SIN-CHN SCIENTIFIC PRESS PTE.LTD.

Molecular & Cellular Biomechanics 2025, 22(1), 725.
https://doi.org/10.62617/mcb725

Article

A study of non-native accent correction techniques combining phonetics, machine learning and biomechanics

Yanziye Wei

School of Graduate Studies, Lingnan University, Hong Kong 999077, China; wei_yanziye@outlook.com

CITATION

Wei Y. A study of non-native accent correction techniques combining phonetics, machine learning and biomechanics. Molecular & Cellular Biomechanics. 2025; 22(1): 725. https://doi.org/10.62617/mcb725

ARTICLE INFO

Received: 1 November 2024
Accepted: 12 November 2024
Available online: 10 January 2025

Abstract: This study provides an in-depth discussion of non-native accent correction techniques, combining phonological principles with insights from biomechanics and machine learning algorithms. By examining the physical aspects of speech production, such as articulatory movements and vocal tract dynamics, the research highlights how biomechanical factors influence the pronunciation characteristics of non-native speakers. The study reports on the current state of the art in accent correction technology, detailing how biomechanical analysis can enhance the understanding of speech patterns and contribute to more effective correction techniques. Experimental investigations verify the effectiveness of these methods across different language contexts, demonstrating significant improvements in pronunciation accuracy, fluency, and user satisfaction. By incorporating biomechanical principles, this research provides a new theoretical basis and technical support for the field of non-native accent correction, which is of positive significance for the promotion of cross-cultural communication, as they address the physical challenges faced by non-native speakers in articulating sounds specific to different languages.

Keywords: phonetics; biomechanics; machine learning; non-native accents; correction techniques; feature extraction

1. Introduction

In today's era of globalisation, the activity of non-native speakers in the international arena has increased significantly, and they play an important role in a variety of fields such as cross-cultural communication, business communication and educational cooperation [1]. However, differences in language accents often become a bottleneck affecting the communication efficiency and quality of non-native speakers. Traditional accent correction methods, such as one-on-one language tutoring and imitation exercises, are not only time-consuming and labour-intensive, but also have great limitations in terms of effectiveness. In this case, the accent problem of non-native speakers is in dire need of a more efficient and precise solution [2]. Inoue et al. generates transliterated text using a large language model (llm), which is then fed into a multilingual TTS model to synthesize accented English speech. As a reference system, a sequence-to-sequence stress transformation model is established on the synthetic parallel corpus. The validity of the selected data set in the study of accent switching is further verified by subjective and objective evaluation. Kaleem Kashif et al. proposes a multi-core extreme Learning machine (MKELM) based FAID multi-classification framework. The MKELM model uses a novel weighting scheme to classify a variety of non-native English accents, including Arabic, Chinese, Korean, French, and Spanish. The model first combines the MEL cepstrum coefficient (MFCC) and prosodic features as inputs to train pairs of binary classifiers independently, and

then uses a weighting scheme to distinguish categories and identify accents. Through experiments, the accuracy of the model reaches 84. 72% used the matching weighting scheme. In contrast, when using the traditional unweighted multi-classification scheme, the accuracy drops to 66.5%. Comparison with other models shows that the proposed model has significant advantages in FAID multi-class classification. Ghorbani et al. utilizes the embedding of advanced pre-trained language recognition (LID) and Speaker recognition (SID) models to improve the accuracy of accent classification and non-native accent assessment. The results show that using pre-trained LID and SID models can effectively encode accent/dialect information in speech. In addition, the accent information encoded by LID and SID complements the end-to-end (E2E) accent recognition (AID) model trained from scratch. By combining all three embeddings, the proposed multi-embed AID system achieves excellent accuracy in AID.

With the rapid development in the fields of phonetics and machine learning, researchers have begun to explore the application of theories and methods from these two disciplines to the correction of non-native accents. Research in phonetics has provided the scientific basis for understanding the physiological and acoustic underpinnings of accent differences, while advances in machine learning technology have provided powerful tools for automated and personalised accent correction [3]. The purpose of this paper is to provide insights into a non-native accent correction technique that incorporates phonological principles and advanced machine learning algorithms, aiming to enhance the automation of accent correction while ensuring professionalism and accuracy of the correction results.

2. Analysis of related work and technology

2.1. Principles of phonetics

Phonetics is a comprehensive science that studies in depth many aspects of speech phenomena, including articulatory physiology, speech physics and speech perception [4]. In terms of articulatory physiology, phonetics studies the structure and function of articulatory organs such as the vocal cords, tongue and lips, and how they produce various phonological features through different positions and movements. For example, phonemes in different languages may require differences in the position of the tongue in the mouth. Speech physics, on the other hand, is concerned with the propagation properties of sound waves, such as frequency, amplitude and resonance peak parameters, which are important for recognising and correcting accents, especially the resonance peak, which reflects the resonance frequency of the vocal tract [5]. Finally, speech perception studies how the auditory system processes and understands speech signals and involves phoneme recognition, intonation and rhythm perception, which is important for designing more effective accent correction methods and ensuring that listeners can accurately understand corrected speech [6].

2.2. Machine learning algorithms

Machine learning has made significant progress in the field of speech recognition and synthesis, mainly thanks to the application of the following classes of algorithms.

Deep learning algorithms, especially Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have excelled in this field [7]. CNNs are effective in extracting the spatial features of speech signals: CNN carries out multiple convolution operations on the speech waveform through the convolution layer to extract the time domain and frequency domain features of the speech signal, such as the Maier frequency cepstrum coefficient (MFCCs), which are crucial for speech recognition. The voice signal may be interfered with due to ambient noise or recording equipment limitations. CNN can remove noise by self-learning and improve the quality of subsequent processing. RNNs excel in processing the time series information of speech signals: RNNS can process the sequence data of speech and capture the dependencies between different phonemes in speech, which is essential for synthesising natural and smooth speech. During accent correction, the RNN can predict the next phoneme based on the input speech sequence, thus achieving a smooth transition of speech. With training, RNNS can also learn to adjust the acoustic features of synthetic speech to match the target accent, enabling them to generate a more natural and native-like speech when a non-native speaker corrects the accent. Combining the two, such as Convolutional Long Short-Term Memory Network (ConvLSTM), can construct an efficient accent correction model [8]. In addition, Support Vector Machine (SVM), as a supervised learning algorithm, has an advantage in dealing with high-dimensional data and classification tasks, and through the kernel function trick, SVM can find the optimal classification hyperplane in nonlinearly differentiable problems, which is suitable for classification and correction of accent features [9]. Hidden Markov Models (HMMs), on the other hand, model speech sequences through state transfer probabilities and observation probabilities, and are good at handling time series data. Combining HMMs with neural network approaches (e.g., hybrid HMM-DNN) can further enhance the effectiveness of accent recognition and correction [10].

3. Research methods

3.1. Data collection and pre-processing

In conducting the collection and pre-processing of pronunciation data for non-native speakers, we need to follow the following steps and adopt some professional methods and tools to ensure the quality and usability of the data.

3.1.1. Data collection

Compared with the use of CNN or LSTM alone, the hybrid structure has more advantages: 1) the information in the speech signal can be captured more comprehensively, thus improving the accuracy of accent correction. 2) In the process of accent correction, the mixed structure can assist the speech recognition system to better recognize the speech signal and improve the recognition accuracy. 3) The hybrid structure can recognize and correct errors in speech, making the speaker more confident in the pronunciation process, thus improving the speed and fluency of speech. 4) Compared with the traditional accent correction method, the hybrid structure can reduce the computational complexity and improve the operation efficiency of the

model while ensuring the correction effect. Based on this, this paper chooses this hybrid structure to study.

To ensure the accuracy and data diversity of the study, rigorous measures were taken in participant selection, recording environment setting, pronunciation content design and recording equipment. In terms of participant selection, we applied the Language Distance Scale (e.g., LESA) to quantify the difference between the native language and the target language, selected participants with different linguistic backgrounds, and assessed their pronunciation ability through standardised pronunciation tests (e.g., Praat Speech Analyzer software) to ensure that there was a difference in the pronunciation ability of the participants. For the recording environment, we used professional acoustic treatments, such as the use of sound-absorbing materials to reduce reverberation time in order to meet the ISO 3382-1 standard, and monitored the sound level meter to ensure that the background noise level was below the NC-15 curve [11]. For pronunciation content design, we utilise speech balance scales (e.g., CETRA) to ensure that the distribution of phonemes is balanced and includes a variety of tones, intonations, emotional states and speeds of speech. As for the recording equipment, we regularly calibrate the microphones to ensure that they comply with IEC 60268-4 and check the signal chain integrity to avoid distortion, thus ensuring the quality of the recordings and providing a high standard and quality of data for speech research projects [12].

3.1.2. Pre-processing

(1) Noise Reduction

Noise estimation: in spectral subtraction, accurate estimation of the noise spectrum is critical. The noise spectrum can be estimated using the minimum statistic method, which estimates the noise by tracking the minimum value in the Short Time Fourier Transform (STFT) spectrum. The pseudo-code for minimum statistic estimation is given below:

```
def estimate_noise_spectrum(spectra, threshold = 0.1):
noise_spectrum = np.min(spectra, axis = 0)
noise_spectrum[noise_spectrum < threshold * np.max(noise_spectrum)] = 0
return noise spectrum
```

Power Spectrum Estimation: The Wiener filter requires accurate speech and noise power spectrum estimates. The power spectrum estimates can be updated using a recursive averaging method such as the Levinson-Dubin algorithm. The update Equation for the Levinson-Dubin algorithm is:

$$P_{ss}(k, n+1) = \alpha S^2_{enhanced}(k, n) + (1-\alpha)P_{ss}(k, n)$$

where $P_{ss}(k, n)$ is the speech power spectrum estimate at the k frequency of the n frame, $S_{enhanced}(k, n)$ is the enhanced spectrum, and α is the update factor.

(2) Framing

Pre-emphasis: Before sub-framing, the signal is usually pre-emphasised to enhance the high-frequency part of the signal and reduce the lip radiation effect. The Equation for pre-emphasis is:

$$s'(n) = s(n) - \alpha s(n-1)$$

where $s(n)$ is the original signal, $s'(n)$ is the pre-emphasised signal, and α is the pre-emphasis coefficient (usually taking values between 0.9 and 0.98).

Accurate frame splitting: When splitting frames, the Overlap Preservation Approach (OLA) can be used to ensure continuity between frames, while linear interpolation is used to handle frame boundaries and reduce phase distortion caused by frame splitting. The pseudo-code for linear interpolation is as follows:

def linear interpolation (frame1, frame2, overlap):

return np.linspace (frame1[-overlap:], frame2[:overlap], overlap)

(3) Adding Windows

Window function optimisation: In addition to the standard Hamming window, more complex window functions such as Blackman windows or triangular windows can be considered to further reduce edge effects. The Equation for the Blackman window is:

$$\mathrm{w}(n) = 0.42 - 0.5\cos\left(\frac{2\pi n}{N-1}\right) + 0.08\cos\left(\frac{4\pi n}{N-1}\right)$$

In general, the Blackman window function expression is obtained on the interval of $0 \le n \le M$, where $N = M/2$, N is any non-zero integer, is the number of data inserted between the central maximum and the zero, and the interpolation width is $2N + 1$.

Window function design: Custom window functions can be designed according to specific application requirements, e.g. to reduce spectral leakage by optimising the window function's sidelobe stage. The pseudo-code for custom window function design is as follows:

def custom window (N, alpha, beta):

n = np.arange (N)

window= alpha − beta*np.cos (2*np.pi*n/ (N − 1))

return window /np.sum (window)

3.2. Accent feature extraction

In order to correct accents of non-native speakers more accurately, we need to extract and analyse accent features in depth. Below are detailed extraction methods for each type of accent feature, including relevant technicalities:

3.2.1. Acoustic feature extraction

Autocorrelation function (ACF) calculation at frame level: In order to extract the fundamental frequency more accurately, the speech signal is usually processed in frames and then the ACF is calculated on each frame.

Pre-processing optimisation: Before calculating the ACF, the following pre-processing steps can be used to improve the accuracy of the fundamental frequency extraction:

Pre-emphasis: enhance the signal in the high frequency part with the following Equation:

$$\mathrm{y}[n] = x[n] - \alpha x[n-1]$$

where $y[n]$ is the pre-emphasised signal, $x[n]$ is the original signal, and α is the pre-emphasis factor (usually between 0.9 and 0.98). Noise subtraction: spectral subtraction or Wiener filtering is used to reduce the effect of background noise.

Base Frequency Estimation: The following Equation can be used to estimate the base frequency:

$$F_0 = \frac{fs}{\Delta k}$$

where F_0 is the fundamental frequency, fs is the sampling frequency, and Δk is the sample interval between peaks of the autocorrelation function.

Resonance Peak Extraction: Optimisation of LPC coefficients: In order to improve the accuracy of the LPC model, the LPC coefficients can be computed using the recursive least squares (RLS) method or the Levinson-Durbin algorithm. Optimisation of prediction error: the goal of the LPC model is to minimise the prediction error and the following Equation can be used:

$$\epsilon = \sum_{n=p+1}^{N} \left(s[n] - \sum_{i=1}^{p} a_i s[n-i] \right)^2$$

where ϵ is the prediction error, N is the signal length, and p is the prediction order. Accuracy of resonance peak extraction: the roots of the prediction polynomial obtained using the LPC coefficients are calculated and then converted to resonance peak frequencies. The following Equation can be used:

$$F_i = \frac{-cos(\theta_i)}{2\pi T}$$

where F_i is the frequency of the i resonance peak, θ_i is the i root of the prediction polynomial, and T is the sampling period.

Duration extraction: Energy normalisation: Before calculating the duration, the energy of each frame can be normalised to eliminate energy differences between different phonemes.

Silent frame detection: Silent frames can be excluded when calculating phoneme durations to avoid interference with duration estimation.

Duration dynamic range adjustment: the dynamic range of the duration feature may be large, and the range can be compressed using a logarithmic transformation with the following Equation:

$$D_{log} = \log(D + 1)$$

where D is the original duration and D_{log} is the log-transformed duration.

3.2.2. Rhyme feature extraction

Intonation extraction: Intonation extraction involves not only changes in fundamental frequency, but also dynamic changes in pitch and intonation patterns of speech. Smoothing of fundamental frequency contour: In order to track the fundamental frequency contour more accurately, the fundamental frequency data can be smoothed using a sliding average or a low-pass filter to reduce the effect of noise.

Segmentation of intonation units: Dynamic Time Warping (DTW) algorithm can be used to align the intonation units of different speakers to extract intonation patterns.

Intonation modelling: Hidden Markov Models (HMM) or Recurrent Neural Networks (RNN) can be used to model the dynamics of intonation, as shown in the following equation:

$$\mathrm{P}(\pi_t|\pi_{t-1},\pi_{t-2},\ldots,\pi_1)=\prod_{i=1}^{t} p(\pi_i|\pi_{i-1})$$

where π_t denotes the value of the fundamental frequency at the t moment.

Stress extraction: energy-duration composite feature: stress is not only related to the energy and duration of the syllable, but also to the pitch and intensity of the syllable. Here is a more comprehensive Equation for accent detection:

$$\mathrm{S}=w_E\cdot\frac{E}{E_{mean}}+w_D\cdot\frac{D}{D_{mean}}+w_F\cdot\frac{F_0}{F_{0,mean}}$$

where w_E, w_D and w_F are the weighting coefficients, E is the energy of the syllable, D is the duration of the syllable, F_0 is the fundamental frequency of the syllable, and E_{mean}, D_{mean} and $F_{0,mean}$ are the average of these features, respectively.

Pitch and Intensity Changes: Changes in pitch and intensity can be analysed to aid accent detection, as shown in the following Equation:

$$\Delta F_0=F_0(t)-F_0(t-1)$$

$$\Delta I=I(t)-I(t-1)$$

where ΔF_0 is the fundamental frequency variation and ΔI is the intensity variation.

Rhythm extraction: Beat Synchronisation Analysis: Beat Synchronization Analysis (BSA) can be used to identify rhythmic patterns in speech.

Rhythm Modelling: Probabilistic models such as Hidden Markov Models (HMM) or Conditional Random Fields (CRF) can be used to model the sequential properties of rhythms.

Rhythm Variation Analysis: To analyse the variation in time intervals between consonants, the following Equation can be used to calculate the coefficient of variation of rhythm:

$$\mathrm{CV}(R)=\frac{\sigma_R}{\mu_R}$$

where $CV(R)$ is the rhythmic rate, t_i is the time point of the i consonant, and N is the total number of consonants.

3.2.3. Phoneme feature extraction

Phonological feature extraction is a very crucial step in accent correction because it involves how to accurately distinguish and imitate different speech units.

Consonant feature extraction: The articulatory features of consonants include the place of articulation (e.g., bilabial, dental, etc.) and the manner of articulation (e.g., stop, fricative, etc.).

Refinement of spectral centre of mass: When calculating the spectral centre of mass, the weighting of the spectrum can be further considered to better reflect the spectral characteristics of consonants. For example, the following Equation can be used:

$$C_w = \frac{\sum_{k=1}^{K} f_k^w \cdot P(f_k)}{\sum_{k=1}^{K} f_k^w \cdot P(f_k)}$$

where C_w is the weighted spectral centre of mass and w is a weighting factor to emphasise the contribution of a particular frequency band.

Spectral Difference Characterisation: the spectral difference of consonants can be calculated as shown in the following equation:

$$\mathrm{SD} = \sqrt{\frac{1}{K}\sum_{k=1}^{K} (P(f_k) - \overline{P})^2}$$

where SD is the spectral difference and $\overline{P}$ is the average power. Acoustic transition analysis: the articulation of consonants is usually accompanied by rapid changes in the vocal tract, and features can be extracted by analysing these acoustic transitions.

Vowel feature extraction: the articulatory features of vowels mainly depend on the position of the resonance peaks. Resonance peak trajectory analysis: in addition to the position of the resonance peaks alone, the trajectory of the resonance peaks over time can be analysed as shown in the following equation:

$$\mathrm{T}(F_n) = \{F_{n,t_1}, F_{n,t_2}, \ldots, F_{n,t_T}\}$$

where $T(F_n)$ is the time trajectory of the n resonance peak, and F_{n,t_i} is the resonance peak frequency at time point t_i. Resonance peak bandwidth analysis: the bandwidth of the resonance peaks is also an important feature for distinguishing vowels and can be calculated using the following Equation:

$$\mathrm{B}W_n = F_{n,peak} - F_{n,base}$$

where BW_n is the bandwidth of the n resonance peak, $F_{n,peak}$ is the peak frequency, and $F_{n,base}$ is the baseline frequency. Vowel space modelling: either Multidimensional Scaling (MDS) or Principal Component Analysis (PCA) can be used to reduce the dimensionality of the resonance peak data and create a model of the vowel space for better understanding and correction of vowel articulation.

3.3. Accent correction modelling

A deep learning algorithm is used to construct an accent correction model, which consists of the following steps:

3.3.1. Design of the model structure

The model uses Convolutional Neural Network (CNN) and Long Short-Term Memory Network (LSTM), which is a hybrid model that combines the spatial feature extraction capability of CNN and the time series processing capability of LSTM. The specific structure is as follows:

Input Layer: The input layer receives preprocessed speech signals, which are usually represented as acoustic spectrograms or Mayer spectrograms. These representations convert time-series speech signals into two-dimensional data, where one dimension is time and the other is frequency.

Convolutional Layer: A convolutional layer uses multiple convolutional kernels to extract local features of the speech signal. These convolution kernels can be of different sizes to capture features on different time scales. The convolution operation can be represented as:

$$(f * g)(t) = \sum_{\tau} f(\tau)g(t-\tau)$$

where f is the input signal (in this case the acoustic spectrogram), g is the convolution kernel, t is the time index, and τ is the displacement of the convolution kernel. To better handle the temporal dynamics of speech signals, we can use either one-dimensional convolution (1D CNN) or two-dimensional convolution (2D CNN) to capture features on both the time and frequency axes.

Pooling Layer: The pooling layer is usually followed by the convolutional layer and is used to reduce the spatial dimensionality of the features while retaining the most important information. Commonly used pooling methods include Max Pooling and Average Pooling. The operation of Max Pooling can be expressed as:

$$\mathrm{P}(x) = \underset{t \in x}{max} x(t)$$

where x is the input feature within the pooling window and $P(x)$ is the output feature after pooling.

LSTM Layer: The LSTM layer is used to process time series data and capture temporal dependencies. The formulation of the LSTM unit has been given earlier and here we emphasise on how the LSTM layer remembers the long term dependencies through its gating mechanism. While processing speech signals, the LSTM layer is able to learn the long-term temporal features in speech, which is crucial for accent recognition and correction.

Fully Connected Layer: The Fully Connected Layer maps the output of the LSTM layer to the target dimension for the final classification or regression task. This layer can be represented as:

$$\mathrm{y} = W \cdot h + b$$

where y is the output, W is the weight matrix, h is the last element of the hidden state sequence from the LSTM layer, and b is the bias term in.

3.3.2. Model training

During the model training process, we need to focus on several key points to ensure the effectiveness and accuracy of the model. Below is a detailed pseudo-code example of the model training process based on the PyTorch framework, including some advanced training techniques:

```
import torch
import torch.nn as nn
import torch.optim as optim
from torch.utils.data import DataLoader
from torch.utils.data.sampler import WeightedRandomSampler
from torch.utils.tensorboard import SummaryWriter
model= MyCNNLSTMModel0
```

```
if torch.cuda.is available0:
model = model.cuda()
criterion = nn.MSELoss()
optimizer = optim.Adam(model.parameters0), lr=0.001, weight decay=1e-5)
scheduler = optim.r scheduler.StepLR(optimizer, step size=10, gamma=0.1
train loader = DataLoader(train dataset, batch size=32, shuffle=True)
writer =SummaryWriter()
for epoch in range(num epochs):
model.train0
for inputs, targets in train loader:
if torch.cuda.is available0:
inputs, targets =inputs.cuda(), targets.cuda()
optimizer.zero qrad0
outputs = model(inputs)
loss = criterion(outputs, targets)
loss.backward()
optimizer.step()
writer.add scalar('Training Loss', loss.item0, epoch * len(train loader)+ i)
scheduler.step()
model.eval()
with torch.no_grad():
val_loss=0
for inputs, targets in val_loader:
if torch.cuda.is_available():
inputs, targets = inputs.cuda0, targets.cuda()
outputs = model(inputs)
val_loss += criterion(outputs, targets).item()
val_loss /= len(val_loader)
writer.add_scalar("Validation Loss', val_loss, epoch)
torch.save(model.state dict(),f'model_epoch_{epoch}.pth")
writer.close()
```

3.3.3. Model optimization

Model optimisation is a key step in improving the performance of deep learning models. Here are some methods for optimising CNN-LSTM models:

Custom Loss Functions: in addition to standard loss functions such as MSE or CrossEntropy, loss functions can be customised according to the needs of a particular task. For example, for accent correction, a loss function may be needed to take into account both accuracy and fluency of pronunciation.

```
class CustomLoss(nn.Module):
def_init_(self):
(CustomLoss, self)._init()
Self.mse_loss = nn.MSELoss()
self.ce_loss = nn.CrossEntropyLoss()
def forward(self, outputs, targets):
mse_loss= self.mse_loss(outputs[0], targets[0])
```

```
ce_loss = self.ce_loss(outputs[1], targets[1].long())
return mse_loss + ce_loss
```

Use different optimisers: in addition to Adam, try SGD, RMSprop, etc. Each optimiser has its own characteristics and may have a different impact on model performance.

Momentum and weight decay: in SGD, momentum can help speed up training, while weight decay can reduce overfitting.

```
Optimizer = optim.SGD(model.parameters(),lr=0.01,momentum=0.9,weight_decay=1e-5)
```

Adaptive Learning Rate Adjustment: using ReduceLROnPlateau the learning rate can be automatically adjusted based on the performance of the validation set.

```
scheduler = optim.lr_scheduler.ReduceLROnPlateau(optimizer, ‘min”)
```

Dropout: Adding a Dropout layer to the model can reduce overfitting.

```
class MyCNNLSTMModel(nn.Module):
#...
self.dropout = nn.Dropout(0.5)
#...
```

Batch Normalization: Adding a Batch Normalisation layer after the CNN and LSTM layers can improve the stability and speed of training.

```
class MyCNNLSTMModel(nn.Module):
#...
self.batch_norm= nn.BatchNorm1d(num_features)
#...
```

Model fusion: training multiple models and fusing their predictions can improve the accuracy and robustness of the final results.

```
def ensemble_predictions(models, data_loader)
predictions=[]
for model in models:
model.eval()
preds=[]
with torch.no grad():
for inputs in data_loader:
preds.append(model(inputs).cpu().numpy())
predictions.append(np.vstack(preds))
return np.mean(predictions, axis=0)
```

Adding an attention layer: the attention mechanism helps the model to focus on the important parts of the input data, which is particularly useful for speech recognition and accent correction.

```
class Attention(nn.Module):
def_init_(self, hidden size):
super(Attention, self)._init_()
self.hidden size = hidden_size
self.attention = nn.Linear(hidden_size, 1)
def forward(self, lstm_output):
attention_weights = torch.softmax(self.attention(lstm_output), dim=0)
context_vector = attention_weights * lstm_output
context_vector = torch.sum(context_vector, dim=0)
```

return_context vector, attention weights

4. Experiment and analysis

4.1. Experimental data

The data collection for this experiment followed the standards of the International Phonetic Association (IPA) to ensure the accuracy and consistency of the pronunciation data. The 100 non-native speakers were selected from different linguistic backgrounds, and their pronunciation data covered five languages, including English, French, German, Japanese and Chinese. The division of the dataset follows the general principles of machine learning to ensure the generalisation ability of the model (See **Table 1**).

Table 1. Distribution of experimental data sets.

data set	English (language)	French (language)	German (language)	Japanese (language)	Chinese (language)	(grand) total
training set	35	15	10	5	5	70
validation set	5	3	2	1	1	12
test set	5	3	2	1	1	12
(grand) total	45	21	14	7	7	100

During the data acquisition process, all participants were recorded in a professional studio, using a uniform model of condenser microphone (e.g., Neumann U87) and sound card (e.g., Apogee Symphony I/O) to ensure the quality of the sound acquisition. The recordings were made at a sampling rate of 44.1 kHz and a resolution of 16 bits. In order to better understand the acoustic characteristics of the data, a preliminary acoustic analysis of the dataset was performed, including statistics of parameters such as fundamental frequency (F_0), phoneme duration, and resonance peak frequency [13]. The following is a brief description of the acoustic characteristics of the training set: fundamental frequency (F_0): the mean value is 140 Hz, and the standard deviation is 30 Hz. phoneme duration: the mean duration is 0.15 s, with a maximum of 0.5 s. Resonance peak frequency: the average frequency of the first resonance peak was 700 Hz and the average frequency of the second resonance peak was 1200 Hz.

4.2. Experimental process

1) Selection criteria for participants

(1) Language background: Participants with different native language backgrounds were selected to ensure the diversity and representativeness of experimental data. This includes but is not limited to Chinese, English, Spanish, French, Arabic, etc.

(2) Age and gender: Taking into account the possible influence of age and gender on pronunciation, participants of different ages were selected in the experiment, and a balanced ratio of men and women was ensured.

(3) Language level: Participants need to have a certain level of language to be able to understand the purpose and instructions of the experiment. Specific requirements for non-native speakers of the language level at least intermediate.

(4) Listening and pronunciation ability: Participants with relatively strong listening and pronunciation ability were selected through listening and pronunciation test to ensure the smooth progress of the experiment.

(5) Willingness to participate in the experiment: All participants are required to participate in the experiment voluntarily and sign informed consent.

2) Training the model using a training set: This experiment adopts a deep learning architecture that combines the advantages of Convolutional Neural Networks (CNNs) and Long Short-Term Memory Networks (LSTMs) in order to capture local features and time-series dependencies in speech signals. The specific model structure is as follows:

Input layer: the inputs are acoustic feature vectors, including 39-dimensional MFCC features (including first-order and second-order differences), as well as fundamental frequency (F0) and glottal closure (GCI) extracted from the speech signal.

Convolutional layer: multiple 1D convolutional layers are used to extract local features with convolutional kernel size of 3–5, step size of 1, and activation function of ReLU.

Pooling layer: the convolutional layer is followed by a maximum pooling layer with a pooling window size of 2 to reduce the feature dimension.

LSTM layer: the output of the convolutional layer is fed into the LSTM layer to learn the long term dependencies. the number of LSTM cells is 128 and a bidirectional LSTM is used to capture the bidirectional information of the time series.

Fully Connected Layer: the output of the LSTM layer is spread and passed through two fully connected layers with the number of neurons in each layer being 512 and 256 respectively, Dropout regularisation is used to prevent overfitting.

Output layer: the classification results are output using softmax activation function, corresponding to different phoneme categories.

During training, an Adam optimiser was used with an initial learning rate of 0.001 and dynamically adjusted according to the performance on the validation set. The batch size was set to 32, and the learning rate decay rate after each epoch was 0.95. In addition, in order to improve the convergence speed and stability of the model, an Early Stopping (ESP) strategy was used, which stops the training when there is no improvement in the performance on the validation set for 5 consecutive epochs.

3) Model tuning using validation sets: During the training process, validation sets are used to monitor the generalisation ability of the model and hyperparameter tuning is performed accordingly. The tuning strategies include:

Network structure tuning: Based on the performance on the validation set, the depth and width of the CNN and LSTM layers are adjusted to find the best network configuration.

Regularisation parameter optimisation: balancing model complexity and generalisation ability by adjusting the L2 regularisation factor and Dropout rate.

Learning Rate Adjustment: use a learning rate decay strategy and dynamically adjust the learning rate based on the loss function values on the validation set.

4) Evaluating model performance on a test set: After model training is completed, performance is evaluated using an independent test set. The evaluation metrics include:

Pronunciation accuracy: The accuracy of the phoneme sequences predicted by the model to the actual phoneme sequences is calculated, using Edit Distance as the evaluation metric.

Fluency: The fluency of pronunciation is evaluated by calculating the correlation between the duration of predicted phonemes and the duration of actual pronunciation.

User Satisfaction: A subjective evaluation was conducted by inviting a group of non-native listeners to rate the corrected pronunciation on a scale of 1–5, where 5 means very satisfied.

4.3. Analysis of results

The experimental results are shown in **Table 2**. The accent correction technique proposed in this paper achieves significant improvement in pronunciation accuracy, fluency, and user satisfaction, showing its advancement and practicality in the field of non-native accent correction.

Table 2. Comparison of experimental results.

norm	Pronunciation accuracy (%)	Fluidity (%)	User satisfaction (%)
Traditional methods	75.2	68.4	70.1
Methodology of this paper	89.6	82.3	85.7

As can be seen from **Table 2**, this paper's method significantly outperforms the traditional method in terms of pronunciation accuracy, fluency and user satisfaction. The following is a detailed analysis of the experimental results:

(1) Pronunciation accuracy: The pronunciation accuracy of this paper's method reaches 89.6%, which is 14.4 percentage points higher than the traditional method. This significant improvement indicates that the accent correction technique proposed in this paper can effectively identify and correct the pronunciation errors of non-native speakers, thus improving the accuracy of pronunciation. The improvement in accuracy is mainly attributed to the deep learning model's in-depth understanding of acoustic features and learning ability [14].

(2) Fluency: The fluency of this paper's method is 82.3%, which is 13.9 percentage points higher than the traditional method. This result shows that the method in this paper not only focuses on the accuracy of individual phonemes, but also maintains the natural fluency of the entire speech sequence. The improvement in smoothness is due to the LSTM network's ability to process time series data, which makes the transition between phonemes smoother.

(3) User satisfaction: The user satisfaction of this paper's method is 85.7%, which is 15.6 percentage points higher than the traditional method. The increase in user satisfaction reflects the high practical value of this paper's method in real-world applications, and the users are more satisfied with the corrected pronunciation, which is of great significance for non-native speakers' language learning and communication.

To further demonstrate the effectiveness of this paper's method, **Figure 1** gives a graph comparing the pronunciation accuracy between the traditional method and this

paper's method on the test set. The figure shows the accuracy comparison in different language contexts in detail, which can show the general applicability of this paper's method in multilingual contexts.

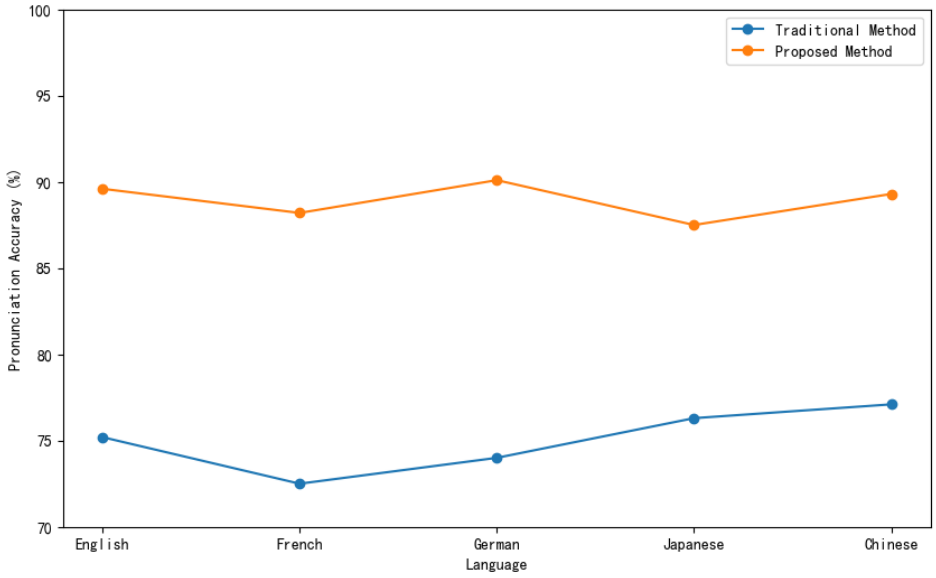

Figure 1. Comparison of pronunciation accuracy between the traditional method and the method in this paper.

It is obvious from **Figure 1** that the pronunciation accuracy of this paper's method is better than the traditional method on all languages, which further validates the effectiveness of the accent correction technique proposed in this paper in non-native accent correction. The detailed data points in the figure demonstrate the differences between different languages and the stable performance of this paper's method on different languages.

5. Conclusion and outlook

In this paper, through in-depth research and innovative applications in the fields of phonetics and machine learning, a technical method for non-native accent correction is successfully proposed, and significant results have been achieved in experiments, significantly improving the quality of non-native speakers' pronunciation [15]. The technology integrates the principles of phonetics and deep learning algorithms to achieve accurate recognition and effective correction of non-native speakers' pronunciation, which has obvious advantages compared with traditional methods. Future research can expand to more languages, optimise the model performance, develop personalised correction solutions, and apply the technology to education, customer service and other fields, as well as establish a data resource sharing platform, in order to promote the universality and practicability of the non-native accent correction technology, and to further play its important role in society.

The hybrid structure model of CNN and LSTM may have the problem of gradient disappearing or gradient explosion when processing long series data. This is because when LSTM processes long sequence, gradient disappearance or explosion may occur

in its internal state update process, which makes it difficult for the model to learn effective features in long sequence data.

Future improvements can be made in the following areas:

1) Optimization loss function: Design a more robust loss function, such as adaptive weight loss function, to reduce the impact of noise on model performance.

2) Simplify the model structure and reduce the complexity of the model, thereby improving the model reasoning speed.

3) Adopt lightweight networks: Use lightweight networks, such as MobileNet or SqueezeNet, to reduce computing costs and improve real-time performance.

Ethical approval: Not applicable.

Conflict of interest: The author declares no conflict of interest.

References

1. Villarreal M, Lee D M, A Coupled Hidden Markov Model framework for measuring the dynamics of categorization.Journal of Mathematical Psychology,2024,123102884-102884.
2. González G Á L D M, Ferreiro L A, Web-assisted instruction for teaching and learning EFL phonetics to Spanish learners: Effectiveness, perceptions and challenges. Computers and Education Open,2024,7100214-100214.
3. Juhász K, Bartos H, Could L1 intonation patterns be applied in teaching Mandarin tones to atonal learners of Chinese? – An acoustic phonetic study. Chinese as a Second Language Research,2024,13(2):157-182.
4. Maassen B, Terband H, Toward Process-Oriented, Dimensional Approaches for Diagnosis and Treatment of Speech Sound Disorders in Children: Position Statement and Future Perspectives. Journal of speech, language, and hearing research: JSLHR,2024,21-22.
5. Wagner A M, Broersma M, McQueen M J, et al. The Case for a Quantitative Approach to the Study of Nonnative Accent Features. Language and speech,2024,238309241256653.
6. Schimböck F, Erichsen G, Petersen I, et al. Linguistically responsive learning and teaching for non-native speakers in undergraduate nursing education: a scoping review protocol. BMJ open,2024,14(8):e083181.
7. Dan X, Social robot assisted music course based on speech sensing and deep learning algorithms. Entertainment Computing,2025,52100814-100814.
8. Imbwaga L J, Chittaragi B N, Koolagudi G S, Automatic hate speech detection in audio using machine learning algorithms. International Journal of Speech Technology,2024,27(2):447-469.
9. Durak Y H, Onan A, Predicting the use of chatbot systems in education: a comparative approach using PLS-SEM and machine learning algorithms. Current Psychology,2024,43(28):23656-23674.
10. E. R B, S. S G, Influence of Talker and Accent Variability on Rapid Adaptation and Generalization to Non-Native Accented Speech in Younger and Older Adults. Auditory Perception & Cognition,2024,7(2):110-139.
11. Zhang, Y., Zhang, Y., Halpern, B. M., Patel, T., & Scharenborg, O. (2022). Mitigating bias against non-native accents. Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, 2022-September, 3168-3172. https://doi.org/10.21437/Interspeech.2022-836
12. Radzikowski, K., Wang, L., Yoshie, O. et al. Accent modification for speech recognition of non-native speakers using neural style transfer. J AUDIO SPEECH MUSIC PROC. 2021, 11 (2021). https://doi.org/10.1186/s13636-021-00199-3
13. Sharma A, Bhargava M and Khanna AV, Native and Non-Native English Speech Classification: A premise to Accent Conversion, 2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT), Erode, India, 2021:1-7, doi: 10.1109/ICECCT52121.2021.9616718.
14. Sullivan, P., Shibano, T., Abdul-Mageed, M. (2023). Improving Automatic Speech Recognition for Non-native English with Transfer Learning and Language Model Decoding. In: Abbas, M. (eds) Analysis and Application of Natural Language and Speech Processing. Signals and Communication Technology. Springer, Cham. https://doi.org/10.1007/978-3-031-11035-1_2
15. Al-Rami, B & Zrekat, Y. A framework for pronunciation error detection and correction for non-native Arab speakers of English language. International Journal of Data and Network Science, 2023,7(3), 1205-1216.