IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 6, 2020, accepted February 11, 2020, date of publication April 16, 2020, date of current version April 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2988365

Improving Low-Resource Speech Recognition Based on Improved NN-HMM Structures

XIUSONG SUN, QUN YANG, SHAOHAN LIU, AND XIN YUAN
College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing 210023, China

Corresponding author: Xiusong Sun (949935857@qq.com)

This work was supported in part by the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, under Grant 1015–KFA19035, and in part by the project Research and Application of Intelligent Technology for Real-time Dispatching of Power Grid Based on Artificial Intelligence.

ABSTRACT The performance of the ASR system is unsatisfactory in a low-resource environment. In this paper, we investigated the effectiveness of three approaches to improve the performance of the acoustic models in low-resource environments. They are Mono-and-triphone Learning, Soft One-hot Label and Feature Combinations. We applied these three methods to the network architecture and compared their results with baselines. Our proposal has achieved remarkable improvement in the task of mandarin speech recognition in the hybrid hidden Markov model - neural network approach on phoneme level. In order to verify the generalization ability of our proposed method, we conducted many comparative experiments on DNN, RNN, LSTM and other network structures. The experimental results show that our method is applicable to almost all currently widely used network structures. Compared to baselines, our proposals achieved an average relative Character Error Rate (CER) reduction of 8.0%. In our experiments, the size of training data is ∼10 hours, and we did not use data augmentation or transfer learning methods, which means that we did not use any additional data.

INDEX TERMS Low-resource, speech recognition, multitask learning, acoustic modeling, feature combinations.

I. INTRODUCTION

A. BACKGROUND

Speech is the most important means for humans to transmit information to each other. A voice carries rich information such as the speaker's intention, identity, and emotion. This makes automatic speech recognition with the goal of human-computer interaction popular, and it has been a research hotspot in recent decades [1]. Automatic Speech Recognition (ASR) refers to the task of an automatic conversion from speech to text by computer. In real life, speech recognition can provide a natural and smooth human-computer interaction method. ASR has many applications, such as Apple's Siri, Microsoft's Cortana, and Xiaomi's Xiao Ai. In recent years, with the improvement of computer hardware capability and the development of neural network theory, deep learning has been applied to Automatic Speech Recognition.

Speech recognition systems based on phoneme level are currently mainly composed of acoustic models (AM), language models (LM) and Pronunciation models (PM). The acoustic model maps the acoustic features of each frames to the modeling unit, which is the phoneme. The language model corresponds the phoneme sequence obtained from the acoustic model to the sentence with the highest probability. The acoustic model is a kind of neural network structure, and it is also the main research point. Acoustic models are mainly divided into two types, one is the Neural Network Hidden Markov Models (NN-HMMs), and the other is the End-to-end models, such as Encoder-decoder structure [2]–[4] and Neural Network structure with Connectionist temporal classification (CTC) loss [5], [6]. The hybrid hidden Markov model (HMM) - neural network (NN) approach on phoneme level always need to take time to train GMM to align data before network training, and it can get better results in many tasks. Although the End-to-end models are currently developing rapidly, they always need a large amount of data to make the network convergence and their performance has not exceeded

The associate editor coordinating the review of this manuscript and approving it for publication was Xiao-Yu Zhang.

the NN-HMM structure, especially in low-resource speech recognition tasks, the models based HMM are much ahead.

B. RELATED WORK

As we all know, deep-learning [7] relies on a large amount of data, so the performance of the ASR system will be unsatisfactory in a low-resource environment. Therefore, improving the ASR under the condition of low resource has become a research hotspot because the acquisition of labeled speech data is usually difficult [8]. A common problem in low-resource environments is that the lack of training data often leads to overfitting of the neural network, which makes the model's performance on the test set worse. To prevent this problem, methods such as transfer learning, data augmentation, and unsupervised pre-training were born.

Transfer learning has been proposed for a long time, and SJ Pan *et al.* made a complete summary of it [9]. It can make full use of the data in the non-target domain to train a better initial model. It has shown promising results in many tasks such as image recognition [10], speech recognition [11], etc. Unsupervised pre-training also uses additional data to train a better initial model, but unlike transfer learning, transcribed data is not necessary. Those data without label helps networks to and capture more intricate dependencies between parameters and get a good initial marginal distribution. It has shown promising results in several areas, including Computer Vision (CV) [12], [41]–[44], Natural Language Processing (NLP) [13], [14] and so on [15]–[17].

Data augmentation [18]–[20] has been proposed for the purpose of studying low-resource language speech recognition for a long time. Kanda *et al.* investigated three distortion methods – vocal tract length distortion, speech rate distortion and frequency-axis random distortion. They evaluated those methods with Japanese lecture recordings and get lower word error. [21]. Jaitly *et al.* used Vocal Tract Length Perturbation (VTLP) to expand training data. When this technique is applied to TIMIT using Deep Neural Networks of different depths, the Phone Error Rate (PER) improved by an average of 0.65% on the test set [22]. Ko *et al.* proposed a method that changing the speed of the audio signal, producing 3 versions of the original signal with speed factors of 0.9, 1.0 and 1.1. They present results on 4 different LVCSR tasks with training data ranging from 100 hours to 960 hours, to examine the effectiveness of audio augmentation in a variety of data scenarios. An average relative improvement of 4.3% was observed across the 4 tasks. As far, the method of changing speed has the lowest implementation cost and achieve state-of-the-art performance [23]. In [24], A new method called SpecAugment is proposed and it consists of warping the features, masking blocks of frequency channels, and masking blocks of time steps. Data augmentation has been proved to be a simple and effective technique, not only in speech recognition but also in other fields such as image recognition [25] and keyword search [26], [27]. However, these methods are equivalent to adding training data, and do not solve the problem of overfitting.

Regularization is a technique to discourage the complexity of the model. It does this by penalizing the loss function. This helps to solve the overfitting problem. L1 and L2 are the most common types of regularization. These update the general cost function by adding another term known as the regularization term. Due to the addition of this regularization term, the values of weight matrices decrease because it assumes that a neural network with smaller weight matrices leads to simpler models. Therefore, it will also reduce overfitting to quite an extent. In many tasks, L2 has proved to achieve better results than L1, so the L2 regularization is widely used.

Inspired by the above research results, we adjusted the NN model structures and investigated the effectiveness of three approaches to improve the performance of the acoustic models in low-resource environments. The Mono-and-Triphone learning (MAT) is based on multitask learning. Multitask learning is an approach to inductive transfer that improves generalization by using the domain information contained in the training signals of related tasks as an inductive bias. It does this by learning tasks in parallel while using a shared representation; what is learned for each task can help other tasks be learned better [30]. We set up a second task to make both context-dependent (CD) and context-independent (CI) targets the learning goals of the network. Besides, we also investigated the effectiveness of the Soft One-hot Label (SOL). We used a new label encoding method based on Gaussian distribution to prevent the over-confidence of the models. We also compared the effect of different acoustic features on the acoustic model. At last, we applied all the three methods on the AMs based on HMM and achieve a remarkable result on the tasks of Mandarin speech recognition in a low-resource environment.

C. OVERVIEW

This paper is organized as follows: In Sect. 2, we will introduce the Mono-and-Triphone learning method (MAT) based on multitask learning. In Sect. 3, we will introduce our new label encoding method named Soft One-hot label (SOL). In Sect 4, the result of our choice of feature combinations is be shown and discussed. In Sect. 5, we will present our experimental setup, behaviors of the method and results of the experiments. In Sect. 6, we will list the contributions of the proposed method explicitly and summarize this paper.

II. MONO-AND-TRIPHONE LEARNING BASED ON MULTITASK LEARNING

In this subsection, we compare the Monophone (Context-Independent) target and Triphone. (Context-Dependent) target. Besides, we introduce the Mono-and-triphone learning method based on multitask learning.

The pronunciation of a word can be given as a series symbols that correspond to the individual units of sound that make up a word. These are called 'phonemes' or 'phones '. A monophone refers to a single phone. A triphone is simply a group of 3 phones in the form "$L - X + R$", where the "L" phone (i.e. the left-hand phone) precedes "X" phone

TABLE 1. Comparison of the monophone declaration and triphone declaration of the sentence "i like dog".

Sentence	**I like dog.**
Monophone-declaration	AY1 L AY1 K D AO1 G
Triphone-declaration	Sil-AY1-L L-AY1-K AY1-K-D K-D-AO1 D-AO1-G AO1-G-Sil

and the "R" phone (i.e. the right-hand phone) follows it. Table 1 shows an example of the conversion of a monophone declaration of the sentence 'I LIKE DOG.' to a triphone declaration. Sil denotes Silence, which means that this phoneme no left or right context phone. Lexical stress is indicated by means of a numeral {0,1,2} attached to a vowel.

Because triphone can better represent contextual information, when the triphone was proposed, it replaced the monophone and became the mainstream modeling method [28], [29]. It has also been shown to achieve better results than the monophone. However, there is a disadvantage to using triphones as DNN targets: there is no distinction between discrimination between different phones, and between different contexts of the same phone. The latter discrimination has a much more limited benefit to producing a more accurate phone hypothesis at test time, because our ultimate goal is just to get the correct phone, not the correct contextual information. However, the two discriminations are both treated equally in cross-entropy DNN training.

In addition, the number of modeling units of the triphone model is several times greater than that of the monophone model. In the CMU English dictionary, which has close to 130,000 word pronunciations, there are only 43 monophones, but there are close to 6000 triphones. This is also the case in Mandarin and any other language. Therefore, a second problem with triphone compared to monophone being the inherent data sparsity issue in having a large output layer. Increasing the number of output units obviously increases the number of weights to be trained between the output layer and final hidden layer, with fewer samples with fewer samples available to train each weight. Therefore, too many triphone modeling units and very little training data can easily lead to overfitting of the neural network.

To solve these problems, we investigated a new network structure based on multitask learning [30]. Our proposed structure is not only trained to optimize a triphone cross-entropy (CE) based loss and we give the network a second optimization task, which is the CE of monophone. The first task "Tri-task" is effectively a mapping from a set of T training frames to a set of Tri-labels, that is:

$$\begin{aligned}\text{Tri-}task : \{t : 1 \leq t \leq T\} &\Rightarrow \{\text{Tri-labels}\}\\ t &\Rightarrow LB_t^{Tri}\end{aligned} \tag{1}$$

where t denotes one frame, LB_t^{Tri} denotes its label under Tri-task.

The second task "Mono-task" is similar, except that the set of labels is different, and replaced with Mono-labels LB_t^{Mono}. These two tasks are combined by a hyper-parameter α, sharing the hidden layer of the neural network, and jointly optimizing the parameters of the neural network. Therefore, the final loss function as follow.

$$\begin{aligned}Loss = &-\sum_t^T \log p(LB_t^{Tri}|x_t;\theta^{Tri})\\ &-\alpha\sum_t^T \log p(LB_t^{Mono}|x_t;\theta^{Mono})\end{aligned} \tag{2}$$

where x_t denotes the acoustic features of each frame, $\theta^{\mathrm{Tri}}$and θ^{Mono}denote the parameters of the networks with different out layer. α denotes the weight of monophone loss. The loss function is minimized with respect to parameters θ when learning.

Although we have two output layers during training, we still use the triphone output layer as the final prediction result during prediction, which mainly considers that the triphone has a great advantage over monophone. The purposes of the monophone task are to effectively limit the complexity of the natural network and improve the generalization of it. Figure 1 shows our network structure.

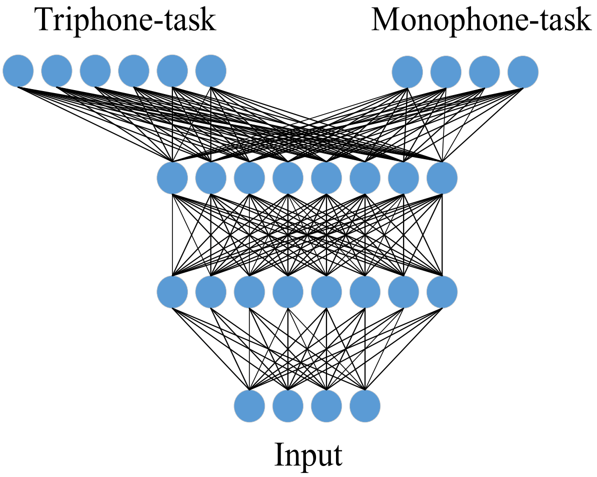

FIGURE 1. The mono-and-triphone learning framework which is similar to multitask learning.

A shared representation between tri-task and mono-task is central to the MAT approach. When computing the gradients, the forward pass can be shared between both tasks, up to the two output layers. We think a single mono-task or a single tri-task is not sufficient. Mono-task is well-defined but not informative enough to guide to the model to a good hidden representation. Tri-task is high-dimensional and can provide more details about the contexts, but a high degree of contexts noise is there especially in a low-resource environment. Therefore, optimizing the two tasks together is a good option. Due to the addition of mono task, compared with the traditional structure, MAT attaches more importance to the correctness of intermediate phonemes. Although the correctness of context is also important, the correctness of intermediate phonemes is what we want. Besides, the addition of a second task also limits the complexity and improves the generalization of the acoustic model.

In order to prove the effectiveness of MAT in preventing neural networks from overfitting, we have conducted several comparative experiments on various network structures, such as DNN, BiRNN, BiGRU, and BiLSTM (hereinafter referred to as RNN, GRU and LSTM). Experiments show that the method achieves better results than baselines on the low-resource Mandarin recognition task. The details and results will be given in Sect. 5.

III. SOFT ONE-HOT LABEL

Here we propose a mechanism based on Gaussian distribution to regularize the classifier layer of the network during training.

In classification tasks, one-hot is the most commonly used label encoding method. One-hot encoding can be defined as a process of converting categorical variables into a distribution that could be provided to ML algorithms to do a better job in prediction. The encoded label method and its distribution shows in Formula. 3.

$$class(\mathrm{k})(one-hot)=\begin{cases}1, & (kislabel)\\ 0, & (kisnotlabel)\end{cases}\ (0\leq k\leq K)$$

$$\begin{aligned}&Distribution(one-hot)\\&\quad=[class(0),\ldots,class(k),\ldots,class(K)]\\&\quad=[0,\ldots,0,1,0,\ldots 0]\end{aligned}\tag{3}$$

where K denotes the number of the classes and k denotes each category of all.

For each training example, our model uses the Softmax layer to compute the probability of each label $k\in\{0\ldots K\}$

$$p(k|x)=\frac{\exp(z_k)}{\sum_{i=0}^{K}\exp(z_i)}\tag{4}$$

here, z_i are the logits or unnormalized log probabilities.

Then we will get a predicted probability distribution. Our optimization goal is to minimize the cross-entropy (CE) loss between the predicted distribution and the ground-truth label distribution.

Model over-confidence is promoted by the CE training criterion. For the baseline network, the training loss is minimized when the model concentrates all of its output distribution on the correct ground-truth category. This leads to very peaked probability distributions, effectively preventing the model from indicating sensible alternatives to a given triphone or monophone. Therefore, one-hot encoding labels often also leads to over-confidence and overfitting of the AM.

In addition, in speech recognition tasks, language models (LM) are often needed to re-score the probability scores derived from acoustic models by fusion as Formula. 5. The language model can linguistically correct the phoneme sequences generated by the acoustic model, and finally get better results.

$$y*=\arg\max_{y}\log p(y|x)+\lambda\log P_{LM}(y)\tag{5}$$

where $P_{LM}(y)$ is provided by the LM, y^* denotes the final score, λ denotes the proportion of language score in the final score, x denotes the training example.

However, the ability to language model rescoring is limited. Sometimes, the model-confidence and overfitting can cause the acoustic model (AM) scores of some wrong sequences too high, which may impact the ability of language model to find good solutions and to recover from errors.

In [40], the authors consider a simple technique of adding time-dependent Gaussian noise to the gradient at every training step. The added Gaussian noise improves the generalization of complicated neural networks because it can prevent the model from falling into the local minima during training. However, adding Gaussian noise to the gradient can not solve the problem of over-confidence.

Inspired by [40], we investigated a new label encoding method named "Soft One-hot Label (SOL)". It is a regularization mechanism to prevent the acoustic model to making over-confident predictions. The goal of our proposal is to reduce the gap between the probability of the correct category and the wrong categories. SOL can prevent peaked probability distributions and improve the generalization of the acoustic models. Besides, since we reduced the gap of correct and wrong categories, this reduces the AM score and enhances the ability to language model rescoring. Because we have very little audio data, the language model plays a very important role in correct the phoneme sequences.

In SOL, we don't use directly 0 and 1 to encode our labels into vectors. We give it more randomness. For the true classification, we still assign it a high probability, but for other classifications, we will not make them 0. Instead, they are assigned a small random variable that obeys the Gaussian distribution. We don't think it's a good idea to have a constant value for each category. This will make the neural network try to fit this invariant distribution and impact the adaptability, so the Gaussian distribution which increases the diversity of label vectors is a good choice. Formula. 6 shows the bottom of the next page, our label encoding method and one example of the label vector. where the hyper-parameter δdenotes the value of a high probability, the parameterμ is the mean or expectation of the distribution (and also its median and mode); and σ is its standard deviation,$x$denotes a random value in a range and is used to calculate a random number that obeys the Gaussian distribution, Kdenotes the number of categories.

In this case, if we use the traditional Gaussian distribution, some generated random numbers *Ran* will be

$$Ran<-\frac{1-\delta}{K-1}\tag{7}$$

and this causes some of the values in the label vector to be negative. As we all know, negative numbers will lead to logarithmic errors, so we add some restrictions to the random number that obeys the Gaussian distribution. To ensure that all values in the SOL vector are positive, we limit the generated Gaussian random numbers *Ran* in a range:

$$-\frac{1-\delta}{K-1} < Ran < \frac{1-\delta}{K-1} \tag{8}$$

In order to implement this limitation, we need to determine whether the random number meets the requirement every time it is generated. If it does not meet the requirement, it will be dropped and regenerated.

The encoding result of SOL for the same label is also different because we add Gaussian perturbation. Without the SOL and MAT, the loss function is:

$$\begin{aligned} loss &= -\sum_{t}^{T} \log(ce(p(t), q(t))) \\ &= -\sum_{t}^{T} \log \sum_{i=1}^{n} (p(t_i) \log q(t_i)) \end{aligned} \tag{9}$$

where $ce()$ denotes the cross-entropy function, $p(t)$ denotes the predicted probability distribution calculated by the neural network for the input x, $q(t)$ denotes represents the Vector 0,1 encoded on the label by one-hot. t represents each frame in the training set. T denotes all the frames. n denotes the dimensions of the $p(t)$ and $q(t)$ distribution.

We can get the gradient of backpropagation by calculating the partial derivative of the loss function. Then, we apply SOL and MAT to the final loss function and make some changes to Formula. 9. Therefore, we firstly define the predicted probability distribution,

$$p(t)^{Tri} = \theta^{Tri}(x_t) \tag{10}$$

$$p(t)^{Mono} = \theta^{Mono}(x_t) \tag{11}$$

where $\theta^{tri}$and θ^{mono} denote the parameters of the networks.

They share all the hidden and input layers of the network but have a different output layer. BiLSTM, for example, they can also be represented as follows,

$$\begin{aligned} &p(t)^{Tri} \\ &\quad = (soft\max{}^{Tri} \circ dropout_4 \circ batchnorm_4 \circ \tanh \circ BiLSTM_4 \\ &\qquad \circ \cdots \circ dropout_1 \circ batchnorm_1 \circ \tanh \circ BiLSTM_1)(x_t) \end{aligned} \tag{12}$$

$$\begin{aligned} &p(t)^{Mono} \\ &\quad = (soft\max{}^{Mono} \circ dropout_4 \circ batchnorm_4 \circ \tanh \circ \\ &\qquad BiLSTM_4 \circ \cdots \circ dropout_1 \circ batchnorm_1 \circ \tanh \circ \\ &\qquad BiLSTM_1)(x_t) \end{aligned} \tag{13}$$

where,

$$BiLSTM = linear \circ concat(LSTM_{forward}, LSTM_{backward}) \tag{14}$$

Then we define the ground-truth label vector,

$$q(t)^{Tri} = SOL(LB_t^{Tri}) \tag{15}$$

$$q(t)^{Mono} = SOL(LB_t^{Mono}) \tag{16}$$

where SOL function denotes the Formula. 6.

Finally, the loss function with SOL and MAT can be got by modifying the Formula. 9:

$$\begin{aligned} loss &= -\sum_{t}^{T} \log(ce(p(t)^{Tri}, q(t)_{SOL}^{Tri})) \\ &\quad -\alpha \sum_{t}^{T} \log(ce(p(t)^{Mono}, q(t)_{SOL}^{Mono})) \\ &= -\sum_{t}^{T} \log \sum_{i=1}^{n} (p(t_i)^{Tri} \log q(t_i)_{SOL}^{Tri}) \\ &\quad -\alpha \sum_{t}^{T} \log \sum_{i=1}^{n} (p(t_i)^{Mono} \log q(t_i)_{SOL}^{Mono}) \end{aligned} \tag{17}$$

We apply SOL and MAT to the final loss function. The Gaussian perturbation in SOL changes the final loss function and makes the networks inclined to fit a more flexible and less peaked probability distribution. This perturbation can make the network not lead to overconfidence during training, so that the network has a better generalization. This method will increase our final loss but improve the performance of the AM.

We apply SOL to the baselines and the multitask learning mentioned in the Sect. 2. Experimental results will be present in Section. 5.

IV. FEATURE COMBINATIONS

In this subsection we explore the effects of different features and combinations of features on the performance of AM.

In ASR, the most commonly used acoustic features are Mel Frequency Cepstral Coefficents (MFCC) and Filter banks (Fbank). Almost all speech recognition tasks choose one of these two. Although they have been shown to achieve good results in speech recognition, these two features do not eliminate the differences between different speakers and affect the performance of the acoustic model. Therefore, we propose to combine the traditional features with FMLLR features as the input of the neural networks.

$$\begin{aligned} class(\mathrm{k})(SOL) &= \begin{cases} 1 - \sum\limits_{i \neq label} class(i)(SOL), & (k \ is \ label) \\ \frac{1-\delta}{K-1} + \frac{1}{\sqrt{2\pi}\sigma} \exp \frac{(x-\mu)}{2\sigma^2}^2, & (k \ is \ notlabel) \end{cases} \quad (0 \leq k \leq K) \\ Distribution(SOL) &= [class(0)(SOL), \ldots class(k)(SOL), \ldots, class(K)(SOL)] \\ &= [0.0015, \ldots, 0.0034, 0.934, 0.0019, \ldots, 0.003] \end{aligned} \tag{6}$$

Feature-space Maximum Likelihood Linear Regression (FMLLR) was explored in [32], [33] for speaker adaptive training and it is a feature space transform where we transform acoustic features for better fit to a speaker-independent (SI) model. We can get FMLLR features vector according to this formula:

$$\bar{o}_t = A_{(n)} o_t + b_{(n)} = W_{(n)} \xi(t) \tag{18}$$

where $W_{(n)} = [A_{(n)}, b_{(n)}]$ stands for the transformation matrix and $\xi(t) = [o_t^T, 1]^T$ represents the extended feature vector.

Before training the SI model, we have an initial matrix $W_{(n)}$, then construct the transformed features iteratively train the new parameters of SI model. After many iterations, we can get a better $W_{(n)}$ for us to perform FMLLR feature transformation.

The combination of different acoustic features can make the speech signal of each frame more detailed and accurate. In particular, the addition of FMLLR features improves the generalization of the model to different speakers.

In Sect. 5, a lot of experiments were conducted to select the combination of features. More details of the them will be shown.

V. EXPERIMENT

A. BASELINE NN-HMM SYSTEM

All experiments are conducted on Pytorch-Kaldi platform [33]. We use a single Nvidia TITAN Xp GPU to do single running. Most of the acoustic models are trained with 40-dimensional high resolution MFCC, 40-dimensional Fbank and 40-dimensional FMLLR feature. Those features were computed with a 25ms window and shifted every 10ms. The raw features are normalized via mean subtraction and variance normalization per speaker side.

Our baseline systems use Natural Networks (DNN, RNN, GRU or LSTM), modelling frame posterior probabilities over triphone units. All of our RNN structures are bidirectional. Unidirectional Recurrent neural networks of that type has the potential disadvantage that it can only take advantage of context information in one direction (usually the past). However, the bi-directional RNN structure makes full use of the context information in both directions, so it is proved to achieve better results. Figure. 2 shows our baseline framework and its components. Dropout [34] is applied in our baselines. Dropout is an effective way to prevent neural networks from over-fitting. The key idea of dropout is to randomly drop units (along with their connections) from the neural network during the training process.

The standard test result in Mandarin speech recognition tasks is Character Error Rate (CER) and Word Error Rate (WER). CER is more convincing than WER in our task. The details of the four different baselines are shown in Table. 2. Splice stands for whether we splice the features of a frame with adjacent frames, so that the model can learn more sequence characteristics. Because of the particularity of RNN structures, SPLICE does not need to be used on them. We set the learning rate as the number of iterations decays to ensure that the network can reach the global minima faster.

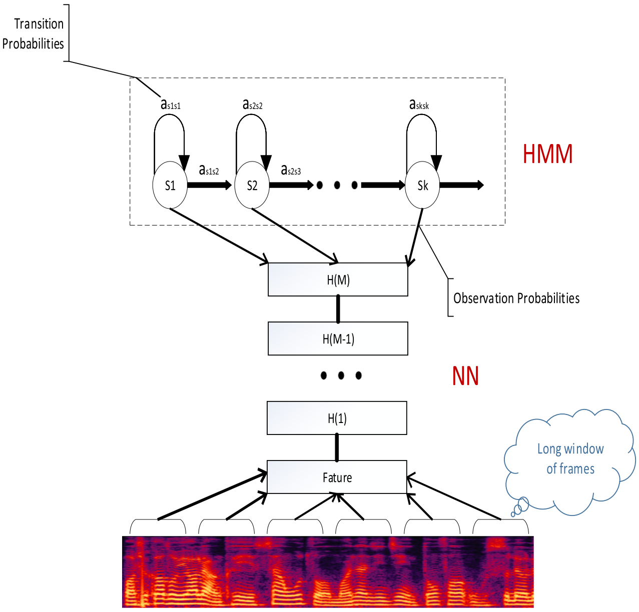

FIGURE 2. The NN-HMM framework and its components.

TABLE 2. The details of our four baselines.

NN structure	DNN	RNN	GRU	LSTM
Layers	5	4	5	4
Units per layer	1024	550	550	550
Activation	Relu	Relu	Tanh	Tanh
Splice	Yes	No	No	No
Epochs	10	10	10	10
BatchNorm	Yes	Yes	Yes	Yes
Dropout rate	0.15	0.2	0.2	0.2
Bi-Direction	------	Yes	Yes	Yes

B. THE-STATE-OF-ART

Since our proposal (MAT + SOL) resembles a regularization method, our experiment compares the results with L2 regularization, which is the-state-of-art method. L2 regularization [35] is a technique to discourage the complexity of the model. It does this by penalizing the loss function and the regularization term is the sum of the square of all feature weights like Formula. 19.

$$loss\_function = loss + \phi * \sum \|w\|^2 \tag{19}$$

where ϕ denotes regularization parameter.

This helps to prevent the overfitting problem by forcing the weights to be small but does not make them zero and does non-sparse solution.

To verify the effectiveness of our framework, we also compared it with the-state-of-art framework in a low-resource environment, TDNN-HMM based on Kaldi platform [39]. TDNN-HMM has been proved to work much better than other models when there is very little data. It uses a method of

TABLE 3. Comparison results of the experiments with mono-and-triphone learning and baselines. "Mono 0.9" means that the weight of mono loss is 0.9. it turn out to be baselines when the weight is 0.0.

	CER(%)			
	DNN-HMM	RNN-HMM	GRU-HMM	LSTM-HMM
Baseline (mono 0.0)	29.26	28.08	27.23	27.16
MAT (mono 0.5)	28.81	27.49	26.55	26.89
MAT (mono 0.9)	**28.50**	**27.10**	26.23	**26.56**
MAT (mono 1.0)	29.10	27.32	**26.20**	**26.56**

sequence-discriminative training and the objective function we used in the training is LF-MMI (Lattice-Free Maximum Mutual Information) [33], [34], which aims to maximize the probability of the target sequence, while minimizing the probability of all other sequences:

$$\begin{aligned} F_{MMI} &= \sum_{u=1}^{U} \log \frac{p(o^u|w^u;\theta)^k p(w)}{p(o^u)} \\ &= \sum_{u=1}^{U} \log \frac{p(o^u|w^u;\theta)^k p(w)}{\sum_{w} p(o^u|w'^u:\theta)^k p(w')} \end{aligned} \tag{20}$$

where o^u and w^u denote the observed sequences and the correct sequence labels. $p(w)$ represents the prior probability of word sequence w and $p(w')$ represents a feasible sequence in the search space. θ represents the hyper-parameters of the model.

C. DATASET

Our experiments are conducted on a ~10 hours training set consisting of 3000 Mandarin utterances. The training set is a subset of THCHS-30 [36], the dev set and test set are the same as those of THCHS-30. THCHS-30 involves more than 30 hours of speech signals recorded by a single carbon microphone at the condition of silent office. Most of the participants are young colleague students, and all are fluent in standard Mandarin. The sampling rate of the recording is 16, 000 Hz, and the sample size is 16 bits.

The language model used in our experiments involves 48k words and is based on word 3-grams. The LM was trained using a text collection that was randomly selected from the corpus and Aishell-2 [37] corpus. The training text involves 772, 000 sentences, amounting to 18 million words and 115 million Chinese characters. The LM was trained with the SRILM tool [38].

D. RESULTS

In order to verify the effectiveness of our proposed methods, we have done a few of comparative experiments. We divided the experiments into three groups, each corresponding to a method to verify the effectiveness of a single method. Finally, we combined the three methods to calculate the best results we achieved. Doing so not only guarantees that all three methods can achieve positive results, but also proves which method has the greatest benefit on our baselines.

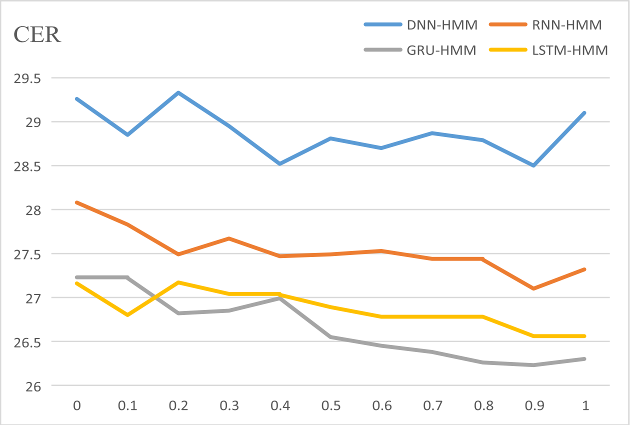

FIGURE 3. The CER curves for the test set with different value of mono-weight in the range [0.0, 1.0] on four baselines.

1) MAT

In this subsection, we verified the effectiveness of Mono-And-Triphone learning (MAT). We performed comparative implementation on all four tasks. The experimental results are shown in Table. 3.

The CER results in the table strongly prove the effectiveness of MAT on four different structures.

Performance of acoustic models with different values of mono-weight (that isα, in Formula. 2) is present in Figure 3. It shows clearly for CER curves on the test set when the value of mono-weight is increased from 0.0 to 1.0. If mono-weight is equal to 0.0, then it turns out to be the baselines. It can be seen that most experiments have improvement compared to baselines, which proves the effectiveness of MAT training. The best acoustic model is obtained when 0.9 is provided, with 2.6% (DNN), 3.5% (RNN), 3.7% (GRU) and 2.2% (LSTM) relatively CER reduction over baselines. It's easy to understand that when the value of mono-weight is too large, it performs worse than the baseline, which is due to the dominance of mono-loss in training.

2) SOL

This set of experiments look at comparing the performance of the Soft One-hot label (SOL) and One-hot label (OL) on

TABLE 4. Comparison results of the experiments with soft one-hot label and the without. There are two groups of comparison in the table, one is on the basis of baseline, and the other is on the structure of mat applied.

	CER (%)			
	DNN-HMM	RNN-HMM	GRU-HMM	LSTM-HMM
Baseline+L2	28.83	27.60	26.86	26.80
Baseline	29.26	28.08	27.23	27.16
Baseline+SOL	**28.73**	**27.56**	**26.98**	**26.88**
MAT	28.50	27.10	26.23	26.56
MAT+SOL(Tri)	28.32	26.98	26.15	26.32
MAT+SOL(Mono)	28.44	27.05	26.23	26.48
MAT+SOL(Tri+Mono)	**28.12**	**26.87**	**25.90**	**26.10**

TABLE 5. Comparison results of the three experiments with different features and for experiments with different combinations of features.

	CER (%)			
	DNN-HMM	RNN-HMM	GRU-HMM	LSTM-HMM
MFCC	29.26	28.08	27.23	27.16
Fbank	29.76	32.99	28.88	28.13
FMLLR	27.52	26.35	26.45	26.67
MFCC+Fbank	28.77	28.57	27.38	27.35
MFCC+FMLLR	27.49	**25.85**	25.88	26.23
FBank+FMLLR	27.19	26.35	25.99	**25.87**
ALL	**27.02**	26.00	**25.73**	25.93

the test set. In experiments with SOL, we set the value of δ in Formula. 6 to 0.95, because it is necessary to ensure that a high probability is assigned to the ground-truth label.

We apply SOL to triphone learning task and MAT. On MAT, We have two types of labels "mono-label" and "tri-label", so we can apply SOL to a single task or to all tasks. SOL(Tri) denotes that we only apply SOL method on the tri-labels. The experimental results are shown in Table. 4.

The experimental results show that SOL can effectively alleviate over-fitting and improve the performance of the model, whether on MAT tasks or not. And it is better to use SOL on both mono-task and tri-task. On the Triphone learning tasks, SOL achieved a relative 1.9% reduction in CER on DNN-HMM, 1.8% on RNN-HMM, 0.9% on GRU-HMM and 1.1% on LSTM-HMM. On the MAT tasks, SOL(Tri+Mono) achieved 1.3% reduction on DNN-HMM, % 0.8 on RNN-HMM, 1.3% on GRU-HMM and 1.7% on LSTM-HMM. Not only that, but our results go beyond the L2 regularization method.

3) FEATURE CHOOSE

At last, we conduct experiments to compare the performance of feature combinations and gather all experimental results. We choose two or three of MFCC, FBANK, FMLLR to combine and train the acoustic model, then choose the best performing one as our final feature combination. As shown in Table. 5, there are seven experiments, including three initial features and four combined features. These experiments are based on baselines, and neither MAT nor SOL is applied to them.

It can be seen in the table that the combination of features brings great benefits. In terms of a single feature, FMLLR achieves the best experimental results due to speaker adaptation. Fbank works least, especially on RNN and its variant structure. When the three features are combined, the model gets the best effect because the multiple features represent a frame of speech signal better. When we use all three features to train the AM, 7.6% (DNN), 7.4% (RNN), 5.5% (GRU) and 4.5% (LSTM) relative CER reduction over baselines are obtained.

It can be seen that different features give different representations of the same frame of speech signals. Although this slightly increased the complexity of the model, great gains were made. Therefore, we choose the combination of "MFCC + FBANK + FMLLR" as our model input features.

4) ALL THE METHODS

Finally, we applied all the three methods mentioned in this paper to our acoustic model modeling. We conducted experiments on all four baselines, and the experimental results are shown in table 6. It can be seen from the table that the three methods all can improve the hybrid hidden Markov

TABLE 6. **Comparison of the final results on four baselines. The value of mono-weight is 0.9 and the input features are MFCC + Fbank + FMLLR. The hyper-parameter of SOL is 0.95.**

Method	CER (%)
TDNN-HMM (Kaldi) [36]	28.07
Baseline (DNN-HMM)+L2	28.83
Baseline (DNN-HMM)	29.26
+MAT	28.50
+Feature Combination	27.17
+SOL	**26.98**
Baseline (RNN-HMM)+L2	27.60
Baseline (RNN-HMM)	28.08
+MAT	27.10
+Feature Combination	25.16
+SOL	**25.00**
Baseline (GRU-HMM)+L2	26.86
Baseline (GRU-HMM)	27.23
+MAT	26.23
+Feature Combination	25.17
+SOL	**24.90**
Baseline (LSTM-HMM)+L2	27.16
Baseline (LSTM-HMM)	27.16
+MAT	26.56
+Feature Combination	25.35
+SOL	**25.12**

model - neural network approach on phoneme level. Compared to baseline, our proposals have achieved relative CER reductions of 7.8% (DNN), 11.0% (RNN), 8.6% (GRU) and 7.5% (LSTM) respectively. Compared with the-state-of-art L2 regularization method, our proposal MAT+SOL has also achieved some improvements. Compared to the result of the TDNN-HMM framework, all the four frameworks exceed it. The GRU-HMM framework achieves the-state-of-art result and relative CER reductions of 11.3% compared to TDNN.

VI. CONCLUSION

In this paper, we investigated the effects of three methods on the performance of acoustic modeling in low-resource environments. We conducted separate comparison experiments on each method on the Mandarin speech recognition task, and finally combined the three methods together. Experimental results show that all three methods can effectively improve the recognition accuracy. MAT+SOL is a new regularization method that can improve overfitting. It works better than L2 regularization especially in a low-resource environments, and our experiments prove that. SOL is a new label encoding method with Gaussian perturbation, which can prevent over-confidence of the model. Feature combination provides a new feature selection scheme for acoustic modeling. We believe they can also be applied to end-to-end models.

We only conducted experiments on the NN-HMM structure and not on the end-to-end model because the end-to-end model performed too poorly in low-resource environments. In future research, we will continue to explore how to better limit the complexity of network models.

ACKNOWLEDGMENT

The authors would like to thank Bo Wang from NARI Group Corporation (State Grid Electric Power Research Institute). The author Xiusong Sun would like to thank Die Hu for her support of English grammar and spelling.

REFERENCES

[1] G. Hinton, L. Deng, D. Yu, G. Dahl, A. R. Mohamed, N. Jaitly, and T. Sainath, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[2] A. Graves, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1764–1772.

[3] 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.

[4] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 4945–4949.

[5] 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.

[6] T. Hori, S. Watanabe, Y. Zhang, and W. Chan, "Advances in joint CTC-attention based End-to-End speech recognition with a deep CNN encoder and RNN-LM," 2017, *arXiv:1706.02737*. [Online]. Available: http://arxiv.org/abs/1706.02737

[7] Y. LeCun and Y. Bengio, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, May 2015.

[8] S. Thomas, M. L. Seltzer, K. Church, and H. Hermansky, "Deep neural network features and semi-supervised training for low resource speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 6704–6708.

[9] S. Jialin Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[10] A. Quattoni, M. Collins, and T. Darrell, "Transfer learning for image classification with sparse prototype representations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[11] D. Wang and T. F. Zheng, "Transfer learning for speech and language processing," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA)*, Dec. 2015, pp. 1225–1237.

[12] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1422–1430.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*. [Online]. Available: http://arxiv.org/abs/1810.04805

[14] S. Edunov, A. Baevski, and M. Auli, "Pre-trained language model representations for language generation," 2019, *arXiv:1903.09722*. [Online]. Available: http://arxiv.org/abs/1903.09722

[15] A. Radford, K. Narasimhan, and T. Salimans, "Improving language understanding with unsupervised learning," OpenAI, San Francisco, CA, USA, Tech. Rep., 2018.

[16] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "Wav2vec: Unsupervised pre-training for speech recognition," 2019, *arXiv:1904.05862*. [Online]. Available: http://arxiv.org/abs/1904.05862

[17] Z. Lian, Y. Li, J. Tao, and J. Huang, "Improving speech emotion recognition via transformer-based predictive coding through transfer learning," 2018, *arXiv:1811.07691*. [Online]. Available: http://arxiv.org/abs/1811.07691

[18] M. A. Tanner and W. H. Wong, "The calculation of posterior distributions by data augmentation," *J. Amer. Stat. Assoc.*, vol. 82, no. 398, pp. 528–540, Jun. 1987.

[19] D. A. Van Dyk and X. L. Meng, "The art of data augmentation," *J. Comput. Graph. Statist.*, vol. 10, no. 1, pp. 1–50, 2001.

[20] M. A. Tanner, *Tools for Statistical Inference: Observed Data and Data Augmentation Methods*, Vol. 67. New York, NY, USA: Springer, 2012.

[21] N. Kanda, R. Takeda, and Y. Obuchi, ''Elastic spectral distortion for low resource speech recognition with deep neural networks,'' in *Proc. IEEE Workshop Autom. Speech Recognit. Understand.*, Dec. 2013, pp. 309–314.

[22] N. Jaitly and G. E. Hinton, ''Vocal tract length perturbation (VTLP) improves speech recognition,'' in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2013, pp. 1–8.

[23] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, ''Audio augmentation for speech recognition,'' in *Proc. Interspeech*, 2015, pp. 1–12.

[24] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, ''SpecAugment: A simple data augmentation method for automatic speech recognition,'' 2019, *arXiv:1904.08779*. [Online]. Available: http://arxiv.org/abs/1904.08779

[25] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, ''AutoAugment: Learning augmentation policies from data,'' 2018, *arXiv:1805.09501*. [Online]. Available: http://arxiv.org/abs/1805.09501

[26] R. Prabhavalkar, R. Alvarez, C. Parada, P. Nakkiran, and T. N. Sainath, ''Automatic gain control and multi-style training for robust small-footprint keyword spotting with deep neural networks,'' in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 4704–4708.

[27] A. Raju, S. Panchapagesan, X. Liu, A. Mandal, and N. Strom, ''Data augmentation for robust keyword spotting under playback interference,'' 2018, *arXiv:1808.00563*. [Online]. Available: http://arxiv.org/abs/1808.00563

[28] R. Schwartz, Y. Chow, O. Kimball, S. Roucos, M. Krasner, and J. Makhoul, ''Context-dependent modeling for acoustic-phonetic recognition of continuous speech,'' in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Apr. 1985, pp. 1205–1208.

[29] S. J. Young, J. J. Odell, and P. C. Woodland, ''Tree-based state tying for high accuracy acoustic modelling,'' in *Proc. Workshop Human Lang. Technol. (HLT)*, 1994, pp. 307–312.

[30] R. Caruana, ''Multitask learning,'' *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, 1997.

[31] D. Povey and G. Saon, ''Feature and model space speaker adaptation with full covariance Gaussians,'' in *Proc. 9th Int. Conf. Spoken Lang. Process.*, 2006, pp. 1–4.

[32] A. Sankar and C.-H. Lee, ''A maximum-likelihood approach to stochastic matching for robust speech recognition,'' *IEEE Trans. Speech Audio Process.*, vol. 4, no. 3, pp. 190–202, May 1996.

[33] M. Ravanelli, T. Parcollet, and Y. Bengio, ''The pytorch-kaldi speech recognition toolkit,'' in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 6465–6469.

[34] J. Xiong, K. Zhang, and H. Zhang, ''A vibrating mechanism to prevent neural networks from overfitting,'' in *Proc. 15th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2019, pp. 1929–1958.

[35] B. Bilgic, C. Fan, K. Setsompop, S. F. Cauley, L. L. Wald, and E. Adalsteinsson, ''Fast image reconstruction with L_2 regularization,'' *J. Magn. Reson. Imag.*, vol. 40, no. 1, pp. 181–191, 2014.

[36] D. Wang and X. Zhang, ''THCHS-30: A free chinese speech corpus,'' 2015, *arXiv:1512.01882*. [Online]. Available: http://arxiv.org/abs/1512.01882

[37] J. Du, X. Na, X. Liu, and H. Bu, ''AISHELL-2: Transforming mandarin ASR research into industrial scale,'' 2018, *arXiv:1808.10583*. [Online]. Available: http://arxiv.org/abs/1808.10583

[38] A. Stolcke, ''SRILM-an extensible language modeling toolkit,'' in *Proc. 7th Int. Conf. Spoken Lang. Process.*, 2002, pp. 1–4.

[39] V. Peddinti, D. Povey, and S. Khudanpur, ''A time delay neural network architecture for efficient modeling of long temporal contexts,'' in *Proc. 16th Annu. Conf. Int. Speech Commun. Assoc.*, 2015, pp. 1–18.

[40] A. Neelakantan, L. Vilnis, Q. V. Le, I. Sutskever, L. Kaiser, K. Kurach, and J. Martens, ''Adding gradient noise improves learning for very deep networks,'' 2015, *arXiv:1511.06807*. [Online]. Available: http://arxiv.org/abs/1511.06807

[41] X.-Y. Zhang, C. Li, H. Shi, X. Zhu, P. Li, and J. Dong, ''AdapNet: Adaptability decomposing encoder-decoder network for weakly supervised action recognition and localization,'' *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jan. 23, 2020, doi: 10.1109/TNNLS.2019.2962815.

[42] X.-Y. Zhang, H. Shi, C. Li, K. Zheng, X. Zhu, and L. Duan, ''Learning transferable self-attentive representations for action recognition in untrimmed videos with weak supervision,'' in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 9227–9234.

[43] X.-Y. Zhang, S. Wang, and X. Yun, ''Bidirectional active learning: A two-way exploration into unlabeled and labeled data set,'' *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 12, pp. 3034–3044, Dec. 2015.

[44] X.-Y. Zhang, H. Shi, X. Zhu, and P. Li, ''Active semi-supervised learning based on self-expressive correlation with generative adversarial networks,'' *Neurocomputing*, vol. 345, pp. 103–113, Jun. 2019.

XIUSONG SUN received the B.S. degree in mechanical engineering from the Nanjing Institute of Technology, Nanjing, China, in 2017. He is currently pursuing the M.S. degree with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing.

His current research interests include signal processing, speech recognition, and machine learning.

QUN YANG received the Ph.D. degree from the College of Computer Science and Technology, Nanjing University, Nanjing, China. She is currently a Professor with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing.

Her current research interests include natural language processing, speech recognition, and machine learning.

SHAOHAN LIU received the Ph.D. degree from the College of Computer Science and Technology, Nanjing University, Nanjing, China. He is currently a Lecturer with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing.

His current research interests include natural language processing, speech recognition, and machine learning.

XIN YUAN received the B.S. degree from the Nanjing Institute of Technology, Nanjing, China, in 2017. He is currently pursuing the M.S. degree with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing.

His current research interests include signal processing, speech recognition, and machine learning.