IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 16, 2020, accepted November 26, 2020, date of publication December 2, 2020,
date of current version December 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3041901

Japanese Pronunciation Evaluation Based on DDNN

DEGUO MU[1], **(Member, IEEE), WEI SUN**[2], **(Senior Member, IEEE), GUOLIANG XU**[2], **(Member, IEEE), AND WEI LI**[1]
[1]National Key Laboratory of Software Development Environment, Beihang University, Beijing 100083, China
[2]Software College, Beihang University, Beijing 100083, China

Corresponding author: Deguo Mu (mudg@buaa.edu.cn)

ABSTRACT In recent years, speech recognition technology based on deep learning model has made great progress, and the accuracy of speech recognition has reached more than 90%. In foreign language learning, speech evaluation is an important application. Billions of foreign language learners need to practice effective pronunciation. However, due to the different goals between speech recognition and speech evaluation, a single speech recognition model cannot be directly applied to pronunciation evaluation. This paper proposes a DDNN (double-layer deep neural network) model, which includes the speech text alignment model and speech recognition model. In the first layer of the speech alignment model, a new Viterbi algorithm method is proposed to find the best path for the alignment of speech and text. In the second layer of speech evaluation and scoring, we are the first to use the CNN (Convolutional Neural Network) and RNN (Recurrent Neural Network) on the encoding part of Attention. The accuracy of CTC model reaches 76.7%, and that of attention model is 81.2%. The experimental results show that the speech and text alignment method is effective, and the speech evaluation results based on the Attention model are better. The FRR (false rejection rate), FAR (false acceptance rate), and DER (diagnostic rate) in the Attention model were 4.5%, 5.1%, and 17.9%, respectively. At the same time, the evaluation of each sentence of the DDNN model in the online experiment is within 1 second, so the model can also be applied to the online real-time evaluation of speech pronunciation.

INDEX TERMS CTC Viterbi, LSTM, attention, pronunciation evaluation, Japanese speech recognition.

I. INTRODUCTION

With the advent of globalization, the number of people are learning foreign languages are increasing. In learning a foreign language, oral practice stands as the most important part. However, it is very expensive, yet expensive, for most second language learners to find an application environment or a foreign language teacher to practice with. With the rapid development of the internet and mobile internet, almost every foreign language learner has a smartphone. It's possible to train your pronunciation by talking to a smartphone. More and more researchers begin involved in the study of CALL (Computer-Aided Language Learning), a research field of speech recognition. The technology overview in [1] reviewed the help of language technologies in education. In particular, computer-aided speech training (CAPT) is applied to language learning as a special type of speech recognition. There are many inaccuracies in speech recognition caused by noise, stress, dialect, and other difficulties. This paper is targeted at the study of Japanese assisted speaking learning.

In the past few decades, there have been plenty of researches on speech error detection, which has made great progress and has been successfully applied to the industry. A technique that imposes the prosody characteristics of the native speaker's utterance in the same sentence to non-native speaker's utterance is proposed in [2] to help Korean learner learn English; the automatic pronunciation correction introduced in [3] helps Chinese learn English; an English prosody training method based on speech conversion technology helps Japanese learn English is studied in [4]; methods for Italian learner of German is presented in [5]. As a language widely used in Asia, there are many researches on Japanese learning. The paper [6] gives an overview of the English and Japanese CALL systems developed at Kyoto University. A dialogue-based CALL system is studied in [7] focusing on the correction of lexical and grammatical errors. the

The associate editor coordinating the review of this manuscript and approving it for publication was Baozhen Yao.

research by [8] on Japanese learners of Italian [8] has realized self-imitation in rhythm training. This paper will focus on the data set of the Chinese learning Japanese. We have collected the speech corpus of millions of Japanese words pronounced by Chinese learners, and the data in our system is growing by 10,000 sentences every day. Therefore, as far as we know, this is the largest data set of Japanese pronunciation learning as a second language.

CAPT technology is mainly about the detection and diagnosis of pronunciation errors. The detection focuses on the finding of pronunciation errors according to learners' pronunciation, while the diagnosis aims to provide corrective feedbacks and facilitate learning. Therefore, in a sense, MDD (mispronunciation detection and diagnosis) is more challenging than ASR (automatic speech recognition), in that ASR is only responsible for directly outputting speech recognition results and can correct some pronunciation errors through language model or dictionary. MDD, on the other hand, is not about overlooking or automatically correcting pronunciation, but about finding out the error and diagnosing the problem. Traditionally, the implementation of MDD is divided into the voice segment and supra segment as in [9]. The voice segment mainly includes the evaluation of sentence, word, syllable, and other factors; and the evaluation of supra segment will be quite comprehensive, including word stress. A model of automatic sentence stress detection is put forward in [10], an automatic syllable stress detection based on prosody features is proposed in [11] and a multi-release deep neural network for automatic stress detection is studied in [12]. In terms of intonation, pitch stress prediction based on integrated machine learning is proposed in [13] and prosody event detection using context information in [14], as well as the method of tone automatic evaluation in [15], voice rhythm in [16], etc. We believe that the primary goal for a second language learner is to learn the correct pronunciation of words and phonetic symbols, so as to achieve the purpose of proper communication. Therefore, this paper focuses on the evaluation of word phonetic symbols at the segmental level.

A review on phonetic pronunciation evaluation is presented in [17]. Phonetic error detection and phonetic evaluation started in the 1990s, which can be divided into evaluation based on mother tongue and L2 pronunciation error detection based on the non-mother tongue. With reference to [18], the whole detection and diagnosis of phonetic errors are classified into three parts; research based on pronunciation scoring, speech recognition network based on forced alignment, and study on acoustic characterization and modeling.

A. SCORING OF PHONETIC FEATURES BASED ON THE HMM SPEECH EVALUATION MODEL

Automatic pronunciation scoring of specific phone segments for language instruction of [19] studies the log-likelihood score, log posterior probability score, segment duration classification, and compares the speech quality evaluation. It is concluded that the scoring by the posterior probability score is closer to that by humans in [20]. In this study, the correct pronunciation of the mother tongue is regarded as the standard of comparison, and the result is better than the posteriori probability score. Based on the log-likelihood score, a famous way to evaluate the "voice grace" is proposed in [21]. The standard comes from the scores of native language pronunciation and the rejection statistics of human evaluation. This method of GOP has been widely used in speech error detection and diagnosis later. The GOP is further extended and the wGOP method is proposed in [22]. The wGOP, combined with multiple LLRs, has achieved better results than GOP. The above-mentioned methods have made great achievements in the detection of speech errors, but they fail to detect the diagnosis errors.

Based on the study of score pronunciation, it is very difficult to find out errors in a very short pronunciation segment, no matter whether it is the log-likelihood score, log posterior probability score, segment duration, or LLR, GOP, wGOP and other evaluation standards. For us human, it's also a big challenge to judge whether we are right or wrong without the continuity of the whole word or sentence when we just listen to the pronunciation on a phoneme level, which is the limitation of voice score as well.

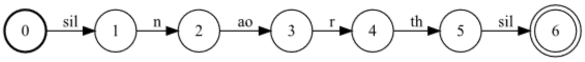


FIGURE 1. Standard recognition network of "north."

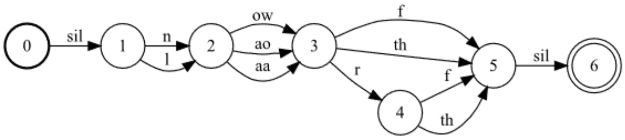


FIGURE 2. Extended recognition network of "north."

B. THE EXTENDED RECOGNITION NETWORK (ERN)

ERN includes not only the right pronunciation model (Figure 1) but also the pronunciation model where language learners are prone to make mistakes (Figure 2). When the output path of the speech after forced alignment falls into the wrong speech model, it will be detected as the wrong pronunciation, so the speech errors can be effectively diagnosed. On the basis of cross-language (Cantonese and English) analysis in [23], a set of context-sensitive phonetic rules are established and verified through the common pronunciation errors in learners' inter-language. These rules are represented as finite-state sensors, which can generate an extended recognition network (ERN) based on any standard pronunciation. In this study, more than half of the speech errors are detected. Various patterns of wrong speech have been studied in [24]–[26]. Correct speech recognition itself is a big challenge, but the wide variety of error models for different learners, especially some very rare ones, are very difficult to build, which is the biggest challenge for ERN.

In addition, if there are too many definitions of error mode, the performance of the system will be greatly reduced.

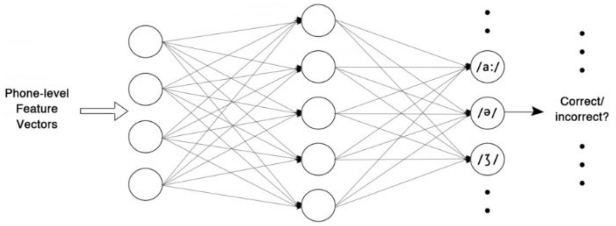


FIGURE 3. Logistic regression classifier based on a Neural network.

C. ACOUSTIC MODEL

With the development of speech recognition technologies, more and more speech recognition models have been applied to the detection of speech pronunciation errors, including the classical hidden Markov model (HMM) and the recently popular deep neural network (DNN) (Figure 3). The whole data was set into subsets by its phoneme label. For each subset, a 2-class logistic regression classifier is trained for the correct or incorrect classification decisions. Decision tree and linear discriminant analysis (LDA) are used in [27] to find different pronunciation errors of Danes while learning the second language based on some distinguishing features and define classifiers for different error patterns based on formant, duration, etc. The results show that the model is good at recognizing vowel pronunciation errors, but not good at identifying consonants.

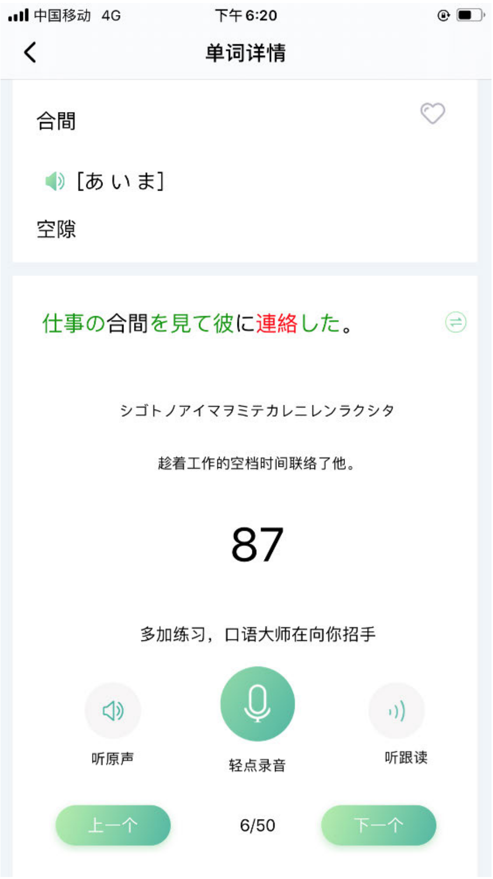
In recent years, deep learning network technology has made significant progress in various identification tasks. The ASR automatic speech recognition technology has made outstanding improvements in replacing the GMM model. We will introduce the progress in this field in detail later. In the field of speech evaluation application, it is the first to apply deep learning neural network to speech error detection and diagnosis of second language learning in [28], comparing it with the original ASR that uses annotation data for training in a supervised way. Experiments show that DBN-HMM can significantly improve the rate of pronunciation errors. It can improve system performance by replacing the system input of MFCC or gaussian posteriori diagrams obtained in a completely unsupervised way by DBN posteriori diagrams as studied in [29]. Some papers like [30] have also proposed the application of DNN in speech error detection, trained the multi-layer superposition constrained Boltzmann machine (RBMs) as a nonlinear basis function to concisely represent the speech signal, and conducted discriminant training on the output layer to optimize the posterior probability of the correct sub-phoneme "senone" state. The research of [31] is inherited by [32], where a deep neural network based on logical regression is used to detect pronunciation errors in L2 language learning and its performance exceeds GOP and SVM research in isolated words. A deep learning neural network of GAN is proposed in [32]. By training the spectral images of short sentences pronounced by native speakers and non-native speakers, the generator can successfully convert the input of non-native language spectrum into a spectrum with self-simulating feedback characteristics. It shows that periodic confrontation training is also a promising method for speech correction.

In any case, large number of current studies simply use the DNN model to represent the HMM model. Based on the innovative research results of ASR, a lot of work needs to be done in terms of speech sound pronunciation detection and diagnosis.

Deep Neural Network (DNN) attempts to model the high-level abstractions in the data, which significantly improves the recognition capacity of the acoustic model in speech recognition as studied in [33]. The research of this paper is mainly based on automatic speech recognition, so we refer to some speech models based on the deep neural network. RNN and CNN improved the performance of HMM and GMMS in the automatic speech recognition system. A new LSTM structure based on RNN was proposed in [34], which can train the acoustic model for large vocabulary speech recognition more effectively by using model parameters. A large-scale analysis is conducted on eight LSTM variants of the three representative tasks of speech recognition, handwriting recognition, and polyphonic music modeling in [35]. Under the framework of the neural network hidden Markov model, CNN neural network is applied to speech recognition in [36]. In order to achieve higher performance of the multi-speaker speech recognition, local filtering, and maximum pool are used to normalize the speaker variance in the spectrum. QCNN is proposed in [37] for phoneme level speech recognition. For modular ASR system, which includes acoustic modeling, pronunciation dictionary and language modeling components trained separately, the end-to-end model is simpler in concept and has the potential benefits of training the whole system in the final task. ASR has two main end-to-end architectures: one is based on attention, using attention mechanism to perform alignment between acoustic framework and recognition symbols as in [38]–[40]; the other is the connectionist temporal classification (CTC), using Markov hypothesis to solve sequence problem effectively through dynamic planning as discussed in [41]–[43]. Although CTC requires several conditional independence hypotheses to obtain the probability of a tag sequence, the attention-based methods do not use these hypotheses. This feature is good for sequence modeling, but we note that the attention mechanism is too flexible, and in its sense, it allows very discontinuous. Alignment, like machine translation, is usually monotonous in speech recognition.

Just as the three parts of the speech evaluation we introduced early, for the isolated recognition of the simple speech morphemes in the first part A, there is the loss of context information before and after the pronunciation, resulting in inaccurate recognition. For the recognition network expands in the second part B, it is very difficult to take all the error paths into consideration. In the third part C, a deep learning model of speech depth is applied, which emphasizes more on the accuracy of speech recognition, and automatically corrects wrong pronunciations. In view of the above problems, this paper proposes to conduct speech recognition in units of words, retaining the phonetic context information, which improves the recognition efficiency of phoneme levels. The alignment method of CTC model results is used to solve the errors of ERN in the second part Path problem. Finally, after the word-level alignment in a sentence, the speech recognition of the Attention model focuses on the pronunciation of one word. It won't automatically correct the pronunciation of a wrong word, which greatly improves the accuracy of detecting the pronunciation of wrong words.

Based on the different characteristics of CTC and the Attention mechanism, this paper proposes a two-layer model, that is, using the CTC speech model for sequence alignment. After sequence alignment, using the attention model for phoneme-level word recognition, giving full play to the advantages of each model and greatly reducing the error recognition rate.



合間

[あ い ま]

空隙

仕事の合間を見て彼に連絡した。

シゴトノアイマヲミテカレニレンラクシタ

趁着工作的空档时间联络了他。

87

多加练习，口语大师在向你招手

FIGURE 4. The system screenshot on mobile phone.

As shown in Figure 4, our system first gives users an example sentence and correct pronunciation. After listening to the correct pronunciation of the announcer, the user imitates the pronunciation and the sound is recorded by the system and uploaded to the server for speech evaluation Finally, the system gives the pronunciation score of each word part for the Japanese sentence. For users, their input is sound, which is following the sound of the original sentence, and the voice file will be uploaded to the server in the format of wav file.

After receiving the sentence text and the user's audio file on the server, the system first divides the sentence into words, and at the same time aligns the sound with the text. Finally, through the speech recognition technology, the user's pronunciation phonetic symbol is recognized. The accuracy of pronunciation can be calculated by calculating the editing distance between the phonetic symbol of correct pronunciation and the pronunciation of the user. For example, if the phonetic of original sentence is「あいま」and the user's pronunciation by speech recognition is あいか, the text editing distance between them is 1, and the total length of phonetic symbols is 3, so the correct pronunciation rate of users is 66.7%. The system will provide users with the pronunciation of each part of the word for scoring. As shown in the figure, if the accuracy rate is lower than 50%, the color of the word will be changed to red; if the accuracy rate is higher than 80%, the text will be shown in green.

The main contributions of this paper include the following four aspects: 1. The two-layer deep learning neural network model based on CTC and Attention is proposed to detect Japanese pronunciation errors, and the state-to-art effect is achieved; 2. The Viterbi decoding alignment algorithm based on CTC is proposed to complete the phoneme-level alignment results in the aspect of forced alignment; 3. The word-level phoneme recognition combining CNN with LSTM and Attention is proposed to detect pronunciation errors, and compared with the detection results of CTC based on LSTM, the former is better; 4. Learners of Japanese as second language in this paper mainly are Chinese users, providing millions of Japanese speech data, which is open for future researchers to participate in Japanese pronunciation error detection. This paper is composed of the following parts: 1. Introduction on the research status of speech error detection and ASR; 2. The system model architecture; 3. CTC algorithm and Viterbi alignment; 4. Attention-based point-to-point speech recognition model; 5. The introduction of the Japanese speech data set based on theGojūon Ordering; 6. The experiment and result analysis of the speech text alignment; 7. The experiment and result analysis of the speech evaluation; 8. Conclusion and the future works prospects.

II. THE FRAMEWORK OF DOUBLE-LAYER DNN MODEL

Automatic speech recognition (ASR) is a point-to-point task based on time series, so the deep model RNN, deep learning network is quite suitable for processing this type of task. In the encoding process in our framework, we use a two-layer LSTM for encoding from the speech signal to the hidden layer. There are many representations of phonetic features. Mel frequency cepstrum coefficient (MFCC) is a set of features widely used in the automatic speech recognition system proposed by Davis and Mermelstein in 1980. This model uses MFCC as the input of speech features. Connectionist temporal classification (CTC) is used to solve the problem that the input sequence and the output sequence are difficult to correspond one-to-one. In speech recognition and speech evaluation, our goal is to have a one-to-one correspondence between speech features and text output features. Therefore, in the first model, we take CTC results to perform decoding operation and output the one-to-one text.

From the actual pronunciation of various languages, whether it is English, Japanese, or Chinese, if it is recognized at the phoneme level, it is easy to overlook individual phonemes due to some pronunciation habits such as words connection and stress. Therefore, the phonemes that are missing due to the habit of language expression should also be judged to be correct. On the other hand, at the word level, there are few linguistic expression scenarios that miss a complete word. According to these pronunciation characteristics of the language, although we have achieved phoneme-level alignment in the first model, we still output in word units in order to avoid the inconsistent effect of phonemes and reduce the accuracy of forced alignment. Another key advantage is that speech in words does not cause the loss of information about the phoneme context. We can build a second-deep network neural model within the range of word pronunciation. This model can use the deep learning neural network of the LSTM + CTC consistent with the first model, or the model of the LSTM + Attention mechanism can output text in phoneme levels. In the experimental part, we have performed experiments on the above two models, analyzed and compared the experimental results.

A. CTC-BASED SPEECH AND TEXT ALIGNMENT MODEL

The main goal of this model is to enforce phoneme-level alignment of sentences to be evaluated and to output alignment results in word units.

As shown in Figure 5, the MFCC transform is performed on the input voice signal. The human ear, according to the study of the hearing mechanism, has different hearing sensitivities to sound waves of different frequencies. The speech signal from 200Hz to 5000Hz has a great impact on speech intelligibility. When two sounds of different loudness are acting on human ear, the presence of frequency components with higher loudness will affect the perception of frequency components with lower loudness, making it difficult to perceive. This phenomenon is called the masking effect.

Secondly, a bidirectional BLSTM operation is performed on the characteristics of the MFCC for each sentence audio generation. Long short-term memory (LSTM) proposed by [49] is a special RNN, mainly to solve the problem of gradient disappearance and gradient explosion during long sequence training. In simple terms, LSTM can perform better in longer sequences than ordinary RNNs. Since speech evaluation is not real-time speech recognition, BLSTM can be used for a bidirectional decoding operation. LSTM is mainly composed of forget gate, input gate, and output gate.

The end of the CTC-based speech text alignment model is CTC decoding and forced alignment output. For the speech recognition task, if we now have a clipped speech and corresponding text, we don't know how to map the speech segment to the text, and this make it difficult to train a speech recognizer. CTC decoding is applied in speech recognition tasks to solve the alignment problem. The task of speech recognition CTC is to find the sum of the probabilities of all output text paths and the best text output path. The goal of our forced alignment is that in the BLSTM output result, the output target text is known, and we have to find the path of the largest possible output to the target text. In order to solve this problem, this paper is inspired by the decoding of Viterbi in [46] and combines the CTC decoded output features to find the output path to the target text and complete the phoneme-level forced alignment. In the word-level alignment result test, a near-perfect alignment is achieved. We will introduce the decoding method of CTC in detail in section 3 and section 4.

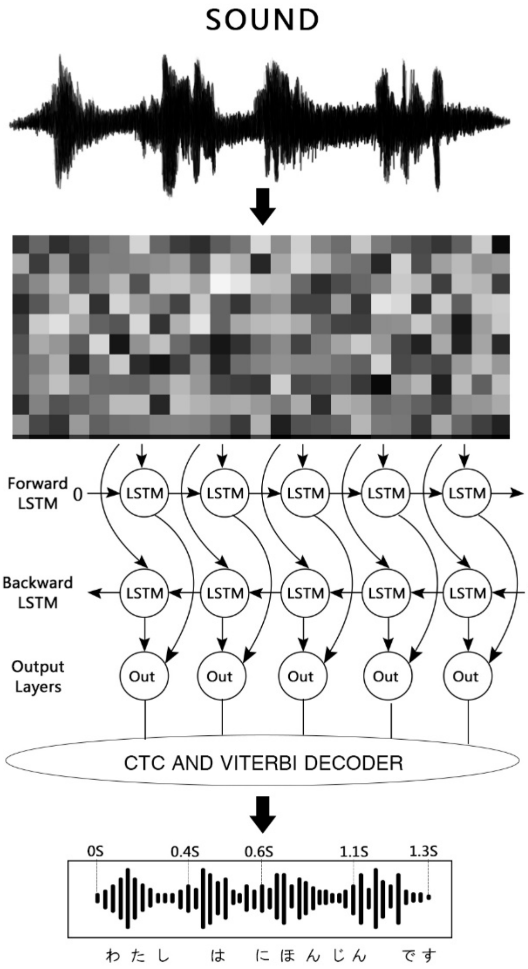


FIGURE 5. One of the Double-layer DNN models for word-level forced alignment.

B. ATTENTION-BASED SPEECH RECOGNITION MODEL

Model 2 implements the phoneme speech recognition model based on the Attention mechanism.

The Encoder-Decoder model can predict any sequence correspondences, but at the same time, a major problem is that the accuracy from encoding to decoding depends heavily on a fixed-length semantic vector. There is a loss of information during the compression of the input sequence to the semantic vector. In a longer sequence, the previous input information is easily covered by the subsequent input information. In order to solve this problem, an attention mechanism is added to the Seq2Seq model. The context used in predicting the output at each moment is the context related to the current output, rather than the same vector. In this way, when each outputs the prediction result, the elements in each semantic vector will have different weights, making the prediction result more targeted. The part of Model 2 in the encoder is mainly composed of two modules, CNN and BiLSTM, and the decoding part is based on the RNN output of Attention. The unit for output in this paper is word-level, so the length of the sequence is shorter, and the accuracy will be better than the sentence model.

In the Attention model, the original audio is first subjected to MFCC feature vector transformation, and then a standard CNN convolution and pooling operation is performed. Using a CNN model similar to [50], our CNN part is not processed at the full link layer because we want to perform a visual operation after Attention to record the historical process of attention. The MFCC transformed feature vector is an ordered feature representation. The feature vectors in the same batch input have the same rows and columns. Each column is represented from left to right, that is the phoneme order of the sound. Each column of input in our model is a single vector. Localized convolution, pooling, activation function operations, and their transfers keep invariant. Therefore, a series of rectangular-like vectors generated by the CNN's convolution operation are consistent with the MFCC input from left to right. The receptive field is used to represent the size of the original image's perception range of different neurons in the network, or the size of the area where the pixels on the feature map output by each layer of the CNN are mapped on the original image. As shown in Figure 7, each vector is related to the timing of the phoneme and represents the sound information in this area.

CNN's encoder processing is to convert rough speech signals into regular data. The length and width of the input vector are consistent, that is, coarse-to-fine. This will be more conducive to the bidirectional LSTM encoding processing. The principle for bidirectional LSTM module is the same as that of model one. The last part and also the most important one is Attention decoded output; we will introduce it in detail in section 5.

Both the aligned model and the word recognition model use bidirectional BLSTM for encoder processing. The difference is that the encoder in Figure 4 is for sentence level, while that in Figure 6 is for word-level. Therefore, both models need to be trained separately. In the next two sections, we will introduce the core algorithms of each model in detail.

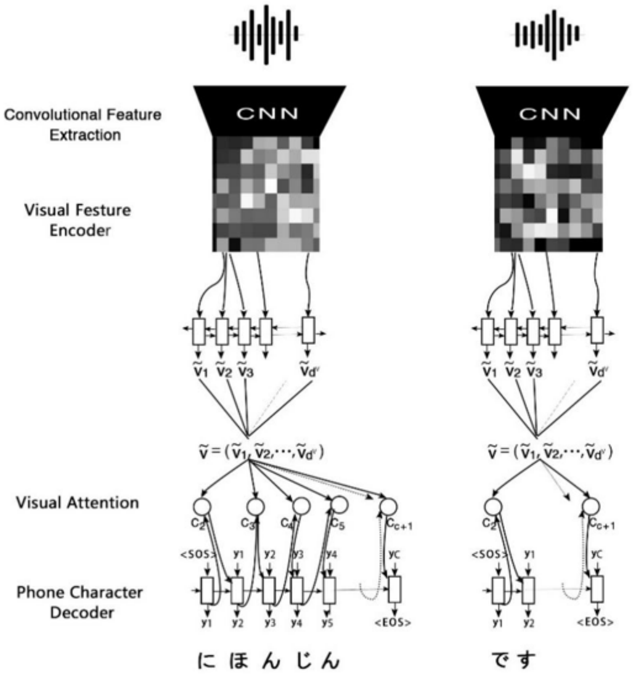

Convolutional Feature Extraction
Visual Festure Encoder
Visual Attention
Phone Character Decoder
CNN
CNN
$\tilde{v}_1$ $\tilde{v}_2$ $\tilde{v}_3$ $\tilde{v}_{d'}$
$\tilde{v} = (\tilde{v}_1, \tilde{v}_2, \cdots, \tilde{v}_{d'})$
c_2 c_3 c_4 c_5 c_{c+1}
<SOS> y1 y2 y3 y4 yC
y1 y2 y3 y4 y5 <EOS>
に ほ ん じ ん
です

FIGURE 6. Phonetic-level word pronunciation evaluation model based on attention.

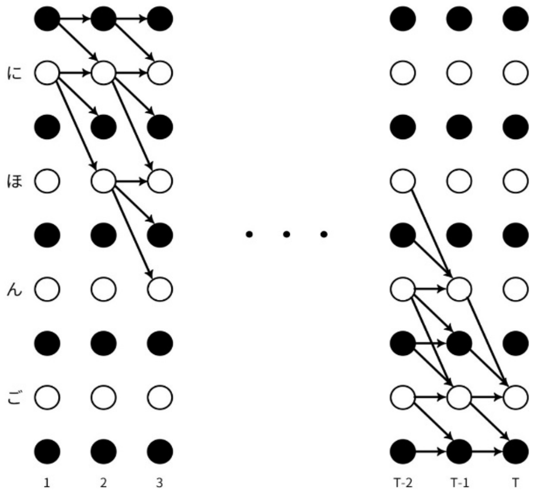


FIGURE 7. CTC forward-backward algorithm for Japanese.

III. CTC ALGORITHM AND VITERBI ALIGNMENT

In the speech recognition task, if we now have a clipped speech and corresponding text, we don't know how to map the speech segment to the text, and this will make it difficult to train a speech recognizer. In order to solve this problem, we can first formulate a rule, such as "one character corresponds to a certain length of language fragment input". For different people, they speak at different speeds, which will make the above rules infeasible. Another solution is to manually align the position of each character in the audio. This method is extremely effective to train our model, but there is no deny that this method is very time-consuming. Connectionist Temporal Classification (CTC) is suitable for this algorithm used when the input and output are not aligned, so CTC is suitable for speech recognition. The method of

CTC focuses on the results of an input sequence to an output sequence, so it only cares whether the predicted output sequence is close to the real sequence, rather than whether each result in the predicted output sequence is exactly aligned with the input sequence at the time point. In a task with high training accuracy, as its name suggests, CTC is specifically designed for temporal classification tasks in [44]; that is for sequence labeling problems where the alignment between the inputs and the target labels is unknown.

A. CTC ALGORITHM

The CTC algorithm can assign a probability for any $[y_1, y_2, y_3, \ldots y_u]$ given an $[x_1, x_2, x_3, \ldots x_t]$. The key to computing this probability is how CTC processes alignments between inputs and outputs. We'll start by looking at these alignments and then show how to use them to compute the loss function and perform inference.

To identify the specific form of the CTC alignments, we should first consider a simple approach. Let's use an example. Assume the input has length of 10 and Y = [に,ほ,ん,ご], in Japanese phonetic. One way to align XXXX and YYYY is to assign an output character to each input step and collapse repeats (Table 1).

TABLE 1. One simple example of CTC alignment.

Input (X)	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
align	に	に	に	ほ	ほ	ん	ん	ご	ご	ご
Output (Y)	に			ほ	ん			ご		

There are two problems for this approach. Firstly, it doesn't make sense to force every input step to align with some outputs. In speech recognition, for example, the input can have stretches of silence with no corresponding output. Secondly, we cannot generate outputs with multiple characters in a row. Considering the alignment [お,お,お, き,き,い,い,], collapsing repeats will produce "おきい" instead of "おおきい."

To address these problems, CTC introduces a new token to the set of allowed outputs. This new token is sometimes called the blank token. We'll refer to it here as ϵ. The ϵ token doesn't correspond to anything and is simply removed from the output.

The alignment length allowed by CTC are the same as the input. We allow any alignment which maps to YYY after merging repeats and removing ϵ tokens, as shown in Table 2:

TABLE 2. One example of CTC alignment with blank token.

	お	お	お	ϵ	お	ϵ	き	き	い	い
Merge	お			ϵ	お	ϵ	き		い	
remove ϵ	お				お		き		い	
output	お	お	き	い						

If Y has two same characters in one row, a valid alignment must have an ϵ between them. With this rule in place, we can differentiate the alignments which collapse to "hello" from those collapse to "helo."

Both CTC and attention are training and decoding processes based on LSTM. The greatest advantage of LSTM is that it can remember the input at each step. Therefore, in the speech model, although を and お have same pronunciation, the model can calculate which one should be the correct output according to the content of the pronunciation before and after.

Let's go back to the output [に,ほ, ん,ご] with an input length of 10. Here are a few more examples of valid and invalid alignments. Table 3 is a valid alignment of CTC output, and Table 4 is an invalid alignment of CTC output.

TABLE 3. Valid alignments of CTC.

に	に	に	ほ	ほ	ん	ん	ご	ご	ご
に	に	ϵ	ほ	ほ	ん	ん	ϵ	ご	ご
ϵ	に	に	ほ	ほ	ん	ん	ご	ご	ϵ
に	ϵ	ϵ	ほ	ほ	ん	ん	ご	ご	ϵ

TABLE 4. Invalid alignments of CTC.

[ににほんご]	に	ϵ	に	ほ	ほ	ん	ん	ご	ご	ご
has length 9	に	に	ϵ	ほ	ほ	ん	ん	ϵ	ご	■
Missing 'h'	に	ϵ	ϵ	ほ	ほ	ϵ	ご	ご	ご	ϵ
[ににんほご]	ϵ	に	に	ん	ほ	ほ	ほ	ご	ご	ϵ

CTC alignments have a few prominent features. First, the allowed alignments between X and Y are monotonic. If we advance to the next input, we can keep the corresponding output the same or advance to the next one. The second feature is that the alignment of X to Y is many-to-one. One or more input elements can align to a single output element but not vice-versa. This implies a third feature: the length of Y cannot be greater than the length of X.

As shown in Figure 7, Black circles represent blanks, and white circles represent labels. Arrows signify allowed transitions. Forward variables are updated in the direction of the arrows, and backward variables are updated against them.

B. VITERBI ALIGNMENT BASED ON CTC RESULTS

The result of CTC algorithm calculates the sum of all possible output paths. The purpose of speech and text alignment is to find the output path with the highest probability among all output target text paths. In this section, we use に ほ ん ご as an example to perform dynamic decoding output on CTC results. After BiLSTM + CTC calculation, output weights are available at any time and at any node. We perform a softmax operation on the output weights, as shown in Figure 8:

First, the alignment operation also follows the calculation rules of CTC. For example, our Japanese output label "に ほ ん ご" is added to the initial transformation of CTC, so an output sequence of length 9 can be obtained, Which

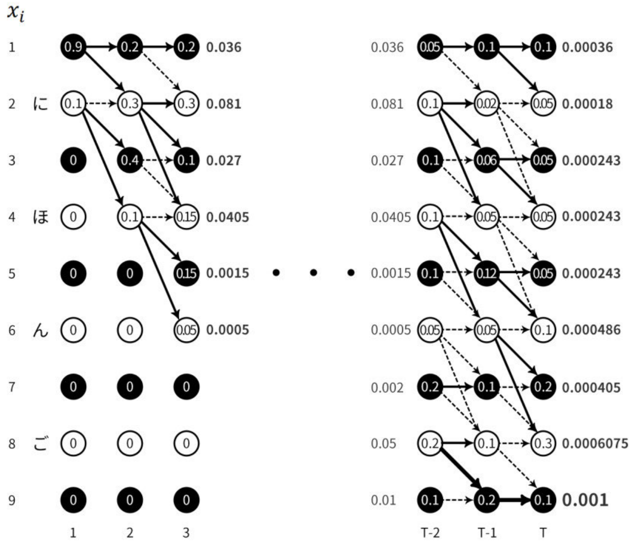


FIGURE 8. CTC Decode Result with the weights.

is [$x_{1,}x_2$, x_4, x_5, x_5, x_5, x_6, x_6, x_6, x_7, x_7, x_8, x_9], as well as [space, に, space, ほ, space, ん, space, ご, space]. The calculation of its maximum possible path is completed by the following six steps:

1) At T = 1, only the empty string or the first character of labels can be outputted.
2) At T = 2, the CTC algorithm rule is that if T = 1 is a space, then the space x_1 or the first string x_2 is output at T = 2; If T = 1 is x_2, then T = 2 can be x_2, spaces x_3, x_4. On the same path to x_2, there are two paths: path(x_2, x_2) with an output probability of 0.27, and path(x_1, x_2) with the output probability of 0.03. According to the dynamic programming algorithm based on Viterbi, the output probability of path(x_2, x_2) is bigger than path(x_1, x_2), so path(x_1, x_2) will be deleted, path (x_2, x_2) is reserved.
3) At T = 3, there are two paths to the x_2 point at the same time: path(x_1, x_1, x_2) and path(x_1, x_2, x_2). The corresponding output probabilities are 0.054 and 0.081. Output path (x_1, x_2, x_2) is reserved; at the second space x_3 has two paths path(x_1, x_2, x_3) and path(x_2, x_3, x_3). The output probability is 0.027 and 0.004, so the former output path (x_1, x_2, x_3) is reserved. There are three paths to the x_4 point, path(x_1, x_2, x_4), path(x_2, x_3, x_4), path(x_2, x_4, x_4), and the corresponding probabilities are: 0.0405, 0.006, 0.0015, so the largest Path path($x_{1,}x_2$, x_4) is reserved.
4) At step T-2, the output we consider does not have the limitation of applying the CTC inverse algorithm. The reason is that, in the actual voice evaluation process, there may be situations where the user does not finish reading. Therefore, we keep the output probabilities of all x values.

At step T-1, according to the rules of CTC forward algorithm, continue to calculate the probability value of each output point at the 14th time point. If there are multiple paths output at the same point, according to the Viterbi rule with only one maximum path, the path with less weight will be deleted, which is showed by the dotted line in Figure 8.

TABLE 5. Max probabilities to each path in the CTC results.

Time	Paths	probabilities
1	(x_1,x_1,x_1,x_1,x_1,x_1,x_1,x_1,x_1,x_1,x_1,x_1,x_1)	0.00036
2	($x_{1,}x_2$,x_2,x_2,x_2,x_2,x_2,x_2,x_2,x_2,x_2,x_2,x_2)	0.00018
3	($x_{1,}x_2$,x_2,x_3,x_3,x_3,x_3,x_3,x_3,x_3,x_3,x_3,x_3)	0.000243
4	($x_{1,}x_2$,x_4,x_4,x_4,x_4,x_4,x_4,x_4,x_4,x_4,x_4,x_4)	0.000243
5	($x_{2,}x_4$,x_5,x_5,x_5,x_5,x_5,x_5,x_5,x_5,x_5,x_5,x_5)	0.000243
6	($x_{1,}x_2$,x_4,x_5,x_5,x_6,x_6,x_6,x_6,x_6,x_6,x_6,x_6)	0.000486
7	($x_{1,}x_2$,x_4,x_5,x_5,x_5,x_6,x_6,x_6,x_7,x_7,x_7,x_7)	0.000405
8	($x_{1,}x_2$,x_4,x_5,x_5,x_5,x_6,x_6,x_6,x_7,x_7,x_8,x_8)	0.0006075
9	($x_{1,}x_2$,x_4,x_5,x_5,x_5,x_6,x_6,x_6,x_7,x_7,x_8,x_9)	0.001

5) At step T, we got the final output, and calculated the top output path, as shown in Table 5:

Path($x_{1,}x_2$, x_4, x_5, x_5, x_5, x_6, x_6, x_6, x_7, x_7, x_8, x_8, x_9, x_9) = 0.001. Therefore, a complete one-to-one correspondence between the audio and target text is also achieved, and the task of aligning speech with text is completed.

TABLE 6. Viterbi algorithm on CTC output.

```
Algorithm 1 find the best path for speech and text
Input: CTC Output Result (3-D matrix)
Output: the path with probability (List)
for t in range(T): #t is the Time step, T is the length of output
  if t==0: #at the first step, only the blank and first char
    outpath.append(0)
    outpath.append(1)
  else:
    if outPath[index] is "blank": #index is the last label pos
        outPath.append(index)
        outPath.apened(index+1)
    else: #if the last label is a character
        outpath.append(index)
        outpath.append(index+1)
        outpath.append(index+2)
  for path in (outPath): #merge the same road
    if newPath have the path:
        if newPath.value<path.value:
            newsPath=path
    else:
        newPath.add(path)
  outPath=newPath
for p in outPath #export the best path with max probability
  if p.value>maxValue:
    maxValue=p.value
```

As shown in Table 6, this is the pseudocode, which finished to find the max probability path on the CTC output results. The core algorithm is consistent with that of CTC. If the output meets "blank", the next step can have two results: one is still blank itself; the other is the next character. When the output is the label, the next output has three results: one is still the character itself; the other two are the next character or "blank." Based on Viterbi algorithm, the path with the highest probability passes a certain point of the fence network, then the sub-path from the starting point to this point must also be the most probable path from the beginning to this point. We can delete the path that has a smaller probability on the same point, to improve the time complexity. This algorithm can be calculated in a real-time environment. In the last part of the pseudocode, we export the one path with probability.

Viterbi algorithm in [45] is a dynamic programming algorithm used to find the Viterbi path, the hidden state sequence that is most likely to produce the sequence of observation events, especially applied in the context of Markov sources and hidden Markov models. Through the combination of CTC and Viterbi algorithms, the optimal path that is consistent with the target text is found. Because second language learners cannot complete the target sentence in the evaluation task, we have not added the constraint function of the CTC reverse algorithm. In the above example, the path probability is relatively ideal. Even the probably multiplied value can be calculated. In practical applications, some concepts will be very small. If they are multiplied consecutively, the values will be out of bounds, so we can take the Log value and change the multiplication to the addition, which can effectively solve the problem of multiplication of small values.

Because speech alignment and speech recognition targets are different, the largest possible path of our alignment model output is inconsistent with the beam search results of speech recognition. Our model calculates the path that is most likely to be consistent with the target text when the target text is known. The beam search result of speech recognition is the most likely text to be obtained when the target text is unknown. In the next section, we will perform speech error detection based on the alignment results, and also verify the accuracy of our speech evaluation results.

IV. END-TO-END SPEECH RECOGNITION MODEL BASED ON ATTENTION

Our attention model combines several standard neural components from vision and natural language processing. Unlike most visual attention models, our model uses a full grid encoder over the input wav files, so that it not only makes the data from Coarse to Fine but also supports the left-to-right order.

The model first changes wav file into image features by MFCC, then extracts image features using a convolutional neural network (CNN) and arranges the features in a grid. Each row is then encoded, using a special recurrent neural network: Long short-term memory (LSTM). These encoded features are then used by an RNN decoder with a visual attention mechanism. The decoder implements a conditional language model over the Japanese pronunciation. The whole process is as follows.

A. CONVOLUTIONAL NETWORK

The visual features of a wav MFCC results are extracted with a multi-layer convolutional neural network interleaved with max-pooling layers. This network architecture is now standard; we model it specifically after the network used by [47] for visual images (the specification is given in Table 6.) Unlike some recent CNN works, we do not use fully connected layers, since we want to preserve the locality of CNN features in order to use visual attention.

B. ROW ENCODER

In attention-based visual captioning, the image feature grid can be directly fed into the decoder [48]. For MFCC features, the visual features fed in the decoder contain significant relative sequential order information. Therefore, we use an additional RNN encoder module that re-encodes each row of the grid. The reasons are as following: (1) the MFCC features are in a left-to-right order, which can be easily learned by the encoder, (2) RNN can utilize the surrounding context to refine the hidden representation. Formally, a recurrent neural network (RNN) is a parameterized function RNN that recursively maps an input vector and a hidden state to a new hidden state.

C. DECODER

As shown in Figure 9, h^1, h^2, h^3, h^4 are the encoded input which are from LSTM results. Attention is actually a match between the current input and output. In Figure 9, the first attention is the match between h^1 and z^0. h^1 is output vector for the hidden layer of RNN at the current moment, and z^0 is initialization vector, such as initial memory in RNN. Match is a module that calculates the similarity of two vectors; α_1^0 is the similarity calculated by match.

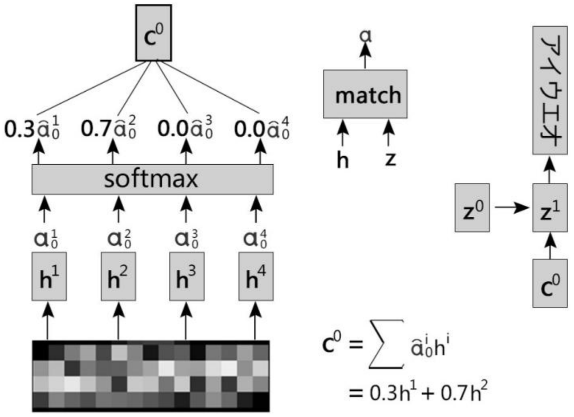


FIGURE 9. Attention-based model for Japanese recognition.

There are many methods for the calculation of the similarity between two vectors: cosine similarity, a Simple Neural Network, or Matrix transformation. After knowing how to compute the similarity, we can get all inputs similarity to each output. Then, we use SoftMax function to normalization, and let the sum of all weights at the output be 1. When we get c^0, as the input of decoding RNN, then we will get the first timing output z^1, which is decided by c^0 and z^0. After we get the value of z^1, then instead of z^0 computing the similarity with the encoder vector, this cycle lasts until the end.

The calculation formula is as follows:

$$u_t^i = v^T \tanh\left(W_1 h_i + W_2 Z_t\right) \tag{1}$$

$$\alpha_i^t = \mathrm{softmax}\left(u_t^i\right) \tag{2}$$

$$c^t = \sum\nolimits_{i=1}^{T} a_t^i h_i \tag{3}$$

The vector v and matrices W_1', W_2' are learnable parameters of the model. The vector u_t has length T and its i-th item contains a score of how much attention should be put on the i-th hidden encoder state h_i. These scores are normalized by SoftMax to create the attention mask at over encoder hidden states.

D. ATTENTION MECHANISM AND TRAINING

In this paper, a deep neural network is implemented as encoder and attention-based sequence-to-sequence model as decoder. This model is similar to that in [48], using LSTM or GRU as encoder. In the model of normalization, it uses sampled SoftMax in [49] to output. These scores are normalized by SoftMax to create the attention mask α^t over encoder hidden states.

In all our experiments, we use the same hidden dimensionality (256) at the encoder and the decoder, sov is a vector and W_1', W_2' are square matrices. Lastly, we concatenatec$_t$, which becomes the new hidden state from which we make predictions, and then fed it to the next time step in our recurrent model.

In the Attention model, z^0 stands for the <start> identifier, z^t is the <end> identifier, so they are not involved in the calculation of the correct rate in the formal output results. After the speech recognition task is completed, the phonetic symbols recognized by each word are outputted through the attention model. We calculate the edit distance of the characters between the output phonetic symbol and the reference phonetic symbol to get the correct rate of the pronunciation of the word. The experimental data and results will be introduced in detail in next sections.

V. INTRODUCTION OF JAPANESE SPEECH DATASETS

The data set in this paper consists of two parts: one is the correct pronunciation of the announcer (a native speaker) and the correct pronunciation of its example sentence; the second is the pronunciation of words and sentences that users read every day, and about 20,000 sentences of Japanese pronunciation are generated every day. The user's pronunciation is either correct or wrong, so we use the correct part data, which is scored by the HMM model. The data set will have a very large impact on the model's results. The experimental results will prove that it will have a very good effect on training our first model to perform the alignment, using the user's data set. The reason should be the same as the training of speech recognition. The more extensive the pronunciation data from various spoken languages is, the wider the coverage, and the better the effect will be.

The data is from the wordbooks for the Japanese Language Proficiency Test (JLPT). It includes ten thousand words and their example sentences. And there are six classic standard Japanese textbooks, which include words with sentences and sentences in lessons. A total number of 26,950 words and 19,398 sentences. All words and sentences are pronounced in standard Japanese. We use Japanese word segmentation system in [50] for each sentence, which can complete the word segmentation of Japanese sentences and the pronunciation of hiragana, as shown in Table 7.

TABLE 7. Sample japanese sentence composition.

Sentence	毎日果物を食べています。
Words	毎日',果物','を','食べ','て','い','ます',
hiragana	マイニチ','クダモノ','ヲ', 'タベ','テ','イ','マス'

In the Japanese fifty phonetic diagrams, each kana represents a phone, so it belongs to syllabic letters. Japanese kana include voiceless, voiced, half-voiced, and dialed sounds. Among them, there are 5 basic vowels, 41 consonants, and 4 non-spell able vowels. A total of 80 phone kana are used in this paper.

VI. EXPERIMENT FOR SPEECH TEXT ALIGNMENT

The most important step in speech pronunciation evaluation is to align speech and text. Our first model is basic. Only when the speech and text are aligned correctly, the recognition evaluation of the second model will be correct. In this experiment, we applied 10,000 sentences for pre-training to output in words. The experimental method uses the Viterbi text alignment based on the results of the CTC algorithm proposed in section 4.1 to obtain the largest similar speech-to-text path.

As input features, we use 80 Mel-scale filter bank coefficients with pitch features as suggested in [51], [52] for the BLSTM encoder, and add their delta and delta features for the BLSTM encoder [53]. The encoder is a 2-layer BLSTM with 256 cells in each layer and direction, and the linear projection layer is followed by each BLSTM layer. The 2nd and 3rd bottom layers of the encoder are reading every second hidden state in the underlying network, reducing the utterance length by the factor of 4 (subsampling).

The AdaDelta algorithm with gradient clipping is used for the optimization. The beam width is set to 10 in decoding under all conditions. The joint CTC ASR is implemented by using the Chainer deep learning toolkit [54].

As Table 8 shows: the sentence column represents the number of sentences tested; the percent column represents the proportion of the number of sentences. We divide the sentence-level speech recognition results into 1-6 Levels according to the correct speech recognition rate of each sentence. The accuracy rate of sentence speech recognition is about 81.6%.

During the speech recognition of sentences, we also output the results of speech and text alignment. The system will save the voice of each word as a wav file after alignment. For wav

TABLE 8. The Japanese speech recognition result for sentence.

level	score	sentence	percent
1	100	62	0.024721
2	90-99	339	0.135167
3	80-89	905	0.360845
4	70-79	826	0.329346
5	60-69	276	0.110048
6	~-60	100	0.039872
Total		2508	1

file of each word, we use the same model for word-level speech recognition training and testing. The results are shown as follows in Table 9:

TABLE 9. The Japanese speech reorganization result for words.

level	100	90-99	80-89	70-79	60-69	~-60	
1	233	0	0	8	20	58	319
2	1918	0	7	85	122	600	2732
3	5811	0	24	251	354	1673	8113
4	5342	0	12	276	321	1564	7515
5	1618	0	3	84	94	568	2367
6	391	0	3	24	26	123	567
total	15313	0	49	728	937	4586	21613

In Table 9, the levels are the same as in Table 8, scoring level of the corresponding sentence. The values in the table represent the number of words in different segments. For example, the 233 words means the word speech recognition for a number of 233 words after segmentation in a sentence of level 1 (100 points for speech recognition) is completely correct. We have noticed that 90-99 words are segmented and the number of words is 0. This is because a word does not reach 10 phonemes in a sentence, so once a phoneme speech recognition error occurs, it will cause the entire word speech recognition accuracy rate to be less than 90%. Word-level scoring accuracy is based on the character editing distance.

We sample the words in Table 9 to manually check the alignment results. After manual inspection, it is confirmed that in the speech and text alignment results with a word score of more than 60 points, all the speech and text consistency is 100%, even if the speech recognition is inaccurate. It is because the speech recognition rate cannot reach 100%. For the points below 60, we conduct a sample survey of the alignment of speech and text from level 1 to level 6. The statistical results are shown in Table 10:

As shown in Table 10, there are 17,027 words in the word score of 60-100, all of which have the same pronunciation and text alignment. In the word evaluation test results of speech less than 60, about 465 words will have misalignment. Most of these alignment deviations come from intercepting some syllables of the previous word together. In this case, we also treat them as alignment errors. Overall, the speech-to-word alignment results reach more than 97%. Therefore, the experimental results verify the effectiveness of the alignment output based on CTC speech recognition results. This provides basis for our next word-level morpheme speech recognition and evaluation.

TABLE 10. Speech and text alignment results.

level	60-100	error	words	~-60	error	words
1	261	0	0	58	0	0
2	2132	0	0	600	0.1	60
3	6440	0	0	1673	0.1	167.3
4	5951	0	0	1564	0.1	156.4
5	1799	0	0	568	0.1	56.8
6	444	0	0	123	0.2	24.6
Total	17027		0	4586		465.1

Error rate： 0.021

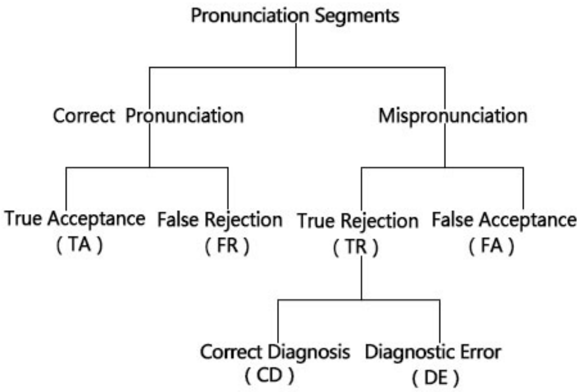


FIGURE 10. hierarchy tree of mispronunciation in pronunciation segments.

VII. SPEECH EVALUATION EXPERIMENTS AND RESULTS

In this section we will introduce our speech evaluation experiments and results. Experiments are performed on two speech models: CTC and Attention, and the experimental results are compared and analyzed.

There are many evaluation methods for computer-aided pronunciation training technologies (CAPT). For example, the correlation coefficient is commonly used when the scores given by the evaluators are continuously valued and when the discrete or class-wise prediction is given by the evaluators, however, confusion matrix-based metrics such as Cohen's Kappa value. The false acceptance rate (FAR)/false rejection rate (FRR) or precision/recall are often used. Some adopt modified or weighted versions of these basic metrics. User studies are another popular approach for evaluating the effectiveness of a CAPT system.

In our experiments in this section, we basically follow the previously defined hierarchy of mispronunciation detection. As shown in Figure 10:

In Figure 10, correct pronunciation includes true acceptance (TA) and false rejection (FR). The wrong pronunciation can be divided into correct rejection (TR) and wrong acceptance (FA). In the case of true rejection, it can be divided into correct diagnosis (CD) and diagnosis error (DE).

$$\mathrm{FRR} = \mathrm{FR}/(\mathrm{TA}+\mathrm{FR}) \tag{4}$$

$$\mathrm{FAR} = \mathrm{FA}/(\mathrm{FA}+\mathrm{TR}) \tag{5}$$

$$\mathrm{DER} = \mathrm{DE}/(\mathrm{CD}+\mathrm{DE}) \tag{6}$$

Among them, FR is the number that identifies the correct phoneme as the wrong one; TA is the number of phonemes that identify the correct phoneme as the wrong phoneme; FA is the number of correct phonemes diagnosed as wrong. TR means correctly diagnosis errors. CD is based on the diagnosis of the wrong phoneme, which means the wrong pronunciations can be diagnosed successfully; While, DE stands for the number of phonemes diagnosed incorrectly.

The experiment is based on the Attention speech recognition model. The words in MP3 audio file are preprocessed and converted into a 16,000 Hz wav file. Then the input data is transformed into vector data by MFCC processing. The vector length is 13. On this basis, convolution and maximum pooling of vector data are performed. The parameters are shown in Table 11.

TABLE 11. CNN definition for speech reorganization.

CONV				Pool	
c: 64	k: (3,3)	s: (1,1)	p: same	po: (2,2)	s: (1,2,2,1)
c: 128	k: (3,3)	s: (1,1)	p: same	po: (2,2)	s: (1,2,2,1)
c: 256	k: (3,3)	s: (1,1)	p: same	po: (2,1)	s: (1,2,1,1)
c: 512	k: (3,3)	s: (1,1)	p: same	po: (2,1)	s: (1,2,1,1)
c: 512	k: (2,2)	s: (1,1)	p: valid		

'Conv': convolution layer, 'Pool: max-pooling layer. 'c': number of filters, 'k': kernel size, 's': stride size, 'p': convolution style, 'po':, kernel size, in pooling layer.

Our experiment runs on TensorFlow, and the max input size is 780*32, and the output phone number is 60. The gradient optimization algorithm is Adadelta.

Our Attention model has a total of 7,901,254 training learning parameters. The first phonetic alignment model was trained using about 17,776 Japanese sentences, which is about the total data volume generated by second language users in one month. After sentence training was completed, the first model was used to perform word-level segmentation of the sentence, and 109,716 Japanese word audios were generated. Based on this data set, we applied a GPU G 2080Ti to train about 40 epoch CTC models and Attention models. The experimental results were compared with the direct output of the first model as follows in Table 12:

TABLE 12. Result for Japanese speech assessment.

	Result	TA	FR	FA	TR	CD	DE
CTC(1)	94.5%	1465	64	40	147	107	40
CTC(2)	76.8%	1213	281	30	168	79	89
Attention	83.8%	1399	66	12	223	183	40

Ctc(1) indicates the first CTC model for alignment, ctc(2) indicates the second CTC model for speech reorganization, attention(2) indicates the attention model for speech reorganization.

The Result column represents the speech recognition accuracy rate of each model. TA, FR, FA, TR, CD, and DE of Table 12 correspond to the number of results for each leaf shown in Figure 10. As shown in the test results data set, although the first model can achieve a high accuracy of speech recognition, it is easy to overlook incorrect pronunciations because sentence-level recognition contains information between word contexts. FA is the highest value of False Accept, reaching 40. The models of CTC and Attention in Model 2 are completely independent of the context information training in the sentence, so they do not contain any word context information to better recognize the pronunciation of independent words. From the experimental results, in the comparison with the second model, the Attention model is significantly better than the CTC model, and its phoneme-level speech recognition accuracy rate has reached 83.3%. It works very well in phoneme recognition of independent words. At the same time, the value of FA, False Accept, is much lower than the value of the first model, because the second model is word-level training, and it will not automatically accept wrong pronunciations through context. Its TR, True Rejection, reaches 223, which is also significantly higher than the values of model 1 CTC and model 2 CTC. More importantly, the CD value reachess 183, which can better identify the wrong pronunciation, that is, the correct diagnosis.

According to the statistical results in Table 13, we can get the error recognition rate of each dimension as follows:

TABLE 13. Error result for Japanese speech assessment.

	FRR(%)	FAR(%)	DER(%)
CTC(model 1)	4.1	21.3	27.2
CTC(model 2)	18.8	15.5	52.9
attention (model 2)	**4.5**	**5.1**	**17.9**

The experimental results of Tables 13 show that although Attention's model is 0.4 percentage points slightly higher in FRR, both FAR and DER are significantly better than CTC-based model 1 and model 2. Japanese speech recognition based on the two-layer model is more effective for Japanese speech pronunciation recognition and correct diagnosis.

As the saying goes, one coin has two sides. Since the dual model mechanism is enabled, the efficiency of the whole calculation time is slower than that of the single model. If the purpose of speech evaluation is to judge whether the words are correct or not at the speech recognition level, then Model 1 is more suitable for applications. If the goal is to require correct diagnostic results, then the dual model's first word-level segmentation and then independent recognition and diagnosis of the Attention model are a better choice. With the continuous improvement of hardware computing performance, the score for each sentence will be completed in 0.1 seconds for each model. Therefore, for the whole system, whether it is a single model or two models, it can be called a real-time speech evaluation and diagnosis system.

VIII. CONCLUSION AND FUTURE WORKS

This paper presents a Japanese speech evaluation system based on a two-layer deep learning model. The first CTC model is used to segment and align speech and text, and the input sentence is divided into speech files in words. The second Attentions model trains word-level speech recognition models and performs recognition and evaluation. The experimental results show that the two-layer model can achieve solid results in speech evaluation and diagnosis. The results of the second Attention test model show that FRR can identify the correct phoneme as the wrong phoneme with an error rate as low as 4.5%, and FAR, the recognition rate of identifying the wrong phoneme as the correct phoneme is as low as 5.1%. DER means that after identifying the wrong phoneme, the diagnostic recognition error rate is as low as 17.9%. It is very effective in providing correct feedback results for the learning of Japanese as a second language.

Another contribution of this paper is that all speech data comes from Japanese learners. Our system can collect the pronunciation of more than 3,000 sentences and about 15,000 words every day. There is no doubt that a model trained on a user-level pronunciation data set has a significantly higher speech recognition performance than a speech recognition model based on a standard announcer's pronunciation training. Therefore, the model trained with the Japanese learners' data set can perform speech segmentation and word text alignment accurately. In the first model experiment, our speech and text alignment accuracy can reach 97.5%.

The user's voice is a one-to-one sentence corresponding to the original text, so the entire data set does not need to be manually labeled. In the future, with more and more speech data contributed by users, the accuracy of our model will be further improved. In addition, the data set in this paper is from effective and complete user-followed sentences, so in the future, sentences that are not completely read by users can be further analyzed to improve the robustness of the model. Furthermore, with the future optimization of our model, we can analyze words that are easy to be pronounced incorrectly in different regions, and even the Japanese phonetic pronunciations that are easy to be pronounced incorrectly. The collection of big voice data can further diagnose the Japanese voices of people in different regions and allow the artificial intelligence speech test model to help people better learn language pronunciation.

Our dual model evaluation system has achieved good results in Japanese language learning, and it can also be applied to English and Chinese phonetic evaluation, to help people learn a second language such as English and Chinese. The different performances of various languages on the two-layer model are also worthy of future research. Due to limited time and the capacity of our research team, we also look forward to and welcome more researchers or teams to join the deep learning-based speech recognition and evaluation in the future. We would love to provide all our experimental user data and possible help.

REFERENCES

[1] M. Eskenazi, "An overview of spoken language technology for education," *Speech Commun.*, vol. 51, no. 10, pp. 832–844, Oct. 2009.

[2] K. Yoon, "Imposing native speakers' prosody on non-native speakers' utterances: The technique of cloning prosody," *J. Mod. Brit. Amer. Lang. Literature*, vol. 25, no. 4, pp. 197–215, 2007.

[3] M. Peabody and S. Seneff, "Towards automatic tone correction in non-native mandarin," in *Proc. Int. Conf. Spoken Lang. Process*, 2006, pp. 602–613.

[4] K. Nagano and K. Ozawa, "English speech training using voice conversion," in *Proc. 1st Internat. Conf. Spoken Lang. Process.*, Kobe, Japan, 1990, pp. 1169–1172.

[5] M. P. Bissiri, H. R. Pfitzinger, and H. G. Tillmann, "Lexical stress training of german compounds for Italian speakers by means of resynthesis and emphasis," in *Proc. 11th Austral. Int. Conf. Speech Sci. Technol.* Auckland, New Zealand: Univ. Auckland, 2006, pp. 24–29.

[6] T. Kawahara, H. Wang, Y. Tsubota, and M. Dantsuji, "English and Japanese CALL systems developed at Kyoto University," in *Proc. APSIPA ASC*, 2010, pp. 804–810.

[7] O.-P. Kweon, A. Ito, M. Suzuki, and S. Makino, "A grammatical error detection method for dialogue-based CALL system," *J. Natural Lang. Process.*, vol. 12, no. 4, pp. 137–156, 2005.

[8] E. Pellegrino and V. Debora, "Self-imitation in prosody training: A study on Japanese learners of Italian," in *Proc. SLaTE*, Leipzig, Germany, vol. 5, 2015, pp. 53–57.

[9] H. Meng, C.-Y. Tseng, M. Kondo, A. Harrison, and T. Viscelgia, "Studying L2 suprasegmental features in Asian Englishes: A position paper," in *Proc. Interspeech*, 2009, pp. 1715–1718.

[10] K. Imoto, Y. Tsubota, A. Raux, T. Kawahara, and M. Dantsuji, "Modeling and automatic detection of English sentence stress for computer-assisted English prosody learning system," in *Proc. 7th Int. Conf. Spoken Lang. Process.*, 2002, pp. 749–752.

[11] J. Tepperman and S. Narayanan, "Automatic syllable stress detection using prosodic features for pronunciation evaluation of language learners," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Mar. 2005, pp. 937–940.

[12] K. Li and H. Meng, "Automatic lexical stress and pitch accent detection for L2 English speech using multi-distribution deep neural networks," *Speech Commun.*, vol. 96, pp. 28–36, Feb. 2018.

[13] X.-J. Sun, "Pitch accent prediction using ensemble machine learning," in *Proc. Int. Conf. Spoken Lang. Process.*, 2002, pp. 953–956.

[14] J. Zhao, W.-Q. Zhang, H. Yuan, M. T. Johnson, J. Liu, and S. Xia, "Exploiting contextual information for prosodic event detection using auto-context," *EURASIP J. Audio, Speech, Music Process.*, vol. 2013, no. 1, p. 30, Dec. 2013.

[15] J. P. Arias, N. B. Yoma, and H. Vivanco, "Automatic intonation assessment for computer aided language learning," *Speech Commun.*, vol. 52, no. 3, pp. 254–267, Mar. 2010.

[16] K. Kyriakopoulos, K. M. Knill, and M. J. F. Gales, "A deep learning approach to automatic characterisation of rhythm in non-native English speech," in *Proc. Interspeech*, Graz, Austria, Sep. 2019, pp. 1836–1840.

[17] S. M. Witt, "Automatic error detection in pronunciation training: Where we are and where we need to go," in *Proc. Int. Symp. Autom. Detection Errors Pronunciation Training*, vol. 1, 2012, p. 133.

[18] K. Li, X. Qian, and H. Meng, "Mispronunciation detection and diagnosis in l2 English speech using multidistribution deep neural networks," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 25, no. 1, pp. 193–207, Jan. 2017.

[19] H. Franco, L. Neumeyer, Y. Kim, and O. Ronen, "Automatic pronunciation scoring for language instruction," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 1997, pp. 645–648.

[20] H. Franco, L. Neumeyer, M. Ramos, and H. Bratt, "Automatic detection of phone-level mispronunciation for language learning," in *Proc. Eurospeech*, Budapest, Hungary, 1999, pp. 851–854.

[21] S. M. Witt and S. J. Young, "Phone-level pronunciation scoring and assessment for interactive language learning," *Speech Commun.*, vol. 30, nos. 2–3, pp. 95–108, Feb. 2000.

[22] J. van Doremalen, C. Cucchiarini, and H. Strik, "Using non-native error patterns to improve pronunciation verification," in *Proc. Interspeech*, 2010, pp. 590–593.

[23] A. M. Harrison, W.-K. Lo, X.-J. Qian, and H. Meng, "Implementation of an extended recognition network for mispronunciation detection and diagnosis in computer-assisted pronunciation training," in *Proc. 2nd Workshop Speech Lang. Technol. Educ. (SLaTE)*. Warwickshire, U.K.: ISCA, Sep. 2009.

[24] G. Kawai and K. Hirose, "A method for measuring the intelligibility and nonnativeness of phone quality in foreign language pronunciation training," in *Proc. Int. Conf. Spoken Lang. Process. (DBLP)*, Jan. 1998, pp. 782–785.

[25] Y.-B. Wang and L.-S. Lee, "Improved approaches of modeling and detecting error patterns with empirical analysis for computer-aided pronunciation training," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2012, pp. 5049–5052.

[26] Y.-B. Wang and L.-S. Lee, "Supervised detection and unsupervised discovery of pronunciation error patterns for computer-assisted language learning," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 3, pp. 564–579, Mar. 2015.

[27] K. Truong, A. Neri, C. Cucchiarini, and H. Strik, "Automatic pronunciation error detection: An acoustic-phonetic approach," in *Proc. InSTIL/ICALL*, 2004, pp. 3040–3051.

[28] X.-J. Qian, H. Meng, and F. Soong, "The use of DBN-HMMs for mispronunciation detection and diagnosis in L2 English to support computer-aided pronunciation training," in *Proc. Interspeech*, 2012, pp. 755–758.

[29] A. Lee, Y. Zhang, and J. Glass, "Mispronunciation detection via dynamic time warping on deep belief network-based posteriorgrams," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 8227–8231.

[30] W. Hu, Y. Qian, and F. Soong, "A new DNN-based high quality pronunciation evaluation for computer-aided language learning (CALL)," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (Interspeech)*, Jan. 2013, pp. 1886–1890.

[31] W. Hu, Y. Qian, and F. K. Soong, "A new neural network based logistic regression classifier for improving mispronunciation detection of l2 language learners," in *Proc. 9th Int. Symp. Chin. Spoken Lang. Process.*, Sep. 2014, pp. 245–249.

[32] S. H. Yang and M. Chung, "Self-imitating feedback generation using GAN for computer-assisted pronunciation training," in *Proc. Interspeech*, Sep. 2019, pp. 1881–1885.

[33] A.-R. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. Audio, Speech, Language Process.*, vol. 20, no. 1, pp. 14–22, Jan. 2012.

[34] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," *Comput. Sci.*, pp. 338–242, Feb. 2014.

[35] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.

[36] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2012, pp. 4277–4280.

[37] T. Parcollet, Y. Zhang, M. Morchid, C. Trabelsi, G. Linares, R. de Mori, and Y. Bengio, "Quaternion convolutional neural networks for end-to-end automatic speech recognition," in *Proc. Interspeech*, Sep. 2018, pp. 1–5.

[38] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, "End-to-end continuous speech recognition using attention-based recurrent NN: First results," *arXiv:1412.1602*, 2014. [Online]. Available: https://arxiv.org/abs/1412.1602

[39] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 4960–4964.

[40] L. Lu, X. Zhang, and S. Renais, "On training the recurrent neural network encoder-decoder for large vocabulary end-to-end speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 5060–5064.

[41] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2014, pp. 1764–1772.

[42] Y. Miao, M. Gowayyed, and F. Metze, "EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand. (ASRU)*, Dec. 2015, pp. 167–174.

[43] D. Amodei *et al.*, "Deep speech 2: End-to-end speech recognition in English and Mandarin," in *Proc. ICML*, 2015.

[44] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, 2006, pp. 369–376.

[45] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 2, pp. 260–269, Apr. 1967.

[46] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 11, pp. 2298–2304, Nov. 2017.

[47] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," *Comput. Sci.*, vol. 1409, pp. 2048–2057, Feb. 2015.

[48] O. Vinyals, L. Kaiser, T. Koo, S. Petrov, I. Sutskever, and G. Hinton, "Grammar as a foreign language," vol. 7449, Dec. 2014, *arXiv:1412.7449*. [Online]. Available: https://arxiv.org/abs/1412.7449

[49] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," vol. 1409, Sep. 2014, *arXiv:1409.0703*. [Online]. Available: https://arxiv.org/abs/1409.0473

[50] T. Kudo. (2005). *Mecab: Yet Another Part-of-Speech and Morphological Analyzer*. [Online]. Available: http://mecab.sourceforge.net/

[51] P. Ghahremani, B. BabaAli, D. Povey, K. Riedhammer, J. Trmal, and S. Khudanpur, "A pitch extraction algorithm tuned for automatic speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 2494–2498.

[52] Y. Miao, M. Gowayyed, X. Na, T. Ko, F. Metze, and A. Waibel, "An empirical exploration of CTC acoustic models," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 2623–2627.

[53] Y. Zhang, W. Chan, and N. Jaitly, "Very deep convolutional networks for end-to-end speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 4845–4849.

[54] S. Tokui, K. Oono, S. Hido, and J. Clayton, "Chainer: A next-generation open source framework for deep learning," in *Proc. Workshop Mach. Learn. Syst. (LearningSys) 29th Annu. Conf. Neural Inf. Process. Syst.*, 2015, pp. 1–6.

DEGUO MU (Member, IEEE) was born in Rizhao, Shandong, China, in 1980. He received the M.S. degree in software engineering from the Software College, Beihang University, where he is currently pursuing the Ph.D. degree in computer science and artificial intelligence with the School of Computer.

Since 2011, he has been working as an Assistant Professor with the Software College, Beihang University, and the Director of iYuba Lab. He had led the iYuba team to create more than 50s apps for learning language. In China, there have been about 50 million users learning English by iYuba apps until now. His research interests include deep learning on language education, the mobile Internet, and social networks. His awards and honors include the First prize of innovation and Entrepreneurship of MIIT, in 2018, and ten years outstanding graduates of Beihang Software College, in 2012.

WEI SUN (Senior Member, IEEE) was born in Sichuan, China, in 1961. He received the M.S. degree in computer science from Beihang University, and the Ph.D. degree from the University of Illinois, Chicago.

Since May 2003, he has been the Dean and a Professor with the Software College, Beihang University, training tens of thousands of software engineering masters in more than ten years. He has published and edited nine books, and more than 70 academic articles. He has served as the Chairman for the conference or a program committee for the four major international conferences of ACM and IEEE. He has also served as a member for more than 30 international conference program committees. He is also the Executive Chairman of the Beijing Software Industry Association, and a Professor with the FIU School of Computer Science, Florida State University, Miami, FL, USA.

GUOLIANG XU, photograph and biography not available at the time of publication.

WEI LI, photograph and biography not available at the time of publication.