion identification methods using high-level statistical functions (HSFs), which show high accuracy in emotion identification, can be improved by extracting and combining HSFs from windows with multiple temporal resolutions instead of a single fixed window length [38][42]. In this paper, we aim to improve the accuracy by extending the HSFs extracted from a single fixed window in the existing methods to HSFs generated from multiple windows with temporal resolutions of 30 or more. In addition, to reduce the number of parameters to be learned simultaneously for a small amount of data, stacking with Gradient Boosting Decision Trees (GBDT) is applied when combining features of multiple temporal resolutions [43][44]. As a result, the highest emotion classification performance was obtained for the American emotional speech database. In addition, although the method initially uses multiple temporal resolutions of more than 30, it is found that the same classification performance can be obtained with only 15 temporal resolution features based on analyzing by GBDT.

1.6 Contribution

The following is a list of problems and contributions for each task.

Overall

- Target: Realize real-time paralinguistic and nonverbal information extraction from speech signal towards empathetic dialogue systems.
- Contribution: Improve the performance of speaker attribute recognition, phoneme recognition, acoustic event detection and recognition, and acoustic emotion recognition in the aspects of recognition performance and real-time decoding.

Chapter 2 Speaker Attribute Recognition

- Problem: Separately consider speech detection and speaker attributes. Calculation cost is high, and no real-time processing.
- Contribution: Propose simultaneous speech detection and speaker attribute recognition method. It leads to low cost and real-time calculation.

Chapter 3 Phoneme Recognition

- Problem: Need a phoneme clustering tree that is highly dependent on each language. Therefore, the training cost is high.
- Contribution: Propose a new phoneme discriminator without a phoneme clustering tree.

Chapter 4 Acoustic Event Detection and Recognition

- Problem: Need word-dictionary with filler and word fragment and need second-pass decoding.
- Contribution: Propose simultaneous speech recognition and detection of filler and word fragments. It leads to real-time decoding without a specific word dictionary.

Chapter 5 Acoustic Emotion Recognition

- Problem: The classification performance of emotion is low. There is no real-time decoding for it.
- Contribution: Propose multiple discriminators with multiple temporal resolution features. The classification performance is improved using a method that can have a possibility for real-time decoding.

1.7 Total System

Our target is to construct the detection and recognition modules that are smoothly connected to each other. Figure 1.1 shows the covered modules in this paper. Furthermore, each module should work in real-time, and the calculation cost should be low. We propose methods for speaker attribute recognition, speech recognition, phoneme recognition, acoustic event detection and emotion recognition. Combining all components, we can get closer to similar to real-time human communication.

Figure 1.2 shows the total system which we will construct. Each chapter explain the detail of the method.

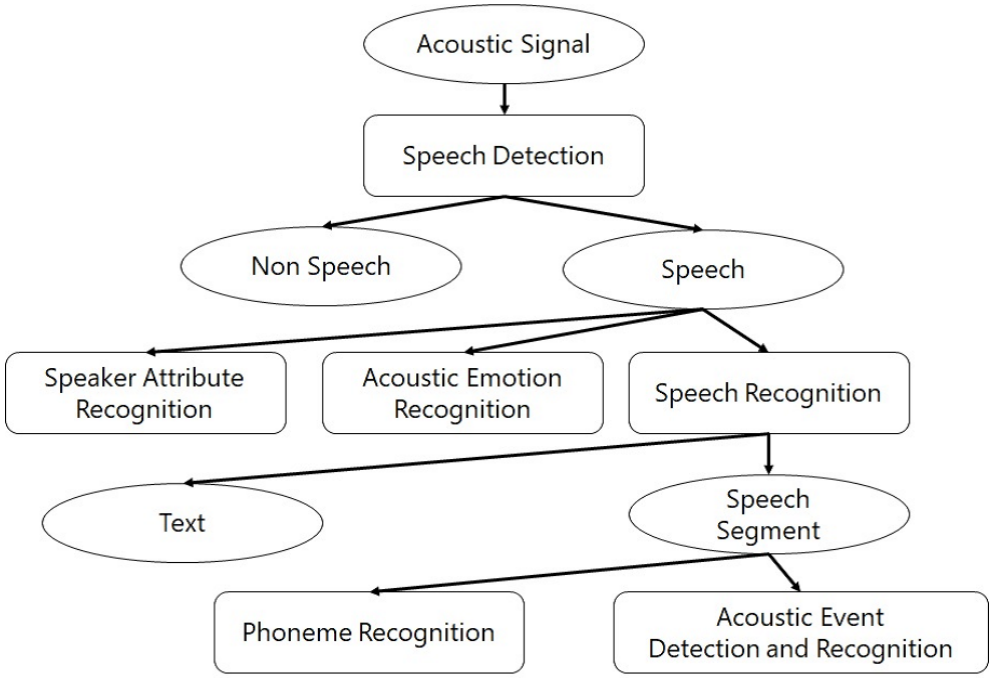


Figure 1.1: Position of our themes

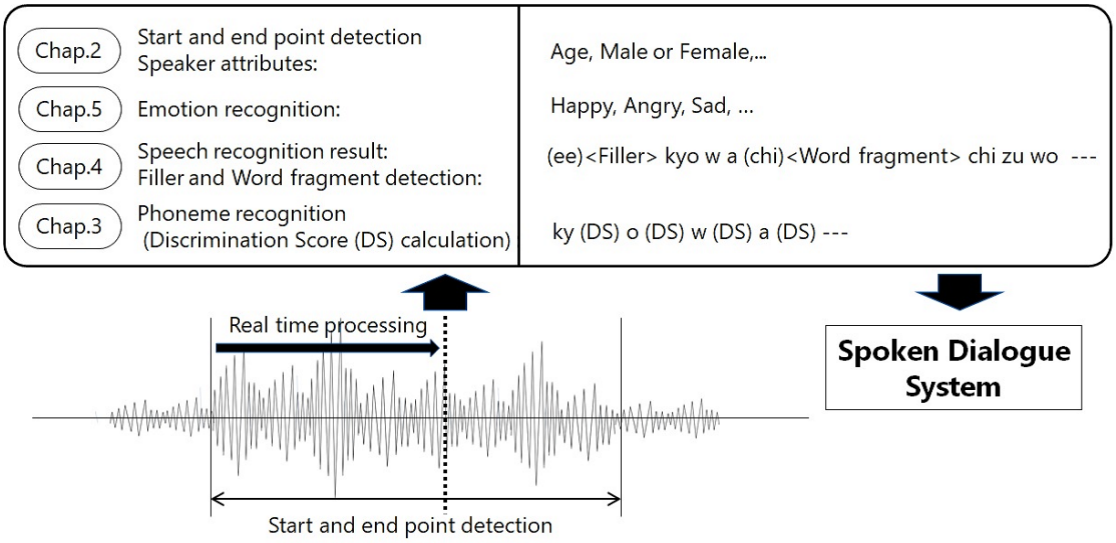


Figure 1.2: Total System

Chapter 2

Speaker Attribute Recognition

(Reference[1])

2.1 Introduction

Applications utilizing speech recognition systems have been released by many companies. In particular, the market of speech recognition applications has been rapidly growing for the last two or three years. It is important for the companies to release an accurate speech recognition system. However possibility for extending the market remains if we utilize other information in addition to the results of the speech recognition from voice.

In this chapter, we consider the problem of estimating speaker attributes from an input voice. In particular, gender is one of the most important speaker attributes. Gender classification is a simple problem, however it can be used for a wide variety of applications, such as recommendation systems and trend analysis. For some applications such as trend analysis, it is not necessary to execute gender classification in real-time. However, for other applications such as recommendation systems, real-time gender classification is required. We focus on gender classification for the real-time applications.

Many studies are reported for gender classification. When we consider gender classification, we have to select input features, voice activity detection method, and a classifier to achieve it. Gaussian Mixture Models (GMMs) in combination with RASTA-PLP and pitch features achieve good performance for gender classification in noisy conditions [46]. However, this study assumes that voice activity detection is performed in advance. In [47], Artificial Neural Networks are applied to MFCC features extracted from voiced frames which are detected using amplitude. This method also considers speech recognition. Therefore, the same features as speech recognition are used for gender classification. In [48], a method of simultaneous gender classification and voice activity detection is proposed, where the same

[1]This dissertation is based on "Simultaneous gender classification and voice activity detection using deep neural networks" [45], by the same author, which appeared in the Proceedings of INTERSPEECH2014, Copyright(C)2014 ISCA. The material in this paper was presented in part at the Proceedings of INTERSPEECH2014 [45], and all the figures of this paper are reused from [45] under the permission of the ISCA.

GMMs are used for both of them. Male, female, and silence GMMs are trained using male, female, and silence signal sections, respectively. Likelihoods for each frame are calculated using male, female, and silence GMMs. A frame classification result is obtained by comparing the likelihoods. A frame is classified as speech if the likelihood of the male or female GMMs is larger than that of the silence GMMs. Speech sections are detected by Voice Activity Detection (VAD) process where the frame classification results are smoothed; if a silence section is shorter than a predefined max pause length, the section is merged to the preceding speech section. The total likelihoods of the male and female GMMs are calculated for each speech section, which may include silence sections shorter than the max pause length. This method can execute gender classification and VAD simultaneously. However, GMM is weaker than other classifiers such as Deep Neural Networks (DNNs) [7] or Support Vector Machine [49] for a classification problem in general. DNNs achieve high accuracy for not only speech recognition [7] but also gender classification. Nishimura et. al [27] classifies each frame into male, female, and silence by a comparison among the posterior probabilities using DNNs, and the final result is obtained by counting frame results. However this method cannot detect a voice activity section, hence it cannot classify a sentence into male and female speech sections when male and female speech are included in the sentence. For VAD algorithms, DNNs are also adopted recently, and obtain good performance [50][51]. DNNs achieve good accuracy for both gender classification and VAD. However the calculation cost becomes high if DNNs are calculated many times for real-time applications.

Simultaneous gender classification and VAD are proposed, but single frame-based GMMs are used in [48]. On the other hand, DNNs method for gender classification is proposed, but VAD is not considered in [27]. Furthermore, the posterior probabilities are only used for the comparison in a frame. We propose that results of a single frame-based DNNs classifier are used for both gender classification and VAD. Posterior probabilities of male, female, and silence classes are calculated for each frame by the DNNs. We achieve high accuracy based on these posterior probabilities for gender classification and VAD. The point is that speech posterior probability is calculated using male and female posterior probabilities for both of them.

The remainder of this chapter is organized as follows: Section 2.2 introduces the DNN gender classification. Section 2.3 introduces simultaneous gender classification and VAD method using DNNs. Section 2.4 shows experimental setups for the evaluations. Section 2.5 shows the results of the proposed simultaneous classification and VAD method. Conclusions are presented in section 2.6.

2.2 DNN Gender Classification

This section introduces our gender classification method using a frame-based DNNs classifier. To simplify problems, a two-class which consists of male and female problem is considered for an input signal (Fig. 2.1). The overview of gender classification is shown in Fig. 2.2. One of the key essences for gender classification is to detect frames which include information for the classification. The best way of a frame selection is to extract voiced frames that don't include

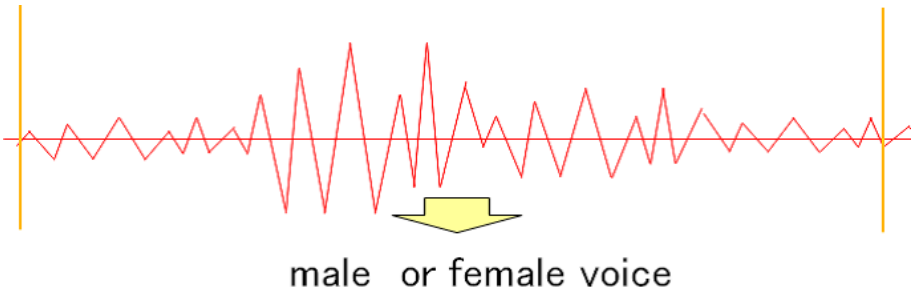


Figure 2.1: A two-class problem for an input signal; male or female classification

unvoiced and silent frames. However voiced/unvoiced classification is more difficult than speech/non-speech classification [52]. Therefore it is often replaced by speech/non-speech detection for each frame. We also adopt it for a pre-processing of our gender classification method.

For a frame-based classifier, three clusters which consist of male, female and silence are prepared. Our proposed method uses a DNNs classifier for each frame. DNNs are trained at the output layer to separate these three clusters for each frame in Fig. 2.3. In this case, the number of units is three at the output layer. Output value at the output layer expresses posterior probability using soft max processing [7]. Posterior probabilities for three clusters $(p_n^{(\mathrm{m})}, p_n^{(\mathrm{f})}, p_n^{(\mathrm{s})})$, where $p_n^{(\mathrm{m})}$ is the male posterior probability, $p_n^{(\mathrm{f})}$ is the female posterior probability and $p_n^{(\mathrm{s})}$ is the silence posterior probability, respectively , are calculated using the DNNs for the nth frame in total N frames of the input signal. Gender classification for the input signal is based on the results of the frame-based classification.

In [27], frame based classification is executed by a comparison among posterior probabilities of three clusters from DNNs. A cluster which has max posterior probability is selected as the output of the classification of the frame. The final result is obtained to count the number of each output clusters for all frames.

We propose a gender classification based on the sum of posterior probabilities for speech frames. On the other hand, the speech frames is detected by the sum of the male and female posterior probabilities (Fig. 2.3). The detection method is shown in the following algorithm. We assume that a bias of the number of each cluster is small enough to ignore it when DNNs are trained. $p_n^{(\mathrm{v})}$ and c_n denote the speech posterior probability and the estimated label (0:non-speech, 1:speech) of the nth frame.

$$c_n = \begin{cases} 0, & (p_n^{(\mathrm{v})} < p_n^{(\mathrm{s})}), \\ 1, & (p_n^{(\mathrm{v})} \geq p_n^{(\mathrm{s})}), \end{cases} \tag{2.1}$$

where

$$p_n^{(\mathrm{v})} = p_n^{(\mathrm{m})} + p_n^{(\mathrm{f})}, \tag{2.2}$$

$$p_n^{(\mathrm{m})} + p_n^{(\mathrm{f})} + p_n^{(\mathrm{s})} = 1. \tag{2.3}$$

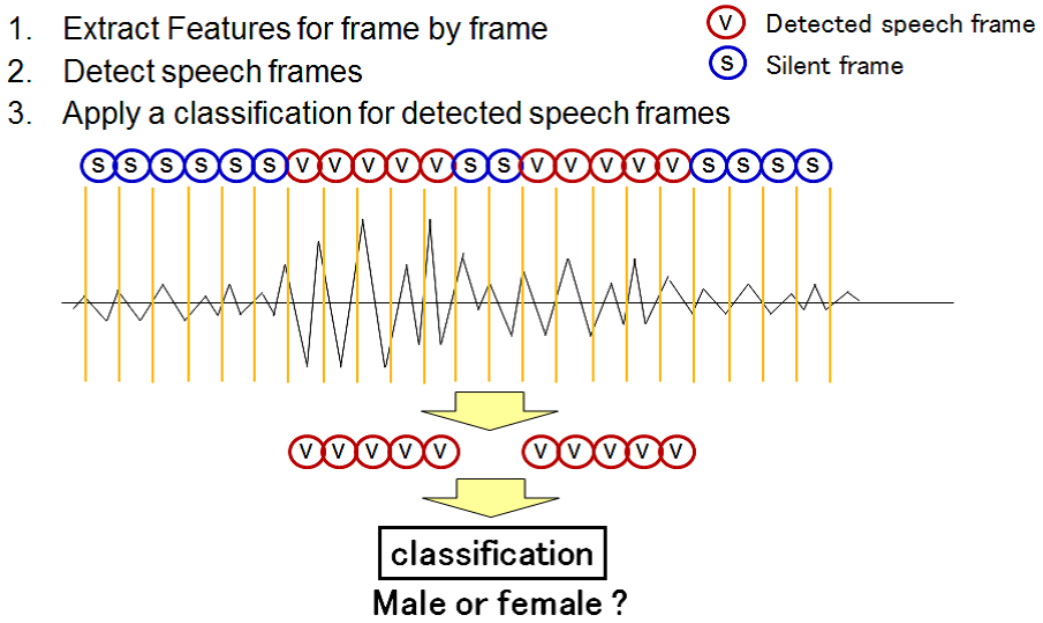


Figure 2.2: The overview of gender classification

The male and female log-posterior probabilities are calculated for the speech frames after detecting them, respectively. $P^{(\mathrm{m})}$ and $P^{(\mathrm{f})}$ denote the sum of male log-posterior probability and female log-posterior probability.

$$P^{(\mathrm{m})} = \sum_{n:c_n=1} \log p_n^{(\mathrm{m})}, \tag{2.4}$$

$$P^{(\mathrm{f})} = \sum_{n:c_n=1} \log p_n^{(\mathrm{f})}. \tag{2.5}$$

Finally, gender classification is executed by comparing $P^{(\mathrm{m})}$ and $P^{(\mathrm{f})}$. C denotes a class label (0:female, 1:male) for the input signal.

$$C = \begin{cases} 0, & (P^{(\mathrm{m})} < P^{(\mathrm{f})}), \\ 1, & (P^{(\mathrm{m})} \geq P^{(\mathrm{f})}). \end{cases} \tag{2.6}$$

2.3 Simultaneous VAD and gender classification

This section introduces simultaneous gender classification and VAD using frame-based DNNs classifier which is also used for gender classification. The gender classification algorithm is described in Section 2.2. However, the method can only classify an input signal of a given segment into male and female. We'd like to detect speech segments, and classify the speech segments into male and female, respectively (Fig. 2.4). It can be executed after other VAD

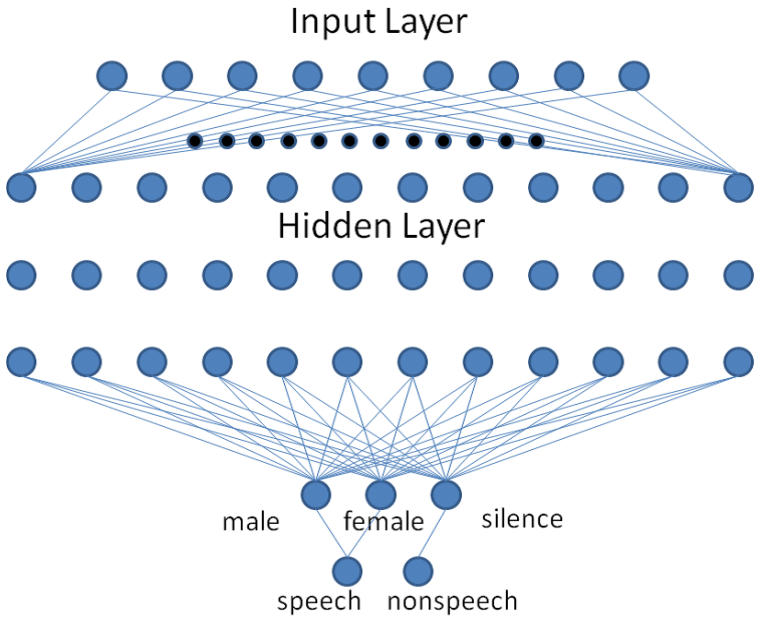


Figure 2.3: DNNs for gender classification

method is applied to an input signal. However the calculation cost becomes high because other feature extraction and other classifier are applied to the VAD method before the gender classification. Therefore, we propose a VAD method that uses frame-based DNNs classifier which is also used for gender classification. In the previous section, posterior probabilities for male $p_n^{(\mathrm{m})}$, female $p_n^{(\mathrm{f})}$, and silence $p_n^{(\mathrm{s})}$ are calculated for the nth frame to solve gender classification problems. The posterior probabilities of speech $p_n^{(\mathrm{v})}$ is also calculated by the sum of male and female probabilities for the nth frame. The speech probability $p_n^{(\mathrm{v})}$ is used for the proposed VAD. In general, a VAD process is executed using a threshold for speech likelihood of frames. In our method, speech likelihood for the nth frame is replaced by the speech probability $p_n^{(\mathrm{v})}$. Fig. 2.5 shows the waveform and the speech probability time series of the file "MNL_O79_961O_718_8_Z_1_3_734O_133.raw" in CENSREC-1C [53]. Speech or non-speech is decided by a threshold to the speech probability of each frame. After the thresholding, a simple finite-state automaton decides the start-of-speech and end-of-speech points. The automaton gives time constraint to the speech section.

2.4 Experimental Setup

2.4.1 Evaluation Data

Experiments were conducted using Japanese, English and Chinese speech databases for gender classification. Two databases are evaluated in Japanese. One is the news domain reading-script corpus. Another is the CSJ monologue corpus [54] which is a Japanese standard evaluation set

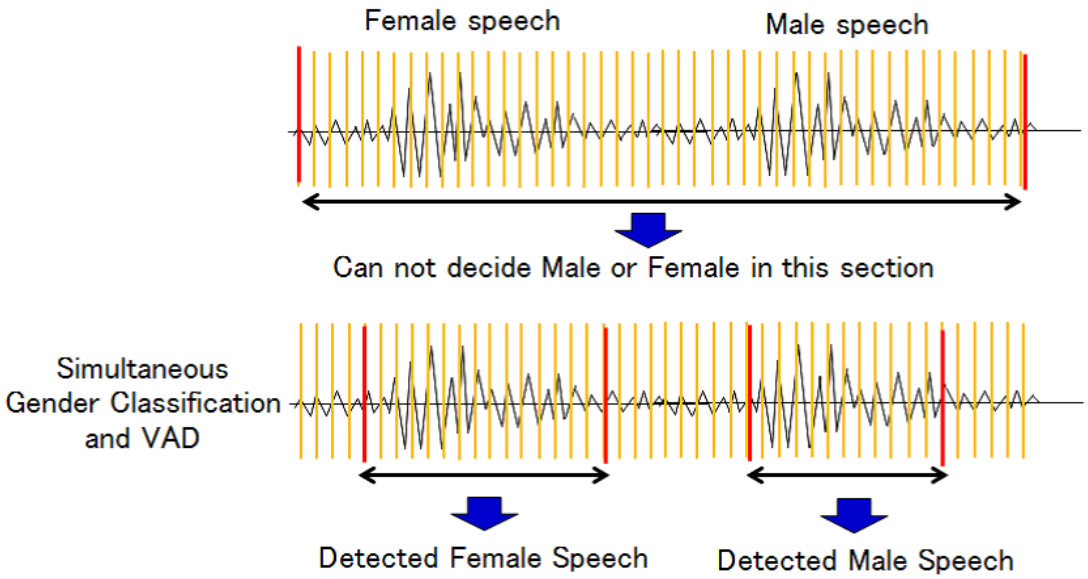


Figure 2.4: Simultaneous gender classification and VAD method

for the continuous speech recognition. CSJ testset3 was used for the evaluations. The travel domain corpus is used for English and Chinese gender classification. CENSREC-1C was used for both gender classification and VAD evaluation. CENSREC-1C [53] is the common VAD evaluation. Speech sampling rate was set to 8000 [kHz] for all evaluation databases.

2.4.2 Training Data

DNNs and GMMs were trained using ATR Japanese database [55]. This database includes utterances of phonetic balanced sentence and dialogue. Babble noises were added to the speeches by SNR 0-30 for VAD robustness.

2.4.3 DNNs Training

The basic feature was 12 MFCC, and normalized power. In addition to them, non-normalized Pitch feature and the Δ, $\Delta\Delta$ [56] were used. Hence, the total number of dimensions is 16. For 12-MFCC, moving window cepstral means normalization and ARMA [57] are applied. A simple spectral subtraction which subtracts a estimated noise spectrum from a target spectrum is also applied. The noise spectrum was estimated from the first 10 frames of a sentence for this training and evaluation. An input feature vector for DNNs was created by concatenating 5 neighboring frames which totaled 11 frames. Therefore, the number of dimensions is $16 \times 11 = 176$. The structure of hidden layers was $512[\text{unit}] \times 2[\text{layer}]$. A final layer of three units included male, female and silence clusters. DNNs were trained by backpropagation without pre-training. Alignment information was created by HMM-GMMs in advance.

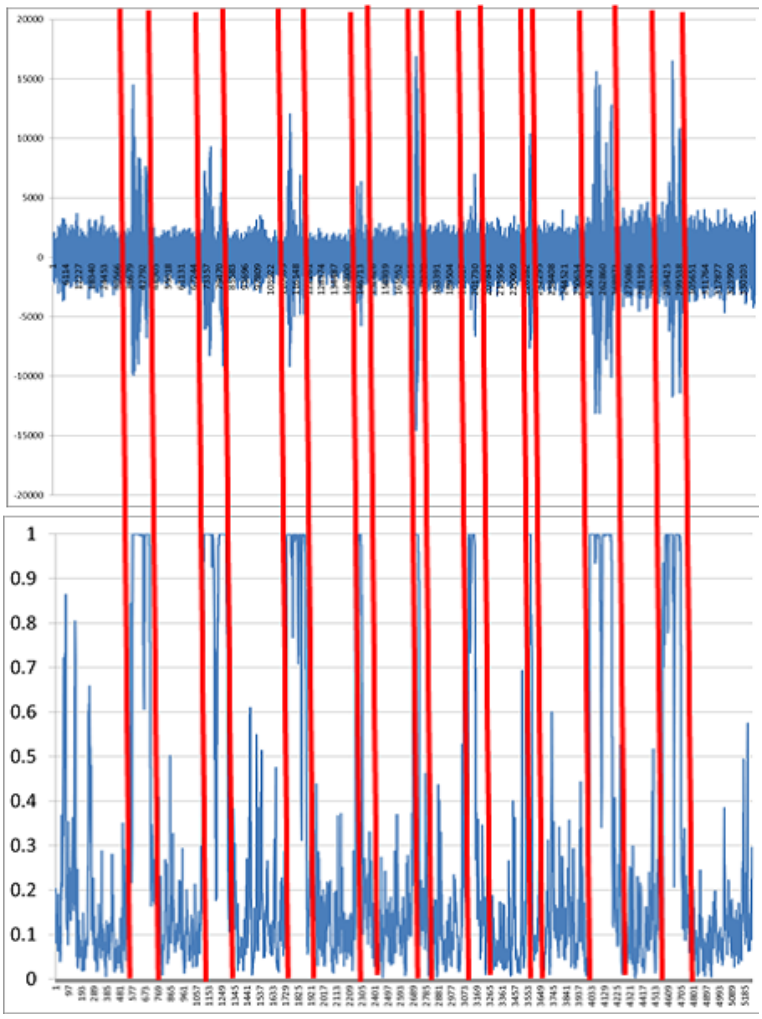


Figure 2.5: An input signal and the speech posterior probability

2.4.4 GMMs Training

GMMs were trained using the same database and feature as DNNs. However frames were not concatenated. The structure was 16 mixture models. Initial means were calculated by a LBG algorithm. Male, female and silence GMMs were trained using the same alignment information as DNNs.

Table 2.1: Training and Evaluation Data

DNN Training Data	ATR corpus 200 hours
#Evaluation utterances	Japanese News domain 2000 (16speakers) CSJ testset3 2484 (10speakers) Chinese Travel domain 4400 English Travel domain 1000 CENSREC-1C
Basic Feature	12-dimensional MFCCs + normalized power + Non-normalized Pitch + Δ + $\Delta\Delta$

2.5 Experiment

2.5.1 Gender Classification

The performance of gender classification is shown in this section. For the GMMs experiment, the likelihoods of male, female and silence GMMs were used for the detection of speech frames and gender classification in [48]. The algorithm is the following. $l_n^{(\mathrm{m})}$, $l_n^{(\mathrm{f})}$ and $l_n^{(\mathrm{s})}$ denote the likelihoods of male, female and silence GMMs for the nth frame. c_n denote the estimated label (0:non-speech, 1:speech) of the nth frame.

$$c_n = \begin{cases} 0, & (\max(l_n^{(\mathrm{m})}, l_n^{(\mathrm{f})}) < l_n^{(\mathrm{s})}), \\ 1, & (\max(l_n^{(\mathrm{m})}, l_n^{(\mathrm{f})}) \geq l_n^{(\mathrm{s})}). \end{cases} \tag{2.7}$$

The male and female likelihoods are calculated for the speech frames after detecting them, respectively. $L^{(\mathrm{m})}$ and $L^{(\mathrm{f})}$ denote the sum of male log-likelihood and female log-likelihood,respectively.

$$L^{(\mathrm{m})} = \sum_{n:c_n=1} \log l_n^{(\mathrm{m})}, \tag{2.8}$$

$$L^{(\mathrm{f})} = \sum_{n:c_n=1} \log l_n^{(\mathrm{f})}. \tag{2.9}$$

Finally, gender classification using GMMs is executed by comparing $L^{(\mathrm{m})}$ and $L^{(\mathrm{f})}$. C denotes a class label (0:female, 1:male) for the input signal.

$$C = \begin{cases} 0, & (L^{(\mathrm{m})} < L^{(\mathrm{f})}), \\ 1, & (L^{(\mathrm{m})} \geq L^{(\mathrm{f})}). \end{cases} \tag{2.10}$$

Tab. 3.2 shows the GMMs and DNNs performances for the evaluation data which is shown in Tab. 2.1. "GMM" means the results of GMMs described in this section. "DNN (count)" means the results of the method in [27]. "DNN (post-probability)" means the results of

Table 2.2: The performance comparison by classification rate[%]

	J News	CSJ	E travel	C travel
GMM	96.55	90.26	94.20	98.95
DNN (count)	100.00	95.65	97.50	99.82
DNN (post-probability)	100.00	96.01	97.70	99.86

our method. "J News" and "CSJ" shows the results for Japanese news corpus and Japanese monologue corpus (CSJ). "E travel" shows the result for English travel corpus. "C travel" shows the result for Chinese travel corpus. Apparently, DNN performance is better than GMM performance. In DNN methods, "DNN (post-probability)" is slightly better than "DNN (count)". For English and Chinese, DNN methods keep high performances for gender classification although the model is trained using Japanese databases.

2.5.2 VAD Experiment

VAD performance was evaluated using the CENSREC-1-C dataset. The DNNs for a classification of speech and non-speech for each frame was the same as gender classification. Tab. 2.3 shows the VAD accuracy. The performance is from [58]. Sohn's method is the VAD method based on Gaussian statistical model. PAR is a speech periodic to a periodic component ratio-based VAD [59]. SKF is switching Kalman Filter based VAD. PAR+SKF is a combination method of PAR and SKF [58]. "DNN" means the VAD method using DNNs which has not a male, female and silence but only a speech and non-speech output layer. The DNNs was trained using the same features, training databases, and structure in Sec. 4.4.1. "GMM gender" means the VAD method using the following GMMs log-likelihoods ratio. R_n denotes GMMs log-likelihood ratio for the nth frame.

$$R_n = \frac{\log(\max(l_n^{(\mathrm{m})}, l_n^{(\mathrm{f})}))}{\log(l_n^{(\mathrm{s})})}. \tag{2.11}$$

Speech likelihood for the nth frame is replaced by R_n. "Proposed" means our proposed method by DNNs. The proposed method cannot reach the SKF performance. However our approach just used MFCC and Pitch features with a classical spectral subtraction. Some adaptive noise reduction methods can be embedded into it. In addition, the models are trained using babble noises only in these experiments. "DNN" performance is little worse than "Proposed". We have to add some experiments for the result in the future.

2.5.3 Gender Classification for Detected Speech Sections

This section shows the performance of the gender classification for the correctly detected speech section in Sec. 2.5.2. The detected speech section is from our proposed method. Tab.

Table 2.3: The performance of VAD accuracy rate [%]

	Rest.Hi.	Rest.Lo.	St.Hi.	St.Lo.	Ave.
Baseline	21.45	-43.48	-15.65	-33.91	-17.90
Sohn	45.51	-6.38	94.49	57.39	47.75
PAR	24.35	-6.67	64.35	54.49	34.13
SKF	68.41	12.46	97.68	93.62	68.04
PAR+SKF	72.75	19.71	99.13	94.78	71.60
DNN	67.83	2.03	98.26	62.61	57.68
GMM gender	64.64	11.01	87.54	39.42	50.65
Proposed	75.65	21.45	95.94	49.86	60.73

Table 2.4: The performance of gender classification rate for detected speech sections [%]

	Rest.Hi.	Rest.Lo.	St.Hi.	St.Lo.	Ave.
#Detected	317	215	335	255	*
GMM	88.96	69.30	85.97	94.90	84.78
DNN (count)	94.64	82.33	97.31	98.82	93.28
DNN (post-probability)	97.48	86.98	97.61	98.43	95.13

2.4 shows the results of gender classification for the detected speech sections. The number of correctly detected sections in 345 speech sections is also shown. Apparently, DNNs results are also better than GMMs in 2.5.1. In DNNs methods, "DNN (post-probability)" is better than "DNN (count)".

2.6 Conclusions

We proposed simultaneous voice activity detection and gender classification method using single deep neural networks (DNNs). A frame based classifier of DNNs classifies each frame into male, female and silence clusters. These frame based results are used for both gender classification and VAD. This method is competitive compared to existing techniques for both gender classification and VAD. However the calculation cost is almost the same as a gender classification because the same DNNs classifier is applied to VAD.

2.7 Acknowledgements

The present study was conducted using a CENSREC-1 database and a CENSREC-1-C database developed by the IPSJ-SIG SLP Noisy Speech Recognition Evaluation Working

Group.

Chapter 3

Phoneme Recognition

(Reference[1])

3.1 Introduction

Many applications that exploit automatic speech recognition (ASR) demand better performance for large vocabulary ASR. The ASR system has to discriminate the target phonemes from others because only a few different phonemes are the clues to discriminate similar words. Therefore, we focus on phoneme discrimination in speech recognition.

The hidden Markov model (HMM) with Gaussian Mixture Model (GMM) is the principal technique for automatic speech recognition. Many efforts have been made to improve performance of phoneme discrimination for GMM-HMM. Some studies succeeded in the improvement of the accuracy by Minimum Phone Error (MPE) and Maximum Mutual Information (MMI) discriminative training for GMM-HMM [61] [62] [63]. These are based on GMM-HMM. On the other hand, some studies try to replace GMM by a discriminative model directly. Ganapathiraju *et al.* [29] combined HMM and the support vector machine (SVM) which is one of the effective discriminative models. Their technique applies SVMs to phoneme segments obtained from the alignment of HMM, and N-best hypotheses are rescored by scores of SVMs. Padrell-Sendra *et al.* [30] applied SVM to the frame-level discrimination, and obtained better performance than a standard GMM-HMM. SVMs are constructed for three classes per monophone. Ragni and Gales [64] exploited generative model and discriminative model with structured discriminative models. Recently HMM with Deep Neural Network (DNN-HMM) showed remarkable improvement to GMM-HMM [7]. We also proposed AdaBoost rescoring for the second pass of GMM-HMM [65], which improved the isolated word recognition performance for the discrimination of monophones.

For speech recognition, the feature of a monophone changes depending on the contexts.

[1]This dissertation is based on "N-best rescoring by phoneme classifiers using subclass adaboost algorithm" [60], by the same author, which appeared in the Proceedings of INTERSPEECH2013, Copyright(C)2013 ISCA. The material in this paper was presented in part at the Proceedings of INTERSPEECH2013 [60], and all the figures of this paper are reused from [60] under the permission of the ISCA.

Hence, a model is constructed for the triphone unit. However many speech recognition systems use tied-state triphones which are tied by the context clustering as subclasses because the number of triphones is large, and the samples for model training become sparse. Therefore, the context clustering for triphones is the key essence for phone modeling. For SVM phone modeling [29] [66], SVMs are constructed for monophones. DNN-HMMs [7] are modeled for tied-state triphones which are constructed by tree clustering with Maximum Likelihood criterion. However the tied-state triphones from tree clustering are not optimized for the discriminator. In some studies, discriminative criterion is used for creating the tied-state triphones. Wiesler *et al.* [67] achieved 10% relative improvement by tree clustering with the minimization of the classification error. Sim [68] proposed probabilistic state clustering with Conditional Random Field for HMM with GMM. These studies create tied-state triphones with the discriminative criteria. They can be applied to DNN-HMM and other discriminative methods. However they are not optimized for phoneme discriminators themselves. Therefore, it is desired that the tied-state triphones are directly optimized for phoneme discriminators.

In our method, this is achieved by AdaBoost [69] [70]. In image recognition field, some ideas of subclass AdaBoost [71] [72] [73] are proposed. In these studies, subclass AdaBoost is applied after the estimation of subclasses for input vectors. Hence, they are not the techniques which exploit subclass label such as triphone contexts which are given without estimation. In many speech recognition systems, subclass attribution, which is the key of split, is given such as phoneme contexts in advance. In our method, weak classifiers of AdaBoost are designed for exploiting subclass labels such as phoneme contexts. The training samples of AdaBoost is split into subclasses by the phoneme contexts, and the best discriminators are selected for the subclasses for each weak classifier training step of AdaBoost. It is iterated for all prepared subclass-splits. In this step, the subclasses and the discriminators which have the least error for training samples are selected as the weak classifier. The AdaBoost which is constructed by such weak classifiers is jointly optimized for subclasses and the discriminative criterion, implicitly.

As a first try of subclass AdaBoost application, classification scores are calculated using subclass AdaBoost classifiers for the phoneme segments of N-best hypotheses given by HMMs, and the N-best hypotheses are re-ordered based on the classification scores. The subclass AdaBoost classifies whether the phoneme of a segment is correct or not as a binary classifier.

The remainder of this chapter is organized as follows: Section 3.2 introduces the proposed subclass AdaBoost algorithm. Section 3.3 introduces the proposed rescoring method by AdaBoost. Section 3.4 shows experimental setups and results of the proposed rescoring method. Conclusions are presented in section 3.5.

3.2 Subclass AdaBoost

3.2.1 Algorithm

The overview of the subclass AdaBoost is shown in Fig.3.1. The difference between conventional and proposed AdaBoost is mainly data split at each weak classifier training. Each

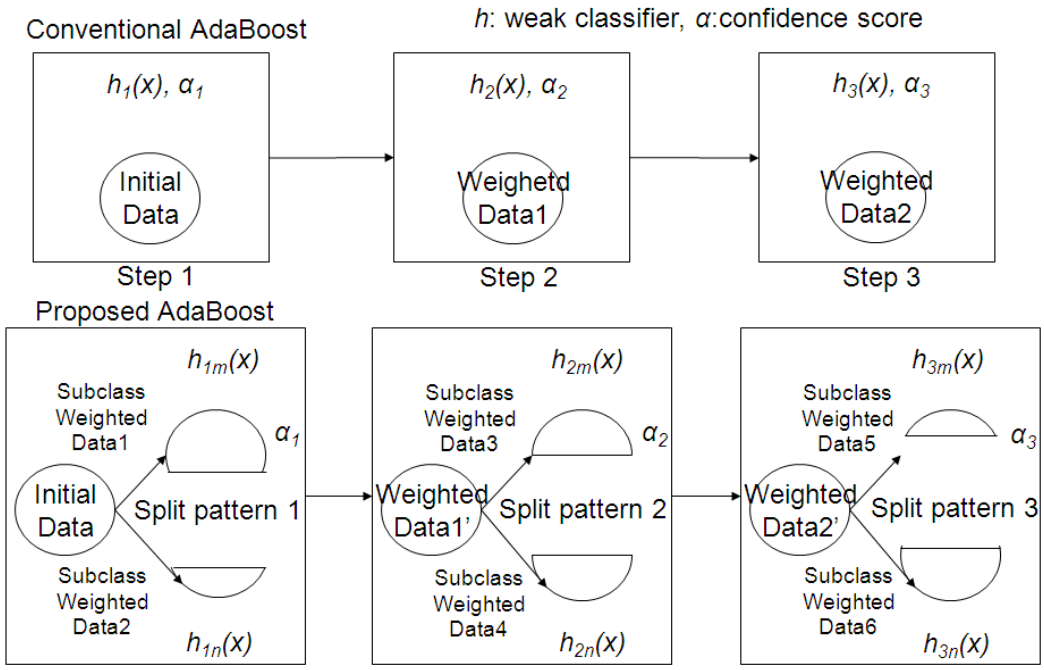


Figure 3.1: The overview of the proposed AdaBoost training

weak classifier is optimized by data split and the classification error. For the purpose of the data split, some rules are prepared in advance before the subclass AdaBoost training. In the speech recognition field, the rule is defined by the triphone context as the conventional tree clustering uses it.

Assume that there are N training samples $(x_1, y_1), ..., (x_N, y_N)$, where x_i and y_i denote the feature vector and the class label of sample i. Normal AdaBoost classifiers are trained by the following steps:

- Step 1. Initialize sample weight distribution $D_0(i)$ by the following equation:

$$D_0(i) = \begin{cases} \dfrac{1}{2\sum_{j:y_j=1} 1}, & y_i = 1, \\ \dfrac{1}{2\sum_{j:y_j=-1} 1}, & y_i = -1. \end{cases} \tag{3.1}$$

- Step 2. Train a weak classifier $h_t(x_i)$ minimizing error rate ε_t on sample weight distribution D_t,

$$\varepsilon_t = \sum_{i:y_i \neq h_t(x_i)} D_t(i). \tag{3.2}$$

- Step 3. Compute voting weight α_t as

$$\alpha_t = \frac{1}{2} \log \frac{1 - \varepsilon_t}{\varepsilon_t}. \tag{3.3}$$

- Step 4. Recompute sample weight distribution as

$$\hat{D}_{t+1}(i) = D_t(i) \exp(-\alpha_t y_i h_t(x_i)). \tag{3.4}$$

- Step 5. Normalize the summation of sample weights to 1

$$D_{t+1}(i) = \frac{\hat{D}_{t+1}(i)}{Z_{t+1}}, \tag{3.5}$$

  where

$$Z_{t+1} = \sum_{i=1}^{N} \hat{D}_{t+1}(i). \tag{3.6}$$

- Step 6. Iterate from step 2 to step 5 T times, and obtain T weak classifiers.

  Finally the strong classifier $H(x)$ is obtained by weighted sum of T weak classifiers as

$$H(x) = \text{sign} \left\{ \sum_{t=1}^{T} \alpha_t h_t(x) \right\}, \tag{3.7}$$

  where each weak classifier outputs 1 (true) or -1 (false) by comparing a threshold and values of a selected dimension of the segment feature vectors.

In our proposed method, Step 2 is different from normal AdaBoost training step. First some classification rules are prepared in advance. Let K denote the classification rules. Samples in D_t are divided into subclasses by a rule $k \in K$. Let M_k denote the subclasses constructed by the split rule k. The best weak classifier $h_k^{(m)}$ (m: class label for subclasses M_k) is detected for each subclass m. The total error ε_k is calculated by sum of subclass errors. The minimum total error $\varepsilon_{k_{min}}$ is found in all $k \in K$. Finally, the subclasses $M_t = M_{k_{min}}$ and the best classifiers $h_t^{(m)}$ are restored as they minimize the total error. The new Step2 is the following:

- Step 2'-1. For all $k (\in K)$, the best weak classifiers $h_k^{(m)}$ for subclasses M_k split by the rule k are detected.

$$\varepsilon_t^{(k)} = \sum_{m \in M_k} \sum_{\substack{i: z_i = m \\ y_i \neq h_k^{(m)}(x_i),}} D_t(i), \tag{3.8}$$

  where z_i is the label of ith sample.

Algorithm 1 The training step of the t-th iteration

There are 3 binary questions for an example:
Binary question 1 :previous phoneme is "a"; $true$ or $false$,
Binary question 2 :previous phoneme is "i"; $true$ or $false$,
Binary question 3 :previous phoneme is "u"; $true$ or $false$.

for k = 1 to 3 **do**
- Apply "Binary question k" to samples.
- Samples divided into subclass "k:true" and "k:false".
- Find the best discriminators $h_k^{(true)}$ and $h_k^{(false)}$
to minimize $\varepsilon_t^{(k)}$,
$\varepsilon_t^{(k)} = \sum_{\substack{i:z_i=true \\ y_i \neq h_k^{(true)}(x_i),}} D_t(i) + \sum_{\substack{i:z_i=false \\ y_i \neq h_k^{(false)}(x_i),}} D_t(i).$

end for
- Find $k_{min} (\in \{1, 2, 3\})$ to minimize $\varepsilon_t^{(k)}$.
- Calculate α_t using $\varepsilon_t^{(k_{min})}$
- Calculate the distribution D_{t+1} using $\varepsilon_t^{(k_{min})}$.
- Restore k_{min}, α_t, $h_{k_{min}}^{(true)}$, $h_{k_{min}}^{(false)}$ as the weak classifier parameters of this step.
- Go to the $(t + 1)$-th iteration.

- Step 2'-2. Find k to minimize the $\varepsilon_t^{(k)}$.

$$k_{min} = \arg\min_k \varepsilon_t^{(k)} \tag{3.9}$$

- Step 2'-3. Decide $\varepsilon_t = \varepsilon_t^{(k_{min})}$, $M_t = M_{k_{min}}$ and $h_t^{(m)} = h_{k_{min}}^{(m)}$, where $m \in M_t$.

For the application of the speech recognition, the split rule is the triphone context which is "a-*" or not for example. The simplest way is to split by binary questions which are usually applied to tree clustering by ML estimation. The binary questions are prepared in advance, and all questions are applied to data split at each weak classifier training. The best weak classifiers in an iteration are decided by total errors after all patterns by the questions are tried. A sample of the algorithm is shown in Algorithm1. To do classification using a subclass AdaBoost classifier, given an input vector and left and right contexts, for each weak classifier, the input vector is assigned to either of the two subclasses by the binary question, and the discriminator correspoiding to the subclass is applied to the input vector.

3.2.2 Loss Function

AdaBoost Algorithm is equivalent to a forward stagewise additive modeling algorithm that minimize a exponential loss. This section explain about the loss function of the subclass AdaBoost.

$T-1$ linear combinations $F(x)$ of a weak classifier $\alpha_t h_t(x)$ is

$$F(x) = \sum_{t=1}^{T-1} \alpha_t h_t(x). \tag{3.10}$$

The loss function of the $F(x)$ is defined as the following

$$L(F) = \frac{1}{N} \sum_{i=1}^{N} \exp(-y_i F(x_i)), \tag{3.11}$$

where the Tth weak classifier $h_T(x)$ is added to $F(x)$ after multiplying the weight $\alpha_T (> 0)$

$$F(x) + \alpha_T h_T(x). \tag{3.12}$$

In the case of subclass AdaBoost, $h_T(x)$ is the sum of $h^{(m)}(x)$, where $m \in M_T$,

$$h_T(x) = \sum_{m} h^{(m)}(x). \tag{3.13}$$

The loss function $L(F + \alpha_T h_T)$ is minimized. This loss function can be re-written by

$$
\begin{aligned}
& L(F + \alpha_T h_T) \\
&= \frac{1}{N} \sum_{t=1}^{N} \exp(-y_i(F(x_i) + \alpha_T h_T(x_i))) \\
&= \frac{1}{N} \sum_{t=1}^{N} \exp(-y_i(F(x_i)) + \alpha_t \sum_{m_t} h_t^{(m_t)}(x_i)) \\
&= \frac{1}{N} \sum_{t=1}^{N} \exp(-y_i F(x_i)) \exp(\alpha_t \sum_{m_t} h_t^{(m_t)}(x_i)) \\
&= \frac{1}{N} \sum_{m_t} \sum_{\substack{i:z_i=m_t \\ y_i=h_t^{(m_t)}(x_i),}} \exp(-y_i F(x_i)) \exp(-\alpha_t) \\
&+ \frac{1}{N} \sum_{m_t} \sum_{\substack{i:z_i=m_t \\ y_i\neq h_t^{(m_t)}(x_i),}} \exp(-y_i F(x_i)) \exp(\alpha_t) \\
&= \frac{\exp(-\alpha_t)}{N} \sum_{m_t} \sum_{i:z_i=m_t} \exp(-y_i F(x_i)) \\
&- \frac{\exp(-\alpha_t)}{N} \sum_{m_t} \sum_{\substack{i:z_i=m_t \\ y_i\neq h_t^{(m_t)}(x_i),}} \exp(-y_i F(x_i)) \\
&+ \frac{\exp(\alpha_t)}{N} \sum_{m_t} \sum_{\substack{i:z_i=m_t \\ y_i\neq h_t^{(m_t)}(x_i),}} \exp(-y_i F(x_i)) \\
&= \frac{\exp(-\alpha_T)}{N} \sum_{i=1}^{N} \exp(-y_i F(x_i)) \\
&+ \frac{\exp(\alpha_T) - \exp(-\alpha_T)}{N} \sum_{m} \sum_{\substack{i:z_i=m \\ y_i\neq h_T^{(m)}(x_i),}} \exp(-y_i F(x_i)).
\end{aligned}
\tag{3.14}
$$

h_T only depends on the second term.

$$
D_T(i) = \frac{\frac{1}{N} \exp(-y_i F(x_i))}{Z_T}, \tag{3.15}
$$

$$
Z_T = \frac{1}{N} \sum_{i=1}^{N} \exp(-y_i F(x_i)), \tag{3.16}
$$

The second term is expressed using the error rate ε_T on the distribution D_T

$$\varepsilon_T = \sum_{i:y_i \neq h_T(x_i)} D_T(i) = \sum_m \sum_{\substack{i:z_i=m \\ y_i \neq h^{(m)}(x_i),}} D_T(i) \tag{3.17}$$

$$\begin{aligned} &\frac{\exp(\alpha_T) - \exp(-\alpha_T)}{N} \sum_m \sum_{\substack{i:z_i=m \\ y_i \neq h^{(m)}(x_i),}} \exp(-y_i F(x_i)) \\ &= (\exp(\alpha_T) - \exp(-\alpha_T)) Z_T \varepsilon_T. \end{aligned} \tag{3.18}$$

After h_T is decided, the loss function is

$$\begin{aligned} &L(F + \alpha_T h_T) \\ &= Z_T(\exp(-\alpha_T) + \exp(\alpha_T) - \exp(-\alpha_T))\varepsilon_T). \end{aligned} \tag{3.19}$$

The α_T minimizing the loss function is derived by equaling the derivation of it 0

$$\alpha_T = \frac{1}{2} \log \frac{1 - \varepsilon_T}{\varepsilon_T}. \tag{3.20}$$

If $\varepsilon_t < 0.5$, the loss function is

$$L(F + \alpha_T h_T) = L(F) \cdot 2\sqrt{(1-\varepsilon)\varepsilon} < L(F). \tag{3.21}$$

It can reduce the value of the loss function. The relation between ε' from the normal AdaBoost and ε from the subclass AdaBoost is always $\varepsilon \leq \varepsilon'$. Finally the loss function is always smaller because of

$$\begin{aligned} &L(F + \alpha_T h_T) \\ &= L(F) \cdot 2\sqrt{(1-\varepsilon)\varepsilon} \leq L(F) \cdot 2\sqrt{(1-\varepsilon')\varepsilon'} < L(F). \end{aligned} \tag{3.22}$$

3.3 Rescoring Using AdaBoost Classifiers

3.3.1 Overview of the Rescoring Process

The subclass AdaBoost can be applied to many applications. An example is shown for applying the subclass AdaBoost to a speech recognition system [65]. This section describes the proposed method to rescore the N-best hypotheses using subclass AdaBoost phoneme classifiers as [74]. Our method consists of the first pass and the second pass. The first pass outputs N-best hypotheses with scores by a conventional speech recognition system using HMM. The process of the second pass is shown in Fig.3.2. First, phoneme segments are obtained by forced alignment of phoneme HMMs for each hypothesis. Then, segment features are extracted from each phoneme segment, and classification scores are calculated by subclass AdaBoost classifiers using the segment features. For the subclass AdaBoost, an input label has left and right contexts. In Fig.3.2, the input labels of the hypothesis word "nijou" are "sil-n+i, n-i+j, i-j+o, j-o+u, o-u+sil", where "sil" means silence phoneme. Finally, classification scores are added to the N-best scores and N-best hypotheses are re-ordered.

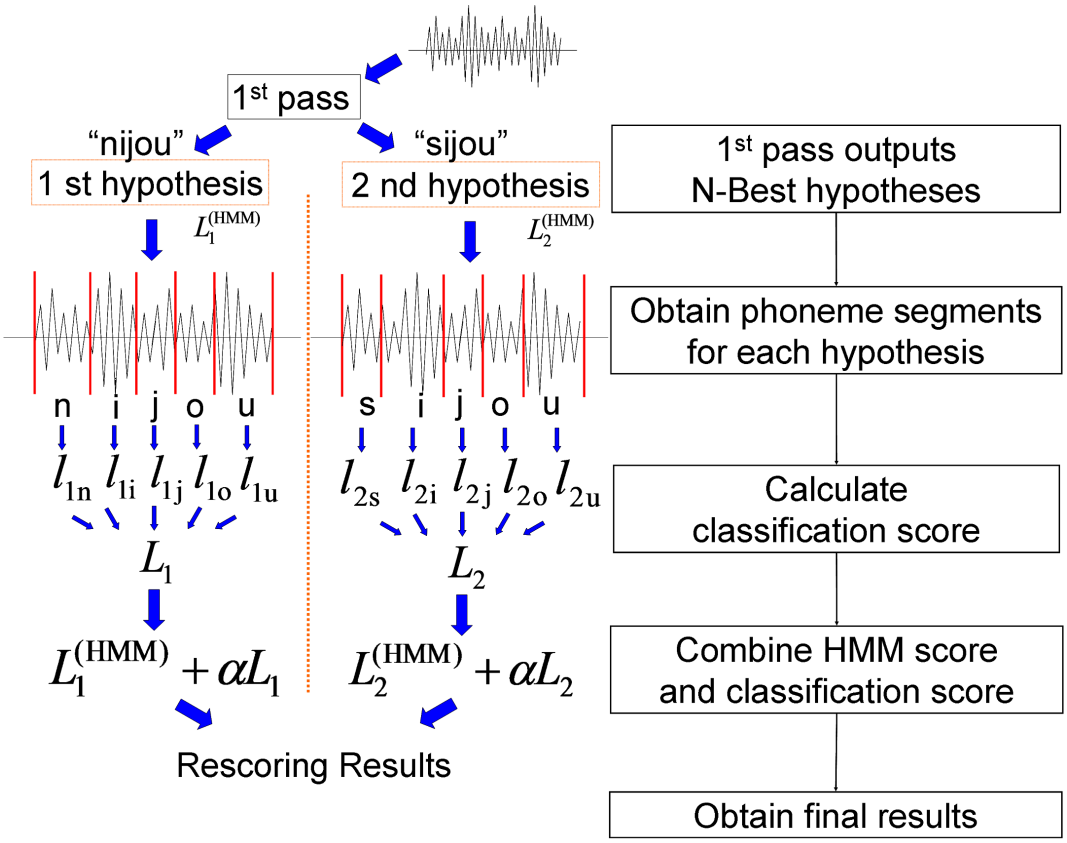


Figure 3.2: The flowchart of the second pass

3.3.2 Segment Feature

Although lengths of phoneme segments are variable, fixed-length features are desirable for most classifiers. Hence, fixed-length features are extracted from phoneme segments with variable length. Alignment information of HMM states is exploited for this purpose. Figure 3.3 shows the segment feature used for AdaBoost. Assume without loss of generality that phoneme HMMs are three-state left-to-right models and frame-based features are extracted in advance. In this report, LPC-spectrum and MFCCs are extracted as frame-based features in advance. First, the phoneme segment for time and LPC-spectrum plane is normalized to average 0 and variance 1. Then, mean and variance are calculated in each Mel scale block of the phone segment. The segment is extended by one frame for the start and end of the state alignment because frames of calculating variance are allocated. Second, mean and variance are calculated in each Mel scale block of each state segment. Third, mean is calculated at each dimension in the phone segment for MFCCs. Forth, mean is calculated at each dimension in each state segment for MFCCs. Finally, all vectors from LPC-spectrum and MFCCs are concatenated to form a fixed-length vector expressing the phoneme segment information.

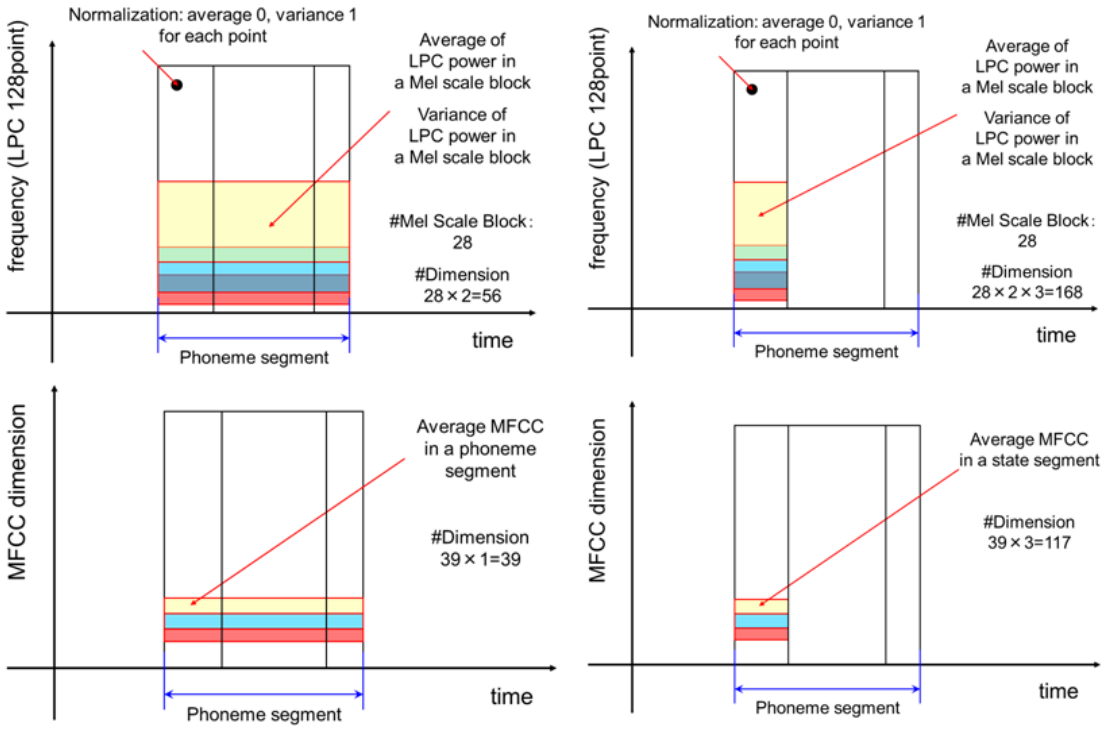


Figure 3.3: Fixed length feature extraction from variable length segment

3.3.3 Classification Score

Phoneme classifier is constructed for each phoneme by an AdaBoost algorithm. The process of the calculation of classification score is explained using the conventional AdaBoost case. Each phoneme classifier discriminates whether the given segment is the phoneme or not, and outputs the AdaBoost score for the segment. As described in [75], the AdaBoost score $S_p(x)$ of a phoneme p for segment feature x is derived from the following equation,

$$S_p(x) = \frac{1}{\sum_{t=1}^{T} \alpha_t^{(p)}} \sum_{t=1}^{T} \alpha_t^{(p)} h_t^{(p)}(x). \tag{3.23}$$

Note that the range of $S_p(x)$ is $-1 \leq S_p(x) \leq 1$ due to the normalization. Classification scores are calculated from the AdaBoost scores. First the classification score is calculated at each phoneme segment in all N-best hypotheses. Classification score l_i of i-th phoneme segment in a hypothesis is obtained by the following equation:

$$l_i = \log \frac{S_{ip_i} + 1}{\sum_{k=1}^{|P|} (S_{ik} + 1)}, \tag{3.24}$$

where p_i denote the phoneme of the i-th segment in the hypothesis, and S_{ik} and $S_i p_i$ are AdaBoost scores for phoneme k of i-th segment and for phoneme p_i of i-th segment, respec-

Table 3.1: Training and Evaluation Data

Language	Japanese
Acoustic Model Training Data	CSJ corpus 500 hours
Language Model Training Data	CSJ corpus and Travel domain text are mixed
#Evaluation utterances	Travel domain 3200 (16speakers) CSJ testset3 2484 (10speakers)
Evaluation Task	30 thousands continuous words recognition
Basic Feature	12-dimensional MFCCs + normalized power, their Δs and ΔΔs

tively. In order to guarantee the normalized score is positive, a constant 1 is added to S_{ik} and $S_i p_i$. Note that the score l_i can be considered as a logarithm of a posteriori probability based on AdaBoost scores.

The classification score L for a hypothesis is obtained by averaging l_i in the hypothesis,

$$L = \frac{1}{K} \sum_{i=1}^{K} l_i, \tag{3.25}$$

where K is the number of phonemes in this hypothesis. Finally, the rescoring score L_{re} is obtained by weighted sum of L and HMM score $L^{(\mathrm{HMM})}$ of the hypothesis,

$$L_{\mathrm{re}} = L^{(\mathrm{HMM})} + \alpha L, \tag{3.26}$$

and the N-best hypotheses are re-ordered using the score L_{re}.

3.4 Experiments

3.4.1 Experimental Setup

Evaluation Data

Experiments were conducted using Japanese speech database. Two databases are evaluated. One is the travel domain corpus which includes utterances by 21 speakers. Each speaker utters 200 sentences about the travel domain. The data set is divided into a development set of 5 speakers and an evaluation set of 16 speakers. Another is the Corpus of Spontaneous Japanese (CSJ) [54] which is a Japanese standard evaluation set for the continuous speech recognition.

HMM training

The HMMs were trained by 500-hour CSJ speech data which doesn't include test set data. The basic structure of the HMM is three-state continuous density triphones that share 3000 states with 32 Gaussian mixture components. All triphones have a simple left-to-right topology. A feature vector for the HMM consisted of 12 MFCC (Mel-frequency cepstrum coefficient), a normalized power, and their Δs and $\Delta\Delta$s (totally 39 dimensions). HTK [76] was used for the HMM training. HLDA (heteroscedastic linear discriminant analysis) [77] without nuisance dimensions and MPE (minimum phone error) training [61] were applied to the HMM after the MLE training.

AdaBoost Training

Basically, the training process of the subclass AdaBoost is the same as the conventional AdaBoost. AdaBoost was trained using the same data as HMMs'. The feature for AdaBoost were obtained by the method described in Sec.3.3.2 based on forced alignment of HMMs. LPC-spectrum is calculated from 17th order LPC. MFCCs are the same as the HMM training. Consequently, there were 380 dimensions for AdaBoost features. AdaBoost classifier were constructed for each phoneme, and each AdaBoost classifier discriminates if a given segment is the phoneme or not. The weak classifier was constructed by a threshold decision for a dimension [70]. The tied-state triphone models were used for forced alignment to obtain phoneme segments of the training data with the correct phoneme labels. Error phoneme segments were extracted from the lattices which were created in HMM-MPE process. The positive labels are attached to the target phoneme labels of the correct phoneme labels, and the negative labels are attached to the other phoneme labels of the correct phoneme labels and error phoneme segments from the lattices. The number of phonemes excluding silence was 38 in the experiments. For Subclass AdaBoost, the data at each weak classifier was split by binary questions for triphone contexts. The number of weak classifier steps was 1200 for the conventional AdaBoost and 600 for the subclass AdaBoost, respectively.

Language Model Training

The standard 3-gram language model was constructed using CSJ text corpus and travel domain text corpus. The text corpus for evaluation test was not included in the training. The portion of travel domain text is the same as CSJ's.

3.4.2 Phoneme Classification Experiment

A phoneme classification experiment was conducted to confirm the classification performance for the subclass AdaBoost. First, the evaluation utterances were aligned by the tied-state triphone HMMs with the correct labels. Then for each segment, scores in Eq.(3.23) were calculated using AdaBoost models, and the phoneme label with the highest score was output as a classification result. For the subclass AdaBoost, correct left and right contexts were given as an input label. Table 3.2 shows the results of conventional AdaBoost and subclass AdaBoost.

Table 3.2: The classification performance by classification error rate [%]

	Conventional AdaBoost	Subclass AdaBoost
CSJ tesetset3	17.46	8.12

Table 3.3: The performance of rescoring methods by Word Error Rate [%]

	HMM	Rescore(conventional)	Rescore(subclass)
Travel	14.00	13.02	12.42
CSJ testset3	19.08	18.76	18.70

It was evaluated using CSJ testset3. It shows that the performance of the subclass AdaBoost is highly better than the conventional AdaBoost. The error reduction rate is 53.49%.

3.4.3 Rescoring Experiment

Performance of the proposed rescoring technique was evaluated on the continuous word recognition tasks. In this experiment, the first pass output 1000-best hypotheses, and they were rescored by the AdaBoost scores in the classification scores in Eq.(3.25) in the second pass. Table 3.3 shows the performance of the rescoring. "HMM" denotes results of the first pass, "Rescore(conventional)" denotes results of rescoring by the classification scores calculated from conventional AdaBoost scores. "Rescore(subclass)" denotes results from subclass AdaBoost scores. Language model weight and coefficient α in Eq.(3.26) are adjusted to maximize the performance on the development set for the travel corpus. For the CSJ corpus, they are adjusted to maximize the performance on the evaluation set itself. Accordingly, the rescoring results are better than the 1st pass results for both of tasks. For the travel corpus, the word error reduction rate was 11.29% by subclass AdaBoost, which is better than the performance of the conventional AdaBoost. However the performance of the rescoring for CSJ is not improved much. Furthermore, the difference of performance between conventional and subclass AdaBoosts is very small although the difference of classification performance is large. CSJ is the spontaneous speech database, which includes a lot of ambiguous speech utterances. The rescoring method by phoneme segments may not match the solution of the spontaneous speech recognition. This is the future work for us.

3.5 Conclusions

We proposed a novel technique to exploit discriminative models with subclasses for speech recognition. Our method is subclass AdaBoost which does not need splits by any clustering method in advance. Subclass AdaBoost can automatically optimize the discriminator with

triphone contexts. This method can be applied to many applications. As a first try, a phoneme classifier is constructed for each phoneme by the subclass AdaBoost algorithm. Each phoneme classifier discriminates whether the given segment is the phoneme or other phonemes, and outputs the AdaBoost score for the segment as the classification score. In the classification task, the performance of the subclass AdaBoost is highly better than the conventional AdaBoost. The error reduction rate is 53.49%. Furthermore, experimental results showed that the proposed technique consistently improved the continuous word recognition performance. The word error reduction rate was max 11.29% , when the number of weak classifiers was 600 using the proposed subclass AdaBoost.

Chapter 4

Acoustic Event Detection and Recognition

(Reference[1])

4.1 Introduction

Spontaneous speech often includes a lot of fillers and word fragments such as "um", "ah", and "thi-this". They not only lead significant performance deterioration of speech recognition [78], but also decrease readability of real-time captioning for human. To solve these problems, speech recognition systems need to detect and selectively remove fillers and word fragments. Since many applications of spontaneous speech recognition, such as speech-to-speech translation and speech dialogue systems, require real-time computation, it is necessary to detect fillers and word fragments with very low latency.

Some techniques have been proposed [79, 80, 34] to detect word fragments using sub-word language models and confusion networks without modeling acoustic aspects of word fragments. However, since fillers and word fragments have distinct acoustic features, it is better to exploit such acoustic features for filler and word fragment detection. In [33], fillers are independently modeled using acoustic-prosodic features, and the model is applied to calculation of filler scores of filler hypotheses during second-pass decoding, which prevents real-time decoding. In [81], word fragments are directly modeled by adding a fragment tag to phonemes composing word fragments, and phoneme models with the fragment tag are used as garbage models. We also proposed a technique to model filler and word fragment symbols in addition to phonetic symbols as outputs of an LSTM-CTC acoustic model [82]. In this technique, fillers and word fragments were registered to a lexicon and filler and word fragment symbols were added to their pronunciation. However, it is impracticable to register all possible word fragments to the lexicon. To overcome this problem, in this chapter, we propose a new decoding technique that can detect fillers and word fragments using a conventional lexicon.

[1]This dissertation is based on "Simultaneous Speech Recognition and Acoustic Event Detection Using an LSTM-CTC Acoustic Model and a WFST Decoder" [1], by the same author, which appeared in the Proceedings of ICASSP2018, Copyright(C)2018 IEEE. The material in this paper was presented in part at the Proceedings of ICASSP2018 [1], and all the figures of this paper are reused from [1] under the permission of the IEEE.

End-to-end training of an acoustic model and a language model has recently been proposed [83, 84, 85]. Many methods use a bi-directional LSTM model and an attention mechanism. However, it is not suitable for real-time applications because the standard attention mechanism requires whole input speech. In addition, some methods use both end-to-end models and conventional language models by means of a conventional decoder [83, 86]. Thus, WFST decoder [87] is still important for an efficient decoding. Our proposed decoder discriminates fillers and word fragments from normal words using a WFST that can accept filler and word fragment symbols. For fillers, the WFST is generated from a conventional lexicon and a language model, and the decoder determines fillers by checking symbols on paths of a lattice. In addition, the decoder detects word fragments using paths for word fragments added to a WFST of a lexicon.

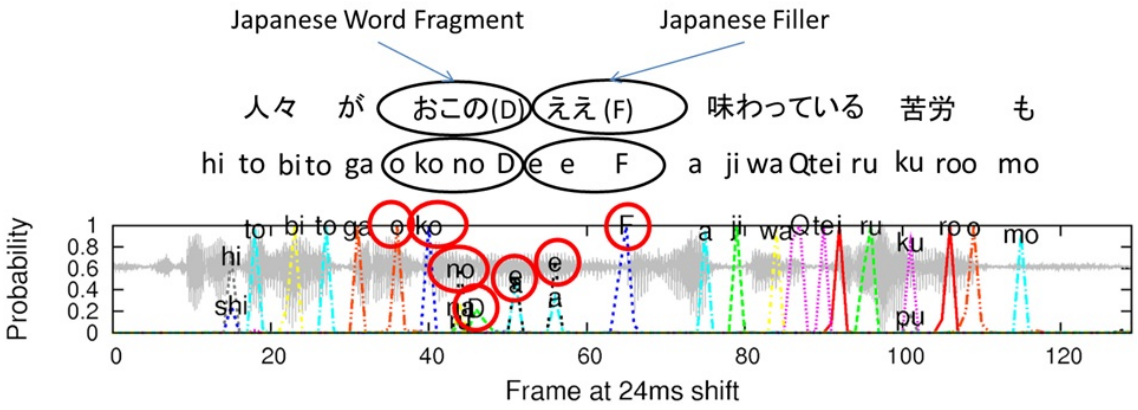
4.2 Acoustic Model

In this section, we introduce an acoustic model developed in [82]. In order to model and discriminate phonetic units and acoustic events simultaneously, we used Long Short-Term Memory trained by Connectionist Temporal Classification (LSTM-CTC) criterion [88]. In an end-to-end approach [89], grammatical events such as "apostrophe" and "space" symbols are used as an output symbols of the model in addition to graphemes. On the other hand, in our approach, a filler symbol and a word fragment symbol are added to the output symbols of the model. Since LSTM can model long-term dependency between input and output sequences and CTC allows to train a model using input and output sequences of different lengths without alignments, we only need to insert filler symbols and word fragment symbols at appropriate positions in output sequences to model these acoustic events. Table 4.1 shows an example of output sequences with a filler and a word fragment. In this table, <F> and <D> represent filler and word fragment symbols, respectively.

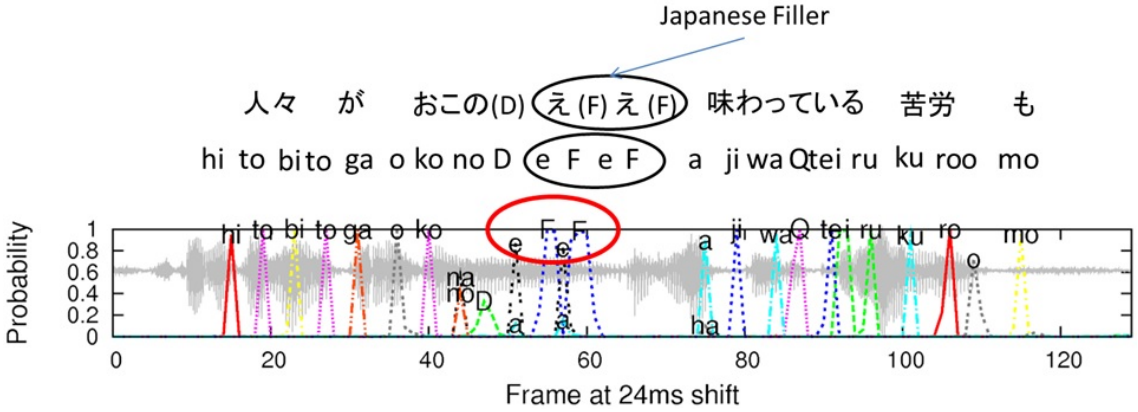
Figure 4.1 shows output probability sequences for Japanese morae and filler and word fragment symbols calculated by LSTM-CTC acoustic models. Figure 4.1(a) shows an output probability sequence using a label in which acoustic event symbols were inserted at the end of fillers and word fragments as described in [82]. On the other hand, in Figure 4.1(b), filler symbols are inserted after each mora in fillers. In this case, it is possible to calculate confidence scores for filler detection by counting the number of filler symbols in word hypotheses as described in the next section. It is noted that the confidence scores cannot be calculated for word fragments because it is thought that initial parts of word fragments do not have distinct features compared to that of normal words, and thus we insert word fragment symbols only at the end of word fragments.

4.3 Decoder

We adopt a WFST decoder with an LSTM-CTC acoustic model [86]. The decoder uses a WFST composed of R, L and G as a decoding graph, where R is a WFST squashing a label



(a) Acoustic event symbols are inserted at the end of fillers and word fragments.



it (b) Filler symbols are inserted after each mora in fillers.

Figure 4.1: Probability sequences of phonetic and acoustic event symbols (Copyright(C)2018 IEEE, [1])

sequence of the acoustic model in a CTC manner, L is a WFST of a lexicon and G is a WFST of a language model. In the proposed method, a filler symbol <F> does not appear in L unlike [82] in order to recognize arbitrary words as filler. For recognizing word fragments, The decoder should accept arbitrary phonetic symbol sequences terminated by a word fragment symbol <D>. This can be achieved by extending a mechanism of dynamic words which are not in L [87, 90, 91].

4.3.1 Filler

In order to detect fillers, transitions accepting the filler symbol are added to the WFST R. The filler symbol <F> appears only on the input side of the transitions as well as a blank. By checking input symbols on the transitions of word hypotheses, it is possible to detect words as fillers. Figure 4.2 shows an example of the WFST R.

Here we introduce a filler confidence score $c = f/p$, where f and p are the numbers of detected filler symbols and phonetic symbols in a word, respectively. The decoder outputs a

Table 4.1: Output sequences with fillers and word fragments (Copyright(C)2018 IEEE, [1])

Transcription	Thi-this is ah a pen
Phonetic symbols	di di s Iz aa a pe N
+filler and word fragment symbols	di <D> di s Iz aa <F> a pe N

word as a filler only when c is higher than a predetermined threshold.

4.3.2 Word Fragment

In order to detect word fragments without registering them to the lexicon, a WFST L should accept arbitrary phonetic symbol sequences terminated by <D>. Figure 4.3 shows an example of L including transitions to recognize arbitrary phonetic symbol sequences. In this example, ε is an empty symbol, a set of phonetic symbols is P ={k, b, ɑː, ʌɪ}, and a set of normal words is {car, bike}. $\#_{\mathrm{D}}$, $\#_{\mathrm{n}}$ and $\#_{\mathrm{p}}$ are auxiliary symbols to help determinization. w_{D} is a symbol to help determining where the word fragments appear in G. w_{n} has the same role as w_{D} but for dynamic words. State 2 has self-transitions, each of which has a phonetic symbol in P on the input side for accepting arbitrary phonetic symbol sequences. Word fragments and dynamic words share these self-transitions.

In order to calculate probabilities of dynamic words and word fragments, the WFST G has to handle w_{n} and w_{D}. In the WFST G, w_{n} is treated as a normal word, and the probability of w_{n} is calculated in the same manner as normal words. This can be achieved by assigning probability of a class of words, such as unknown words or words with the same part-of-speech, to w_{n}. On the other hand, it is difficult to calculate probabilities of the word fragments because of the difficulty of collecting a text corpus including word fragments occurring naturally. Thus, G should accept w_{D} in any context by adding a self-transition with w_{D} to states in G. To prevent excessive enlargement of $L \circ G$, we limit adding the self-transition with w_{D} to states having an outgoing transition with w_{n}.

The construction steps for L and G is modified as

$$RLG = \pi_{\varepsilon}(opt(R \circ opt(\underset{i \to o}{proj}(L \circ G)))),$$

where $\circ$ represents composition, *opt* represents determinization and minimization, π_{ε} replaces auxiliary symbols with ε [92], and $proj_{i \to o}$ projects an input symbol to an output symbol for each transition in the cyclic transition of L for phonemes. Additionally, an output symbol on a transition with input symbol $\#_{\mathrm{p}}$ is replaced by w_{n} for dynamic words and w_{D} for word fragments, respectively, after $proj_{i \to o}$ for decoding efficiency.

Word fragments are recognized through a WFST D by sharing the mechanism of dynamic words without using outputs of RLG directly. The decoder refers $RLGD = RLG \circ D$ during decoding. This composition operation is performed on-the-fly. The WFST D has three

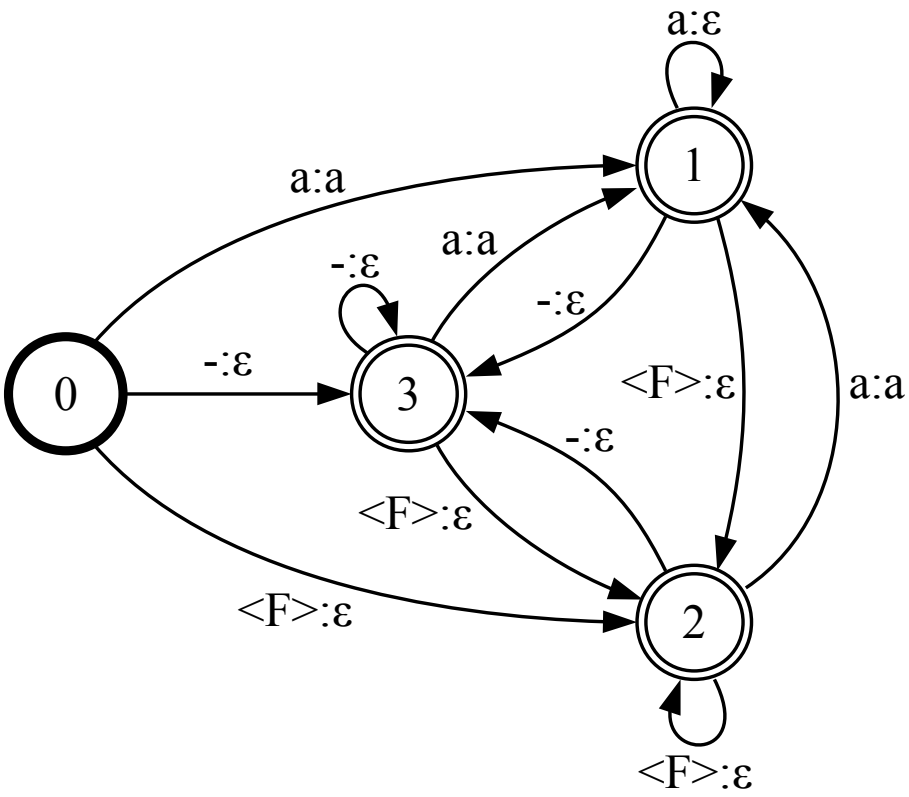


Figure 4.2: WFST R with transitions accepting a filler symbol <F>. (Copyright(C)2018 IEEE, [1])

functions: (1) convert a phoneme sequences to dynamic words, (2) transfer normal words without any conversion, (3) convert phonetic symbol sequences to w_D which is referred in post-processing. Figure 4.4 shows an example of the WFST D. In this example, a dynamic word 'kibe' is recognized by a path $0 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 0$, a self-transition with *:* at state 0 realizes the function (2), and a path $0 \rightarrow 1 \rightarrow 0$ realizes the function (3).

The post-processing collects phonetic symbol sequences of word fragments by the following steps:

1. Find w_D on the output side of a searched path.
2. Find word boundaries of w_D.
3. Retrieve an input symbol sequence on the path specified by the word boundaries.
4. Squash the input symbol sequence in the CTC manner.
5. Get a character sequence from the squashed input symbol sequence such as a phonetic symbol sequence.

In the case of recognizing Japanese language using morae, a squashed input symbol sequence can be easily converted to a kana character sequence because they can be mapped one-to-one.

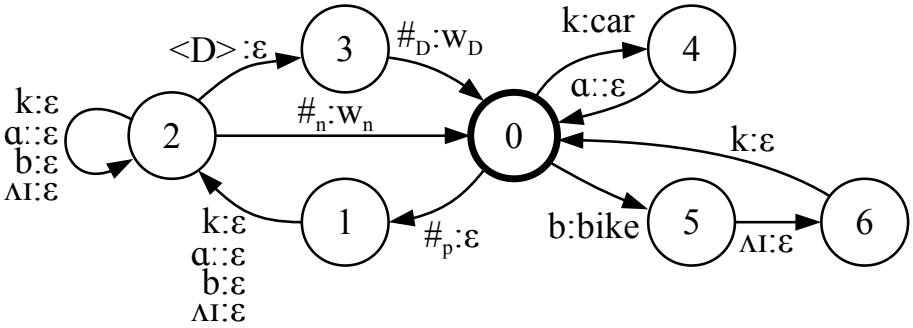


Figure 4.3: WFST L with phoneme loops. (Copyright(C)2018 IEEE, [1])

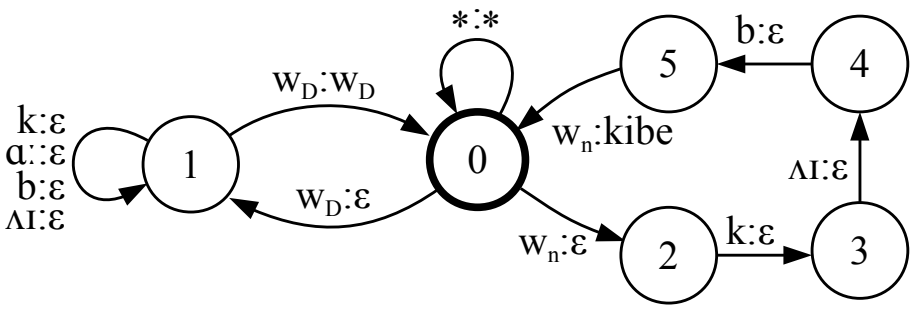


Figure 4.4: WFST D. (Copyright(C)2018 IEEE, [1])

In order to prevent word fragments from dominating recognized words, a penalty is added to the self-transition for word fragments (e.g., at state 1 in Figure 4.4). This penalty should be determined empirically as described in the next section.

4.4 Experiments

4.4.1 Experimental Setup

Evaluation Data

Evaluation experiments were conducted using an in-house Japanese corpus and the Corpus of Spontaneous Japanese (CSJ) [54]. The in-house Japanese corpus is a liaison-meeting evaluation set that consists of half an hour of speech by a speaker. The CSJ is a standard set for an evaluation of spontaneous Japanese speech recognition. We used CSJ testset3 that includes 2,484 monologue utterances by 10 speakers. For both of the evaluation sets, the utterances were recorded by close-talking microphones. The number of fillers and the number of word fragments in each evaluation set are shown in Table 4.2. The evaluation metrics are precision and recall rates for detection performance, and character error rate (CER) [%] for speech recognition performance, which is calculated by $CER = 100 - 100 \times (C - I)/N$, where C is the number of correct characters, I is the number of insertion characters, and N is the total number of characters in the reference.

Table 4.2: The number of fillers and word fragments (Copyright(C)2018 IEEE, [1])

	#Filler	#Word fragment
CSJ testset3	785	169
Liaison-meeting	238	36

Table 4.3: Acoustic models (Copyright(C)2018 IEEE, [1])

Acoustic model	Training label for "こ，ええと この (ko, eeto kono)"
Normal model (NAM)	ko e e to ko no
Filler + word fragment detection model (FDAM1)	ko <D> e e to <F> ko no
Filler + word fragment detection model (FDAM2)	ko <D> e <F> e <F> to <F> ko no

Table 4.4: Lexicons for the conventional decoder (Copyright(C)2018 IEEE, [1])

Lexicon	Word and the pronunciation
Normal lexicon (NLex)	kono: ko no, eeto: e e to
Normal lexicon with word fragments (NLex+D)	kono: ko no, ko: ko, eeto: e e to
Lexicon with filler and word fragment symbols 1 (FDLex1)	kono: ko no, ko: ko <D>, eeto: e e to <F>
Lexicon with filler and word fragment symbols 2 (FDLex2)	kono: ko no, ko: ko <D>, eeto: e <F> e <F> to <F>

Acoustic Model

In the experiments, uni-directional LSTM-CTC models were trained on the complete CSJ training set (about 580 hours) for real-time decoding. The CSJ training set has filler and word fragment tags in transcriptions. The basic feature was a 28-dimensional Mel-filterbank output. The input feature vector for the LSTM-CTCs was created by concatenating current and consecutive 8 previous frames, giving 9 frames in each input feature vector. Hidden layers were composed of 3 LSTM layers with 1024-dimensional memory cells and recurrent projections [93] that reduce the number of dimensions from 1024 to 256 for each layer output. A final layer of 126 units was set for Japanese mora outputs including a blank and a silence outputs. For filler and word fragment outputs, 2 units were added to the final layer. For all experiments, we created three acoustic models listed in Table 4.3. The main differences among them were the training labels. For fillers, two training methods were tested as shown in Figure 4.1. One was to insert filler symbols at the end of fillers as shown in Figure 4.1 (a), and the other was to insert filler symbols after each mora in fillers as shown in Figure 4.1 (b).

Language Model and Lexicon

A 4-gram language model was trained on 200 million sentences extracted from web pages and Toshiba internal spontaneous dataset. It is noted that the CSJ dataset was not included in the training set.

Table 4.4 shows four types of lexicons and sample words included in the lexicons. In this table, the normal lexicon was used for the proposed decoder, and other lexicons are used for the conventional decoder. The vocabulary size was about 200,000 words. The lexicons included fillers extracted from the spontaneous dataset. Word fragments extracted from the CSJ dataset were added to the lexicons for the conventional decoder, while word fragments were not added to the normal lexicon for the proposed decoder. In order to combine LSTM-CTC acoustic models having filler and word fragment outputs with conventional decoder, filler and word fragment symbols were added to pronunciations of words in the lexicons FDLex1 and FDLex2.

WFST

We prepared 5 WFSTs based on the combinations of acoustic models and lexicons. For each WFST, we selected a conventional decoder or the proposed decoder for detection of fillers and word fragments. Table 4.5 shows the combinations and selected methods. "NWT" is the WFST using the conventional language model, the conventional lexicon with word fragments, and the conventional decoder. "FDWT1" and "FDWT2" use lexicons including filler and word fragment symbols for the decoding. The main contributions of this chapter are "FDWT3" and "FDWT4" that use the conventional language model, the conventional lexicon and the proposed decoder. We used a Toshiba original decoder based on [90] for all experiments.

4.4.2 Speech Recognition Performance

Table 4.6 shows the recognition performance of each WFST. Basically, CERs are similar among the WFSTs. However, the CERs of FDWT1-4 are slightly better than NWT's. It seems that the explicit modeling of fillers and word fragments results in better performance.

4.4.3 Detection Performance

We defined a tolerance for filler and word fragment detection based on logistic OR and logistic AND sections between reference and detected sections depicted in Figure 4.5. The tolerance is calculated by $tolerance = (l - c)/c$, where l and c denote the length of the OR section and the length of the AND section, respectively. The value becomes 0 when a detected section is identical to a corresponding reference section. In this experiment, detection of fillers and word fragments were judged to be correct if the tolerance was less than a threshold value. We set the threshold value to 0.7 for filler detection and 0.9 for word fragment detection. Generally, uni-directional LSTM-CTC modeling leads to the delay of output of phonetic symbols. Hence, the positions of the detections are uniformally shifted back compared to the

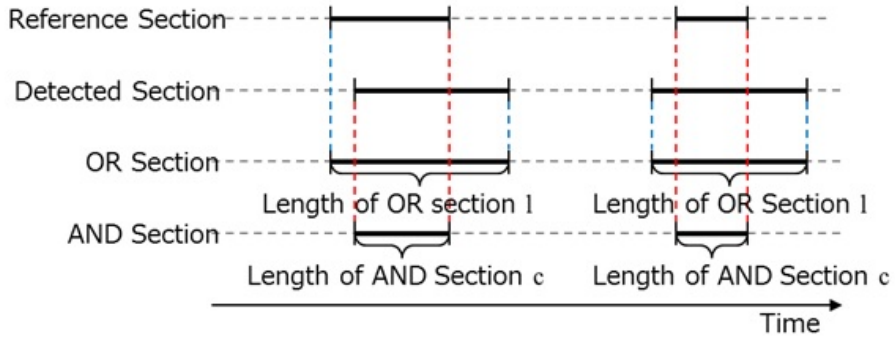


Figure 4.5: The metric for calculating the tolerance (Copyright(C)2018 IEEE, [1])

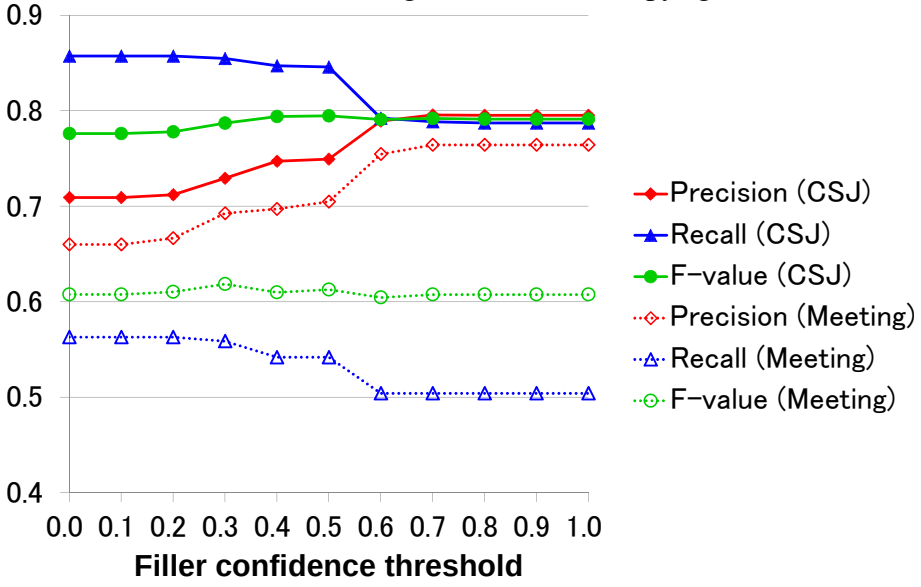


Figure 4.6: Filler confidence and detection performance using FDWT4 (Copyright(C)2018 IEEE, [1])

real positions of the acoustic events on the time axis. We set the time offset to 300 ms to compensate for the difference.

Figure 4.6 shows the relation between the filler confidence threshold and the filler detection performance in terms of the precision and recall rates and the F-value. Table 4.7 and 4.8 show the performance of the filler and word fragment detection of each WFST. The value of filler confidence was set to 0.3 in Table 4.7. The penalty to the self-transition for word fragments was set to maximize the ASR performance. For the filler detection, it can be shown that the performance of FDWT1-4 was better than that of NWT. However, the performance of FDWT3 and FDWT4 was similar to that of FDWT1 and FDWT2. This is because the filler can likely be covered with a conventional language model and lexicon (NWT), and the effectiveness of the acoustic assist for the filler detection is the same for all methods. The advantage of FDWT4 is that it can control the precision and recall rates by the filler confidence score depending on the purposes of applications. On the other hand, for word fragment detection, the performance

of FDWT3 and FDWT4 was apparently better than the performance of FDWT1 and FDWT2. The conventional decoder cannot detect word fragments even if word fragments extracted from the CSJ dataset is added to the lexicon. These results show that our approach is effective for detection of filler and word fragment using conventional language models and lexicons.

4.5 Conclusion

We proposed a technique to detect fillers and word fragments using an LSTM acoustic model and a WFST decoder. The LSTM acoustic model outputs filler and word fragment symbols as well as phonetic symbols, and the proposed decoder can simultaneously recognize speech and detect fillers and word fragments using conventional language models and lexicons without registering all possible word fragments. We showed that in word fragment detection the proposed decoder outperformed a conventional decoder using lexicons including word fragments.

Table 4.5: Created WFSTs ("WF": Word Fragment) (Copyright(C)2018 IEEE, [1])

WFST name	Acoustic model	Lexicon	Decoder
Normal (NWT)	NAM	NLex+D	Conventional
Filler & WF (FDWT1)	FDAM1	FDLex1	Conventional
Filler & WF (FDWT2)	FDAM2	FDLex2	Conventional
Filler & WF (FDWT3)	FDAM1	NLex	Proposed
Filler & WF (FDWT4)	FDAM2	NLex	Proposed

Table 4.6: ASR performance for each WFST (CER [%]) (Copyright(C)2018 IEEE, [1])

WFST name	CSJ testset3	Liaison-meeting	Ave.
NWT	10.34	15.36	12.85
FDWT1	10.35	14.75	12.55
FDWT2	9.83	14.65	12.24
FDWT3	10.53	14.82	12.68
FDWT4	10.16	14.99	12.57

Table 4.7: Filler Detection performance (F:F-value) (Copyright(C)2018 IEEE, [1])

WFST	CSJ testset3			Liaison-meeting		
	Precision	Recall	F	Precision	Recal	F
NWT	0.77	0.61	0.68	0.79	0.45	0.57
FDWT1	0.80	0.77	0.78	0.87	0.57	0.69
FDWT2	0.78	0.81	0.79	0.78	0.58	0.66
FDWT3	0.75	0.85	0.79	0.72	0.58	0.64
FDWT4	0.75	0.85	0.79	0.70	0.54	0.61

Table 4.8: Word Fragment Detection performance (F:F-value) (Copyright(C)2018 IEEE, [1])

WFST	CSJ testset3			Liaison-meeting		
	Precision	Recall	F	Precision	Recall	F
NWT	0.22	0.04	0.07	0.00	0.00	0.00
FDWT1	0.36	0.02	0.04	1.00	0.03	0.05
FDWT2	0.42	0.03	0.06	1.00	0.06	0.11
FDWT3	0.53	0.25	0.34	0.86	0.33	0.48
FDWT4	0.49	0.25	0.34	0.61	0.31	0.41

Chapter 5

Acoustic Emotion Recognition

(Reference[1])

5.1 Introduction

In recent years, applications involving speech recognition have become increasingly common in daily life and the work environment. Recognizing emotion in speech is necessary for speech applications to understand human states. If a speech application using a spoken dialogue system can recognize various human emotions, a machine navigator can provide more appropriate replies for a given situation. However, constructing speech applications that can understand emotions for a variety of situations is a challenging task because it is difficult to prepare a speech database that contains a variety of emotions [94]. Therefore, we focus on "categorical emotion task", which uses a small speech data in this chapter.

As a representative framework of the emotion classification, the emotion is estimated by using neural networks (NNs) that the input is Low-level descriptors (LLDs) which is the short-time frame feature extracted by 5-10msec shift [20][21][22], [23][24][25][26]. On the other hand, some methods using high-level statistical functions (HSFs) as input of NNs, which are statistics of short-time frame feature obtained by using a fixed-length window of more than 100 msec for LLDs, have been proposed [38][42][95].

In the former method, if the data is sufficient, it is possible to successfully extract both global and local information related to emotions by a convolutional neural network (CNN), Long short-term memory (LSTM), NNs with an attention mechanism. However, it is difficult to train a huge NN which can absorb varieties of emotional expression and personal differences using a small amount of data that the number of utterances is about 500 to 1500. In particular, improving emotion recognition for RAVDESS [96] is difficult because it includes a relatively large number of the speaker, but the number of recorded utterances is small.

[1]This dissertation is based on "Speech Emotion Recognition by Combining Multiple Discriminators with Multiple Temporal Resolution Features" [2], by the same author, which appeared in [2], Copyright(C)2022 IEICE. The material in this paper was presented in part at [2], and all the figures of this paper are reused from [2] under the permission of the IEICE.

In the latter method, emotion can be classified by a compact discriminator by intentionally inputting global information. The global information is extracted by taking statistics of the whole utterance or in a fixed length window. In [95], the mixed Gaussian distribution prepared for each emotion class is used, and the likelihood of all classes and all mixed Gaussian distribution is connected as a supervector and used as a feature. The feature called GeMAPs is also effective for emotion recognition [97]. However, these two features do not show performance in neural network-based discriminators compared with simple features such as Mel-Frequency Cepstral Coefficients (MFCC) and Mel-Filterbank (MF) [35].

On the other hand, the average and the variance of various LLDs features based on MFCC and MF using a fixed-length window are extracted, and high classification accuracy is obtained by discriminating them using a neural network in [38][42]. In particular, it results in better accuracy than the method of inputting LLDs directly into the neural network for RAVDESS [38].

For emotion recognition, both short and long-term features are important. In [98], the performance is improved by calculating the modulation-filtered cochleagram feature of four different resolutions in the time direction and the frequency direction for each frame and inputting it to the neural network. It is applied to the "dimensional emotion task".

In [39], multiple raw speech signal sequences augmented by downsampling and shift-average using some parameters are directly input to a convolution layer. Next, a pooling layer and an all-combining layer are applied to obtain emotion classification results for a "categorical emotion task". However, according to the comparison with the results of [39] and [99], the classification accuracy is better when a single MFCC sequence is input to a neural network compared to using the augmented raw speech signal sequences.

In [40], features for emotion classification are extracted using convolutional filters of 8, 16, 32, and 64 frame widths simultaneously for a sequence of MF. However, this method cannot improve the performance of that results from only 32-frame or 64-frame width.

In [41], convolutional layers with kernel sizes of 5, 7, 9, and 11 are prepared for a sequence of MFCC extracted with a frameshift of 10 msec for a window length of 25 msec, and a pooling layer takes the mean, variance, and maximum of their outputs. Finally, the attention layer integrate them and word vectors from a sentence. It achieves high accuracy of emotion classification. However, the integration method for the acoustic part without the text information do not significantly improve the accuracy compared to a single kernel size method.

As mentioned above, several studies focus on using multiple temporal resolutions [39], [99], [40]. However, neural networks with kernels corresponding to windows of four different temporal resolutions for LLDs features, such as MFCCs, do not significantly improve the accuracy compared to using a single temporal resolution [40],[41]. In the case of the existing methods, they easily lead to overfitting because many parameters of multiple temporal resolutions need to be trained for a neural network simultaneously using a small amount of training data. As a result, the variation of the kernel size in the time direction needs to be reduced.

We consider that the performance can be improved by extracting HSF by combining multiple time resolution windows instead of one fixed window length for the discrimination of emotion. On the other hand, the applied method should be simultaneously prevented from

overfitting. Therefore, we propose a new method using HSF with multiple time resolution windows for a small training data.

EmoDB [100] and RAVDESS are used for the classification experiment. They are both small databases. First, long short-term memory (LSTM) is trained for each HSF sequence in which average and variance of windows of different time resolutions are taken. Next, a method integrating the output of LSTM is applied to improve the emotion classification performance by utilizing the features of each window of different time resolution. In this chapter, we use Gradient Boosting Decision Trees (GBDT) as an integration method [44]. XGBoost [101], a kind of GBDT, has been applied to many classification problems for small to large tabular data [102][103][104][105]. The reason for using GBDT as the integration method this time is that it can calculate the importance of the feature used for the decision tree, that is, the feature for each window with different time resolution [106].

In this study, stacking using GBDT is applied to suppress the parameter which simultaneously learns for a small amount of data [43]. In this chapter, the class output of LSTM learned from the feature values of windows of each time resolution is used as the input to learn a GBDT model. Stacking using GBDT is an effective method in the classification problem [103][104][105]. However, it has not been applied to multiple classes of emotion classification so far.

On the other hand, a decision tree node is strongly affected by an effect of specific LSTM-outputs when existing GBDT is applied. Therefore, the generalization performance is degraded. For improving the generalization, comparison values by the median of each dimension of the LSTM output features are applied. The comparison value is 1 if it is bigger than the median, and 0 if it is smaller than it. We call the comparison value median feature in this chapter. When conventional features are applied, a class is decided only by a single dimension of an LSTM output in the upper layer part of the decision tree. Therefore, the relevance to other feature quantities cannot be taken into consideration. In order to prevent it, the median feature is applied to the branching of only the upper layer of the decision tree. The upper layer of the decision tree is roughly classified, and operated to use a plurality of features.

In this chapter, there are two contributions as follows. The first is a construction of an emotion discriminator with high classification performance by integrating features obtained from multiple resolution windows. The combination is executed by GBDT. The second is to improve the integration performance by introducing the median feature into the GBDT in the integration.

5.2 Outline of the proposed method

An outline of the proposed method is shown in Fig. 5.1. A LLD sequence of i of a sample with N is defined as $(r_{i1}, r_{i2}, ..., r_{iL_i})$. Here L_i is the sequence length of the i th sample. For W different time resolution windows, a HSF feature sequence obtained from the LLD sequence using the $w(w \leq W)$th window is defined as $(u_{i1}^{(w)}, u_{i2}^{(w)}, ..., u_{iM^{(iw)}}^{(w)})$. $M^{(iw)}$ is the sequence length of the HSF feature value obtained from window w for the ith sample. Next,

A Bidirectional LSTM is built for a HSF feature sequence by window w. The LSTM of wth outputs a class probability $(p_{i1}^{(w)}, ..., p_{ic}^{(w)}, .., p_{iC}^{(w)})$ for an emotion class $c(c = 1, 2, ..., C)$. Set as x_i to GBDT by connecting C class probabilities outputted respectively from W LSTMs. Therefore, x_i has the dimension of CW. The GBDT receives these inputs, outputs the final class probability for emotion classes. A result is defined by the highest probability in them.

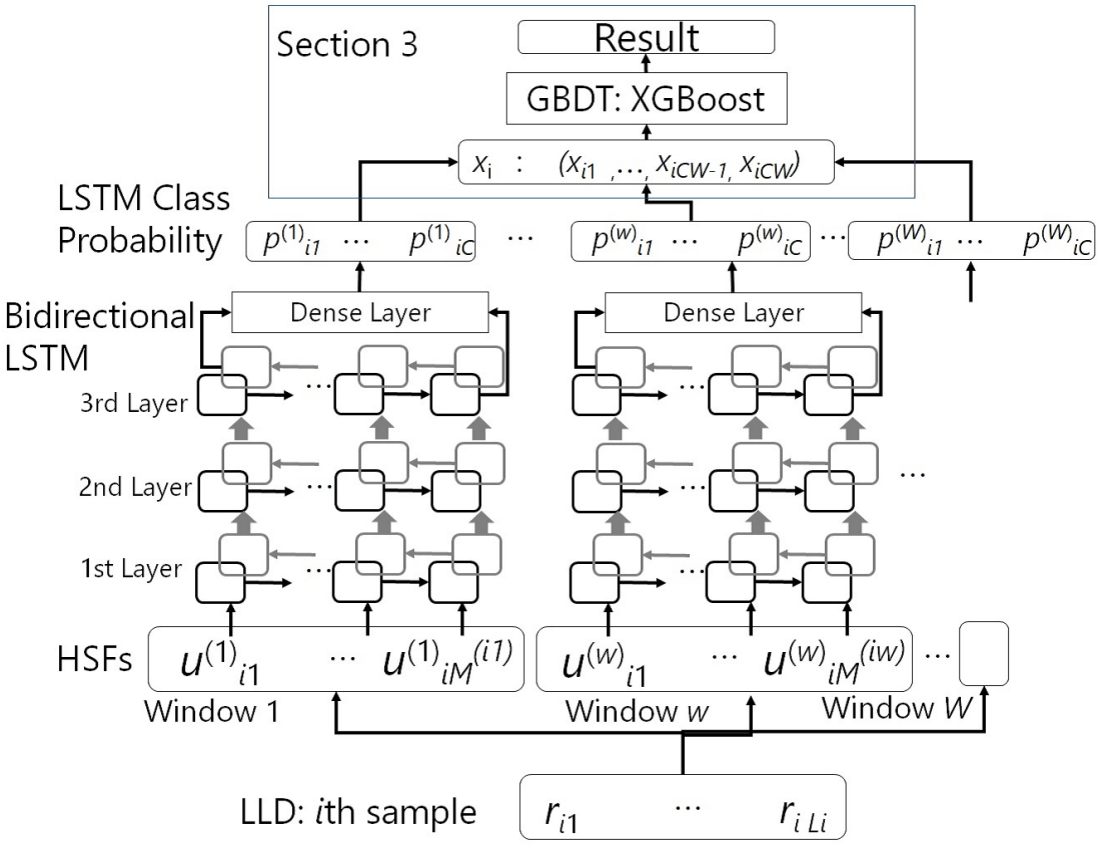


Figure 5.1: Outline of the proposed method (Copyright(C)2022 IEICE, [2] Fig.1)

5.3 Multi-Class GBDT

In this chapter, we combine the results of discriminators using features by windows with different time resolutions by GBDT, as shown in Fig. 5.1. Since the number of emotion classes in this chapter is 7 and 8, the outputs of GBDT are 7 or 8 probability values. We explain the algorithm to multi - class classification using soft max and cross entropy loss functions based on [101]. For the ith sample in N samples, an input feature amount $x_i(x_i \in \mathbb{R}^{CW})$ having a feature amount of CW and $y_i(y_i \in \mathbb{R}^C)$ with labels for C classes as elements are given. In this case, a prediction score value $\hat{y}_{ic}$ for a class $c(c = 1, 2, 3, .., C)$, when x_i is input to a model

$F_c(x_i)$ constituted of the decision tree of K, is obtained. At this time, the decision tree of kth is defined as $f_{kc}(k = 1, 2, ..., K)$, and $a_j^{(kc)}$ is defined as the weight allocated to the leaf of the leaf index j present in the decision tree of kth by $T^{(kc)}$. If $q^{(kc)}(x_i)$ returns the corresponding leaf index when x_i is given, $F_c(x_i)$ can be expressed as follows.

$$\hat{y}_{ic} = F_c(x_i) = \sum_{k=1}^{K} f_{kc}(x_i) = \sum_{k=1}^{K} a_{q^{(kc)}(x_i)}^{(kc)}. \tag{5.1}$$

The model allocates a sample x_i to leaves based on a decision tree f_{kc}, and adds the weight $a_{q^{(kc)}(x_i)}^{(kc)}$ assigned to each leaf for K trees from $k = 1$ to $k = K$. The model uses K decision trees for each class. In order to use a cross entropy using Softmax as a loss function, the loss function $l(y_i, \hat{y}_i)$ is as follows by using the Softmax function $S(\hat{y}_{ic})$.

$$l(y_i, \hat{y}_i) = -\sum_{c} y_{ic} \log S(\hat{y}_{ic}), \tag{5.2}$$

where

$$S(\hat{y}_{ic}) = \frac{\exp(\hat{y}_{ic})}{\sum_{z=1}^{C} \exp(\hat{y}_{iz})}, \tag{5.3}$$

Set Ω as a regularization term that gives penalty to the complexity of a decision tree, and use the following regularization objective function $\mathcal{L}(F_c)$ to optimize the decision tree.

$$\mathcal{L}(F_c) = \sum_{i=1}^{N} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_{kc}), \tag{5.4}$$

where,

$$\Omega(f_{kc}) = \gamma T^{(kc)} + \frac{1}{2}\lambda |a^{(kc)}|^2. \tag{5.5}$$

Function $\mathcal{L}(F_c)$ is replaced as follows because it is difficult to perform direct optimization. $\hat{y}_i{}^{(t)}$ is the predicted value for the ith sample for the tth time, and the function f_{tc} is added for the purpose of minimizing the loss.

$$\mathcal{L}(F_c) = \mathcal{L}_c^{(t)} = \sum_{i=1}^{N} l(y_i, \hat{y}_i^{(t-1)} + f_{tc}(x_i)) + \Omega(f_{tc}). \tag{5.6}$$

This is equivalent to adding f_{tc} to the green to reduce the loss. The second Taylor expansion is applied to this problem. g_{ic} and h_{ic} are first and second order partial differentials of l, respectively, and calculated using the expression 5.2 respectively.

$$g_{ic} = \frac{\partial l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_{ic}^{(t-1)}} = S_{ic}^{(t-1)} - \delta_{cc'}, \tag{5.7}$$

$$h_{ic} = \frac{\partial^2 l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_{ic}^{(t-1)}} = S_{ic}^{(t-1)}(1 - S_{ic}^{(t-1)}), \tag{5.8}$$

where,

$$S_{ic}^{(t-1)} = \frac{\exp(\hat{y}_{ic}^{(t-1)})}{\sum_{z=1}^{C} \exp(\hat{y}_{iz}^{(t-1)})}. \tag{5.9}$$

$\delta_{cc'}$ is a delta function that is 1 when $c' = y_{ic}$ is 1. When g_{ic}, h_{ic}, the expression after the Taylor expansion is as follows.

$$\begin{aligned} \mathcal{L}_c^{(t)} \simeq \sum_{i=1}^{N} [l(y_i, \hat{y}_i^{(t-1)}) + g_{ic} f_{tc}(x_i) \\ + \frac{1}{2} h_{ic} f_{tc}^2(x_i)] + \Omega(f_{tc}). \end{aligned} \tag{5.10}$$

Except for the constant term, the following equation is obtained.

$$\tilde{\mathcal{L}}_c^{(t)} = \sum_{i=1}^{N} [g_{ic} f_{tc}(x_i) + \frac{1}{2} h_{ic} f_{tc}^2(x_i)] + \Omega(f_{tc}). \tag{5.11}$$

If I_l is a subsample set belonging to leaf$l$, expression 11 can be modified as follows.

$$\begin{aligned} \tilde{\mathcal{L}}_c^{(t)} &= \sum_{i=1}^{N} \left[g_{ic} f_{tc}(x_i) + \frac{1}{2} h_{ic} f_{tc}^2(x_i) \right] \\ &+ \gamma T^{(tc)} + \frac{1}{2} \lambda \sum_{l=1}^{T^{(tc)}} a_l^{(tc)2} \\ &= \sum_{l=1}^{T^{(tc)}} \left[(\sum_{i \in I_l} g_{ic}) a_l^{(tc)} + \frac{1}{2} (\sum_{i \in I_l} h_{ic} + \lambda) a_l^{(tc)2} \right] \\ &+ \gamma T^{(tc)}. \end{aligned} \tag{5.12}$$

The optimum weight $a_l^{(tc)*}$ can be calculated by partial differentiating the above expression by $a_l^{(tc)}$ and setting the left side to 0.

$$a_l^{(tc)*} = -\frac{\sum_{i \in I_l} g_i}{\sum_{i \in I_l} h_{ic} + \lambda}. \tag{5.13}$$

When this $a_l^{(tc)*}$ is substituted for expression 12, the following expression is obtained.

$$\tilde{\mathcal{L}}_c^{(t)}(q) = -\frac{1}{2} \sum_{l=1}^{T^{(tc)}} \frac{(\sum_{i \in I_l} g_{ic})^2}{\sum_{i \in I_l} h_{ic} + \lambda} + \gamma T^{(tc)}. \tag{5.14}$$

In practice, it is not realistic to calculate the optimal decision tree from all the combinations of patterns, so the optimal tree structure is obtained by the greedy algorithm. If I_L is a branch set belonging to the left side of the tree and I_R is a set belonging to the right side, and $I = I_L \cup I_R$, then the loss reduction at that time is determined as follows.

$$\mathcal{L}_{c,split} = \frac{1}{2}\left[\frac{(\sum_{i\in I_L} g_{ic})^2}{\sum_{i\in I_L} h_{ic} + \lambda} + \frac{(\sum_{i\in I_R} g_{ic})^2}{\sum_{i\in I_R} h_{ic} + \lambda} - \frac{(\sum_{i\in I} g_{ic})^2}{\sum_{i\in I} h_{ic} + \lambda}\right] - \gamma. \tag{5.15}$$

The formula is used for calculating the split feature candidate of the decision tree, and also for the importance calculation of the feature. The pseudo-code of the splitting algorithm in the case of multI-class extension is shown referring to the literature in Figure 5.2. The following is described according to the pseudo code. Splitting is performed at each node of the decision tree for each class c. When an existing sample set is set to I_c in a node with a class c, the sum of g_{ic}, h_{ic} of the present node is calculated first. Then rotate the loop for the sample's feature dimension m. Sort the sample set with the value of dimension m and try it in order from the smaller value. For dimension m, a sample smaller than the value of sample x_{jm} is L. a larger sample is divided into R. The division score of the expression 15 becomes the maximum value is determined as a division point of the node. The approximation search algorithm is also proposed in the literature, but it is not used in this chapter.

As the whole movement of the multi-class GBDT, one decision tree is created and added in each class for each iteration t at the time of learning. A decision tree is generated by using the splitting algorithm. The final probability value for each class is calculated from Softmax by adding the score of the decision tree of K for each class to the input x_i using expression 5.1, 5.3.

5.4 Median Feature

In this chapter, we introduce a median feature to enhance the generalization performance of the decision tree of GBDT. To solve the problem that only the feature of a specific LSTM output is selected for a small amount of training data or an input of a small number of dimensions and a merit of integrating a plurality of identification results is not received when the feature of a decision tree is defined by using the LSTM output of a 7.8 class like this time. Then, in the upper layer of the decision tree, rough classification is performed so that the leaf is not determined by a single feature, and the leaf is determined by a plurality of feature quantities. As a contrivance for not depending only on a small number of specific feature quantities, the method of Column Sampling has been introduced in XGboost. In this chapter, we propose a method of selecting the feature of each branch node at random. In this chapter, Column Sampling and median feature quantities are used together, and the effect is confirmed.

Next, the median feature is explained. Let $x_{im}(m = 1, 2, .., CW, .., 2CW)$. In order to add a median feature dimension, the total dimension of x_i is set to $2CW$. If the index of the

Input: c, class index
Input: I_c, instance set of current node for class c
Input: m, feature dimension
gain $\leftarrow 0$
$G_c \leftarrow \sum_{i \in I_c} g_{ic}, H_c \leftarrow \sum_{i \in I_c} h_{ic}$
for $m = 1$ to CW **do**
  $G_{Lc} \leftarrow 0, H_{Lc} \leftarrow 0$
  for j in sorted $(I_c, by\ x_{jm})$ **do**
    $G_{Lc} \leftarrow G_{Lc} + g_{jc}, H_{Lc} \leftarrow H_{Lc} + h_{jc}$
    $G_{Rc} \leftarrow G_c - G_{Lc}, H_{Rc} \leftarrow H_c - H_{Lc}$
    score $\leftarrow \max(score, \frac{G_{Lc}^2}{H_{Lc}+\lambda} + \frac{G_{Rc}^2}{H_{Rc}+\lambda} - \frac{G_c^2}{H_c+\lambda})$
  end for
end for
Output: Split with max score for the current node of class c

Figure 5.2: Splitting Algorithm (Copyright(C)2022 IEICE, [2] Fig.2)

median value of the mth dimension in all samples is h_m, the median feature amount can be expressed as follows.

$$x_{i(CW+m)} = \begin{cases} -1 & (x_{im} \leq x_{h_m m}), \\ 1 & (x_{im} > x_{h_m m}). \end{cases} \tag{5.16}$$

If x_{im} is less than the median value of the total training sample, -1 is added to $x_{i(CW+m)}$ as a feature amount. In contrast, if x_{im} is more than the median value of the total training sample, 1 is added to $x_{i(CW+m)}$ as a feature amount. When the median feature value is applied to the depth of the decision tree D, if the target node is the depth of the decision tree D or less, division is performed using only the median value feature value $x_{im}(m > CW)$. If it is deeper than D, split only using $x_{im}(m \leq CW)$. At the time of evaluation, the feature is calculated by using the value of the median value of the learning sample.

5.5 Experiment

5.5.1 Experimental Setup

In this experiment, we used emotional speech databases RAVDESS [96] and EmoDB [100]. RAVDESS is a database in which actors record voices and images expressing eight emotions. The voice includes not only speech but also recorded data of songs. In this chapter, we used only the utterance data of the speech. The eight emotions were neutral, calm, happy, sad, angry, fearful, surprise, disgust, and recorded 1140 speech by 24 actors. All the sounds were recorded with 16 bit, 48 kHz sampling. The downsampling was performed to 16 kHz.

EmoDB is a voice database that records the voice of seven emotions: neutral, happiness, anger, fear, sadness, boredom, disgust. The number of speeches was 535 by 10 speakers. All the sounds were recorded by 16 bit, 16 kHz sampling, and the number of bits and sampling frequency were used in the experiment.

5.5.2 Evaluation Metric

Since RAVDESS and EmoDB are small databases, when a model parameter is tuned in an evaluation set, it easily overfits to the evaluation set, and the correct result is not obtained in practical use. In addition, the performance depends on the selected evaluation set only by one training set and evaluation set, and the correct evaluation is not obtained. In this chapter, we defined a learning set, development set, and evaluation set for all utterances at the rate of 0.75: 0.15: 0.1. A development set and an evaluation set were constructed to keep the proportion of classes of the learning set. This evaluation method was similar to the experiments in [35][38] All evaluation values were obtained from the average value of these 10 trials.

For the parameter tuning, the development set of each trial was used. In order to evaluate the accuracy of emotion classification, a general weighted accuracy was used as a performance evaluation of emotion classification [35][38]. F value was used for the accuracy evaluation of each emotion. “Accuracy” in this chapter represents weighted accuracy.

5.5.3 Feature for experiments

In this chapter, we used a base feature of 152 dimensions, which was a combination of 24 - dimensional Mel - Frequency Cepstral Coeffficients (MFCC) and 128 - dimensional Mel - Filterbank(MF) features. These were generated by Fourier transformation for a window length of 25 msec, and each feature was obtained for each 8 msec shift. The features were commonly used in speech emotion recognition [35][36][38]. They were generated using the librosa toolkit [107]. An average and variance were calculated by using a fixed time window length for the extracted feature, and the features were input to the neural network.

5.5.4 Preliminary Experiments

In order to examine a size of a appropriate neural network for each database, some neural networks of different size and configuration were prepared, and the emotion classification performance was compared. Four neural network parameters created in Table 5.1 are presented. The NN1 in the table was a feedforward neural network of all coupling up to the second layer, and only the third layer was bidirectional-LSTM (Bi-LSTM). The NN2, NN3 and NN4 were all Bi-LSTMs, except for the input layer and output layer, and the number of units was 128, 256, 512, respectively. The Bi-LSTM of the third layer used the output of each unit as the input of the next stage every time the feature was input, and the Bi-LSTM of the third layer used the unit output of the Bidirectional termination state as the input of the final layer. #Params indicates the number of parameters for the neural network. The batch size was 10 for EmoDB and 25 for RAVDESS [36]. Batch Normalization and DropOut were applied to all

Table 5.1: Preliminary Experiments: Trained Neural Netwroks [#units] ((D) means "Dense layer", and (L) means "Bidirectional-LSTM layer"))

Layer	NN1	NN2	NN3	NN4
Input	152(MFCC24 + MF128)			
1^{st}	256 (D)	128 (L)	256 (L)	512 (L)
2^{nd}	256 (D)	128 (L)	256 (L)	512 (L)
3^{rd}	128 (L)	128 (L)	256 (L)	512 (L)
Final	(D) (softmax)			
#Params.	541384	1235144	4304072	15946952

Table 5.2: Preliminary Experiments: The performance of trained Neural Networks (Accuracy[%])

	NN1	NN2	NN3	NN4
EmoDB	79.4	78.9	76.9	77.4
RAVDESS	73.8	72.9	75.6	72.4

the neural networks, and Adam was used for the optimization. The number of epochs was set to be the maximum accuracy in each development set of 10-fold validation. In this chapter, we extracted a base feature by using a time window length of 20 frames and 10 frameshift.

The average accuracy of the 10-fold validation is shown in Table 5.2. The performance of NN1 was the best for EmoDB, and the performance of NN3 was the best for RAVDESS. After this experiment, the experiment was carried out using only the neural network with the best performance.

5.5.5 Emotion recognition results for each different time resolution

In this chapter, we conducted an experiment to classify emotions by using the feature generated for each window with different time resolution. In the emotion classification, it is shown that it is better to use the statistic of the feature of the short time frame as the feature with a certain time width than to use the feature for every short time frame for the input of the neural network. However, in these papers, a fixed time window was used, and the performance of discrimination of emotion and the time resolution of the window which took statistics did not be examined. In this chapter, we compared the accuracy of discrimination of emotions by using an LSTM emotion discriminator to obtain the accuracy using the average and variance calculated from windows of different time resolutions for the feature of a short time frame.

Design of window with different time resolution

As for the time resolution, the time window length was changed in 2 frames from 6 to 80 frames, and the result of the emotion classification and F measure for each feeling were calculated. In all experiments, the feature sequence was extracted by shifting half of the time window length. For example, when the time window length is 6 frames, the shift was three frames, and when the time window length was set to 10 frames, the shift was 5 frames. The average and variance of the feature of the total 152 dimensions which combined the 24-dimensional MFCC and the 128-dimensional MF were calculated for the window of each time resolution, and they were input to each LSTM.

Accuracy

Figure 5.3 shows the accuracy calculated for each window with different time resolution for EmoDB. Figure 5.3 shows the accuracy calculated for each window with different time resolution for RAVDESS. The smoothed accuracy in the figure was averaged using the values of one point before and after. For example, the average value of the window length of 8, 10 and 12 was obtained at the point of 10 frames of the window length. These figures show that the performance of EmoDB was low in low resolution with a long window length. On the other hand, the performance of RAVDESS was high in high resolution with a short window length compared to EmoDB.

5.5.6 Integration of multiple discriminators

In this chapter, we constructed boosting trees that integrate multiple time resolution windows identification results using XGBoost, a kind of GBDT. First, the learning parameter is described. The cross entropy using softmax was used as the loss function. The number of decision trees for one model was set to 200. Since the decision tree exists for each class, it has 1400 decision trees for EmoDB with 7 classes and 1600 decision trees for RAVDESS with 8 classes. Column sampling for each node of the decision tree used only 10% of the total feature at each node. The value of λ was fixed at 1.0, and γ was fixed at 0.1. The maximum value of tree depth was set to 4 in all experiments. The sub-sampling which randomly samples the learning samples used for each decision tree addition was set to be optimum in $0.4, 0.6, 0.8$ for the development set. The learning rate was set to be optimum in the development set in $0.001, 0.0050, 010$. Therefore, the optimum one in the development set was selected from 9 kinds of parameters which were combinations of three sub-sampling and three learning rates. For the median feature, the performance was calculated by changing the parameter of D to $0, 1, 2$. Note that 0 is equivalent to a normal XGBoost that does not add a median feature.

First, we trained the model using the output results of 36 LSTM from 10 to 80 frames in window length. The feature importance was then calculated from the built boosting tree. The feature importance used this time was the sum of the loss reduction values obtained by expression 15 for all trees when the feature was used as a branching condition in the node in the constructed tree. The loss reduction value when the median feature was used in the

node was added to the original feature dimension in which the median value was calculated. Since the loss reduction value was obtained for each Feature dimension, that was, for each class output of each LSTM, the sum of loss reduction values for all classes was taken, and the feature importance for each window was calculated. In the order of the feature importance, 15, 10, 5 windows were selected, and the boosting tree was made again from the LSTM output produced by those windows.

Performance of Integration and Median Feature

Table 5.3 shows the results from LSTM-outputs integration with or without median features based on XGBoost. #Win. in the table indicates the number of windows used for the combination. The experiment results using 36 windows, are shown at the top. "Average" in the table indicates the additive average of the LSTM posterior probabilities, and "XGBoost" in the table is the result of integrating by XGBoost.

Since the values in the table were the average values of the 10-fold validation, the p-values were calculated using a paired t-test for "Baseline" using 10 evaluation sets. The significance level was set at 0.05. The * in the right side of the table indicates that there is a significant difference at $p < 0.05$ for "Baseline". ** indicates that there is a significant trend at $0.05 < p < 0.10$. If there is no * or ** to the right of the number, $p > 0.10$ and there is no significant difference to "Baseline". For each of the XGBoost results under the same conditions as the Average results, p-values were calculated by paired t-test and the significance level was set at 0.05.

The + sign on the right side of the numbers in the table indicates that there is a significant difference between the results of XGBoost under the same conditions as the results of Average at $p < 0.05$. ++ indicates a significant trend with $0.05 < p < 0.1$. If there is no + or ++ on the right side of the number, the result is not significantly different from the "Average" result because $p > 0.10$.

Baseline was selected for the best performing window among the different time resolutions. In Figs. 5.3 and 5.4, "Baseline" was 82.4% for EmoDB, that was the 22 frames of window length and 11 frames of shift, and 76.4% for RAVDESS, that was the 42 frames of window length and 21 frames of shift. When all 36 windows of different temporal resolutions were used, the performance of EmoDB was 84.8% when $D = 0$, which was significantly different from "Baseline"; the performance of RAVDESS was 83.3% when $D = 2$, which was significantly different from "Baseline". These indicated that integrating the results from multiple resolution windows improved the performance over the best performance obtained from a fixed window.

As for the integration method, there was no tendency for "XGBoost" to perform better than the Average result in EmoDB. On the other hand, for RAVDESS, "XGboost" with $D = 0$ did not show any significant improvement over the "Average" results, but "XGBoost" with $D = 1$ and $D = 2$ using the median feature showed a significant improvement in the condition of 36 windows and several window numbers. It indicates that the XGBoost integration method using the median feature tends to perform better than the "Average" method for RAVDESS. Since EmoDB has less than half the training data of RAVDESS, the performance difference

between XGBoost, which learns parameters, and Average, which is simple, is considered to be small. When XGBoost was used as the integration method, the highest accuracy was obtained using the median feature for all window numbers, although there were some cases where the same points were obtained.

For EmoDB, the result of $D = 0$ using XGBoost with 36 windows and the results of each window number smaller than 36 windows were t-tested respectively. As results of the experiment, no significant difference was found. On the other hand, the t-test between the result of $D = 2$, which has the maximum accuracy using XGBoost with 36 windows for RAVDESS, and the result with the maximum accuracy in each window number smaller than 36 windows, showed that there was a significant difference for the 10 windows case. The t-test showed that there was no significant difference between the result of 10 windows and the result of 36 windows for EmoDB. On the other hand, there was a significant difference between the results of 10 windows and the result of 36 windows for RAVDESS. The use of a smaller number of windows has the advantage of reducing the number of LSTMs to be calculated.

Analysis for Window of Different Time Resolution

Figure 5.12 and 5.13 show the top 10 feature importances for EmoDB and RAVDESS for each validation set when $D = 2$. The top row shows the window Length of the time resolution, and each row from the second row shows the selected window length by filling the cell corresponding to the selected window length for one validation set. Since the top 10 were shown for each validation set, each row was filled with 10 cells. These figures show that high-resolution windows with short window length were selected for EmoDB, and low-resolution windows are selected for RAVDESS, except for high-resolution windows with window-length 10-14. In Fig. 5.4, the performance decreases at low resolutions of window length 50 and above, but many low-resolution windows of window length 50 and above are selected in Fig. 5.13. It indicates that multiple windows with a wide range of temporal resolutions, from low to high, are useful, especially for RAVDESS.

Discussion of differences in trends among evaluation databases

RAVDESS is a database of 1140 utterances by 24 American speakers, and EmoDB is a database of 535 utterances by 10 German speakers. In [108], the difference in emotion recognition by speaking rate between American and German was discussed. In this paper, only five emotions (happy, sad, angry, frightened, and neutral) were tested, but the difference in speech rate had a greater effect on emotion recognition in American speech in general, especially for both slow and fast speech. In contrast, the effect of speech rate on the perception of emotion in German was smaller than in American. In the case of RAVDESS, the effect of the proposed method was likely to be higher because of the large number of speaker variations in American. Quantitative experiments for this discussion and experiments with open speakers are future work.

Table 5.3: Performance of the Integration and Median Feature (Accuracy[%])

Input	EmoDB					
	$D = 0$		$D = 1$		$D = 2$	
#Win.	Average	XGBoost	Average	XGBoost	Average	XGBoost
36	84.3 **	84.8 *		84.6 **		84.8 **
15	85.4 *	84.4 ** ++	84.5 **	84.6 **	85.0 *	84.6 *
10	84.4 **	83.9	84.1 **	84.3 *	84.6 *	83.3 +
5	83.2	83.5	84.1 **	83.5	82.4	81.7 ++
Baseline	82.4					
Input	RAVDESS					
	$D = 0$		$D = 1$		$D = 2$	
#Win.	Average	XGBoost	Average	XGBoost	Average	XGBoost
36	82.0 *	83.0 * ++		83.1 * +		83.3 * +
15	81.4 *	81.8 *	81.7 *	82.4 *	81.7 *	83.0 * +
10	81.3 *	81.7 *	81.3 *	81.9 * ++	81.3 *	82.4 * ++
5	80.3 *	80.7 *	80.5 *	82.4 * +	80.2 *	81.7 *+
Baseline	76.4					

5.5.7 Benchmark

The latest evaluation results were compared with our method. Table 5.4 shows the RAVDESS benchmark results, and Table 5.5 shows the EmoDB benchmark results. The literature used in the benchmark has a description that validation has been performed at least five times. The value in the tables was quoted from the literature. Unweighted Accuracy (UA) in the table is the average recall calculated for each emotion. If there is no (UA) description, it means weighted accuracy. For multiple LLDs consisting of MFCC, MF, Chroma vectors, spectral contrast, and tonal centroid features, the performance of HSF LSTM using fixed window statistic features [38] was 78.5% for RAVDESS, the best of conventional methods. On the other hand, the accuracy of our method was 83.3%. As far as we know, it is the highest performance at present. In EmoDB, the performance of QCNN [21], which processes spectrograms as two-dimensional images, was the highest accuracy. In EmoDB, the performance of a window in a relatively short time was better than that of a window for a long time, as shown in Fig. 5.3 and 5.12. In addition, the short-length window was chosen even when multiple windows were integrated. Therefore, even when the data quantity is small, it is possible to construct the highly accurate discriminator only from the feature in the short-time. It is necessary to consider the combination of multI-resolution window and CNN system in the future.

Table 5.4: RAVDESS Bencmark

Medhod	Year	Accuracy [%]
BLSTM [35]	2019	63.9 (UA)
Capsule [35]	2019	68.1 (UA)
Deep CNN [20]	2020	71.7
QCNN [21]	2021	77.9
HSF LSTM [38]	2020	78.5
Proposed Method	2021	83.3

Table 5.5: EmoDB Benchmark

Medhod	Year	Accuracy [%]
CNN LSTM [36]	2019	82.4
Proposed Method	2021	84.8 (83.8 (UA))
Deep CNN [20]	2020	86.1
QCNN [21]	2021	88.8 (UA)

5.6 Conclusion

We proposed a method to improve speech emotion recognition accuracy using the small emotional speech database. The proposed method extracted statistical features that are calculated from windows with multiple time resolutions. Multiple LSTMs which classified emotions were trained using each multiple time resolution feature. By integrating the outputs of these LSTM using XGBoost, a kind of GBDT, the speech emotion discriminator corresponding to multiple time resolutions was constructed. Furthermore, a median feature was introduced to keep the generalization for the integration.

We conducted emotion recognition experiments using EmoDB and RAVDESS. 82.4% for EmoDB and 76.4% for RAVDESS were obtained using a conventional method using a single time-resolution window. On the other hand, 84.8% for EmoDB and 83.3% for RAVDESS were obtained using the proposed method integrating the LSTM-outputs of 36 different time-resolution windows. The results show that the proposed method was effective.

The benchmark of the proposed method and some conventional methods was conducted. As a result, the proposed method achieved the top performance for RAVDESS. On the other hand, the performance of EmoDB was not equal to the top. The high-performance approach in EmoDB made good use of CNN's components. The hybrid method that compensates for the weak points of both methods is considered to be effective in improving the performance more.

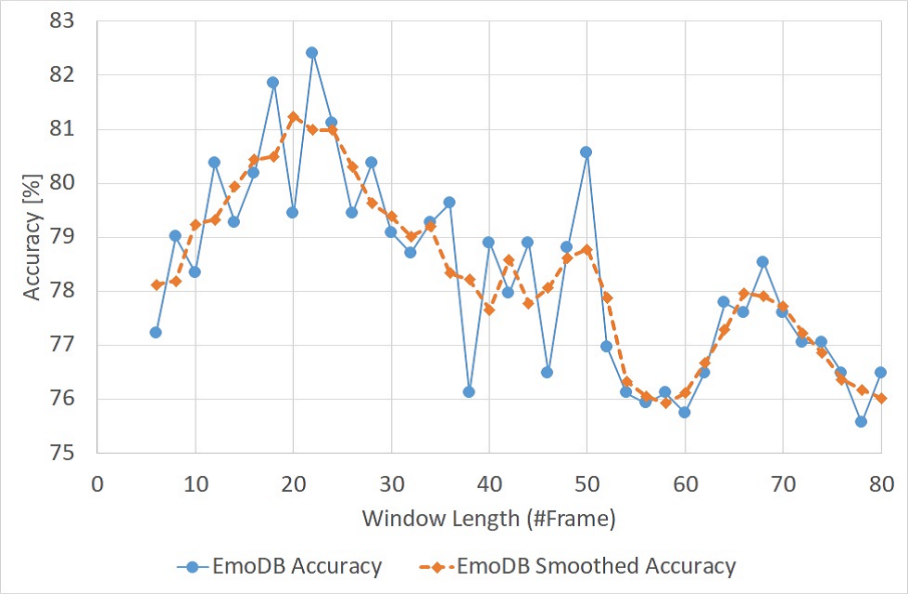


Figure 5.3: Accuracy by Window with Different Time Resolution(EmoDB) (Copyright(C)2022 IEICE, [2] Fig.3)

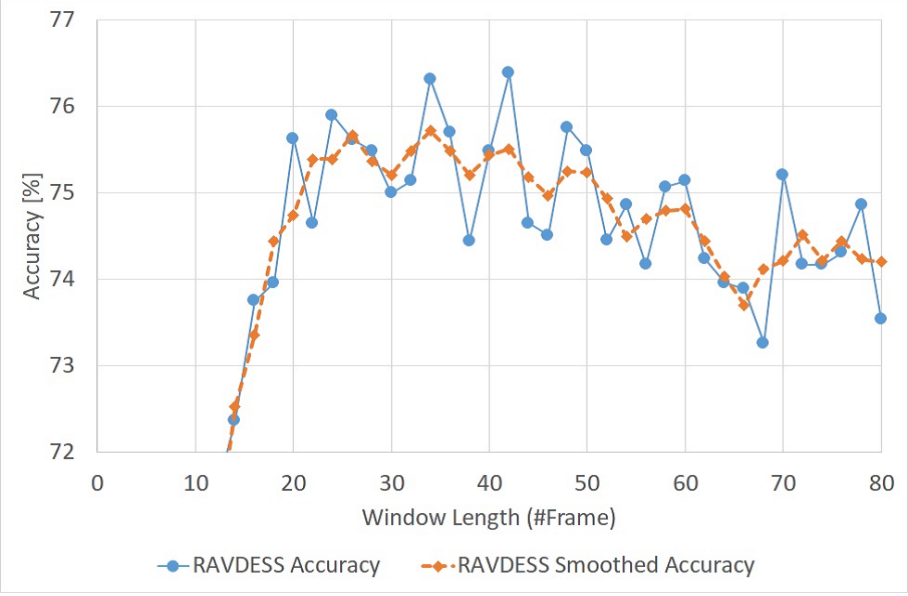


Figure 5.4: Accuracy by Window with Different Time Resolution(RAVDESS) (Copyright(C)2022 IEICE, [2] Fig.4)

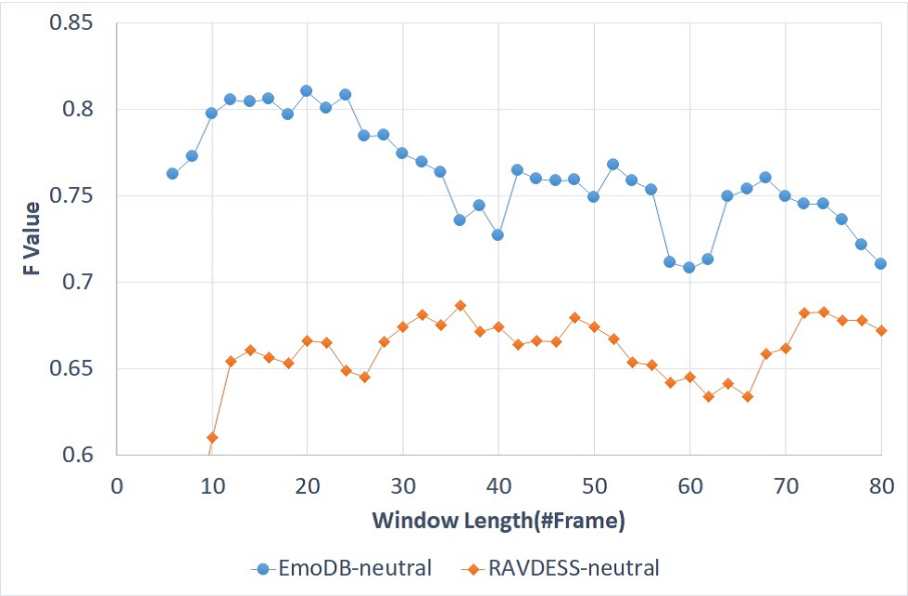


Figure 5.5: “neutral” Emotion Recognition (Copyright(C)2022 IEICE, [2] Fig.5)

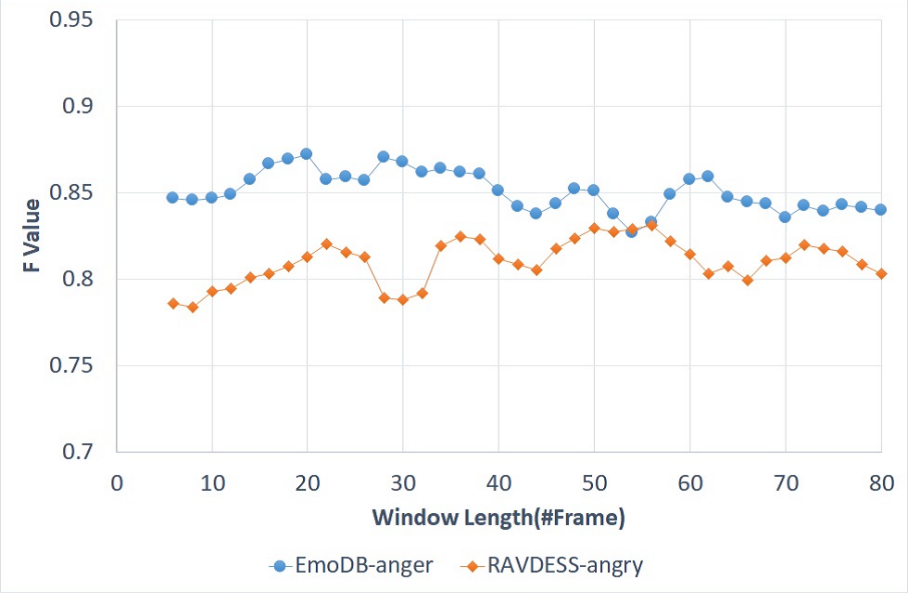


Figure 5.6: “anger” and “angry” Emotion Recognition (Copyright(C)2022 IEICE, [2] Fig.6)

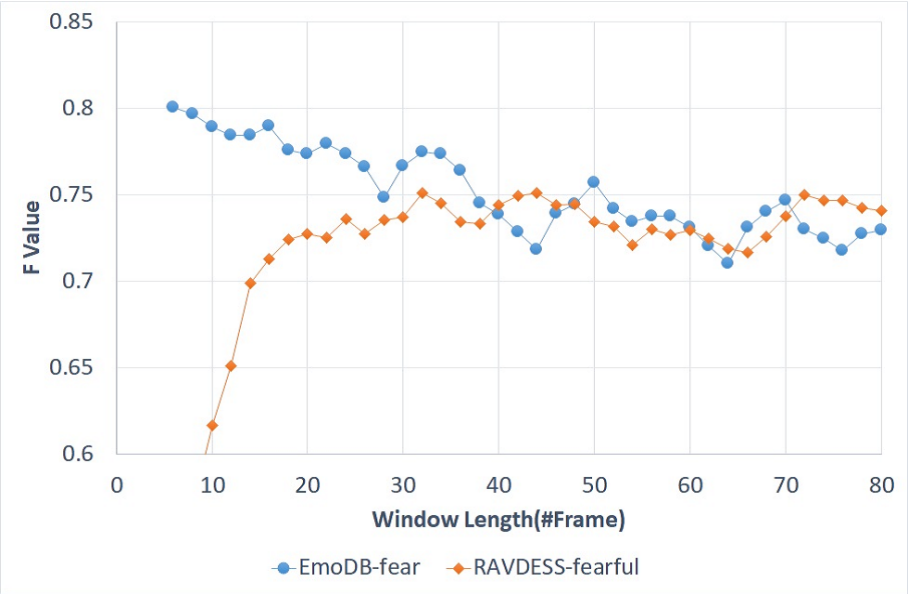


Figure 5.7: “fear” and “feaful” Emotion Recognition (Copyright(C)2022 IEICE, [2] Fig.7)

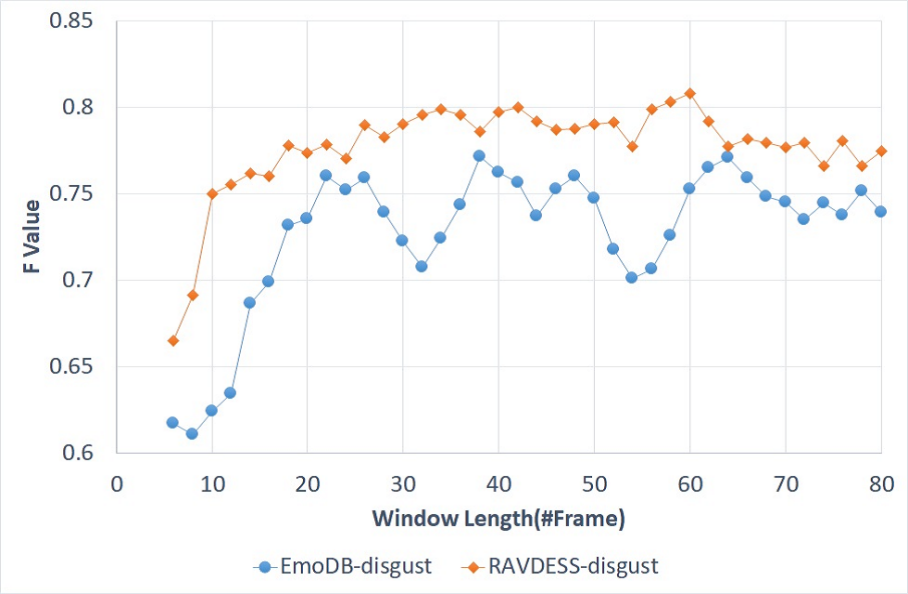


Figure 5.8: “disgust” Emotion Recognition (Copyright(C)2022 IEICE, [2] Fig.8)

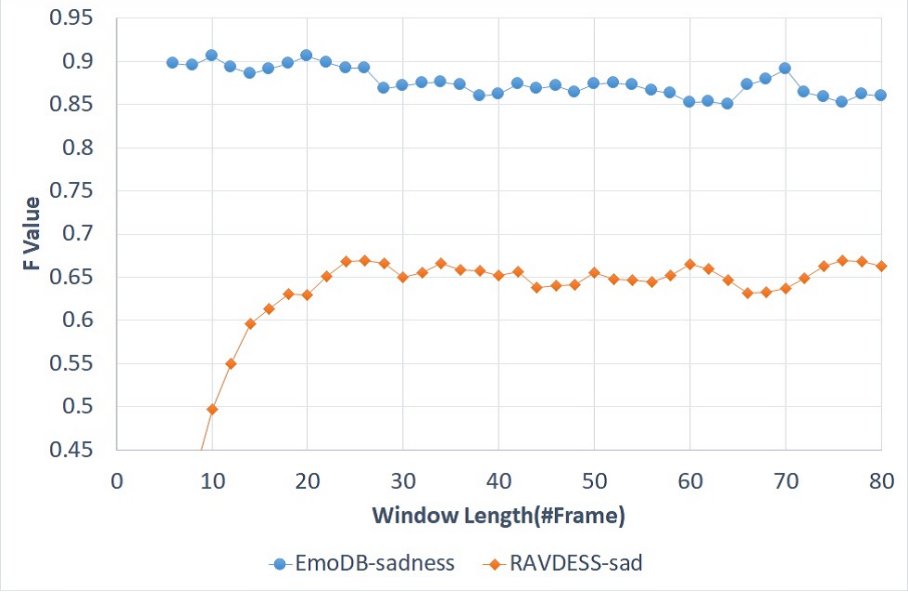


Figure 5.9: “sadness” and “sad” Emotion Recognition (Copyright(C)2022 IEICE, [2] Fig.9)

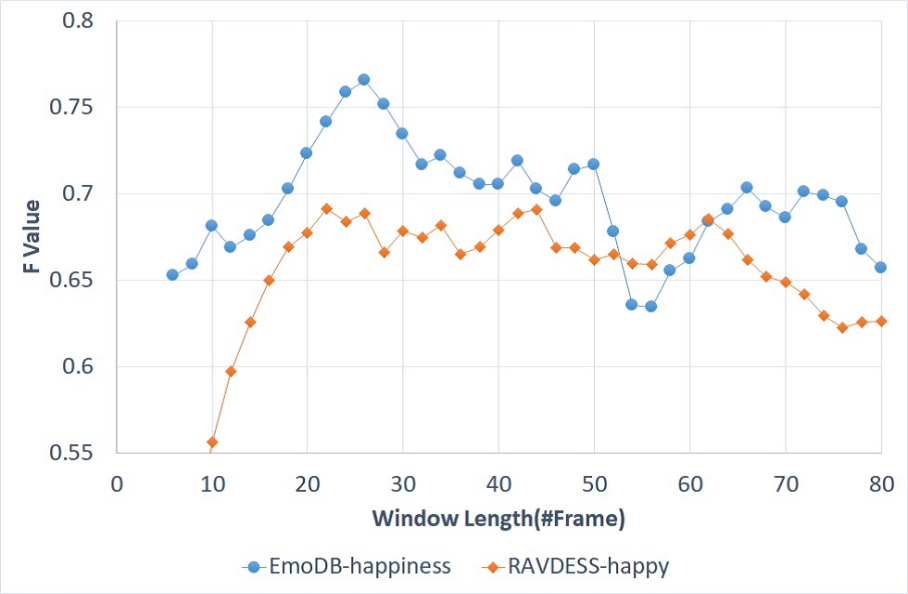


Figure 5.10: “happiness” and “happy” Emotion Recognition (Copyright(C)2022 IEICE, [2] Fig.10)

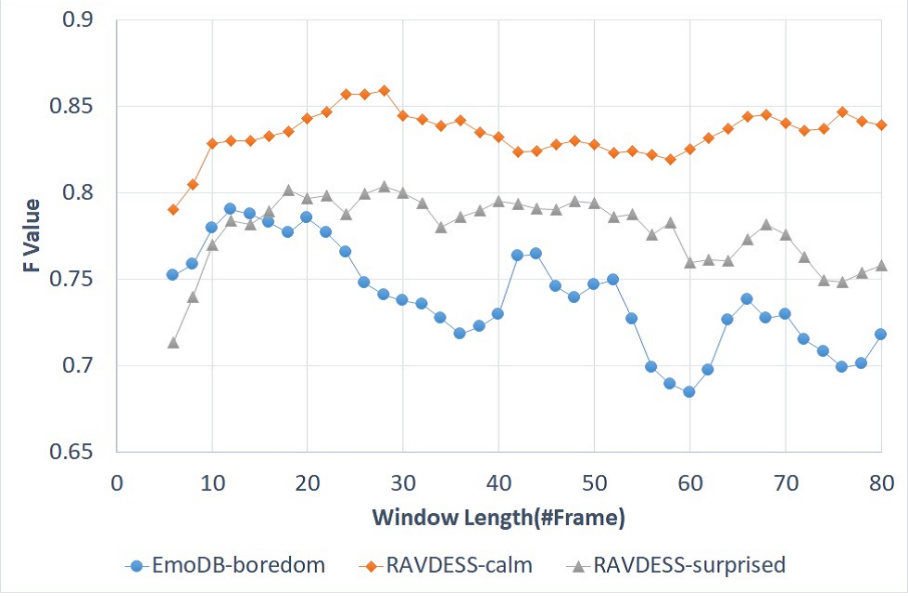


Figure 5.11: “boredum”, “calm” and “surprise” Emotion Recognition (Copyright(C)2022 IEICE, [2] Fig.11)

Window Length (#Frame)	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48	50	52	54	56	58	60	62	64	66	68	70	72	74	76	78	80
set1	0	1	0	0	1	0	1	1	0	1	1	1	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
set2	0	0	0	1	0	1	0	0	1	1	0	0	1	0	0	0	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
set3	0	1	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1	0	1	1	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0
set4	1	0	1	0	0	0	0	1	0	0	1	1	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
set5	0	0	1	0	1	0	1	1	0	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0
set6	1	0	1	1	0	1	0	1	0	1	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
set7	0	0	1	0	0	0	0	1	0	1	0	1	0	1	0	0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0
set8	0	0	1	0	0	0	1	0	1	1	0	1	0	0	0	0	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
set9	1	0	1	1	0	0	1	0	1	0	0	1	0	1	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
set10	0	1	0	1	1	0	0	0	0	0	1	0	1	1	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 5.12: Feature Importance top-10 for EmoDB (Copyright(C)2022 IEICE, [2] Fig.12)

Window Length (#Frame)	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48	50	52	54	56	58	60	62	64	66	68	70	72	74	76	78	80
set1	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	0	0	1	0	1	0	0	0	0	1	1	0	1	0	0	0	0	0	0	1	0
set2	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1	0	1	0	0	1	0
set3	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	1	0	1	0	0	0	0	1	0	1	0	0	1	0	0	0	0	1	1
set4	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	1	0	1	1
set5	0	0	0	0	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	1	1	0	1	1	0	1	0	0	0	0	0	0	0	1	0
set6	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	1	0	1	0	1	0	0	1	1	0	1	0	1
set7	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	1	1	0	0	1	0	0	1	1
set8	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	1	1	0	0	1	0	1	1	0	1	0	0	0	1	0	0	0	0	0	0	0
set9	0	0	0	0	0	0	1	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	1	0	0	0	1
set10	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0	0	1	1	0	0	1	0	1	1	0	1	0	0

Figure 5.13: Feature Importance top-10 for RAVDESS (Copyright(C)2022 IEICE, [2] Fig.13)

Chapter 6

Conclusion

6.1 Summary of Research

The problems and contributions focused in this paper are listed up again.

Overall

- Target: Realize real-time paralinguistic and nonverbal information extraction from speech signal towards empathetic dialogue systems.
- Contribution: Improve the performance of speaker attribute recognition, phoneme recognition, acoustic event detection and recognition, and acoustic emotion recognition in the aspects of recognition performance and real-time decoding.

Chapter 2 Speaker Attribute Recognition

- Problem: Separately consider speech detection and speaker attributes. Calculation cost is high, and no real-time processing.
- Contribution: Propose simultaneous speech detection and speaker attribute recognition method. It leads to low cost and real-time calculation.

Chapter 3 Phoneme Recognition

- Problem: Need a phoneme clustering tree that is highly dependent on each language. Therefore, the training cost is high.
- Contribution: Propose a new phoneme discriminator without a phoneme clustering tree.

Chapter 4 Acoustic Event Detection and Recognition

- Problem: Need word-dictionary with filler and word fragment and need second-pass decoding.
- Contribution: Propose simultaneous speech recognition and detection of filler and word fragments. It leads to real-time decoding without a specific word dictionary.

- Problem: The classification performance of emotion is low. There is no real-time decoding for it.
- Contribution: Propose multiple discriminators with multiple temporal resolution features. The classification performance is improved using a method that can have a possibility for real-time decoding.

6.1.1 Speaker attribute Recognition

In chapter 2, we focused on a method for identifying speaker attributes, which were nonverbal information in speech. We proposed simultaneous voice activity detection and gender classification method using single deep neural networks (DNNs). A frame-based classifier of DNNs classifies each frame into male, female, and silence clusters. These frame-based results were used for both gender classification and VAD. This method was competitive compared to existing techniques for both gender classification and VAD. The calculation cost does not increase much when the conventional DNN-VAD is applied. In addition, it can incrementally output the classification result.

This method can be applied to other speaker attribute classifications. It can provide speaker attribute information to spoken dialogue system with real-time processing. In the future, the method should be combined with a spoken dialogue system.

6.1.2 Phoneme Recognition

In Chapter 3, we proposed a phoneme identification method that leads to the extraction of paralinguistic and nonverbal information. In particular, we focused on calculating a phoneme score to measure the degree of low-intelligibility speech. For that purpose, we proposed a novel technique to exploit discriminative models with subclasses for speech recognition. Our method is subclass AdaBoost which does not need splits by any clustering method in advance. Subclass AdaBoost can automatically optimize the discriminator with triphone contexts. This method can be applied to many applications. As a first try, a phoneme classifier is constructed for each phoneme by the subclass AdaBoost algorithm. Each phoneme classifier discriminates whether the given segment is the phoneme or other phonemes, and outputs the AdaBoost score for the segment as the classification score. In the classification task, the performance of the subclass AdaBoost is highly better than the conventional AdaBoost. The error reduction rate is 53.49%. Furthermore, experimental results showed that the proposed technique consistently improved the continuous word recognition performance. The word error reduction rate was max 11.29% , when the number of weak classifiers was 600 using the proposed subclass AdaBoost. The method was only applied to Japanese. Additional experiments for other languages should be tried in the next step.

This method provided a phoneme classifier without constructing a clustering-tree which was highly dependent on language. Using the discrimination score for phoneme segments, can calculate a discrimination score for each phoneme to decide that the phoneme is part of

low-intelligibility speech. For the next step, the dialogue system with this method should be constructed and have an experiment of it. This method will also reduce the cost of constructing a system because it needs no clustering trees.

6.1.3 Acoustic Event Detection and Recognition

In chapter 4, we focused on a method for extracting paralinguistic and nonverbal information, such as fillers and word fragments. We proposed a technique to detect fillers and word fragments using an LSTM acoustic model and a WFST decoder. The LSTM acoustic model outputs filler and word fragment symbols and phonetic symbols. The proposed decoder can simultaneously recognize speech and detect fillers and word fragments using conventional language models and lexicons without registering all possible word fragments. We showed that the proposed decoder outperformed a conventional decoder using lexicons including word fragments in word fragment detection. The method was only applied to Japanese in this paper. Additional experiments for other languages should be tried in the next step.

The method can provide simultaneous filler, word fragments, and speech recognition results to a spoken dialog system with real-time processing. It will be helpful for it to understand users' states using it. It should be combined with a spoken dialogue system in the next step.

6.1.4 Acoustic Emotion Recognition

In chapter 5, we focused on a method for recognizing emotions, which are paralinguistic and nonverbal information. We aimed at “categorical emotion”, and tackled the improvement of the speech emotion recognition accuracy of the small voice database. In the proposed method, statistical features are calculated from windows with multiple time resolutions, and multiple LSTMs which identify emotions are learned using each calculated feature quantity. By integrating the outputs of these LSTM using XGBoost, a kind of GBDT, the speech emotion classifier corresponding to multiple time resolutions was constructed. The median feature was introduced in the integration.

As an evaluation experiment, each LSTM output and the median feature quantity applied as a contrivance to improve the generalization performance were integrated by GBDT. 82.4% for EmoDB and 76.4% for RAVDESS by a conventional method using a single window, but improved performance with 84.8%, 83.3% by integrating the results of 36 windows. The results showed that the method using multiple time resolution window statistics was effective. When 15 windows that were effective for discrimination were selected using Feature Importance of GBDT, they could be reproduced. When the median feature was used, the best accuracy was obtained for the integration for RAVDESS.

The benchmark of our results and the existing method were carried out, and our method achieved the top performance for RAVDESS at present. On the other hand, the performance of EmoDB was not equal to the top. The high-performance approach in EmoDB makes good use of CNN's components. The hybrid method, which compensates for the weak points of both methods, seems to be effective.

In the experiments, the German emotion-speech database EmoDB and the English emotion-speech database RAVDESS were used as the evaluation sets. There are some other emotion databases in the field. In the future, the Japanese emotion-speech database JTES [109] and English emotion database IEMOCAP[110] will be used for the experiments using our method. Our method adopted bidirectional LSTM for improving performance. However, unidirectional LSTM was better for real-time processing. We will compare them for more practical use.

6.2 Future Research

Our future goal is to build a spoken dialogue system with empathy. For this purpose, it is essential to integrate the methods obtained in this study and make them work as a single system. It is necessary to create a set of modules that are unified in a single language. In many of our experiments, we have created modules in Japanese, but we have not been able to evaluate them in Japanese for emotion recognition. Therefore, we need to conduct experiments in Japanese using a database for emotion recognition such as JTES [109].

Each of these methods can output incrementally when a certain length of speech is obtained, or when one word of the speech recognition result is obtained. However, for the emotion recognition method, incremental output requires a little more effort to integrate multiple time resolutions. A simple method would be to use the longest time window with the smallest time resolution, but it is necessary to check the degree of degradation of recognition accuracy in that case. If the output is incremental, it can be input to a time-series neural network as a feature, and a neural network such as LSTM may have possible to construct an empathetic dialogue system using the features of this paralinguistic and nonverbal information.

In this paper, we focused on estimating the user's state based on paralinguistic and non-verbal information obtained from acoustic signals. When estimating emotions, we should consider methods such as [111] and [41], which combine text and audio to improve performance. However, while there is a large amount of data available for text [112][113][114][115], there are only a few types of databases including both speech and emotion. Therefore, integrating the results of each module built with text and speech may be effective.

6.3 Conclusion

We proposed paralinguistic and nonverbal information extraction methods towards empathetic dialogue systems from speech signals. In this research, we extracted paralinguistic and nonverbal information such as emotions, speaking style, and speaker attributes towards a human-like empathetic dialogue system. Understanding human states is an important factor in realizing empathetic dialogue systems. These components are a clue for understanding human states using speech only. The recognition performance was improved by real-time processing for each component to compensate for no real-time processing and the low performance of conventional methods. The proposed methods' output can be incrementally input to a time series neural network such as LSTM. In the future, we would like to use this research as a basis for further research and development toward the realization of human-like agents.

Acknowledgments

Special thanks are due to my thesis supervisor Professor Katsunobu ITOU for his invaluable support and guidance through the hard moments of the doctor course. I am also grateful to my dissertation committee: Professor Kaoru Uchida and Professor Runhe Huang for their support, valuable feedback, and insightful ideas to this research. I would like to thank co-authors of the publications. Finally, I would like to thank my family for their support.

Publications

Refereed papers

- Fujimura, H. Simultaneous Gender Classification and Voice Activity Detection Using Deep Neural Networks. In *Proc. INTERSPEECH 2014*, 2014, pp. 1139 – 1143.
- Fujimura, H., Shinohara, Y., and Masuko, T. N-best rescoring by phoneme classifiers using subclass adaboost algorithm. In *Proc. Interspeech 2013*, 2013, pp. 3327 – 3331.
- Fujimura, H., Nagao, M., and Masuko, T. Simultaneous speech recognition and acoustic event detection using an LSTM-CTC acoustic model and a WFST decoder. In it Proc. ICASSP 2018, 2018. doi: 10.1109/ICASSP.2018.8461916 pp. 5834 – 5838.
- Fujimura, H., Speech Emotion Recognition by Combining Multiple Discriminators with Multiple Temporal Resolution Features, IEICE Trans. Inf. & Syst. (Japanese Edition), vol. J105-D, no. 03, Mar. 2022.
  doi: 10.14923/transinfj.2021PDP0009 (Student Excellent Paper)

Unrefereed papers

- Fujimura, H., Itou, K., Takeda, K., and Itakura, F. (2004). In-car speech recognition experiments using a large-scale multi-mode dialogue corpus. In *Proc. Independent Components Analysis (ICA)*, 4, 2583-2586.
- Fujimura, H., Itou, K., Takeda, K., and Itakura, F. (2004). Analysis of In-car speech recognition experiments using a large-scale multi-mode dialogue corpus. In *Eighth International Conference on Spoken Language Process (INTERSPEECH, 2004)*.
- Fujimura, H., Miyajima, C., Itou, K., Takeda, K., and Itakura, F. (2005, March). Analysis of a large in-car speech corpus and its application to the multi-model ASR. In *Proc.ICASSP 2005*, IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005. (Vol. 1, pp. I-445). IEEE.
- Fujimura, H., Kawaguchi, N., Matsubara, S., Itou, K., and Takeda, K. (2005, April). Analysis of a large in-car speech corpus. In *21st International Conference on Data Engineering Workshops (ICDEW'05)*, 2005, (pp. 1209-1209). IEEE.

- Fujimura, H., Miyajima, C., Kawaguchi, N., Itou, K., Takeda, K., and Itakura, F. (2006, August). Characterizing in-Car Conversational Speech of Different Dialogue Modes. In *First International Conference on Innovative Computing*, Information and Control-Volume I (ICICIC'06) (Vol. 2, pp. 552-556). IEEE.
- Fujimura, H., Masuko, T., and Tachimori, M. (2010). A duration modeling technique with incremental speech rate normalization. In *Proc. Interspeech 2010*, 2010, pp. 2962 – 2965.
- Fujimura, H., Nakamura, M., Shinohara, Y., & Masuko, T. (2011, December). N-Best rescoring by adaboost phoneme classifiers for isolated word recognition. In *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*, (pp. 83-88). IEEE.
- Fujimura, H., Ding, N., Hayakawa, D., and Kagoshima, T. (2020, January). Simultaneous Flexible Keyword Detection and Text-dependent Speaker Recognition for Low-resource Devices. In *Proc. ICPRAM* (pp. 297-307).
- Takeda, K., Fujimura, H., Itou, K., Kawaguchi, N., Matsubara, S., and Itakura, F. (2005). Construction and evaluation of a large in-car speech corpus. IEICE transactions on information and systems, 88(3), 553-561.
- Fume, K., Ashikawa, T., Watanabe, N., and Fujimura, H. (2018, July). Implementation of Automatic Captioning System to Enhance the Accessibility of Meetings. In *Proc. International Conference on Computers Helping People with Special Needs* (pp. 187-194). Springer, Cham.
- Doddipatla, R., Kagoshima, T., Do, C. T., Petkov, P., Zorila, C., Kim, E., Hayakawa, D., Fujimura, H., Stylianou, Y. (2018, September). The Toshiba entry to the CHiME 2018 challenge. In *The 5th International Workshop on Speech Processing in Everyday Environments (CHiME 2018) in INTERSPEECH*, 2018.
- Hayakawa, D., Masuko, T., and Fujimura, H. (2018, November). Applying Complex-Valued Neural Networks to Acoustic Modeling for Speech Recognition. In *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)* (pp. 1725-1731). IEEE.
- Kobayashi, Y., Yoshida, T., Iwata, K., Fujimura, H., & Akamine, M. (2018, December). Out-of-domain slot value detection for spoken dialogue systems with context information. In *2018 IEEE Spoken Language Technology Workshop (SLT)* (pp. 854-861). IEEE.
- Yoshida, T., Iwata, K., Fujimura, H., & Akamine, M. (2019). Dialog state tracking for unseen values using an extended attention mechanism. In *9th International Workshop on Spoken Dialogue System Technology* (pp. 77-89). Springer, Singapore.
- Kobayashi, Y., Yoshida, T., Iwata, K., & Fujimura, H. (2019). Slot Filling with Weighted Multi-Encoders for Out-of-Domain Values. In *Proc. INTERSPEECH 2019* (pp. 854-858).

- Kagoshima, T., Ding, N., and Fujimura, H. (2020, December). Adaptive Noise Suppression for Wake-Word Detection by Temporal-Difference Generalized Eigenvalue Beamformer. In *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)* (pp. 805-809). IEEE.
- Iwata, K., Yoshida, T., Fujimura, H., & Akamine, M. (2021). Transfer Learning for Unseen Slots in End-to-End Dialogue State Tracking. In *Increasing Naturalness and Flexibility in Spoken Dialogue Interaction: 10th International Workshop on Spoken Dialogue Systems* (pp. 53-65). Springer Singapore.
- Yoshida, T., Iwata, K., Kobayashi, Y., & Fujimura, H. (2021). Dialog State Tracking with Incorporation of Target Values in Attention Models. In *Conversational Dialogue Systems for the Next Decade* (pp. 117-128). Springer, Singapore.

References

[1] H. Fujimura, M. Nagao, and T. Masuko, "Simultaneous speech recognition and acoustic event detection using an LSTM-CTC acoustic model and a WFST decoder," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5834–5838. doi: 10.1109/ICASSP.2018.8461916

[2] H. Fujimura, "Speech emotion recognition by combining multiple discriminators with multiple temporal resolution features," *IEICE Trans. Inf. & Syst. (Japanese Edition)*, vol. J105-D, no. 03, Mar. 2022. doi: 10.14923/transinfj.2021PDP0009

[3] T. Chen and R. Rao, "Audio-visual integration in multimodal communication," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 837–852, 1998. doi: 10.1109/5.664274

[4] U.-H. Kim, "Noise-Tolerant Self-Supervised Learning for Audio-Visual Voice Activity Detection," in *Proc. INTERSPEECH 2021*, 2021, pp. 326–330. doi: 10.21437/Interspeech.2021-43

[5] J. Streeck, "Gesture as communication I: Its coordination with gaze and speech," *Communications Monographs*, vol. 60, no. 4, pp. 275–299, 1993.

[6] T. Kawahara, K. Inoue, D. Lala, and K. Takanashi, "Audio-visual conversation analysis by smart posterboard and humanoid robot," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 6573–6577.

[7] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal processing magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[8] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Association for Computational Linguistics, 2019, pp. 4171–4186. doi: 10.18653/v1/n19-1423. [Online]. Available: https://doi.org/10.18653/v1/n19-1423

[9] Y. Ma, K. L. Nguyen, F. Z. Xing, and E. Cambria, "A survey on empathetic dialogue systems," *Information Fusion*, vol. 64, pp. 50–70, 2020.

[10] H. Fujisaki, "Prosody, models, and spontaneous speech," in *Computing prosody*. Springer, 1997, pp. 27–42.

[11] K. Maegawa and N. Kitagawa, "How does speech transmit paralinguistic information?" *Cognitive Studies (in Japanese)*, vol. 9, no. 1, pp. 46–66, 2002.

[12] P. Mazaré, S. Humeau, M. Raison, and A. Bordes, "Training millions of personalized dialogue agents," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. Association for Computational Linguistics, 2018, pp. 2775–2779. doi: 10.18653/v1/d18-1298. [Online]. Available: https://doi.org/10.18653/v1/d18-1298

[13] C.-S. Wu, S. C. Hoi, R. Socher, and C. Xiong, "TOD-BERT: Pre-trained natural language understanding for task-oriented dialogue," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 917–929. doi: 10.18653/v1/2020.emnlp-main.66. [Online]. Available: https://aclanthology.org/2020.emnlp-main.66

[14] Y. Zhang, S. Sun, M. Galley, Y.-C. Chen, C. Brockett, X. Gao, J. Gao, J. Liu, and B. Dolan, "DIALOGPT : Large-scale generative pre-training for conversational response generation," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Online: Association for Computational Linguistics, Jul. 2020, pp. 270–278. doi: 10.18653/v1/2020.acl-demos.30. [Online]. Available: https://aclanthology.org/2020.acl-demos.30

[15] K. R. Scherer, "Vocal communication of emotion: A review of research paradigms," *Speech communication*, vol. 40, no. 1-2, pp. 227–256, 2003.

[16] S. G. Koolagudi and K. S. Rao, "Emotion recognition from speech: a review," *International journal of speech technology*, vol. 15, no. 2, pp. 99–117, 2012.

[17] M. A. Pilon, K. W. McIntosh, and M. H. Thaut, "Auditory vs visual speech timing cues as external rate control to enhance verbal intelligibility in mixed spastic ataxic dysarthric speakers: a pilot study," *Brain injury*, vol. 12, no. 9, pp. 793–803, 1998.

[18] J. V. Haxby, E. A. Hoffman, and M. I. Gobbini, "Human neural systems for face recognition and social communication," *Biological psychiatry*, vol. 51, no. 1, pp. 59–67, 2002.

[19] F. Lingenfelser, J. Wagner, T. Vogt, J. Kim, and E. André, "Age and gender classification from speech using decision level fusion and ensemble based techniques," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[20] D. Issa, M. F. Demirci, and A. Yazici, "Speech emotion recognition with deep convolutional neural networks," *Biomedical Signal Processing and Control*, vol. 59, p. 101894, 2020.

[21] A. Muppidi and M. Radfar, "Speech emotion recognition using quaternion convolutional neural networks," in *Proc. ICASSP 2021*. IEEE, 2021, pp. 6309–6313.

[22] S. Mirsamadi, E. Barsoum, and C. Zhang, "Automatic speech emotion recognition using recurrent neural networks with local attention," in *Proc. ICASSP 2017*, 2017, pp. 2227–2231.

[23] Z. Zhao, Y. Zheng, Z. Zhang, H. Wang, Y. Zhao, and C. Li, "Exploring spatio-temporal representations by integrating attention-based bidirectional-LSTM-RNNs and FCNs for speech emotion recognition," in *Proc. INTERSPEECH 2018*, 2018, pp. 272–276. doi: 10.21437/Interspeech.2018-1477. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1477

[24] M. Chen, X. He, J. Yang, and H. Zhang, "3-D convolutional recurrent neural networks with attention model for speech emotion recognition," *IEEE Signal Processing Letters*, vol. 25, no. 10, 2018.

[25] Y. Zhang, J. Du, Z. Wang, J. Zhang, and Y. Tu, "Attention based fully convolutional network for speech emotion recognition," in *Proc. APSIPA ASC 2018*. IEEE, 2018, pp. 1771–1775.

[26] Y. Chiba, T. Nose, and A. Ito, "Multi-stream attention-based BLSTM with feature segmentation for speech emotion recognition," in *Proc. INTERSPEECH 2020*, 2020, pp. 3301–3305. doi: 10.21437/Interspeech.2020-1199

[27] R. Nishimura and C. Tafuji, "A proposal of identifying children and adults from the speech based on dnn for voice-enabled web system," in *Proc. Spring Meeting of the ASJ (in Japanese)*. ASJ, 2014, pp. 139–140.

[28] M. Nakamura, K. Iwano, and S. Furui, "Differences between acoustic characteristics of spontaneous and read speech and their effects on speech recognition performance," *Computer Speech & Language*, vol. 22, no. 2, pp. 171–184, 2008.

[29] A. Ganapathiraju, J. E. Hamaker, and J. Picone, "Applications of support vector machines to speech recognition," *IEEE transactions on signal processing*, vol. 52, no. 8, pp. 2348–2355, 2004.

[30] J. Padrell-Sendra, D. Martín-Iglesias, and F. Díaz-de María, "Support vector machines for continuous speech recognition," in *2006 14th European Signal Processing Conference*. IEEE, 2006, pp. 1–4.

[31] K. Maekawa, "Corpus of spontaneous japanese: Its design and evaluation," in *ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, 2003.

[32] H. Inaguma, K. Inoue, M. Mimura, and T. Kawahara, "Social signal detection in spontaneous dialogue using bidirectional LSTM-CTC," in *INTERSPEECH 2017*, 2017, pp. 1691–1695.

[33] K. Hirose, Y. Abe, and N. Minematsu, "Detection of fillers using prosodic features in spontaneous speech recognition of japanese," in *Proceedings of International Conference on Speech Prosody, Dresden*, vol. 2, 2006, pp. 2–5.

[34] T. Asami, R. Masumura, Y. Aono, and K. Shinoda, "Recurrent out-of-vocabulary word detection using distribution of features." in *INTERSPEECH 2016*, 2016, pp. 1320–1324.

[35] M. A. Jalal, E. Loweimi, R. K. Moore, and T. Hain, "Learning Temporal Clusters Using Capsule Routing for Speech Emotion Recognition," in *Proc. INTERSPEECH 2019*, 2019, pp. 1701–1705. doi: 10.21437/Interspeech.2019-3068. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-3068

[36] S. K. Pandey, H. Shekhawat, and S. Prasanna, "Deep learning techniques for speech emotion recognition: A review," in *2019 29th International Conference Radioelektronika (RADIOELEKTRONIKA)*. IEEE, 2019, pp. 1–6.

[37] V. Dissanayake, H. Zhang, M. Billinghurst, and S. Nanayakkara, "Speech emotion recognition 'in the wild' using an autoencoder," *Proc. INTERSPEECH 2020*, pp. 526–530, 2020.

[38] B. T. Atmaja and M. Akagi, "On the differences between song and speech emotion recognition: Effect of feature sets, feature types, and classifiers," in *Proc. 2020 IEEE REGION 10 CONFERENCE (TENCON)*, 2020, pp. 968–972.

[39] T. Sivanagaraja, M. K. Ho, A. W. Khong, and Y. Wang, "End-to-end speech emotion recognition using multi-scale convolution networks," in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2017, pp. 189–192.

[40] Z. Aldeneh and E. M. Provost, "Using regional saliency for speech emotion recognition," in *ICASSP 2017*. IEEE, 2017, pp. 2741–2745.

[41] Z. Peng, Y. Lu, S. Pan, and Y. Liu, "Efficient speech emotion recognition using multi-scale CNN and attention," in *ICASSP 2021*. IEEE, 2021, pp. 3020–3024.

[42] P. Nantasri, E. Phaisangittisagul, J. Karnjana, S. Boonkla, S. Keerativittayanun, A. Rugchatjaroen, S. Usanavasin, and T. Shinozaki, "A light-weight artificial neural network for speech emotion recognition using average values of mfccs and their derivatives," in *Proc. ECTI-CON 2020*, 2020, pp. 41–44.

[43] D. H. Wolpert, "Stacked generalization," *Neural networks*, vol. 5, no. 2, pp. 241–259, 1992.

[44] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[45] H. Fujimura, "Simultaneous gender classification and voice activity detection using deep neural networks," in *Proc. INTERSPEECH 2014*, 2014, pp. 1139–1143. doi: 10.21437/Interspeech.2014-291

[46] Y.-M. Zeng, Z.-Y. Wu, T. Falk, and W.-Y. Chan, "Robust GMM based gender classification using pitch and RASTA-PLP parameters of speech," in *2006 International conference on machine learning and cybernetics*. IEEE, 2006, pp. 3376–3379.

[47] J. Sas and A. Sas, "Gender recognition using neural networks and asr techniques," *Journal of Medical Informatics & Technologies*, vol. 22, 2013.

[48] K. Markov and S. Nakamura, "Never-ending learning system for on-line speaker diarization," in *2007 IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*. IEEE, 2007, pp. 699–704.

[49] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[50] N. Ryant, M. Liberman, and J. Yuan, "Speech activity detection on youtube using deep neural networks." in *Proc. INTERSPEECH 2013*. Lyon, France, 2013, pp. 728–731.

[51] X.-L. Zhang and J. Wu, "Denoising deep neural networks based voice activity detection," in *ICASSP 2013*. IEEE, 2013, pp. 853–857.

[52] Y. Kida, M. Sakai, T. Masuko, and A. Kawamura, "Robust F0 estimation based on log-time scale autocorrelation and its application to Mandarin tone recognition," in *Tenth Annual Conference of the International Speech Communication Association*, 2009.

[53] N. Kitaoka, T. Yamada, S. Tsuge, C. Miyajima, K. Yamamoto, T. Nishiura, M. Nakayama, Y. Denda, M. Fujimoto, T. Takiguchi *et al.*, "CENSREC-1-C: An evaluation framework for voice activity detection under noisy environments," *Acoustical Science and Technology*, vol. 30, no. 5, pp. 363–371, 2009.

[54] T. Kawahara, H. Nanjo, T. Shinozaki, and S. Furui, "Benchmark test for speech recognition using the corpus of spontaneous Japanese," in *ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, 2003.

[55] "ATR-promotions web site," http://www.atr-p.com/sdb.html.

[56] P. Ding and L. He, "Improve the implementation of pitch features for Mandarin digit string recognition task," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[57] C.-P. Chen, J. A. Bilmes, and K. Kirchhoff, "Low-resource noise-robust feature post-processing on aurora 2.0." in *Proc. INTERSPEECH 2002*, 2002.

[58] M. Fujimoto, K. Ishizuka, and T. Nakatani, "A voice activity detection based on the adaptive integration of multiple speech features and a signal decision scheme," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2008, pp. 4441–4444.

[59] K. Ishizuka and T. Nakatani, "Study of noise robust voice activity detection based on periodic component to aperiodic component ratio." in *SAPA@ INTERSPEECH*. Citeseer, 2006, pp. 65–70.

[60] H. Fujimura, Y. Shinohara, and T. Masuko, "N-best rescoring by phoneme classifiers using subclass adaboost algorithm," in *Proc. INTERSPEECH 2013*, 2013, pp. 3327–3331. doi: 10.21437/Interspeech.2013-736

[61] D. Povey and P. C. Woodland, "Minimum phone error and i-smoothing for improved discriminative training," in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1. IEEE, 2002, pp. I–105.

[62] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted MMI for model and feature-space discriminative training," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2008, pp. 4057–4060.

[63] G. Heigold, R. Schluter, and H. Ney, "Modified MPE/MMI in a transducer-based framework," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2009, pp. 3749–3752.

[64] A. Ragni and M. J. Gales, "Inference algorithms for generative score-spaces," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 4149–4152.

[65] H. Fujimura, M. Nakamura, Y. Shinohara, and T. Masuko, "N-best rescoring by adaboost phoneme classifiers for isolated word recognition," in *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*. IEEE, 2011, pp. 83–88.

[66] A. Ganapathiraju, J. Hamaker, and J. Picone, "Support vector machines for speech recognition," in *Proc. ICSLP*, 1998.

[67] S. Wiesler, G. Heigold, M. Nußbaum-Thom, R. Schlüter, and H. Ney, "A discriminative splitting criterion for phonetic decision trees," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[68] K. C. Sim, "Probabilistic state clustering using conditional random field for context-dependent acoustic modelling," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[69] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[70] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, vol. 1. IEEE, 2001.

[71] S. Z. Li, L. Zhu, Z. Zhang, A. Blake, H. Zhang, and H. Shum, "Statistical learning of multi-view face detection," in *European Conference on Computer Vision*. Springer, 2002, pp. 67–81.

[72] C. Huang, H. Ai, Y. Li, and S. Lao, "Vector boosting for rotation invariant multi-view face detection," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, vol. 1. IEEE, 2005, pp. 446–453.

[73] L. Ding and A. M. Martinez, "Features versus context: An approach for precise and detailed detection and delineation of faces and facial features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 11, pp. 2022–2038, 2010.

[74] S. M. Siniscalchi, J. Li, and C.-H. Lee, "A study on lattice rescoring with knowledge scores for automatic speech recognition," in *Ninth International Conference on Spoken Language Processing*, 2006.

[75] K. Schutte and J. Glass, "Speech recognition with localized time-frequency pattern detectors," in *2007 IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*. IEEE, 2007, pp. 341–346.

[76] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey *et al.*, "The htk book," *Cambridge university engineering department*, vol. 3, no. 175, p. 12, 2002.

[77] N. Kumar and A. G. Andreou, "Heteroscedastic discriminant analysis and reduced rank hmms for improved speech recognition," *Speech communication*, vol. 26, no. 4, pp. 283–297, 1998.

[78] T. Shinozaki, C. Hori, and S. Furui, "Towards automatic transcription of spontaneous presentations," in *Proc. Eurospeech*, vol. 1, 2001, pp. 491–494.

[79] A. Rastrow, A. Sethy, and B. Ramabhadran, "A new method for oov detection using hybrid word/fragment system," in *Proc. ICASSP 2009*. IEEE, 2009, pp. 3953–3956.

[80] H.-K. Kuo, E. E. Kislal, L. Mangu, H. Soltau, and T. Beran, "Out-of-vocabulary word detection in a speech-to-speech translation system," in *Proc. ICASSP 2014*. IEEE, 2014, pp. 7108–7112.

[81] Y. Tsvetkov, Z. Sheikh, and F. Metze, "Identification and modeling of word fragments in spontaneous speech," in *Proc. ICASSP 2013*. IEEE, 2013, pp. 7624–7628.

[82] Y. Nasu and H. Fujimura, "Acoustic event detection and removal using LSTM-CTC for speech recognition (in Japanese)," in *IEICE Tech. Rep.,PRMU2016-69*, vol. IEICE-116, 2016, pp. 121–126.

[83] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Proc. ICASSP 2016*. IEEE, 2016, pp. 4945–4949.

[84] S. Kim, T. Hori, and S. Watanabe, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in *Proc. ICASSP 2017*. IEEE, 2017, pp. 4835–4839.

[85] R. Prabhavalkar, K. Rao, T. N. Sainath, B. Li, L. Johnson, and N. Jaitly, "A comparison of sequence-to-sequence models for speech recognition," in *Proc. INTERSPEECH 2017*, 2017, pp. 939–943.

[86] Y. Miao, M. Gowayyed, and F. Metze, "EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding," in *Proceedings of the ASRU 2015*. IEEE, 2015, pp. 167–174.

[87] P. R. Dixon, C. Hori, and H. Kashioka, "A specialized WFST approach for class models and dynamic vocabulary," in *Proc. INTERSPEECH 2012*. ISCA, 2012.

[88] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. ICML*, 2006, pp. 369–376.

[89] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *International conference on machine learning*. PMLR, 2016, pp. 173–182.

[90] M. Nagao, "Generation device, recognition device, generation method, and computer program product," u. S. Patent Publication No. 2016/0155440.

[91] C. Allauzen and M. Riley, "Rapid vocabulary addition to context-dependent decoder graphs," in *Proc. INTERSPEECH 2015*. ISCA, 2015, pp. 2112–2116.

[92] M. Mohri, F. Pereira, and M. Riley, "Weighted finite-state transducers in speech recognition," in *ASR-2000*. ISCA, 2000, pp. 97–106.

[93] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. Fifteenth Annual Conference of the International Speech Communication Association (INTERSPEECH2014)*. ISCA, 2014, pp. 338–342.

[94] A. Nagaoka, H. Mori, and Y. Arimoto, "Utilizing existing labels for automatic cross-corpus emotion labeling (in Japanese)," *The Journal of the Acoustical Society of Japan*, vol. 73, no. 11, pp. 682–693, 2017.

[95] E. Loweimi, M. Doulaty, J. Barker, and T. Hain, "Long-term statistical feature extraction from speech signal and its application in emotion recognition," in *Proc. International Conference on Statistical Language and Speech Processing*, 2015, pp. 173–184.

[96] S. R. Livingstone and F. A. Russo, "The Ryerson audio-visual database of emotional speech and song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English," *PloS one*, vol. 13, no. 5, p. e0196391, 2018.

[97] F. Eyben, K. R. Scherer, B. W. Schuller, J. Sundberg, E. André, C. Busso, L. Y. Devillers, J. Epps, P. Laukka, S. S. Narayanan *et al.*, "The Geneva minimalistic acoustic parameter set (GeMAPS) for voice research and affective computing," *IEEE transactions on affective computing*, vol. 7, no. 2, pp. 190–202, 2015.

[98] Z. Peng, J. Dang, M. Unoki, and M. Akagi, "Multi-resolution modulation-filtered cochleagram feature for LSTM-based dimensional emotion recognition from speech," *Neural Networks*, vol. 140, pp. 261–273, 2021.

[99] Q. Mao, M. Dong, Z. Huang, and Y. Zhan, "Learning salient features for speech emotion recognition using convolutional neural networks," *IEEE transactions on multimedia*, vol. 16, no. 8, pp. 2203–2213, 2014.

[100] F. Burkhardt, A. Paeschke, M. Rolfes, W. F. Sendlmeier, and B. Weiss, "A database of German emotional speech," in *Proc. INTERSPEECH 2005*, 2005, pp. 1517–1520.

[101] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[102] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, "Deep learning for hate speech detection in tweets," in *Proceedings of the 26th international conference on World Wide Web companion*, 2017, pp. 759–760.

[103] B. M. Pavlyshenko, "Linear, machine learning and probabilistic approaches for time series analysis," in *2016 IEEE First International Conference on Data Stream Mining & Processing (DSMP)*. IEEE, 2016, pp. 377–381.

[104] M. Bieshaar, S. Zernetsch, A. Hubert, B. Sick, and K. Doll, "Cooperative starting movement detection of cyclists using convolutional neural networks and a boosted stacking ensemble," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 4, pp. 534–544, 2018.

[105] Z. Ouyang, X. Sun, J. Chen, D. Yue, and T. Zhang, "Multi-view stacking ensemble for power consumption anomaly detection in the context of industrial internet of things," *IEEE Access*, vol. 6, pp. 9623–9631, 2018.

[106] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[107] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, vol. 8, 2015, pp. 18–25.

[108] B. Caterina, L. Diana Van, and D. Irene, "The contribution of speech rate and pitch variation to the perception of vocal emotions in a german and an american sample," *Cognition and Emotion*, vol. 15, no. 1, pp. 57–79, 2001. doi: 10.1080/02699930126095. [Online]. Available: https://doi.org/10.1080/02699930126095

[109] E. Takeishi, T. Nose, Y. Chiba, and A. Ito, "Construction and analysis of phonetically and prosodically balanced emotional speech database," in *2016 Conference of The Oriental Chapter of International Committee for Coordination and Standardization of Speech Databases and Assessment Techniques (O-COCOSDA)*. IEEE, 2016, pp. 16–21.

[110] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "IEMOCAP: Interactive emotional dyadic motion capture database," *Language resources and evaluation*, vol. 42, no. 4, pp. 335–359, 2008.

[111] Z.-J. Chuang and C.-H. Wu, "Multi-modal emotion recognition from speech and text," in *International Journal of Computational Linguistics & Chinese Language Processing, Volume 9, Number 2, August 2004: Special Issue on New Trends of Speech and Language Processing*, 2004, pp. 45–62.

[112] J. Bollen, H. Mao, and A. Pepe, "Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena," in *Proceedings of the international AAAI conference on web and social media*, vol. 5, no. 1, 2011.

[113] R. C. Balabantaray, M. Mohammad, and N. Sharma, "Multi-class twitter emotion classification: A new approach," *International Journal of Applied Information Systems*, vol. 4, no. 1, pp. 48–53, 2012.

[114] B.-K. H. Vo and N. Collier, "Twitter emotion analysis in earthquake situations," *International Journal of Computational Linguistics and Applications*, vol. 4, no. 1, pp. 159–173, 2013.

[115] A. Arora, P. Chakraborty, M. Bhatia, and P. Mittal, "Role of emotion in excessive use of Twitter during COVID-19 imposed lockdown in India," *Journal of Technology in Behavioral Science*, vol. 6, no. 2, pp. 370–377, 2021.