Pronunciation Erroneous Tendency Detection with Combination of Convolutional Neural Network and Long Short-Term Memory

Longfei Yang, Yanlu Xie and Jinsong Zhang
Advanced Innovation Center for Language Resource, Beijing Language and Culture University
15 Xueyuan Road, Haidian District, Beijing 100083, P.R.China
yanglongfei908@gmail.com, xyl@blcu.edu.cn, jinsong.zhang@blcu.edu.cn

Abstract

Computer Assisted Pronunciation Training (CAPT) systems can automatically detect pronunciation problems in the speech by the second language learners, thus are helpful for them to do more pronunciation training. Pronunciation Erroneous Tendencies (PETs) we proposed previously consist of a set of articulation configurations regarding incorrect articulation manners and positions, and their detection could lead to a more instructive guidance than the commonly used scoring ones. Although approaches have shown that PETs could be reliably detected based on Gaussian Mixture-Hidden Markov Model (GMM-HMM) or Deep Neural Network-Hidden Markov Model (DNN-HMM), they also suggested that the proposal be seriously suffering from problems of acoustic variations and data sparsity. To alleviate the problems, we propose a series of techniques for PET detection in this paper: firstly, some features with robustness was extracted by convolutional layer to reduce spectral variation; and then Long Short-Term Memory (LSTM) model was employed for modeling PET in order to handle variations along time. Besides, data augmentation was adopted to lessen the data sparsity; and then All proposals have been experimented and the results suggested that they are effective in the PET detection task.

Keywords

Computer Assisted Pronunciation Training, Mispronunciation detection, Pronunciation Erroneous Tendency, Deep learning.

1. Introduction

Computer Assisted Pronunciation Training (CAPT) technology has been one of the subjects that draws more and more academia's attentions as it can provide some feedbacks information to guide the second language (L2) learner to practice their pronunciation. The conventional feedbacks information proposed for pin-pointing phone mainly contain the pronunciation scores [Zheng et al., 2007; Hu et al.,2013] and the mispronunciations [Harrison et al., 2009; Wang et al., 2012. The pronunciation scores measure the similarity between the L2 learners' pronunciations and the natives' thus it is appropriate to evaluate the overall pronunciation level directly. The mispronunciations are proposed to make the learners pay attention to their erroneous pronunciation by pointing out the phonetic identities of mispronounced phones and providing some information about how to correct the erroneous pronunciations. The conventional mispronunciations are defined as the phone-level insertion, deletion or substitution. However, in reality, our previous research showed that the learners' erroneous pronunciations were difficult to be classified into these certain categories. So, the Pronunciation Erroneous Tendencies (PETs) were proposed in our previous works [Cao et al., 2010]. Compared with the previous two kinds of feedback, PETs represent phone-level erroneous pronunciation ranging from acoustic deviations to category mispronunciations thus can provide more detailed and informative feedbacks. According these feedbacks, the learners can do more targeted practice to correct their erroneous pronunciations rather than stereotype listening and repeating. It is an attractive topic to develop the robust PET detection technologies for building a feedback-rich CAPT system. On account of the similarity of two tasks, the mainstream mispronunciation detection systems are based on automatic speech recognition (ASR) framework. Our previous work demonstrated that PET could be regarded as the additional phones and detected by the statistic-based ASR model, e.g., Gaussian Mixture Model (GMM) model [Duan et al., 2014].

It is a challenge to establish a high-performance PET detection system. There are variety of translational variances in speech signals that make trouble to the detection, such as frequency shift caused by different speakers or different speaking styles, noise and distance. For PET detection, it is necessary to handle these variations. For example, the speaking styles of learners may be different with their common speaking styles since the learners are speaking the second language rather than their native language and it may result in misrecognition of the detection system. So, the acoustic model for PET must be of high accuracy and robustness.

With the success of deep learning method, some state-of-the-art performance for ASR have been achieved. Motivated by this trend, deep neural networks (DNNs) has also been investigated to PET detection and the accuracies of detection are improved [Gao et al., 2015].

But DNNs are not explicitly designed for modeling time sequence and difficult to handle the translation variance which exist in the speech signals. To address this problem, convolutional neural networks (CNNs), which is one of the oldest neural networks and a popular model for handwriting recognition [LeCun et al., 1995], is explored in the field of speech recognition and some improvements are conducted [Sainath et al., 2013; Abdel et al., 2014]. Compared to densely-connected DNNs, CNNs have the ability of reducing translational variance in signals in the manners of the *local connectivity* and the *weight sharing* thus CNNs can learn the feature that of strong robustness against spectral variation in speech signals. However, the deficiency in both DNNs and CNNs is that they utilize less timing information. The speech is the time sequence and some kind of PETs are context sensitive and the spectral representations of speech have strong correlations [Sainath et al., 2013]. In fact, some PETs are the transition state between the standard state and another in the phone level. So, it is important to make use of the context information in the PET detection.

More recently, recurrent neural networks (RNNs) especially of Long Short-Term memory (LSTMs) have been investigated in the field of speech and show its powerful ability of modeling sequence data [Graves et al., 2013; Sak et al., 2014]. LSTMs have some characteristic components referred to as *memory blocks* in the recurrent hidden layer. Within the memory blocks there are some units called *memory cells* with self-connections to store the temporal state of the network in addition to some special units called *gates* to control the flow of activation information. The input and output of memory cells are constructed in a context-sensitive way thus LSTMs can make use of long range context information. Besides, the performance of LSTMs can be further improved by passing some better features to them, such as convolutional features, to them [Bengio et al., 2013].

Additionally, there is another point we pay attention to. The mainstream ASR-based detection frameworks are based on supervised learning which demand plenty of training data and corresponding accurate annotations. But for PET detection, it is very difficult to meet this requirement. The annotation task for PET is time-consuming as the executors must be of proficiency in phonetic and linguistic knowledge and, especially, PET definition [Li et al., 2014]. Insufficiency of training samples and annotations may lead to low performance or over-fitting [Ko et al., 2015].

Given these analysis, we then investigated the approaches to improve the PET detection performance from the aspect of acoustic model and data. We first explored the application architecture of LSTMs to this task. And then, to introduce the convolutional feature to reduce the translational variance, LSTM and CNN were combined into a unified system in which convolutional layers were employed as the feature extractor. Besides, to address the data sparsity problem, the data augmentation was introduced to increase the size of the training

set and enhance the diversity of the data.

The rest of this paper was organized as follows. Description of the PET and the frame work of PET detection system were presented in Section 2. The experiment setup was listed in the Section 3 and Section 4 demonstrated the experimental results and analysis. Finally, Section 5 concluded this paper.

2. Pronunciation Erroneous Tendencies (PETs) detection

2.1. Pronunciation Erroneous Tendencies (PETs)

In some researches, some common mispronunciation patterns in the second language learning are investigated [Wang et al., 2004; Xie et al., 2010; Li et al., 2011]. Most of these patterns are mainly caused by the inaccurate places of articulation or the erroneous uttering manners when pronunciations are produced. In the process of that the learners practice their pronunciation of the target language, the articulation-placement which exists in the learners' mother tongue is adopted consciously or unconsciously due to the effect from the negative transfer of mother tongue [Cao et al., 2010]. Besides, it is also very difficult for learners to master the uttering articulation or manners which do not exist in their mother tongue. Most of their erroneous pronunciation are some of that deviate a little from the canonical pronunciations. They have some difference with the substitution of phoneme categories thus these erroneous pronunciations cannot be simply classified into the certain classes, e.g., substitution, deletion and insertion. To address the problem, Pronunciation Erroneous Tendency (PET) is proposed.

PETs define a set of incorrect articulation configurations from the aspect of the articulators and uttering manners in the phone level, e.g., advancing, backing, rounding, spreading, centralizing, raising, lowering, labio-dentalizing, laminalizing, devoicing, voicing, insertion, deletion, stopping, fricativizing, nasalizing, retroflexing and etc [Cao et al., 2009]. These erroneous articulation tendencies are informative for CAPT systems to provide the instructive feedbacks to the learners and make the CAPT system function more informatively. For example, if the detection system detects a PET of "lip rounding" in the learner's pronunciation, it will send a notification message to tell the learners to pronounce the phone with a "spreading" lip. In this process, the learners will realize the different between pronunciation produced by them and the standard pronunciation and learn how to correct the mispronunciation. An illustration of PET of lip spreading and rounding was shown in Figure 1 as follows.

Lip: spread rounding
Pinyin: 'e' 'o'
Rounded 'e' pronunciation: 'e{o}'
Spreading 'o' pronunciation: 'o{w}'

Figure 1: An Illustration of the PET of lip spreading and rounding.

On the basis of the BLCU-CAPT-1 annotation rules, the annotation task was performed on the Japanese part of our large-scale Chinese as the second language speech database (referred to as the BLCU Inter-Chinese speech corpus). Parts of the PET definition were shown in Table 1 as follows.

PET	**Diacritics**	**E.g.**	**Notation**
Spreading	w	u{w}	The round sound 'u' has a spreading lip.
Rounding	o	e{o}	The spreading lip sound 'e' is pronounced the round sound.
Backing	-	n{-}	The tongue position of phoneme is a little back.
Advancing	+	e{+}n	The tongue position of phoneme 'e' is a little advancing so the pronunciation of 'en' is like that of 'n'.
Shortening	;	p{;}	The aspiration duration of phoneme 'p' is a little shorter.
Lengthening	:	z{:}	The fricativizing duration of phoneme 'z' is a little longer.
Laminalizing	sh	sh{sh}	Balade-palatal phoneme 'sh' is pronounced like the Japanese limina-alveolar.
Labio-dentalizing	f	u{f}	The pronunciation of phoneme 'u' is pronounced like 'v'.

Table 1: Parts of definitions of PETs

2.2 PET detection framework

The data flow illustration of the detection system was shown in Figure 2. The system prompts the learners to speak the specified utterance whose corresponding texts are provided by the system. The learners' speech signals are sent to the ASR-based detection module. The detection module recognizes and decodes the speech in the phone level according to the acoustic model and the extended pronunciation network. And then the system judges the pronunciation according to the difference between the recognized phone-level transcription and the correct utterance. At last, the decision module searches for corresponding information from mispronunciation database and provides feedback messages (e.g. which phone is wrong and how to rectify) to speakers.

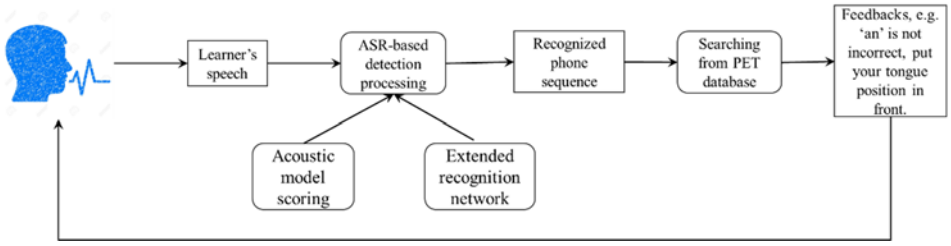

Figure 2: An illustration of PET detection framwork.

2.3 Extended pronunciation network

Extended pronunciation network is a representation of pronunciation variants in the form of a network. The pronunciations of every Chinese word (or Character) are extended in the dictionary according to all of the possible PET annotations. The corresponding extended pronunciation network will be constructed with all of the possible pronunciations when the system gives the prompts to speakers. Figure 3 demonstrates an example for extended pronunciation network.

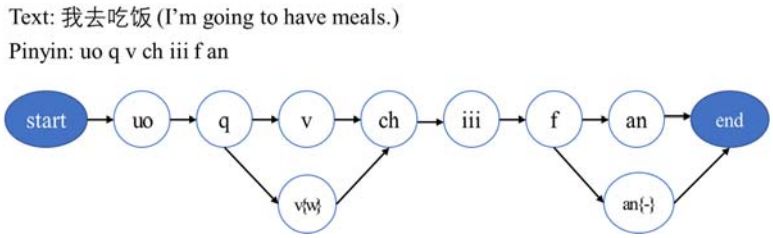

Figure 3: Extended pronunciation network.

2.4 Acoustic model

The acoustic model plays the same role as that of in the ASR system. It is employed to classify the acoustic features into the corresponding classes of the HMM states which have been aligned by force alignment.

2.4.1 Long Short-Term Memory (LSTM)

To model variable-length sequences, Recurrent Neural Network (RNN) h is proposed and has been used for various tasks including language modeling and speech recognition [Graves et al., 2013]. As shown in Figure. 4, unlike some feedforward neural networks (FFNNs) like DNN, the characteristic property of RNN is to allow the connections between the hidden layers at different steps. The architecture of RNN has the cycles sending the activation information from the previous time steps as the input to the network to make the decision for the current time steps. The way of utilizing contextual information between the FFNNs and RNNs is different. In FFNNs, the input features over the fixed contextual windows are spliced into a super-vector sent to the network. The approaches for utilizing contextual information of RNN is that the activation information from the previous time step are stored in the network. In this manner, RNNs employs a dynamically changing contextual windows and it allows RNNs to utilized long rage contextual information and makes RNNs better suited for sequence modeling.

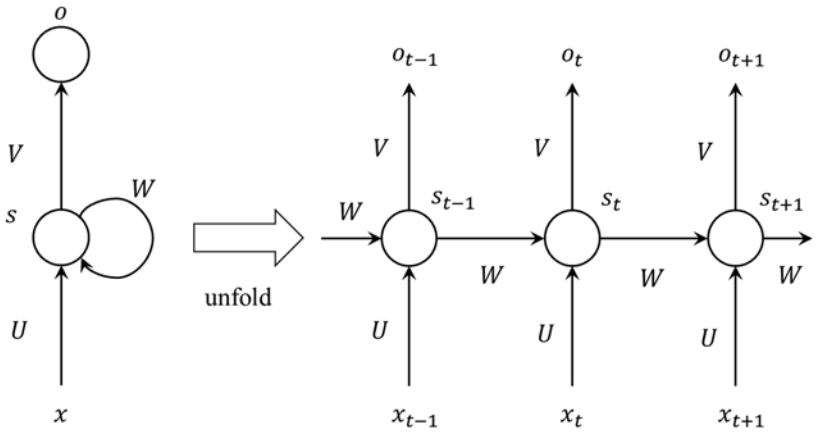

Figure 4: An illustration of Recurrent Neural Networks (RNN).

However, the gradient-based back-propagation through time (BPTT) algorithm, which is the training approach for RNNs, is difficult to handle the vanishing gradient and exploding problems [Bengio et al., 1994]. Besides, long range context dependencies also limit the

capability of RNNs. To address these problems, LSTM is designed in a more elegant way. An illustration of LSTM structure is shown in Figure. 5.

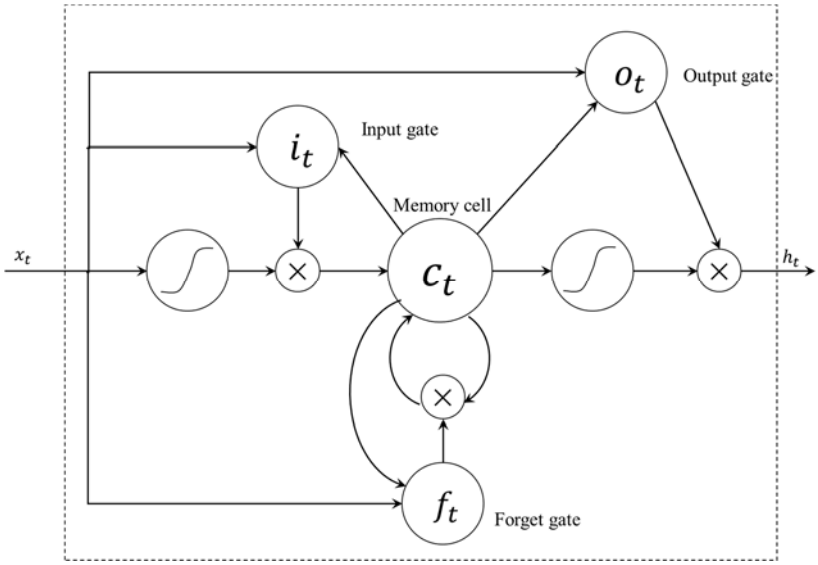

Figure 5: An illustration of the memory blocks of Long Short-Term Memory (LSTM).

LSTM consists of the distinct units called *memory blocks* in the recurrent hidden layers. The *memory blocks* compose of the *memory cells* which are used to store the temporal state of the network and gates which controls the flow of activation information, i.e., the *input gate* controls which input flow in and *output gate* controls which to participate in the computation of the output activation and the *forget gate* determines the extent of the information kept in the memory cells.

Typically, given the input sequence $X = (x_1, x_2, \dots, x_T)$, the computations of activation information in the components of the LSTM layers are performed from the time step $t = 1$ to $t = T$. The computation at the time step t can be described as follows:

$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + W_{ic}c_{t-1} + b_i) \tag{3}$$

$$f_t = \sigma\left(W_{fx}x_t + W_{fh}h_{t-1} + W_{fc}c_{t-1} + b_f\right) \tag{4}$$

$$c_t = i_t \odot \emptyset(W_{cx}x_t + W_{ch}h_{t-1} + b_c) + f_t \odot c_{t-1} \tag{5}$$

$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + W_{oc}c_t + b_o) \tag{6}$$

$$a_t = o_t \odot h(c_t) \tag{7}$$

$$y_t = W_{ya} a_t + b_y \tag{8}$$

Where the i_t, f_t, o_t, c_t denote the state of the input gate, forge gate, output gate and memory cell respctively. The $W_{i(\cdot)}$ and the b_i are corresponding weights matrix and bias between the input gate i_t and the input sequence respectively; the $W_{f(\cdot)}$ and the b_f are parameters of the forget gate f_t; The $W_{o(\cdot)}$ and b_o are weights and bias of the output gate o_t that precites the preceding states with the input x_t at the current time step and the state of the hidden layers h_{t-1} and the state of the memory cell c_{t-1} at the previous tiem step. The y_t is the output of the memory blocks and W_{ya} and b_y are weight matrix and bias of output units respctively. a_t denotes the output activation vector of the memory cell. $\odot$ is the element-wise product of the vector. σ and $\emptyset$ are activation functions, generally $sigmoid$ and $tanh$.

2.4.2 Convolutional Neural Networks (CNNs)

Figure. 6 demonstrates the CNN architecture of the acoustic model for our PET detection system. In detail, compared with fully-connected deep neural networks (DNNs), CNNs have three main properties*: local connectivity*, *weight sharing* in convolutional layers and the pooling layers.

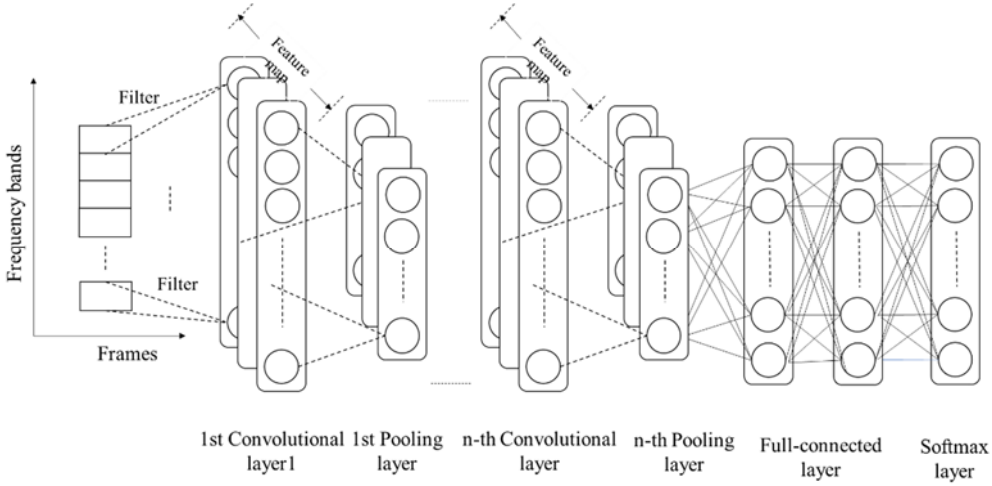

Figure 6: An illustration of Convolutional Neural Network (CNN).

In convolutional layers, feature extraction is performed in a scheme of *local connectivity*. Each convolutional kernel (neuron or unit and generally referred as a weight matrix) in the convolutional layers receives the input feature from only a local region from the previous layer. The window through which we capture the input and multiply the weight matrix will

generally overlap. The outputs of the multiplication with the same kernel are constructed as a feature map in which the convolutional operation share the same weight (referred as *weight sharing*). This process can be described as follows:

$$y_j^l = \varphi\left(\sum_{i \in M_i} x_i^{l-1} k_j^l + b_j^l\right) \tag{1}$$

Where the y_j^l denotes the j-th feature map in the l-th layer, M_i is the set of feature maps in the $(l-1)$-th layer. The x_i^{l-1} is the i-th kernel in the $(l-1)$-th layer, k_j^l denotes the corresponding weight in the weight matrix and b_j^l is the bias in the l-th layer. φ is the activation function. The right-hand side of the Equation (1) can be referred as the convolutional operation. With local connectivity, each kernel of convolutional layer only extracts local information in the lower layer and the variance will be handled when combination of these local features in the higher layer. Besides, the parameters of the network are reduced through weight sharing that can also avoid over-fitting. In this manner, the model will handle robustness against translational variance along frequency in the speech signals, which may be caused by non-white noise, distortion, mismatch due to different speakers and different speaking styles, since such the variance may only occur in some specified frequency bands.

Following the convolutional layer, down-sampling is performed in the pooling layer generally. The structure of pooling layer is similar with that of convolutional layer except that the size of the kernel is smaller and there is no overlap between the kernels. Some common pooling functions are max-pooling, mean-pooling and stochastic pooling. Max-pooling was applied to our model and it can be described as follows:

$$y_j^l = \max_{p \in P} x_p^{l-1} \tag{2}$$

Where the y_j^l denotes the j-th feature map in the l-th layer and the x_p^{l-1} is the input from the $(l-1)$-th layer. P is the pooling size used to determine the length of the pooling window through which the operation outputs the max value. The purpose of pooling operation is to reduce the feature dimension to keep off over-fitting and it can reduce some small variances which exists in the input features as well.

2.4.3 Convolutional Long Short-Term Memory (CNN-LSTM)

As shown in Figure. 7, the CNN-LSTM is constructed with the combination of CNN and LSTM into a unified system. The outputs of the convolutional layers are regarded as the input passed to the LSTM layers. In this scheme, the convolutional layers play the role of feature extractor to provide better features to reduce the spectral variances along the frequency axis

and the LSTM layers handle the variance along the temporal axis. As for the LSTM layers, the LSTMP layers are employed. The difference between the conventional LSTM layers and the LSTMP layers is that the LSTMP layer has a separate linear projection layer after the LSTM layer to reduce the computational complexity.

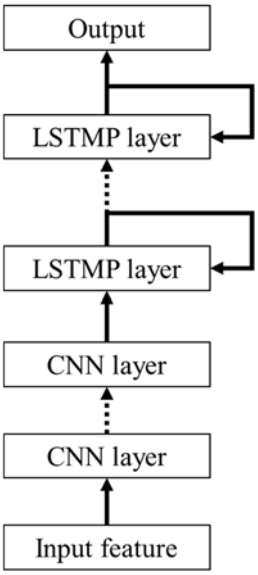

Figure 7: The architecture of CNN-LSTM

3. Experiment

3.1 Database

3.1.1 Corpus

In our experiment, parts of the Japanese side from BLCU Inter-Chinese speech corpus were employed as the database in which 80% were employed as the training set and the rest for testing. Table.2 demonstrated some statistics of the database.

Content	**Quantity**
Text	301
Speakers	7
Utterance (total)	1899
Phonemes (total)	26431
Annotators	6

Table 2: The statistics of training database.

The phonetically annotations with PET diacritics were performed at the segment level. The annotators were 6 post-graduate students majoring in phonetics and they were divided into two groups. Each speech data was annotated twice independently by two groups, with each annotator labeling a continuous 200 utterances on a rotating basis.

3.1.2 Data augmentation

To alleviate the data sparsity problem and construct a more robust system, data augmentation was carried out before the training process. A speed perturbation with the factor of 0.9, 1.0 and 1.1, a stretch perturbation, which changes the speed without changing the speakers' pitch, with the factor of 0.9, 1.0 and 1.1, and the pitch scaling with the factor of 0.85 and 1.25 were performed on the training set. And then the perturbed data were combined with the original training set into a unified large-scale training database.

3.2 Evaluation metric

The recall and precision were employed as the evaluation metric as the number of PETs was much less than that of the correct pronunciation. The recall measured that, among all of the phones labeled as the PETs manually, how many PETs were detected by the detection system. The precision denoted that how many PETs were labeled by both the detection system and the human. Detection Accuracy (DA) was to measure the accuracy for all phones including PETs and the correct pronunciation of the learners. F1-score was used since we consider that the precision and recall were equally important for the language learners when using CAPT systems. Under ideal condition, the recall and the precision should be improved at the same time, but, in reality, there was a trade-off. Among all of the 65 kinds of PETs that occurred in the corpus, 16 most common PETs were selected for analysis since there were fewer samples of the other PETs that might lead to the unreliability of the analysis.

$$Recall = \frac{N_{Shot}}{N_{Human}} \times 100\%, \tag{9}$$

$$Precision = \frac{N_{Shot}}{N_{Machine}} \times 100\%, \tag{10}$$

$$F1 - score = 2 \times \frac{Recall \times Precision}{Recall + Precision}, \tag{11}$$

$$DA = \frac{C}{T} \times 100\%. \tag{12}$$

Where N_{Shot} denoted the number of the phones labeled as the PETs by human at the same time detected as the PETs by the detection system. N_{Human} was the total number of phones that were annotated by the annotators. $N_{Machine}$ was the total number of the phones which

were detected as the PETs by the detection system and it might contain the real PETs and that of the mistakenly recognition. C denoted the numbers of the phones which were correctly recognized by the detection system and it contained the PETs and the correct pronunciation produced by the learners. T was the total number of all the phones.

3.3 Model setup

The 40-dimension Mel-Frequency Cepstral Coefficient (MFCC) feature was employed as the input feature. These spectral features were extracted through a 25ms windows with a 10ms frame shift. All the features sets were applied cepstral mean and variance normalization (CMVN) before training.

DNN-HMM was established as the baseline system. The DNN had 3 hidden layers and each layer consisted of 1,024 Rectified Linear Units. The 40-dimension MFCC features extracted from 4 preceding frames, the current frame and the 4 succeeding frames were spliced into a super-vector feature sent to the network. Minimizing the cross-entropy (CE) loss with the mini-batch stochastic gradient descent (mini-batch SGD) based back-propagation (BP) algorithm was used to train the network. The batch size was set to 128 and the learning rate was initialized as 0.001 and an exponentially decaying manner was adopted to scale the learning rate during the process of training.

The CNN-HMM was constructed with 3 convolutional layers and 2 fully-connected layers. The convolutional layers had 64, 128, 256 convolutional kernels sequentially and the size of all the kernels was 3. The convolutional operation along both the frequency axis and the time axis (referred to as 2-D), only along the frequency axis (referred to as 1-D) were explored. Following each convolutional layer, there was a pooling layer consisting of non-overlapping max pooling units whose size was 2. Pooling operation was performed only along the frequency axis in the case of 1-D convolution was applied along the frequency axis and along both axes when 2-D convolution was used along. Zero padding with corresponding size was performed if necessary for some convolutional layers. The input feature passed to CNN and the training algorithm were the same as that of DNN except that the computations of back-propagation for convolutional layers and pooling layers were different with the version of DNN.

The LSTMP was constructed with 3 layers where each layer had 1024 memory blocks. The LSTMP was unfolded for 20 time steps and the output state label was delayed by 3 frames. The input feature passed to LSTMP layers was a single 40-dimensional MFCC feature vector without splicing. The LSTMP network was trained using the SGD-based back-propagation through time (BPTT) algorithm.

For CNN-LSTM, after the feature extraction performed by the convolutional layers,

the outputs of the convolutional layer were passed to LSTMP layers in the manner of that the LSTMP layers replaced the location of the DNN layers in the CNN architecture mentioned above. The configuration of the convolutional layers was the same as that of CNN model and the LSTMP layers had the same setup as that of LSTMP model. The input feature was the single 40-dimensional MFCC feature vector as well. When computation of back-propagation was performed, the BPTT version was employed for the LSTMP layers and the convolutional layers applied the CNN version.

4. Experimental results and Discussions

The experimental results were shown in Table. 3 below.

Acoustic model	**Recall**	**Precision**	**F1-score**	**DA**
DNN baseline	48.4%	62.6%	54.6	88.0%
DNN (after augmentation)	51.1%	64.5%	57.1	88.4%
1-D CNN	51.7%	64.2%	57.3	88.8%
2-D CNN	52.7%	63.8%	57.7	89.1%
LSTMP	54.1%	66.9%	59.8	89.9%
CNN-LSTMP	55.2%	70.4%	61.9	90.4%

Table 3. Detection performance for different models.

From the Table. 3, we found that the data augmentation was effective for the improving the performance of PETs detection. The recall after the augmentation had a 2.7% absolute improvement and the precision gets a 1.9% absolute improvement and, at the same time, a slight increasement of DA was achieved. CNN got a 0.6% increasement of recall but a 0.3% reducing of precision and the DA of CNN was better than DNN overall. 2-D CNN significantly increased the recall but the cost of precision was too much at the same the training time of 2-D CNN was longer than that of 1-D CNN because the convolutional operation needed to be performed along both axes. So, after the weighting the advantages and disadvantages, 1-D CNN was adopted for the establishment of CNN-LSTMP. It was demonstrated that the LSTMP contributed in a delightful improvement for the PET detection by increasing a 3% absolute increasement of recall, a 2.4% rise of precision and the 1.5% improvement of DA compared to the DNN baseline after augmentation. In addition, the performance of LSTMP could be further improved by adding additional convolutional layers which provided more better features to reduce the spectral variance. It kept consistence with

[Bengio et al., 2014].

For a detailed analysis, we divided the 16 most common PETs into four broad groups, i.e.

- Rounding and spreading of the shape of the lip: pronunciations with spreading lips have problems of rounding tendency or pronunciations with the round lips have wrong articulation manner of spreading tendency.
- Advancing and backing of the position of the tongue: the position of the tongue is a little advance or back.
- Lengthening and shortening: the duration of the aspiration or the constriction is too short or too long.
- Laminalizing and labio-dentalizing: the balade-palatal phonemes are pronounced like the Japanese lamina-alveolar.

Figure. 8-11 demonstrated the detection results performed by the different models for four groups of PETs mentioned above.

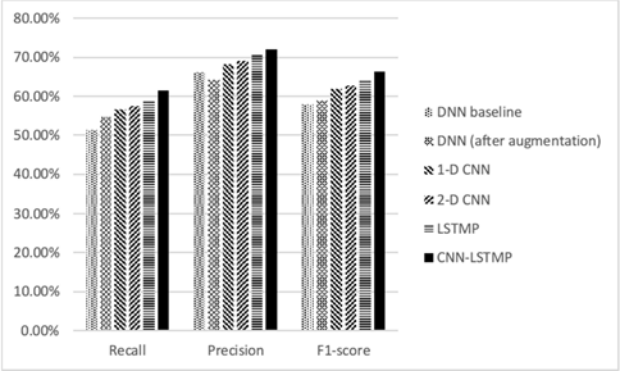

Figure 8: Detection performance for laminalizing and labio-dentalizing.

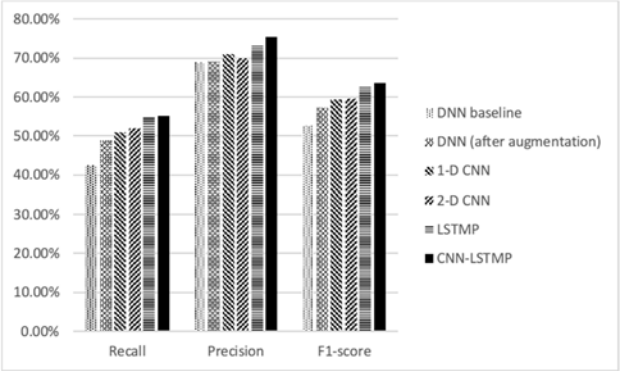

Figure 9: Detection performance for lengthening and shortening.

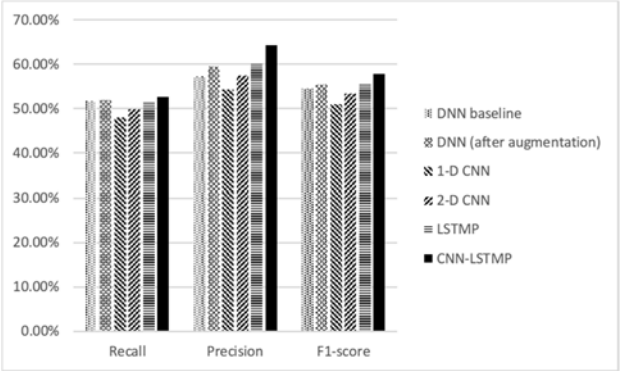

Figure 10: Detection performance for advancing and backing of the position of the tongue.

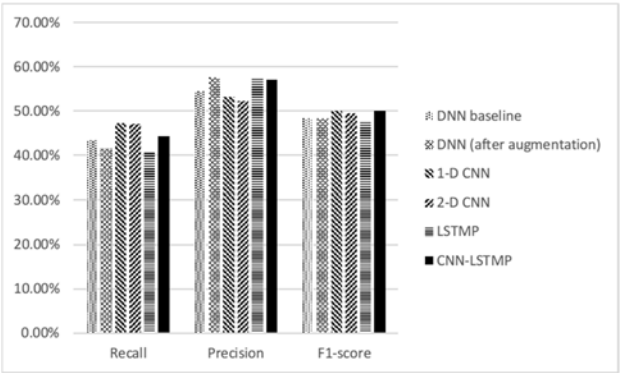

Figure 11: Detection performance for rounding and spreading of the shape of the lip.

In detail, the data augmentation performed very well for detecting shortening and lengthening but mediocrely for the advancing and backing. It may be because that the PETs of shortening and lengthening were mainly reflected in the duration of some phoneme and our augmentation approaches are effective for extend this type of erroneous pronunciation. The recall of rounding and the spreading got an obvious decline and the precision of laminalizing and labio-dentalizing was slightly reduced. For the detection for these two PETs, the augmentation approaches employed in our experiments might be inappropriate. We also found that CNNs were more effective for the detecting the laminalizing and labio-dentalizing and lengthening and shortening but it was not the case for the advancing and backing. The LSTM-based models outperformed the other models significantly except for the rounding and spreading lip. In addition, the convolutional features could further improve the performance of LSTM, our proposed CNN-LSTM model achieved the best performance on the whole.

5. Conclusion and Future Work

Aiming at implementing a high-performance PET detection system, we constructed the system with the CNNs which could provide the better feature with the robustness against the spectral variance and the LSTMs which had strong ability of modeling sequence in to a unified detection system. Besides, data augmentation was introduced to alleviate the data sparsity problem. As a consequence, the best performance from the CNN-LSTM with an absolute improvement of 6.8% in recall, 7.8% in precision and 2.4% in detection accuracy was achieved compared to the DNN baseline. At the same time, the results showed that the detection for some of the PETs was still a challenge. In our future works, more accurate acoustic model will be explored. Besides, more approaches like semi-supervised learning or unsupervised learning will be investigated for PET detection task to deal with the data sparsity problem.

6. References

W. Hu, Y. Qian and F. K. Soong, “A new DNN-based high quality pronunciation evaluation for computer-aided language learning (CALL),” INTERSPEECH 2013, - 14th Annual Conference of the International Speech Communication Association, August 25-29, Lyon, France, Proceedings, 2013, pp. 886-1890.

J. Zheng, C. Huang and M. Chu, “Generalized Segment Posterior Probability for Automatic Mandarin Pronunciation Evaluation,” IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), no. 4, pp. 201-204, 2007.

W. Cao, D. Wang and J. Zhang, “Developing a Chinese L2 speech database of Japanese learners with narrow-phonetic labels for computer assisted pronunciation training,” INTERSPEECH, 2010 – 11th Annual Conference of the International Speech Communication Association, September 26–30, Makuhari, Chiba, Japan, Proceedings, 2010, pp. 1922-1925.

R. Duan, J. Zhang, W. Cao, and Y. Xie, "A Preliminary study on ASR-based detection of Chinese mispronunciation by Japanese learners," INTERSPEECH, 2014.

Y. Gao, Y. Xie W. Cao and J. Zhang, “A Study on Robust Detection of Pronunciation Erroneous Tendency Based on Deep Neural Network,” INTERSPEECH, 2015 – 16th Annual Conference of the International Speech Communication Association, September 6–10, Dresden, Germany, Proceedings, 2015, pp. 693-696.

LeCun, Yann, and Yoshua Bengio. "Convolutional networks for images, speech, and time series." The handbook of brain theory and neural networks 3361.10 (1995): 1995.

Sainath, Tara N., et al. "Deep convolutional neural networks for LVCSR." Acoustics, speech and signal processing (ICASSP), 2013 IEEE international conference on. IEEE, 2013.

Abdel-Hamid, Ossama, et al. "Convolutional neural networks for speech recognition." IEEE/ACM Transactions on audio, speech, and language processing 22.10 (2014): 1533-1545.

Graves, Alex, Abdel-rahman Mohamed, and Geoffrey Hinton. "Speech recognition with deep recurrent neural networks." Acoustics, speech and signal processing (icassp), 2013 ieee international conference on. IEEE, 2013.

Haşim Sak, Andrew Senior, & Françoise Beaufays. (2014). "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition". Computer Science, 338-342.

Bengio, Y., Cho, K., Gülçehre, Ç., & Pascanu, R. (2013). How to Construct Deep Recurrent Neural Networks. CoRR, abs/1312.6026.

K. Li and H. Meng, "Mispronunciation detection and diagnosis in l2 english speech using multi-distribution Deep Neural Networks," IEEE International Symposium on Chinese Spoken Language Processing (ISCSLP), pp. 255-259, 2014.

Ko, T., Khudanpur, S., Peddinti, V., & Povey, D. Audio augmentation for speech recognition. INTERSPEECH2015.

Y. J. Wang and X. N. Shangguan, "How Japanese learners of Chinese process the aspirated and unaspirated consonants in standard Chinese," *Chinese Teaching in the World*, 2004.

Xie, X. "A study on Japanese Learner's Acquisition Process of Mandarin Balade-Palatal Initials." Journal of Jilin Teachers Institute of Engineering and Technology (2010).

F. Y. Li and W. Cao, "Comparative study on the acoustic characteristics of phoneme /u/ in mandarin between Chinese native speakers and Japanese learners," *Chinese Master's Thesis Full-text Database*, No. S1, 2011.

W. Cao, J. Zhang, "The establishment of a CAPL Inter-Chinese Corpus and Its Labeling," Proc. Of NCMMSC, 2009.

Bengio, Yoshua, Patrice Simard, and Paolo Frasconi. "Learning long-term dependencies with gradient descent is difficult." IEEE transactions on neural networks 5.2 (1994): 157-166.

Bilac, S. and Tanaka, H., 2005, Extracting transliteration pairs from comparable corpora, In *Proceedings of the Annual Meeting of the Natural Language Processing Society*, Japan.