Taiwanese Speech Recognition Based on Hybrid Deep Neural Network Architecture

Yu-Fu Yeh , Bo-Hao Su , Yang-Yen Ou , Jhing-Fa Wang
Department of Electrical Engineering
National Cheng Kung University Tainan, Taiwan
n26070106@gs.ncku.edu.tw , xtlettle99360017@gmail.com , ouyang0916@gmail.com , wangjf@mail.ncku.edu.tw

An-Chao Tsai
Department of Computer Science and Entertainment Technology
Tajen University, Pingtung, Taiwan
actsai@tajen.edu.tw

Abstract

In this research, we developed the Taiwanese speech recognition system which used the Kaldi toolkit to implement. The Taiwanese corpus was collected by Taiwan Taiwanese National Reading Competition and Classmate Recording, and a total of about 11 hours of audio files were collected. Because the training data is small dataset, two audio augmentation methods are used to increase the training data, so that the acoustic model can be more robust and more effective training. One method is speed perturbation, which speeds up the original data by 1.1 times and slows it down by 0.9 times. Another method is to use multi-condition training data to simulate reverberation of the original speech and add background noise. The background noise includes music, speech, and noise. The acoustic model is trained for different hybrid deep neural network architectures which can use the advantages of each neural network by hybrid different neural networks, including TDNN, CNN-TDNN and CNN-LSTM-TDNN. In the experimental results, the CER in the domain of language modeling reaches 3.95%, and the CER of online decoding test is 3.06%. Compared with other researches on Taiwanese speech recognition of similar dataset size, the recognition results are better than other studies.

Keywords: Speech Recognition, Taiwanese, Data Augmentation, Deep Neural Network Acoustic Model.

1. Introduction

In the past few years, more and more products using speech recognition technology. Because these speech recognition applications make people's lives more and more convenient, no longer need to type to allow the machine to receive our message input. Taiwanese language is one of the commonly used languages of Taiwanese. From [1], we can know that in 2013, the social change survey results showed that 31.4% of the people in the family spoke Mandarin Chinese most often, and 44.2% of them spoke Taiwanese most often. 19.5% of the people advocate using both Mandarin and Taiwanese, but the proportion of the older generation is much larger than that of the younger generation, which shows that Taiwanese is still the main language for the elderly. Most of them are learning by word of mouth, which leads to relatively scarce resources in Taiwanese. It makes the research of Taiwanese-related technologies much more difficult, and also causes people who speak Taiwanese to not enjoy these conveniences. Therefore, we have established a Taiwanese dataset and a Taiwanese speech recognition system for this problem.

2. Related Work

Establishes a deep neural network architecture in kaldi[2], the input features used in addition to the Mel frequency cepstral coefficients[3] will also concatenate the ivector feature [4], which is a feature vector that can represent the speaker. First, a general background model is trained on the data of all speakers. The universal background model(UBM) is a Gaussian mixture model containing many components, and then the UBM is modified with the speech features of different speakers to achieve the speaker adaptation model, and the expected values of each Gaussian component are concatenated to form a GMM super-vectors. A section of GMM super-vectors can be used to represent the feature vector of a speaker. Finally, the GMM super-Vectors of the general background model are related to the speaker. GMM super-Vectors calculates ivectors.

In recent years, more and more DNNs have been used to replace GMM to increase the modeling capabilities of acoustic models, indicating that DNN-HMM is better than traditional GMM-HMM, and Kaldi continues to update the DNN architecture to build acoustic models, such as Time Delay Neural Network (TDNN) [5], CNN-TDNN or LSTM-TDNN [6], etc. TDNN is a deep neural network structure. It can include historical and future outputs and model long-term dependent speech signals. It was first proposed to classify phonemes in speech signals. Used

for speech recognition [7]. For TDNN, increasing the number of layers allows the network to capture features for a longer period of time; usually it is desirable to deepen the number of network layers of TDNN to achieve better results. However, previous experiments have found that the deeper the network is, the more often the problem of degradation is, so that the increase in the depth of the neural network will result in a decrease in accuracy. Therefore, another TDNN network architecture [8] is proposed. The Matrix Factorization training of the network can make the network training more stable, in order to achieve better speech recognition performance.

Traditional Discriminative Training requires cross-entropy training to obtain a lattice, which must take extra time. Therefore, the extended framework of CTC is proposed, Lattice-free maximum mutual information [9]. The principle is the same as the method of MMI, the formula is as formula (1), and the following changes are made: (a) The denominator FST uses training text to generate a 4-gram phone language model, and does not use backoff less than 3-gram, instead of lattice.(b) Use different training techniques to avoid Overfitting, like: L2 regularization on the network output, Cross-entropy regularization and Leaky HMM

$$F^{MMI} = \sum_u \log P(S_u|O_u, \lambda) = \sum_u \log \frac{P(O_u|S_u,\lambda)P(S_u)}{\sum_{S'} P(O_u|S',\lambda)P(S')} \quad (1)$$

3. Proposed system

The overall architecture of this system is shown in Figure 1, which include Pre-processing, Deep Neural Networks Acoustic model, Decoding Graph and Recognition.

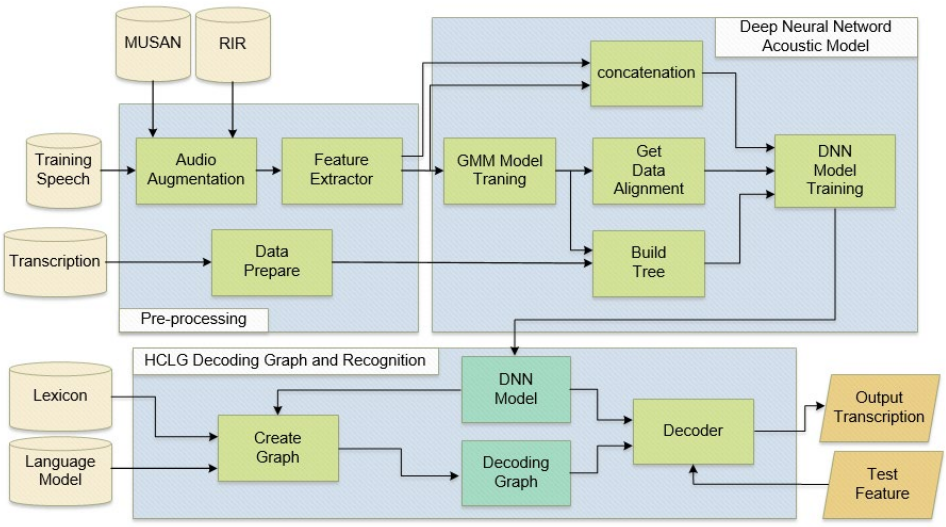

Figure 1. System Flow diagram

3.1 Pre-processing

In the field of speech recognition, the data augmentation is commonly used to increase the quantity of training data, avoid overfitting and improve robustness of the models. The system uses two types of data augmentation, including speed perturbation [10] and using multi-condition training data [11].In this system, the speed perturbation first generates 3 times the amount of original data, and then this data generates 15 times the amount of original data by adding multi-conditional background noise. Increase the original 10 hours of training data to 150 hours.

The 39-dimensional MFCC feature is used in the GMM-HMM system, and with the addition of Cepstral Mean and Variance Normalization, the standard features of mean 0 and Variance 1 are obtained to solve the effects of different microphones and audio channels. The DNN-HMM system uses high resolution MFCC and ivector. The ivector extraction process is: (1) use 40-dimensional features and 512 Gaussian training diagonal universal background model to obtain final.dubm (2) use the obtained UBM to train ivector extractors (3) Use ivector extractors to extract the ivector of each training data. Feature parameter of TDNN architecture is shown in Figure 2. In order to obtain more context information when inputting deep neural network training, the input will be (t-1, t, t+1) three times 40-dimensional high resolution MFCC feature stitching, followed by 100-dimensional ivector features.

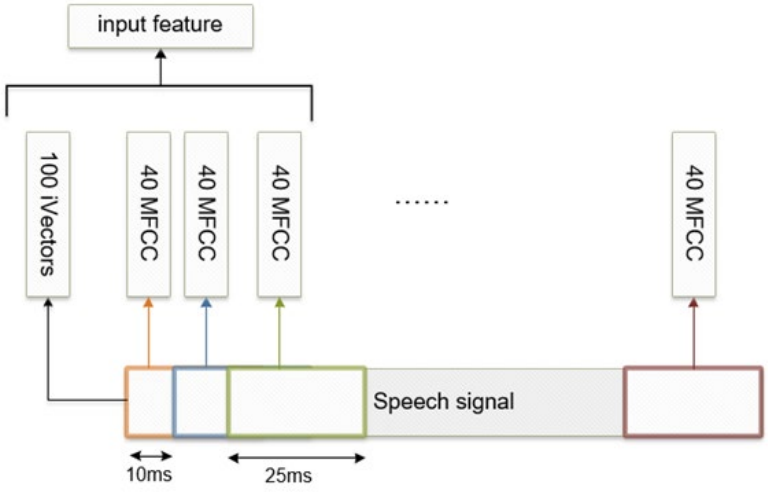

Figure 2. TDNN input feature

The feature parameters of CNN-TDN and CNN-LSTM-TDNN architecture will first convert 40-dimensional high resolution MFCC into Mel-FilterBanks features through Inverse discrete cosine transform layer. Linear transform a 100-dimensional ivector into a 200-dimensional ivector and concatenate with Mel-FilterBanks features. Finally, convert 240-dimensional input features into 40*6. input feature map, such as Figure 3.

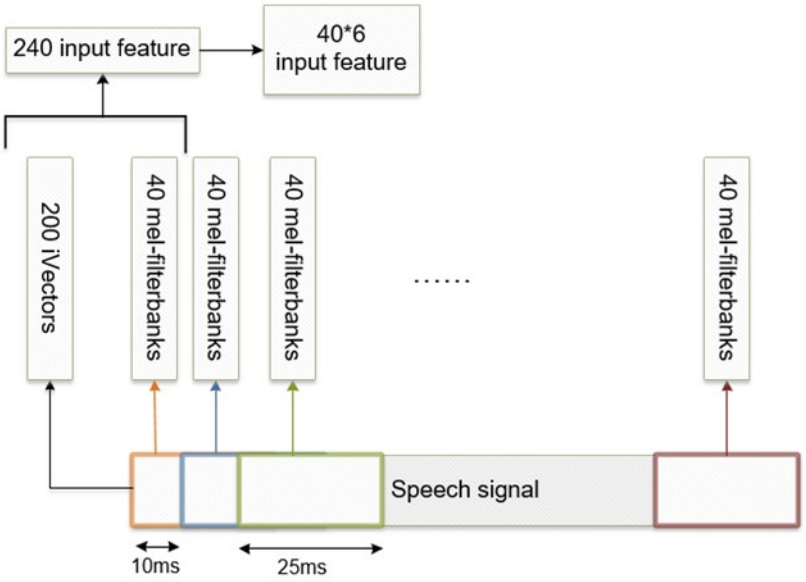

Figure 3. CNN-TDNN and CNN-LSTM-TDNN input feature

3.2 Deep Neural Networks Acoustic model

In this chapter, before training the deep neural network acoustic model, the GMM-HMM system must be pre-trained, and the alignment result obtained by the GMM-HMM system should be used as the training target of the deep neural network acoustic model. This system models HMM at the phone level. Each phone HMM model has 3 states. In the Taiwanese Pinyin system, there are 85 phones including initials and finals. If the tone is considered, the system has 299 HMM models and GMM-HMM model training steps are mono, tri1, tri2, tri3[2].

This research establishes three DNN architectures, including (a) TDNNF, (b) CNN-TDNNF, (c) CNN-LSTM-TDNN.

3.2.1 TDNNF Architecture

The data alignment obtained by the GMM-HMM system is used to establish a decision tree, and the number of leaves corresponds to the output dimension of the deep neural network. Therefore, the architecture output dimension of this chapter is 2776. The TDNNF architecture is shown in Figure 4. This architecture refers to the WSJ recipe and uses the TDNNF architecture proposed by Povey, Daniel, et al. [9], which uses a total of 13 layers of TDNNF layer. The first layer will be 100-dimensional ivector Features and three consecutive 40-dimensional MFCC features make up a total of 220-dimensional input features. Layers 2-4 are (t-1 , t , t+1) three-time input vectors, and layers 6-13 are (t-3 , t ,t+3) three time input vectors, so each frame output can get the information of the first 28 frames and the last 28 frames. The dimension of each layer is 1024, and the SVD decomposition dimension is 128. The internal

architecture of each TDNNF block is shown in Figure 3-10, and the output is divided into chain output and Cross-Entropy output as shown in Figure 4.

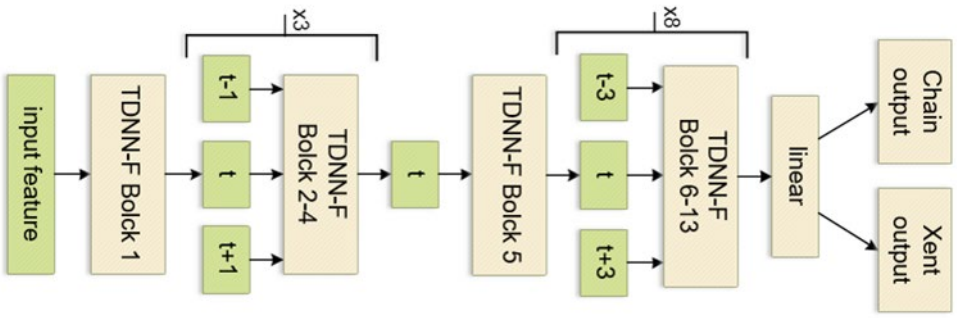

Figure 4. TDNN Architecture

3.2.2 CNN-TDNNF Architecture

In this section, the TDNN architecture used in section 3.3.1 is added to the CNN architecture. The CNN operation method is shown in Figure 5, which is characterized by a 40*6 matrix, and consists of 3 consecutive times to form 3* 40*6 three-dimensional input matrix, before doing convolution, first zero-padding the height to become a 3*42*6 matrix, using 48 3*3*6 size filters for convolution, the output is the first layer The output of the convolutional layer, if there is subsampling, will only reduce the dimension of the height.

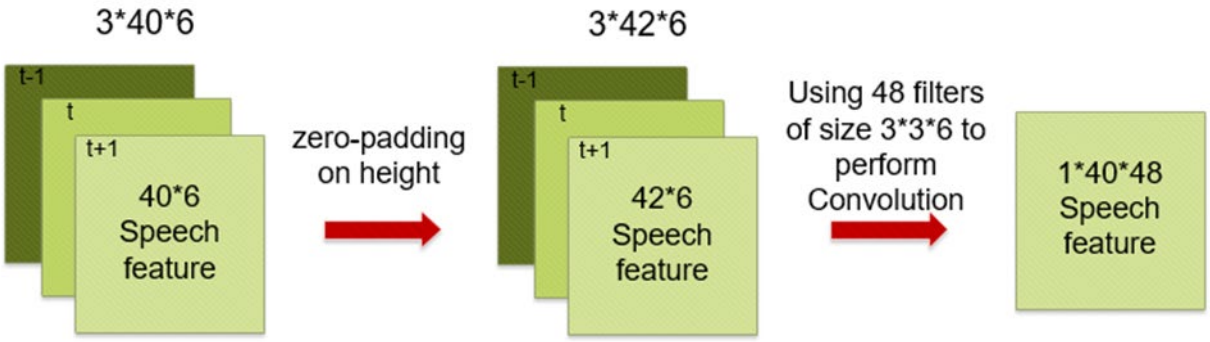

Figure 5. Convolution Neural Network Operation

The output dimension of the CNN-TDNNF architecture in this chapter corresponds to the number of decision tree leaves is 2776. The CNN-TDNNF architecture is shown in Figure 6. This architecture refers to the mini_librispeech recipe, uses 6 layers of convolution layer, and the first layer receives three consecutive times 40*6 The dimension of the speech feature matrix is 3*40*6. After the first layer of convolution layer operation, the output is 1*40*48. After that, each layer uses three consecutive input times, and at the 3rd, 5th and 6th layers, the height will be subsampled, and finally the output dimension will be 1*5*128. After the CNN, 9-layer

TDNNF layer is used, where each layer has a dimension of 1024, and the SVD decomposition dimension is 128 dimensions, and each layer of TDNNF layer uses (t-3, t, t+3) three Time is used as the input vector, so each output can get the information of the first 30 frames and the last 30 frames.

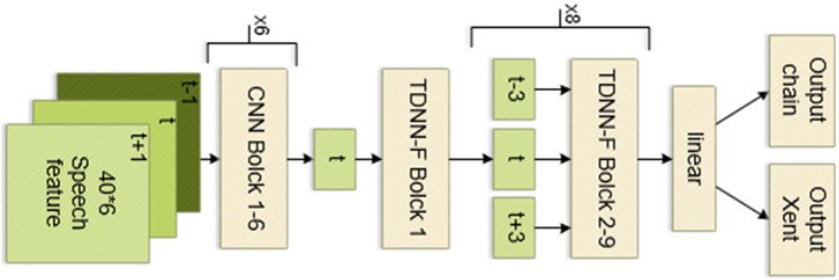

Figure 6. CNN-TDNNF Architecture

3.2.3 CNN-LSTM-TDNN Architecture

CNN-LSTM-TDNN is a deep neural network architecture designed by us. Using CNN can effectively extract feature parameters from a small corpus, and LSTM can model the advantages of long-term sequences to find out this deep neural network architecture. The output dimension of the CNN-LSTM-TDNN architecture in this chapter corresponds to the number of decision tree leaves is 2776. The CNN-LSTM-TDNN architecture is shown in Figure 7, which uses 6 layers of convolution layer, 8 layers of TDNN layer and 2 layers of LSTM layer, among which the convolution layer The architecture parameters are the same as in CNN-TDNNF. The TDNN layer is a general TDNN non-matrix decomposition, and the LSTM cell dimension is 1024. The dimensions of the recurrent and non-recurrent projection layer are all 256, so the input gate, forget gate and output gate input are 1024+256=1280, and then the 1024-dimensional vector is output to the Cell state and Hidden state through the Nonlinear activation function, and the final output is r_t and p_t concatenation. Each TDNN layer has a dimension of 1024 and receives (t-3, t, t+3) three time inputs, so each output can get the information of the first 33 frames and the last 33 frames, and finally output to the chain output and Cross-Entropy output.

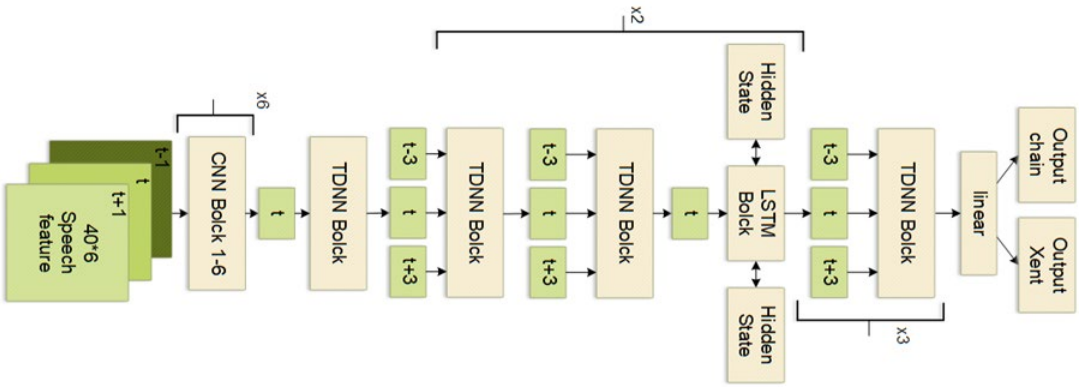

Figure 7. CNN-LSTM-TDNN Architecture

4. Experimental Results

4.1 Taiwanese Dataset

The audio files and the corresponding transcriptions of Taiwanese characters were collected by Taiwan Taiwanese National Reading Competition [12] and recorded by classmates in the laboratory. The sampling frequency is 16kHz, the sampling accuracy is 16bit, and the number of channels is 1 (mono). 11.23 hours, 10439 utterances, 101596 syllables. The experimental part divides the corpus into 10.22 hours of training data and 1.01 hours of testing data.

Lexicon grabs each word and corresponding phone from Taiwanese transcription, and adds the Taiwanese vocabulary of the text of the language model to lexicon, a total of 31331 words are obtained.

The language model text dataset of this system has about 540,000 words, including: 29724 uni-grams, 22123 bi-grams, and 66118 tri-grams.

4.2 Evaluation Method

We use Character Error Rate metrics to evaluate model accuracy in Taiwanese. Character Error Rate (CER), is a common metric of the performance of a speech recognition or machine translation system. The formulas to calculated accuracy as follows:

$$CER = \frac{S+D+I}{N} = \frac{S+D+I}{S+D+C} \tag{2}$$

Where S is the number of substitutions, D is the number of deletions, I is the number of insertions, C is the number of correct words, N is the number of words in the reference (N=S+D+C).

4.3 Experimental Results

The Taiwanese corpus of this system is a small dataset, and the amount of data is far from the size of other language corpus, so consider whether to use tone to label phones. If tones are considered in the part of the dataset marked with phones, the number of phones will increase. This will cause more HMM models, and vice versa, reduce the number of HMM models. And this experiment is to explore whether tones are added to the dataset as the Taiwanese corpus used by this system. The number of HMM models without adding tones is 86, which is much smaller than the HMM with adding tones, which is 299. This experiment uses 1.01 hours of testing data to test the performance of the model. The testing data has a total of 9692 syllables. Table 1 shows the experimental results of whether the corpus adds tones. It can be seen from the experimental results that in the case of the same language model, the traditional GMM-HMM system does not add tones better than tones, but the acoustic model of the deep neural network is the opposite. It can be seen that the architecture of the deep neural network has a better ability to model speech signals, so all subsequent experiments will consider the tone as the pinyin label of the Taiwanese corpus.

Table 1. Compare whether the dataset adds tones

Model	With tone(CER%)	Without tone(CER%)
Mono	36.32	**29.94**
Tri1	27.24	**26.06**
Tri2	24.39	**23.68**
Tri3	19.13	**17.33**
TDNNF	**10.21**	12.12

However, in training speech recognition systems, overfitting problems are often encountered. In order to solve this problem, the easiest way is to add training data. But this is a thorny problem in the case of limited data and manpower, so training data is often obtained through data augmentation. This experiment compares two data augmentation methods, including speed perturbation[10] and using multi-condition training data[11]. The increase in training data can make the model training deeper and more efficient. Table 2 shows the comparison results of adding different data augmentation methods. The first column is a different system. Take the TDNNF architecture as the acoustic model and add SP and multi respectively, where SP stands for speed perturbation and MULTI stands for using multi-condition training data,

The second column shows the amount of data after data augmentation. The TDNNF + SP + MULTI system adds SP first and then MULTI, which increases the total amount of data by 15 times. The third and fourth columns represent the character error rate of the testing data. It can be seen that for the same acoustic model system, the character error rate decreases as the data increases.

Table 2. Comparison of data augmentation methods

System	Duration of data (hours)	CER (%)	Error/Total
TDNNF	10	10.21	991 / 9692
TDNNF + SP	30	8.91	864 / 9692
TDNNF + MULTI	50	8.14	789 / 9692
TDNNF + SP + MULTI	150	**7.94**	770 / 9692

In order to explore the impact of different deep neural network models on acoustic models, four sets of models were set up in this experiment, including TDNNF, CNN-TDNNF, LSTM-TDNN, and CNN-LSTM-TDNN. The recognition results of each deep neural network acoustic model are shown in Table 3. It can be seen that the effect of LSTM-TDNN is worse than the other three, and the best model is the CNN-LSTM-TDNN mixed three deep neural network architecture acoustic models. It can be seen that the effect of the LSTM-TDNN architecture is very poor, possibly because the number of training data in the corpus is too small. Although LSTM is an algorithm for time series training, it requires too many parameters, so a larger amount of training data is needed for training. The disadvantage of CNN is that there is no concept of time series, and the use of too many parameters leads to an increase in training time. But shows that for acoustic models, convolutional neural networks can effectively help feature extraction in small dataset and overall deep neural network learning.

Table 3. Comparison of different model recognition results

Model	CER (%)	Error/Total	ins	del	Sub
TDNNF	7.94	770 / 9692	79	91	600
CNN-TDNNF	7.68	744 / 9692	80	56	608
LSTM-TDNN	10.20	989 / 9692	91	103	796

CNN-LSTM-TDNN	**7.61**	738 / 9692	88	49	601

Finally, a total of 10 people in the laboratory are asked to do an online decoding test. Each person tests 15 sentences. The test text is a daily language in Taiwanese, and the training text of the language model has been added. There are 150 sentences and 1078 syllables in total. The recognition results are shown in Table 4. The online decoding test text example is shown in Table 5.

Table 4. Online decoding test results

	CER (%)	**Error/Total**	**ins**	**del**	**Sub**
Online decoding test	3.06	33 / 1078	6	9	18

Table 5. Online decoding test text

Testing data number	Text
1	gua2 beh4 khi3 tai5 pak4
2	gua2 siunn7 beh4 tshut4 khi3 tshit4 tho5
3	gua2 siunn7 beh4 tsiah8 mih8 kiann7
4	kin1 a2 lit8 thinn1 khi3 be7 bai2
5	u7 siann2 mih4 ho2 tsiah8 e5
…	...
150	gua2 beh4 khi3 siong2 kho3 ah4

5. Conclusions

In this paper, we collected a Taiwanese corpus and use the architecture of the deep learning HMM model to build a Taiwanese speech recognition system. Finally, the CNN-LSTM-TDNN architecture is the best. The language model can be changed according to the domain used to greatly improve the accuracy. In our experiment, the character error rate of the testing data of inside domain is 3.95%. The experimental results show that if the text is inside domain, the Taiwanese speech recognition system does get good results. Finally, in actual application, the laboratory classmates were asked to test the online decoding character error rate of 3.06%.

References

[1] 葉高華. "臺灣歷次語言普查回顧." 臺灣語文研究 13.2 (2018): 247-273.

[2] Povey, Daniel, et al. "The Kaldi speech recognition toolkit." IEEE 2011 workshop on automatic speech recognition and understanding. No. CONF. IEEE Signal Processing Society, 2011.

[3] Davis, Steven, and Paul Mermelstein. "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences." IEEE transactions on acoustics, speech, and signal processing 28.4 (1980): 357-366.

[4] Dehak, Najim, et al. "Front-end factor analysis for speaker verification." IEEE Transactions on Audio, Speech, and Language Processing 19.4 (2010): 788-798.

[5] Peddinti, Vijayaditya, Daniel Povey, and Sanjeev Khudanpur. "A time delay neural network architecture for efficient modeling of long temporal contexts." Sixteenth Annual Conference of the International Speech Communication Association. 2015.

[6] Peddinti, Vijayaditya, et al. "Low latency acoustic modeling using temporal convolution and LSTMs." IEEE Signal Processing Letters 25.3 (2017): 373-377.

[7] Waibel, Alex, et al. "Phoneme recognition using time-delay neural networks." IEEE transactions on acoustics, speech, and signal processing 37.3 (1989): 328-339.

[8] Povey, Daniel, et al. "Semi-Orthogonal Low-Rank Matrix Factorization for Deep Neural Networks." Interspeech. 2018.

[9] Povey, Daniel, et al. "Purely sequence-trained neural networks for ASR based on lattice-free MMI." Interspeech. 2016.

[10] Ko, Tom, et al. "Audio augmentation for speech recognition." Sixteenth Annual Conference of the International Speech Communication Association. 2015.

[11] Ko, Tom, et al. "A study on data augmentation of reverberant speech for robust speech recognition." 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2017.

[12] "台灣國賽台語(閩南語)朗讀篇目整理,"[Online]. Available: http://ip194097.ntcu.edu.tw/longthok/longthok.asp.