IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 29, 2021, accepted February 24, 2021, date of publication March 8, 2021, date of current version March 16, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3064197

Towards Constructing HMM Structure for Speech Recognition With Deep Neural Fenonic Baseform Growing

LUJUN LI, TOBIAS WATZEL, LUDWIG KÜRZINGER, AND GERHARD RIGOLL, (Fellow, IEEE)
Chair of Human-Machine Communication, Technical University of Munich, 80333 Munich, Germany

Corresponding author: Lujun Li (lujun.li@tum.de)

ABSTRACT For decades, acoustic models in speech recognition systems pivot on Hidden Markov Models (HMMs), e.g., Gaussian Mixture Model-HMM system, Deep Neural Network-HMM system, etc., and achieve remarkable results. However, the popular HMM model is the three-state left-to-right structure, without the superiority certainty. There are multiple studies on the HMM structure's optimization, but none of them addresses this problem leveraging deep learning algorithms. For the first time, this paper proposes a new training method based on Deep Neural Fenonic Baseform Growing to optimize the HMM structure, which is concisely designed and computationally cheap. Moreover, this data-driven method customizes the HMM structure for each phone precisely without external assumptions concerning the number of states or transition patterns. Experimental results on both TIMIT and TEDliumv2 corpora indicate that the proposed HMM structure improves both the monophone system and the triphone system substantially. Besides, its adoption further improves state-of-the-art speech recognition systems with remarkably reduced parameters.

INDEX TERMS Deep neural network, HMM topology, speech recognition, vector quantization.

I. INTRODUCTION

Hidden Markov models (HMMs) [1] are a stochastic process for modeling time-series data. Since speech signals possess natural temporality, Gaussian Mixture Models (GMMs)-HMM are the most classic acoustic model of the Automatic Speech Recognition (ASR) system for decades. With the debut of Deep Neural Networks (DNNs), hybrid systems occupy the predominant status, including DNN-HMM systems [2], [3], Convolution Neural Networks (CNNs)-HMM system [4], and Recurrent Neural Networks (RNNs)-HMM [5], etc. In recent years, another round of revolution in machine learning triggers ASR architectures' diversification into a completely new approach, specifically end-to-end models, where HMM is abandoned [6]–[10]. However, the straightforward and challenging problems derived from end-to-end models are the tremendous growth of the model size, the increasing computational complexity, and the weak robustness to the input variations [11]. This drawback is proved by Lüscher *et al.* in [12], where on the LibriSpeech 960h task [13], the hybrid DNN-HMM system outperforms the attention-based system by 15% relative on the clean and 40% relative on the other test sets in terms of Word Error Rate (WER). Moreover, experiments on a reduced 100h-subset of the LibriSpeech training corpus show a more pronounced margin between the hybrid and attention-based architectures. Another argument is [14], where Wang *et al.* demonstrate that their transformer-based hybrid system outperforms the attention-based system by 16.4% relative. Consequently, statistical approaches remain to be essential and still draw considerable attention [12], [14]–[23].

The commonly-employed HMM structure is the left-to-right structure, with three states which model the beginning, the middle, and the end of a phone. Nevertheless, there is no adamant evidence for its suitability and superiority. Since the structure affects the modeling capability considerably, there are multiple studies on optimizing the HMM structure.

Bakis-type HMMs [24] are word-based models, which are derived from sample utterances of the word. The number of states in the model is equal to the average duration of the word in frames. The frame size in Bakis's system is 10 milliseconds, and the average number of states for a word is about 30. Rabiner and Levinson [25] describe

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

another word-based model in which the number of states is reduced to approximately 5. This model results in a substantial reduction in the number of parameters without much deteriorating the accuracy. This is because neighboring states in the Bakis model tend to be quite similar, and reducing several similar consecutive states into a single state does not degrade the model very much. Biem [26] replace Bayesian Information Criterion with Discriminative Information Criterion, where discriminative power among models is maximized together with the likelihood. It achieves a slightly higher recognition rate at the expense of more complicated models. Geiger *et al.* [27] present a method to determine the number of states in HMMs. They propose a modification to the Bakis method [24] and a technique to improve the topology with few iterations.

However, there are three general disadvantages existing in the previous works [24]–[32]. Firstly, as they are all based on statistical methods, HMM topologies are constructed from limited data. Secondly, the statistical methods adopted in these works are computationally-expensive heuristic algorithms (e.g., the tree search algorithms), and not easy to employ. Thirdly, they balance between the state length and the model complexity poorly, leading to either high model complexity or limited performance improvements.

To address these problems, we propose a novel approach to optimize the HMM structure, leveraging deep learning, specifically, a Deep Neural Network Vector Quantizer (DNNVQ). First, we introduce the concept "fenone" for representing sub-phones [33]. Fenone is the building block of phones and is modeled as one state of the HMM, which can be obtained automatically through a vector quantizer. Next, we classify all data against different phones and then apply vector quantization on each phone's data. Finally, DNNVQ generates the fenonic baseforms for every phone, and accordingly, the HMM structure is decided. We refer to this algorithm as Neural Fenonic Baseform Growing (NFBG).

In summary, the main contributions of this paper lie as the following:

- We propose an innovative algorithm leveraging deep learning to customize an HMM structure for every phone.
- Different from previous studies, the proposed method discovers the potential information of the data and contains the model complexity simultaneously, avoiding excessive growth of the number of states.
- Our approach leads to an automatic data-driven state tying. It not only improves the existing state-of-the-art systems but further shrinks their parameter scale considerably.

II. RELATED WORK

The notion of "fenone" is inspired by [34], [35], where they describe a new technique for constructing HMM for the acoustic representation of words. They create the notion of "fenone" to represent sub-word units, and it is derived automatically from one or more utterances of that word. Then the word model is constructed from fenonic forms. Since the word models are all composed of a small inventory of sub-word models, training for large-vocabulary speech recognition systems can be accomplished with a small training script by this technique. A method for combining phonetic and fenonic models is also presented in [35], and impressive improvements are achieved with speaker-dependent and speaker-independent models on several isolated-word recognition tasks.

The Neural Network Vector Quantizer was first proposed by Rigoll and Neukirchen in [36], which is a shallow neural network and is trained with the mutual information criterion. The index of the neuron in the output layer with the highest activation returns the label for the training sample, and thereby, the network performs the quantization to assign the input feature to a specific cluster. This model outperforms a K-means system and nearly matches the performance of a system with continuous (non-quantized) models in terms of word recognition accuracy rate.

Watzel *et al.* [37] extend the neural network vector quantizer to a deep neural network quantizer and introduce a novel approach, a mapping function, to train it in a supervised fashion with an arbitrary output layer size even though suitable target values are not available. The experiments demonstrate that the deep neural network quantizer reduces the WER by 17.6% on monophones and by 2.2% on triphones, respectively, compared to a continuous GMM-HMM system. Inspired by our success in [37], we introduce it to this work as the vector quantizer.

This paper extends the concept from "word" to "phone", substitutes the sophisticated tree search algorithm in [35] with a concise neural network, and confirms its viability on the task of large vocabulary automatic speech recognition.

III. DEEP NEURAL NETWORK VECTOR QUANTIZER

Let $\boldsymbol{\mathcal{D}} = \{(\boldsymbol{x}_i, \hat{y}_i)\}_{i=1}^{N}$ be a dataset comprising feature vectors $\boldsymbol{x_i} \in \mathbb{R}^D$ and their corresponding ground-truth labels $\hat{y}_i \in \mathbb{N}$. The goal of the training is to find a function $f : \boldsymbol{x}_i \rightarrow \hat{y}_i$. In [37], this goal is converted to approximate $g_\theta : \boldsymbol{x}_i \rightarrow \hat{m}_i$, where θ represents the parameters of the network and $\hat{m}_i \in \mathbb{N}$ defines the index of the maximum value in the DNNVQ output layer $\boldsymbol{m}_i \in \mathbb{R}^{N_{\text{clu}}}$ by

$$\hat{m}_i = \underset{1 \leq j \leq N_{\text{clu}}}{\arg\max}\, m_i^j. \tag{1}$$

N_{clu} is the dimension of the output layer, and j describes the jth neuron in the layer. In contrast, the ground-truth label $\hat{y}_i$ is in the range $[1, N_{\text{K}}] = \{\hat{y}_i \in \mathbb{N} \mid 1 \leq \hat{y}_i \leq N_{\text{K}}\}$, where N_{K} denotes the dimension of the ground-truth label space.

Watzel *et al.* [37] employ maximum mutual information (MMI) as the criterion of the training. The mutual information $I(Y; M)$ is defined as

$$I(Y; M) = H(Y) - H(Y|M). \tag{2}$$

Y denotes a ground-truth label, and M is a firing neuron. $H(Y)$ defines the entropy of Y, and $H(Y|M)$ denotes the entropy of Y conditioned on M. Their probability mass functions are

$$P(M=\hat{m}^{j})=\frac{1}{N}\sum_{i=1}^{N}\delta(\hat{m}_i,j)\quad\forall 1\le j\le N_{\text{clu}}\tag{3}$$

and

$$P(Y=\hat{y}^{k})=\frac{1}{N}\sum_{i=1}^{N}\delta(\hat{y}_i,k)\quad\forall 1\le k\le N_{\text{K}}.\tag{4}$$

The probability mass functions are created by counting occurrence numbers of $\hat{m}^j$ and $\hat{y}_k$ based on all samples $\hat{m}_i$ and $\hat{y}_i$, where the index k denotes the kth label in the ground truth label space and $\delta(\cdot)$ refers to Kronecker delta.

As the entropy $H(Y)$ is a constant, we have to minimize $H(Y|M)$ in order to maximize $I(Y;M)$. For this purpose, increasing the dimension of emitted labels $\hat{\boldsymbol{m}}_i$ could be a straightforward solution. However, it causes a new problem for training, where the dimension of the output layer and that of the ground-truth label space are unequal, i.e., $N_{\text{clu}}\neq N_{\text{K}}$. To tackle this problem, Watzel *et al.* introduce the conditional probability $P_b(Y|M)$ of the ground-truth labels $\hat{y}_i$ conditioned on the DNNVQ outputs $\boldsymbol{m}_i$ as

$$\begin{aligned}P_b(Y|M) &= P(\hat{y}_{b,k}|m_{b,j})\\ &\approx \frac{\varepsilon+\sum_{i=1}^{N_b}\delta(\hat{y}_i,k)m_i^j}{\varepsilon N_{clu}+\sum_{i=1}^{N_b}m_i^j},\\ &\quad\forall 1\le k\le N_K,\quad 1\le j\le N_{clu}\end{aligned}\tag{5}$$

where ε is a small constant and the conditional probability $P_b(Y|M)\in\mathbb{R}^{N_K\times N_{\text{clu}}}$. We take minibatches with a sufficient batch size N_{b} to approximate $P_b(y|m)\approx P(Y|M)$. Then, the output $\boldsymbol{m}_i$ is mapped from dimension N_{clu} to dimension N_{K} with $P_b(Y|M)$ as

$$\boldsymbol{m}_{tra,i}=P_b(Y|M)\boldsymbol{m}_i\quad\forall 1\le i\le N_{\text{b}},\tag{6}$$

with $\boldsymbol{m}_{tra,i}$ denoting the transformed outputs of $\boldsymbol{m}_i$. In this way, the prototype size of the vector quantizer can be arbitrary even though the dimension of the ground-truth labels $\boldsymbol{y}_i$ is determinate. During training, we implicitly maximize the mutual information $I(Y;M)$ by minimizing $\mathcal{L}_{CE}(\boldsymbol{m}_{tra,i};\hat{y}_i)$ [38], where

$$\mathcal{L}_{CE}=-\frac{1}{N_b}\sum_{i=1}^{N_{\text{b}}}\sum_{k=1}^{N_{\text{K}}}\delta(\hat{y}_i,k)\log(m_{tra,i}^{k}).\tag{7}$$

The diagram of DNNVQ training is depicted in Fig. 1.

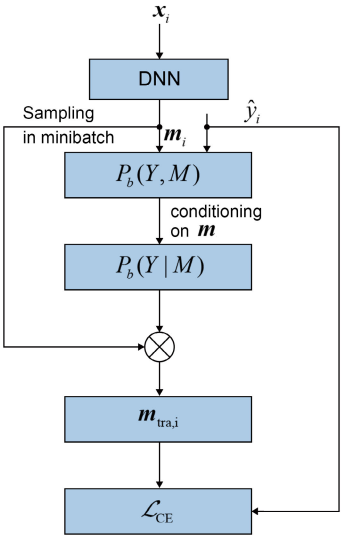


FIGURE 1. Diagram of DNNVQ training.

IV. NEURAL FENONIC BASEFORM GROWING

In this section, the algorithm of the dynamic baseform generation is illustrated and three options of the baseform's HMM topology are presented.

First and foremost, we give the overview of the proposed approach:

Step 1: Train a vanilla GMM-HMM model from flat-start and obtain forced alignments as the ground truth for the sebsequent DNNVQ training.

Step 2: Train a DNNVQ, as described in Section III, using the forced alignments obtained from Step 1 to maximize the mutual information between the ground truth labels and the output units. For each training, the number of the prototype, namely the dimension of the output layer, is specified and fixed.

Step 3: Extract segments of each monophone. Exclude extreme cases in the segment set of each monophone.

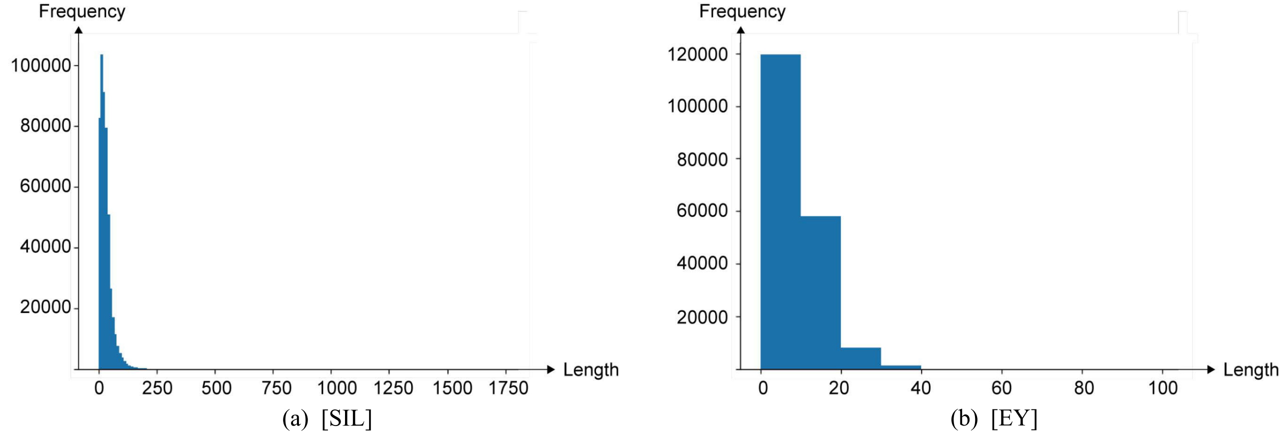
Step 4: Pad the remained segments to the same length.

Step 5: For the same frames of each phone's segments, calculate the products of all posteriors on the same output unit, namely the same prototype. The prototype with the highest product value is the fenone of the current frame. Consequently, the fenone sequence of the phone is acquired.

Step 6: Compact the fenone sequence to the fenonic baseform by eliminating all successive duplicated fenones.

A. SEGMENT LENGTHS PADDING

Intuitively, utilizing all training data must deliver the most accurate result. However, for one thing, computation complexity increases exponentially with the increment of training data; for another, the extreme cases, e.g., the longest ones or the shortest ones, which occur quite rare in the realistic scene mislead the final decision. For this purpose, we plotted histograms for each phone, exhibiting the distribution range of lengths of all segments affiliated to a phone. Fig. 2 demonstrate the histograms of phone [SIL] and [EY]. As shown in Fig. 2, lengths of phone [SIL] differ in the range [0, 1750], while that of phone [EY] is [0, 100]. According to these histograms, we discard the extreme cases for every monophone, and keep at least 80% data for each monophone eventually. For instance, we keep segments ranging in [4, 120] for [SIL] and that for phone [EY] is [4, 30]. We also conducted experiments with 70% or 90% data for each phone, but results indicated no difference, which also proves the robustness of



(a) [SIL]

(b) [EY]

FIGURE 2. Histograms of the segment length distribution of phones (a) [SIL] and (b) [EY].

our proposed approach to noise. Afterwards, all segments affiliated to one phone need to be justified to the identical length for the purpose that all frames representing for the beginning, the middle or the end of the phoneme are aligned together, i.e., we need to justify all lengths of the segments to the maximum one. Instead of zero padding, we use a simple division as the alternative. Assume $l_{max} \div l_n = l_q \cdots l_r$, where l_{max} is the maximum length of all segments, l_n represents the length of any segment in the same set, l_q denotes the quotient, and l_r refers to the remainder. Then we duplicate each frame of the current segment l_q times and the last frame l_r times more. Taking the shortest and the longest segments of phone [EY] as the example, since $30 \div 4 = 7 \cdots 2$, we should duplicate each frame of the 4-frame segment 7 times, while the last frame 2 times more. As the result, the 4-frame segment is padded to be the maximum length.

B. DYNAMIC BASEFORM GENERATION

Fenones represent short speech events and are obtained automatically through the employment of a VQ. Different from [35], where fenones represent sub-word units, we extend its application to sub-phone units, a finer level of details. DNNVQ is deployed for generating the fenone sequence for each phone. Since the fenone sequence of a phone is derived from its utterances, we realign all training utterances against 40 monophones on TEDliumv2 [39] and 48 monophones on TIMIT [40], respectively. In order to distinguish from utterances, the sub-units of utterances are henceforth named as segments. Let $\mathcal{F} = \{f_1, f_2, \cdots, f_{N_F}\}$ $1 \le N_{\mathrm{F}} \le N_{\mathrm{clu}}$ be the alphabet of fenones and $\mathcal{F}^*$ be the set of all finite length strings constructed by concatenating elements of $\mathcal{F}$, namely fenone sequence. $\mathcal{G} = \{g_1, g_2, \cdots, g_s\}$ is the set of s monophones while $\mathcal{K}_i \in \mathcal{K}^* = \{\mathcal{K}_1, \mathcal{K}_2, \cdots, \mathcal{K}_s\}$ denotes the set of all segments affiliated to the corresponding phone g_i $1 \le i \le s$. The goal here is to generate a fenone sequence for phone g_i based on its segments set $\mathcal{K}_i$. The generated fenone sequence $\boldsymbol{f}_{1\rightarrow l_{g_i}} = \{f_1, f_2, \cdots, f_{l_{g_i}}\} \in \mathcal{F}^*$ is spanned up on $f \in \mathcal{F}$, leveraging DNNVQ.

Initially, all segments affiliated to one phone g_i need to be padded to the identical length. For instance, we extract n segments for phone [AW] from all utterances and their lengths vary from l_{min} to l_{max} because of different pronunciation habits or allophones. l_{min} is the minimum length, while l_{max} is the maximum. Consequently, $\mathcal{K}_i = \{\boldsymbol{k}^{(1)}_{1\rightarrow l_1}, \boldsymbol{k}^{(2)}_{1\rightarrow l_2}, \cdots, \boldsymbol{k}^{(n)}_{1\rightarrow l_n}\}$ is converted to $\mathcal{K}'_i = \{\boldsymbol{k}'^{(1)}_{1\rightarrow l_{max}}, \boldsymbol{k}'^{(2)}_{1\rightarrow l_{max}}, \cdots, \boldsymbol{k}'^{(n)}_{1\rightarrow l_{max}}\}$. Afterwards, the tth first frame of $\boldsymbol{k}'^{(t)} \in \mathcal{K}'_i$ $1 \le t \le n$ is fed into the DNNVQ in turn (There are n first frames from n padded segments in total.). As the output, the output vector $\boldsymbol{m}_t$ $1 \le t \le n$ is obtained in turn

$$\boldsymbol{m}_t = [\,p(m_1^1|k_1^{\prime(1)})\; p(m_1^2|k_1^{\prime(1)})\;\cdots\; p(m_1^{N_{\mathrm{clu}}}|k_1^{\prime(1)})\,]. \tag{8}$$

After we get all the outputs of the first frames of $\boldsymbol{k}'^{(t)} \in \mathcal{K}'_i$ $1 \le t \le n$, $\mathcal{A}^{n\times N_{\mathrm{clu}}}$ is acquired as

$$\begin{aligned}\mathcal{A} &= [\boldsymbol{m}_1 \quad \boldsymbol{m}_2 \quad \cdots \quad \boldsymbol{m}_n]^T \\ &= \begin{bmatrix} p(m_1^1|k_1^{\prime(1)}) & p(m_1^2|k_1^{\prime(1)}) & \cdots & p(m_1^{N_{\mathrm{clu}}}|k_1^{\prime(1)}) \\ p(m_2^1|k_1^{\prime(2)}) & p(m_2^2|k_1^{\prime(2)}) & \cdots & p(m_2^{N_{\mathrm{clu}}}|k_1^{\prime(2)}) \\ \vdots & \vdots & \ddots & \vdots \\ p(m_n^1|k_1^{\prime(n)}) & p(m_n^2|k_1^{\prime(n)}) & \cdots & p(m_n^{N_{\mathrm{clu}}}|k_1^{\prime(n)}) \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{q}_1 & \boldsymbol{q}_2 & \cdots & \boldsymbol{q}_{N\mathrm{clu}} \end{bmatrix},\end{aligned} \tag{9}$$

where

$$\boldsymbol{q}_{\mathrm{j}} = \begin{bmatrix} q_{1\mathrm{j}} & q_{2\mathrm{j}} & \cdots & q_{n\mathrm{j}} \end{bmatrix}^T, \ 1 \le j \le N_{clu}. \tag{10}$$

$\boldsymbol{m}_t$ $1 \le t \le n$ is the row vector of matrix $\mathcal{A}^{n\times N_{\mathrm{clu}}}$ and it denotes the output vector of the first frame of the tth segment. $\boldsymbol{q}_j$ is the column vector of matrix $\mathcal{A}^{n\times N_{\mathrm{clu}}}$, and it represents the posterior probabilities of all first frams on fenone j. By the element-wise product of $\boldsymbol{q}_j$, we get the product of the jth column $P_{\boldsymbol{q}_j}$ as

$$P_{\boldsymbol{q}_j} = \prod_{t=1}^{n} p(m_t^{(j)}|k_1^{\prime(t)}) \quad \forall\, 1 \le j \le N_{\mathrm{clu}}. \tag{11}$$

Let $\hat{j}$ be the index of the maximum value of $P_{q_{\mathrm{j}}}$, i.e.,

$$\hat{j} = \mathop{\arg\max}_{1 \leq j \leq N_{\mathrm{clu}}} P_{\boldsymbol{q}_j}, \tag{12}$$

then the first frame of $k'^{(t)}$ $1 \leq t \leq n$ is quantized to the $\hat{j}$th neuron and the corresponding fenone is $f_{\hat{j}}$. Due to the risk of underflow, the logarithm is employed, then

$$\hat{j} = \mathop{\arg\max}_{1 \leq j \leq N_{\mathrm{clu}}} \sum_{t=1}^{n} \log(p(m_t^j | k_1'^{(t)})). \tag{13}$$

Algorithm 1: Pseudo-Code for Dynamic Baseform Growing

```
Input: padded segments
    $\mathcal{K}'_i = \{\boldsymbol{k}'^{(1)}_{1\to l_{max}}, \boldsymbol{k}'^{(2)}_{1\to l_{max}}, \cdots, \boldsymbol{k}'^{(n)}_{1\to l_{max}}\}$ of phone
    $g_i$ $1 \leq i \leq s$;
Output: the corresponding fenone of frame $l$;
Training DNNVQ with $\mathcal{D} = \{(\boldsymbol{x}_i, \hat{y}_i)\}_{i=1}^{N}$;
while $\underline{1 \leq i \leq l_{max}}$ do
    for $\underline{t = 1; t \leq n; t++}$ do
        the output $\boldsymbol{m}_t$:
        $\boldsymbol{m}_t = [p_t^1 \ p_t^2 \ \cdots \ p_t^{N_{\mathrm{clu}}}]$
    end
    the score of the $t$th frame on $N_{\mathrm{clu}}$ clusters:
    $P_{\boldsymbol{q}_j} = \prod_{t=1}^{n} p_t^j \quad 1 \leq j \leq N_{\mathrm{clu}}$;
    if $\underline{\hat{j} = \arg\max P_{\boldsymbol{q}_j}}_{1 \leq j \leq N_{clu}}$ then
        the $t$th frame of $g_i \overset{fenone}{\longleftarrow} f_{\hat{j}}$
    end
    $i = i + 1$;
end
the fenone sequence of $g_i$:
$\{f_{\hat{j}1}, f_{\hat{j}2}, \cdots, f_{\hat{j}l_{max}}\}$;
merge the successive duplicated fenones;
the fenonic baseform of $g_i$:
$\{f_{\hat{j}1'}, f_{\hat{j}2'}, \cdots, f_{\hat{j}l'}\}$ $1 \leq \hat{jl}' \leq N_{\mathrm{clu}}$;
return fenonic baseform
```

Similarly, we repeat the same steps for the remaining $l_{max} - 1$ frames chronologically, and the whole fenone sequence $\{f_{\hat{j}1} f_{\hat{j}2} \ldots f_{\hat{j}l_{max}}\}$ for phone g_i $1 \leq i \leq s$ is acquired. Importantly, the fenone sequence $\{f_{\hat{j}1} \ f_{\hat{j}2} \ \ldots \ f_{\hat{j}l_{max}}\}$ could contain several identical fenones. Subsequently, we eliminate all successive duplicated fenones in the obtained fenone sequence to generate the final fenonic baseform for a phone. Hence the fenonic baseform is refined from the corresponding fenone suquence, without any duplicated fenone. The whole procedure of NFBG is illustrated in Alg. 1. Taking phone [AW] as an example, the length of padded segments of the phone [AW] is 20 and the generated fenone sequence is {27 27 27 27 27 27 27 27 27 27 27 27 92 92 92 92 92 5 5 5}. Thereby, we merge these adjacent identical fenones, and in consequence, the fenonic baseform of phone [AW] appears to be {27 92 5}. We demonstrate the NFBG process of phone [AW] in Fig. 3.

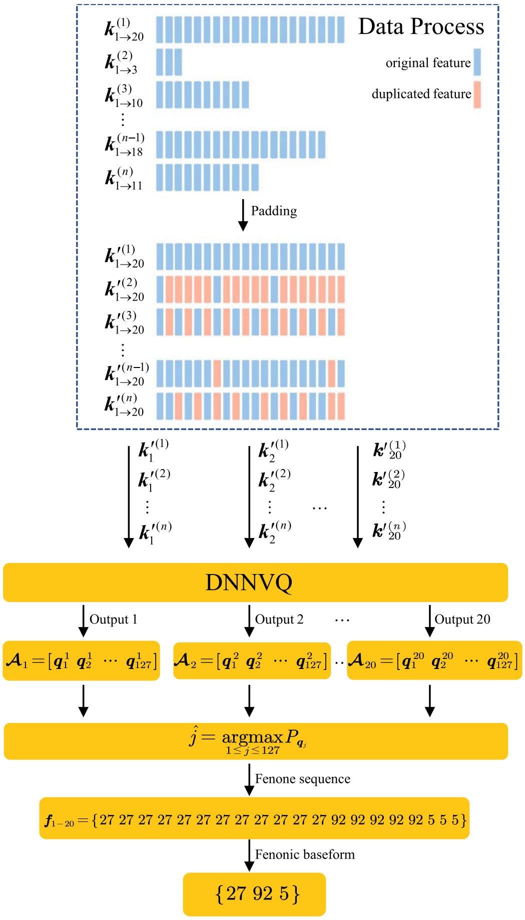

Data Process
original feature
duplicated feature
$k_{1\to20}^{(1)}$
$k_{1\to3}^{(2)}$
$k_{1\to10}^{(3)}$
$k_{1\to18}^{(n-1)}$
$k_{1\to11}^{(n)}$
Padding
$k'^{(1)}_{1\to20}$
$k'^{(2)}_{1\to20}$
$k'^{(3)}_{1\to20}$
$k'^{(n-1)}_{1\to20}$
$k'^{(n)}_{1\to20}$
$k_1'^{(1)}$ $k_1'^{(2)}$ $k_1'^{(n)}$
$k_2'^{(1)}$ $k_2'^{(2)}$ $k_2'^{(n)}$
$k'^{(1)}_{20}$ $k'^{(2)}_{20}$ $k'^{(n)}_{20}$
DNNVQ
Output 1
Output 2
Output 20
$\mathcal{A}_1 = [q_1^1 \ q_2^1 \ \cdots \ q_{127}^1]$
$\mathcal{A}_2 = [q_1^2 \ q_2^2 \ \cdots \ q_{127}^2]$
$\mathcal{A}_{20} = [q_1^{20} \ q_2^{20} \ \cdots \ q_{127}^{20}]$
$\hat{j} = \mathop{\mathrm{argmax}}_{1 \leq j \leq 127} P_{q_j}$
Fenone sequence
$f_{1-20} = \{27\ 27\ 27\ 27\ 27\ 27\ 27\ 27\ 27\ 27\ 27\ 27\ 92\ 92\ 92\ 92\ 92\ 5\ 5\ 5\}$
Fenonic baseform
{27 92 5}

FIGURE 3. Illustration of the NFBG process of phone [AW]. The upper half is the padding process, where the segment lengths of phone [AW] vary from 3 to 20, and the pink frames are the duplicates of the blue original frame before. The lower half is the process of NFBG, which starts from that the 1st - 20th frames are fed into DNNVQ in turn and ends up with compacting the 20-frame fenone sequence to the fenonic baseform.

C. ELEMENTARY MARKOV MODEL FOR FENONES

The HMM of a phone is constructed by concatenating the elementary Markov model of the fenones in its fenonic baseform. The fenonic baseform merely indicates the number of states of an HMM, but the topology remains undetermined. We investigate three sorts of topology for the elementary Markov model: ergodic, Bakis-type [24], Vintsyuk-type [41], as depicted (a), (b), and (c) in Fig. 4. Each state in the ergodic topology can transit to every other state in a single step. Thus the ergodic topology possesses the highest flexibility as well as the highest complexity. By contrast, every state in the Bakis-type [24] topology can only transit to itself or the next one, but the Bakis-type topology dominates the advantage of

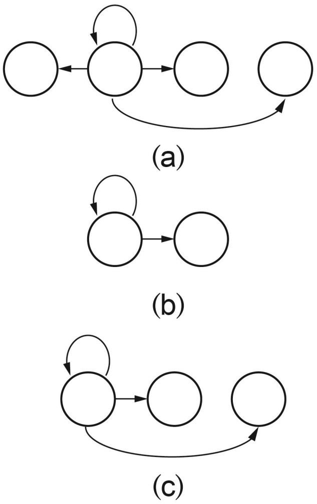


FIGURE 4. Examples of Markov models for fenones. (a) Vintsyuk-type [41], (b) Bakis-type [24], (c) ergodic.

simplicity. The Vintsyuk-type [41] topology is a compromise of the former two, as it allows a maximum shortening by a factor of two. It contains model parameters and preserves the flexibility by a skip arc simultaneously. We execute ablation experiments on the efficacy of each topology.

V. EXPERIMENTAL SETUPS

A. CORPORA AND FEATURES

We test our approach on TIMIT [40] and TEDliumv2 [39].

TIMIT contains a total of 6300 sentences (5.4 hours), consisting of 10 sentences spoken by each of 630 speakers from 8 major dialect regions of the United States. The 462-speaker training set is used. All SA records (i.e., identical sentences for all speakers in the database) are removed as they could bias the results. After realigning the training utterances against 48 monophones, we obtain 220535 segments as the training dataset. Results are reported using the 24-speaker core test set. All of our experiments apply a bigram language model over phones, estimated from the training set.

TEDliumv2 contains 207-hour training data from TED talks, consisting of male and female speakers, native- and nonnative-speakers, and speakers from all age ranges. The contents of the data cover versatile fields. After the realignment of training utterances against 40 monophones, we obtain 8439059 segments as the training dataset. All recognition results are reported on the heavily pruned 4-gram language model and the dictionary with roughly 152k words and 160k pronunciations released by [39].

As for features, we utilize 12-dim Mel frequency cepstral coefficients (MFCC) along with the additional energy feature and their first and second temporal derivatives, hence 39-dim MFCC feature vectors in total. Cepstral mean normalization is employed.

B. DNNVQ SETUPS

The DNNVQ system is trained in the Tensorflow library. The network is composed of four fully-connected hidden layers with 512 neurons and the ReLu activation function followed by a batch-normalization [42] layer, respectively. The dropout [43] layer is discarded due to worse results. A subsequent fully-connected layer with the ReLu activation function is deployed as the output layer. We optimize the DNNVQ with Adam optimizer [44]. An exponentially decaying schedule starts with an initial learning rate of 0.01 and halves the rate when the improvement of the frame accuracy between two successive epochs on a cross-validation set stops.

C. BASELINE

The training of the GMM-HMM and the DNN-HMM baseline systems is pursuant to the Kaldi example recipe [45]. They are trained on the MFCC feature described in Section V-A. The HMM structure adopted on TEDliumv2 is 3-state left-to-right structure for vocal phones while 5-state structure for ''silence'' and ''noise'', leading to 127 Probability Density Functions (PDFs). In contrast, the HMM structure of all 48 monophones is the identical 3-state left-to-right structure on TIMIT, resulting in 144 PDFs. GMM-HMM system is trained from scratch, and 1K Gaussian models are deployed in total; in the DNN-HMM system, the DNN has four hidden layers, each of which has 512 neurons. The number of nodes of the final layer is determined by the number of PDFs. The DNN is initialized randomly with weights drawn from $\mathcal{N}(0, 0.01)$, and the uniform bias drawn randomly from $\mu(-4.1, -3.9)$. Stochastic gradient descent is utilized to minimize the cross-entropy, with the minibatch size of 512 frames. The learning rate is set at 0.0015 initially and decays to 0.00015 progressively. All baseline systems are conducted in the Kaldi toolkit [45].

Primary structures are chosen for baseline systems to force focus on the impact of HMM structures. Besides, the effectiveness of the proposed method in advanced systems will also be given later.

VI. FENONIC BASEFORM RESULTS

In this section, we give the obtained fenonic baseforms for phones included in Tedliumv2 (Table 1) and TIMIT (Table 2) corpora when there are 127 prototypes in the case of Tedliumv2 while 144 prototypes in the case of TIMIT, respectively. There are 40 phones in Tedliumv2 corpus and 48 phones included in TIMIT corpus. In TEDliumv2 corpus, 1 phone gains five states, 6 phones gain four states, 7 phones gain two states, and the remained phones gain three states. In contrast, 8 phones gain four states, 17 phones gain two states, and the rest gain three states, in TIMIT corpus. Consequently, there are 94/127 active fenones in Tedliumve, while 104/144 active fenones in TIMIT. Hence the average numbers of states are 3.025 in Tedliumv2, while 2.8125 in TIMIT. Then it is safe to draw the conclusion that even if the average numbers of states do not change much, the state sharing relations among phones reduce parameters.

After analysing the fenonic baseform results, we can draw two conclusions. First, the shared states tend to appear in the

TABLE 1. Fenonic baseforms for every monophone in Tedliumv2 corpus.

Phone	Fenonic baseform	Phone	Fenonic baseform	Phone	Fenonic baseform	Phone	Fenonic baseform
SIL	17 124 13 17	AW	27 92 5	DH	43 79 28	G	97 52
AA	57 102 24	AY	20 47	EH	111 12 105	HH	86 49
AE	27 82 23	B	125 45 79	ER	84 80 48	IH	4 18 44
AH	6 107 2	CH	38 94 1 40 28	EY	3 64 39 81	IY	16 67 32
AO	87 93 24	D	51 97 42	F	88 0 21	JH	94 40
Phone	**Fenonic baseform**	**Phone**	**Fenonic baseform**	**Phone**	**Fenonic baseform**	**Phone**	**Fenonic baseform**
K	69 108	OW	114 112 46	SH	29 1 40	V	25 68 36
L	106 7 34	OY	87 93 46 81	T	38 121 53	W	122 26 33
M	72 90 104	P	65 115 49	TH	76 0 21 40	Y	101 75
N	77 31 8	R	91 80 63	UH	6 107 2 21	Z	83 60 55
NG	10 56 89	S	73 59 70 11	UW	99 71	ZH	29 1 40

TABLE 2. Fenonic baseforms for every monophone in TIMIT corpus.

Phone	Fenonic baseform	Phone	Fenonic baseform	Phone	Fenonic baseform	Phone	Fenonic baseform
AA	108 26 44 86	AY	100 10 112	DX	36 67	EY	42 50 126 66
AE	136 23 77	B	62 3	EH	109 90 101 32	F	89 11 87 122
AH	142 113 80	CH	37 127 133 25	EL	2 124	G	56 95
AO	103 93 69	CL	28 107	EN	17 96 88	HH	76 46
AW	136 6 116	D	105 56	EPI	79 60 12	ICH	92 35 106
AX	141 58 1	DH	128 18 51	ER	50 5 49	IX	135 57 73
Phone	**Fenonic baseform**	**Phone**	**Fenonic baseform**	**Phone**	**Fenonic baseform**	**Phone**	**Fenonic baseform**
IY	114 24 65	NG	102 83 95	SH	16 133 25	V	98 72
JH	94 25 143	OW	137 97 20	SIL	13 104 110 13	VCL	68 117 70
K	138 46	OY	103 22 112	T	8 123	W	38 3
L	4 140	P	132 60 9	TH	34 53 87 55	Y	19 48
M	115 118	R	7 5	UH	142 113 80 139	Z	29 75
N	82 88	S	33 91 71	UW	120 39 74	ZH	16 133 25

same or similar location of phones. For instance, in TEDliumv2, state /40/ is shared by phones [CH], [JH], [SH], [TH], and [ZH]. It appears as the last state in cases of [JH], [SH], [TH], and [ZH], while as the penultimate state in the case of [CH]. The same phenomenon also appears in TIMIT, where [CH], [JH], [SH], and [ZH] share the last state /25/, and [CH], [SH], and [ZH] even share the last two states, /133/ and /25/. This pattern reveals that a specific state always tends to express a specific part of the phone (the beginning, the middle, or the end). Second, phones with the same suffix tend to share the state in the ending, while phones with the same prefix tend to share the state at the beginning or in the middle. For instance, in TEDliumv2, [EY] and [OY] share the last state /81/, while in TIMIT, [AE] and [AW] share the first state /136/. Nevertheless, the latter half of the pattern is not as common as the former half, and thus we believe that the suffix is more decisive than the prefix for the pronunciation of a specific phone.

Besides, there are some interesting phenomenons in the resultant fenonic baseforms which we highlight here for any possible inspiration to our readers. Firstly, some phones seemingly irrelative share states. For example, [TH] and [F] share two states (/0/ and /21/) in TEDliumv2, and they share one state (/87/) in TIMIT; [B] and [DH] share the state /79/ in Tedliumv2; and [K] and [HH] share the state of /46/ in TIMIT. Secondly, some phones possess a high similarity between them in terms of the state. For instance, it surprised us that the states of [SH] are identical to those of [ZH]. This phenomenon appears in both corpora, so we believe it is not a coincidence. A similar pattern falls on the case of [AH] and [UH], which share three states in both corpora. For these counter-intuitive sharing relationships, we believe it reveals a sort of interior relevance between those phones.

Additionally, there are two phenomenons emerging in the process of compacting the fenonic sequence to the fenonic baseform which are worth noting. For one thing, [SIL] processes the repetitive fenone in its fenonic baseform. For example, the fenonic baseform of [SIL] in TEDliumv2 is derived from its original fenone sequence {17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 124, 124, 124, 124, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 13, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17}. We tried to simplify its fenonic baseform as {17, 124, 13} and added an skipping-back arc from state 13 to state 17. However, the result turned to be slightly worse than keeping it as {17, 124, 13, 17} with the uniform Bakis-type topology for every state. For another, some phones process a singleton state occurrence. For instance, the fenonic sequence of phone [ZH] in TEDliumv2 is {29, 29, 29, 29, 29, 29, 29, 29, 1, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40}. We recognized the single occurrence of state 1 as a fortuity and believed that it should have been 29 or 40 in that location. However, the comparison experiments between setting the fenonic baseform of [ZH] as {29, 40} and {29, 1, 40} indicates that every remained state should be respected even though it is a singleton.

VII. EXPERIMENTAL RESULTS

For the experiments, we first conduct ablation tests on different elementary HMM topologies for the fenone and the dimension of DNNVQ prototypes. Thereafter, we validate the proposed NFBG-based HMM structure in both monophone and triphone systems, with context-independent and context-dependent inputs, and in already advanced systems.

A. EFFECTS OF THE ELEMENTARY HMM TOPOLOGY OF THE FENONE

To execute the test in a fair comparison, we control the nodes of the output layer to stay the same as their respective Kaldi recipe (i.e., 127 nodes for Tedliumv2 while 144 nodes for TIMIT). This rules out the possibility that any performance improvement would come from different dimensions of the output layer. Table 3 shows the effect of different HMM topologies for the fenone. Three sorts of HMM topologies in the table correspond to three HMM model examples in Fig. 4. It is apparently shown that the basic left-to-right topology outperforms the ergodic topology considerably, consistent with the observation in [46]. The full ergodic model tends to overfit the training data since it has large amounts of parameters and the resultant high model complexity, resulting in a poor generalization. The margin between the Bakis topology and Vintsyuk topology is more pronounced on TIMIT than that on TEDlium. We believe that it is owing to less training data on TIMIT. As the Bakis-type topology outperforms both the ergodic topology and the Vintsyuk topology, all the subsequent results in this paper are obtained using it as a fenone topology.

B. EFFECTS OF THE NUMBER OF DNNVQ PROTOTYPES

As NFBG introduces state tying, different numbers of DNNVQ prototypes lead to a different number of HMM states in the NFBG-based model. Accordingly, the model complexity and model strength vary. Fig. 5 highlights the effect of setting different numbers of DNNVQ prototypes. The corresponding HMM states of $N_{\text{clu}} \in \{127, 250, 350, 450, 700, 1000\}$ are {94, 110, 116, 120, 132, 199} on Tedlium, while the HMM states of $N_{\text{clu}} \in \{144, 250, 350, 450, 700, 1000\}$ are {110, 121, 134, 139, 144, 220} on TIMIT. The recognition performance is gradually improved before $N_{\text{clu}} = 250$ on both TEDliumv2 and TIMIT; thereafter, it exposes a downward trend on both corpora. Especially on TIMIT, the curve plunges from $N_{\text{clu}} = 450$, where the model underfits due to large amounts of parameters and insufficient data. Therefore, we take $N_{\text{clu}} = 250$ as the default number of DNNVQ prototypes for the subsequent experiments.

TABLE 3. WER[%] on TIMIT and TEDliumv2 for different elementary HMM topologies in monophone systems.

Corpus	HMM topology	GMM-HMM	DNN-HMM
TEDliumv2	ergodic	59.6	45.9
	Bakis [24]	**54.5**	**36.9**
	Vintsyuk [41]	54.7	37.2
TIMIT	ergodic	35.6	31.1
	Bakis [24]	**30.8**	**23.1**
	Vintsyuk [41]	31.3	24.3

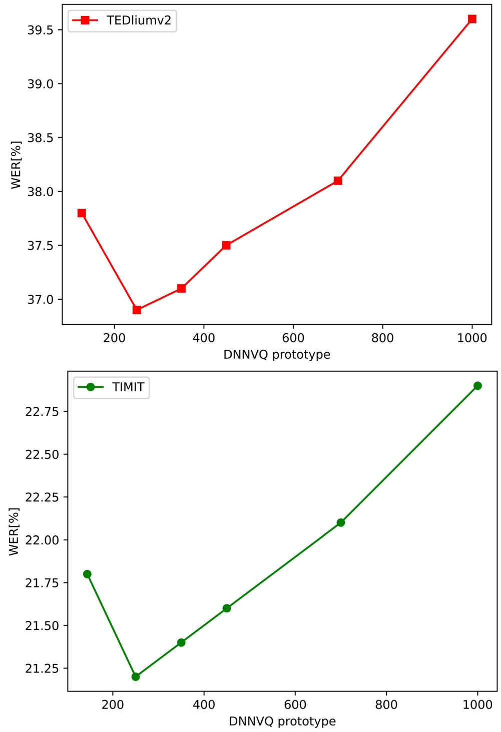


FIGURE 5. Evolution of WER[%] along the number of DNNVQ prototypes on TEDliumv2 and TIMIT, respectively.

C. NFBG VALIDATION IN MONOPHONE SYSTEMS

NFBG introduces a natural state tying in the monophone system. Consequently, the NFBG-based HMM structure reduces the number of PDFs from 127 to 110 on TEDliumv2 while 144 to 121 on TIMIT, leading to the HMM structure's parameters are ∼15% fewer compared to the baseline. Additionally, fewer parameters also make the HMM topology more resistant to overfitting. As presented in Table 4, on TEDliumv2, NFBG-based HMM delivers 2.5% relative improvements in the GMM-HMM system, while 13.8% in the DNN-HMM system with a 15% smaller parameter scale. Comparatively, improvements are more distinct on TIMIT, which are 5.8% and 14.8% relative in the GMM-HMM system and DNN-HMM system, respectively, with more than 15% fewer parameters.

TABLE 4. Impacts of the NFBG-based HMM structure in monophone systems. Results are in WER[%].

Corpus	HMM structure	GMM-HMM	DNN-HMM
TEDliumv2	baseline	55.9	42.8
	our work	**54.5**	**36.9**
TIMIT	baseline	32.7	27.1
	our work	**30.8**	**23.1**

D. NFBG VALIDATION WITH CONTEXT-DEPENDENT INPUTS

We also examine the effectiveness of NFBG-based HMM in the system with context-dependent inputs. By setting $N_{\text{spl}} = m$, inputs are spliced over $(2m + 1)$ frames. It is worth noting that the setups of the context-dependent system stay in accordance with the monophone system (Section VII-C) except for the inputs. Here we only display results in the DNN-HMM system. Tabel 5 displays that the NFBG-based HMM outperforms the corresponding baseline system in all $N_{\text{spl}} \in \{0, 1, 2, 3, 4\}$ circumstances on both corpora. Especially on TEDliumv2, the NFBG-based HMM yields 17.1% relative improvements when $N_{\text{spl}} = 1$. However, we cannot obtain improvements constantly by increasing the input dimension. When $N_{\text{spl}} > 2$, there is no further improvement.

E. TRIPHONE SYSTEMS

The triphone generation leverages the benefit of the Kaldi recipe.[1] Table 6 shows that on Tedliumv2, the relative improvement attained by the NFBG-based HMM is 1.8% in the GMM-HMM system. Similarly, in the DNN-HMM system, the NFBG-based HMM improves WERs in all splice conditions while the most significant improvement appears in the $N_{\text{spl}} = 1$ circumstance, which is 3.3%. As for TIMIT, overall improvements are more distinct compared to TEDliumv2, and the most predominant improvement also appears in $N_{\text{spl}} = 1$, which is 4.5%. Similar to Section VII-D, there is no further improvement when $N_{\text{spl}} > 2$.

F. COMPARISONS WITH ADVANCED MODELS

In this section, we configure our best HMM structure for published state-of-the-art systems on both TEDliumv2 and TIMIT. Here we choose three representative systems for Tedliumv2: the time delay neural network (TDNN) [16], SincNet architecture [47], and the improved RWTH ASR system with SpecAugment [23]. The TDNN models long term temporal dependencies with training times comparable to standard feed-forward DNNs. The network uses sub-sampling to reduce computation during training. It shows a relative WER improvement of 6% on both Switchboard and TEDlumv2 corpus. SincNet is a novel CNN architecture that encourages the first convolutional layer to discover more meaningful filters. In contrast to standard CNNs, which learn all elements of each filter, only low and high cutoff frequencies are directly learned from data with SincNet. Experimental results show that SincNet converges faster and performs better than a standard CNN on raw waveforms. The improved RWTH ASR system with SpecAugmenta is a complete training pipeline to build a state-of-the-art hybrid HMM-based ASR system on the TEDliumv2 corpus. Data augmentation using SpecAugment [50] is successfully applied therein. Their best system achieves a 5.6% WER on the test set, which outperforms the previous state-of-the-art by 27% relative.

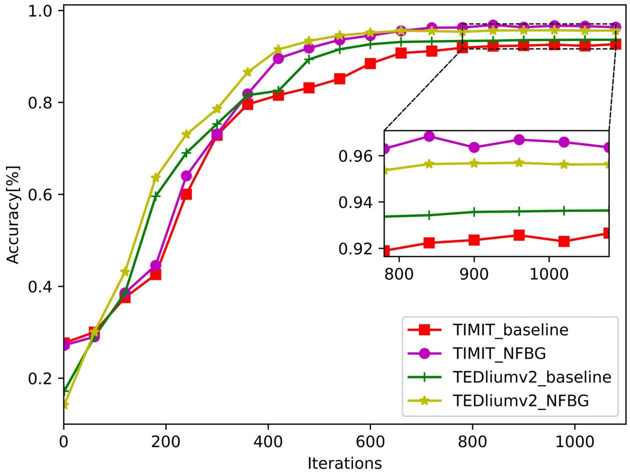


FIGURE 6. Training accuracy and converge speed on both TEDliumv2 and TIMIT.

Besides, we also choose two systems on TIMIT: DNN with a regularization post-layer [48] and DNN with instantaneous frequency features [49]. Vaněk *et al.* [48] propose a regularization post-layer that can be combined with prior techniques, and it brings additional robustness to the DNN. On the TIMIT benchmark task, the adoption of the regularization post layer gives better results than DNN with DBN pre-training. Nayak *et al.* [49] extract features from its time derivative, referred to as instantaneous frequency (IF), to solve the inevitable phase wrapping problem. The combination of IF and MFCC features based systems, using minimum Bayes risk decoding, provides a relative improvement of 8.7% over the baseline system.

Table 7 presents the effectiveness of the NFBG-based HMM in the aforementioned state-of-the-art models. As we observe, these advanced systems are hard to be further improved since they are already exceedingly-optimized. On TEDliumv2, the NFBG-based HMM achieves a 0.3%

[1] https://github.com/kaldi-asr/kaldi/tree/master/egs/wsj/s5/steps

TABLE 5. Impacts of the NFBG-based HMM structure with context-dependent inputs in monophone systems. Results are in WER[%].

Corpus	HMM structure	N_{spl} 0	1	2	3	4
TEDliumv2	baseline	42.8	36.3	35.3	35.1	35.1
	our work	**36.9**	**30.1**	**29.3**	**29.3**	**29.3**
TIMIT	baseline	27.1	23.6	23.2	23.1	23.0
	our work	**23.1**	**21.7**	**21.1**	**21.1**	**21.2**

TABLE 6. Impacts of the NFBG-based HMM structure in triphone systems. Results are in WER[%].

TEDliumv2

GMM-HMM		DNN-HMM				
N_{spl}	0	0	1	2	3	4
baseline	27.3	22.1	21.5	20.2	20.0	19.9
our work	**26.8**	**21.6**	**20.8**	**19.7**	**19.7**	**19.7**

TIMIT

GMM-HMM		DNN-HMM				
N_{spl}	0	0	1	2	3	4
baseline	25.6	20.4	20.2	19.7	19.6	19.6
our work	**24.9**	**19.8**	**19.3**	**19.2**	**19.2**	**19.3**

TABLE 7. The impact of the NFBG-based HMM structures in different advanced models. Results are in WER[%].

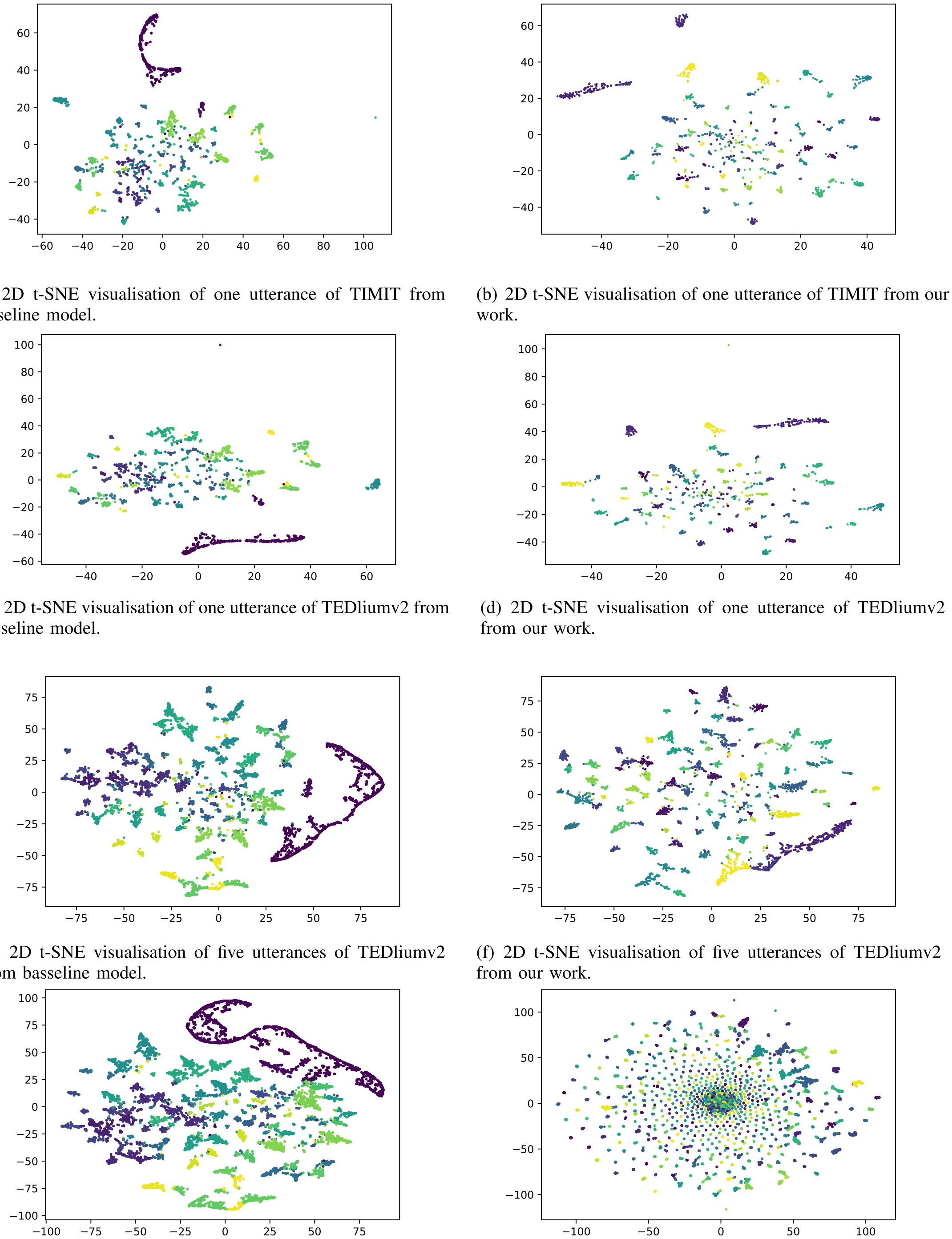
Model	Corpus	Model Structure	Baseline	Our Work
TDNN [16]	TEDliumv2	HMM-TDNN+iVectors+4-gram LM	17.9	**17.6**
SincNet [47]	TEDliumv2	CNN+layer Norm+Dropout+DNN+4-gram LM	21.8	**21.5**
RWTHv [23]	TEDliumv2	HMM-BLSTM+iVectors+SpecAugment+sMBR+Transformer LM	5.6	**5.5**
Regularization [48]	TIMIT	DNN-HMM with last layer regularization	18.3	**17.6**
IF feature [49]	TIMIT	DNN-HMM with MFCC + IF features	17.7	**17.2**

absolute improvement in both the SincNet system and TDNN. Furthermore, the absolute improvement in the improved RWTH ASR system is 0.1%. In comparison, on TIMIT, the NFBG-based HMM performs better. It delivers 3.8% and 2.8% relative improvements in the regularization post-layer and IF feature systems, respectively.

VIII. DISCUSSIONS

From the above results, it is seemingly that the NFBG-generated HMM structure yields improvements in both monophone and triphone systems on both TEDliumv2 and TIMIT. Overall, the improvement on TIMIT is more predominant than that on TEDliumv2. For instance, in the monophone hybrid system, the NFBG-based HMM achieves 13.8% relative improvement on TEDliumv2 while the counterpart of TIMIT is 14.8%. Besides, in the triphone hybrid system, the NFBG-based HMM outperforms the baseline by 3.3% on TEDliumv2 while that is 4.5% on TIMIT. We assume that since the amount of training data of TIMIT is fewer than that of TEDliumv2, TIMIT benefits more from the reduction of parameters. In this section, we provide evidence from the aspect of the convergence speed and the classification performance of the network w/o NFBG-based HMM.

As displayed in Fig 6, introducing NFBG for the HMM construction leads to the accuracy increment on both TEDliumv2 and TIMIT. Additionally, the effectiveness is more distinctive on TIMIT. Besides, the system with the proposed HMM converges faster on both corpora. Moreover, TEDliumv2 is even faster than TIMIT, since there are



(a) 2D t-SNE visualisation of one utterance of TIMIT from basseline model.

(b) 2D t-SNE visualisation of one utterance of TIMIT from our work.

(c) 2D t-SNE visualisation of one utterance of TEDliumv2 from basseline model.

(d) 2D t-SNE visualisation of one utterance of TEDliumv2 from our work.

(e) 2D t-SNE visualisation of five utterances of TEDliumv2 from basseline model.

(f) 2D t-SNE visualisation of five utterances of TEDliumv2 from our work.

(g) 2D t-SNE visualisation of fifteen utterances of TIMIT from basseline model.

(h) 2D t-SNE visualisation of fifteen utterances of TIMIT from our work.

FIGURE 7. 2D t-SNE visualisation from the baseline model and the proposed model. Horizontal axis: the 1st dimension of t-SNE; vertical axis: the 2nd dimension of t-SNE.

110 HMM states on TEDliumv2 while 121 HMM states on TIMIT.

Besides, we also choose t-distributed stochastic neighbor embedding (t-SNE) [51] to visualize the outputs of the network w/o the proposed HMM structure. To begin with, We extract one utterance from the test sets of both TEDliumv2 and TIMIT corpora. Furthermore, we also make a comparison on more test utterances from both corpora. We set the number of prototypes of the DNNVQ the same as the number of PDFs in the baseline model. The perplexity is set to be 30. From every pair of comparisons as dipicted in Fig 7, it is apparent that the employment of NFBG-based HMM reinforces the networks's classification ability.

IX. CONCLUSION

This paper proposes a novel, concise, data-driven, and deep-learning-based method to customize HMM topology for every phone. The proposed algorithm allows the data to reveal their dynamic structure without external assumptions and with a low computational cost. We conduct ablation tests on different HMM topologies and the number of DNNVQ prototypes. Besides, we validate the proposed algorithm on TEDliumv2 and TIMIT in both monophone and triphone systems. Empirical results indicate that the proposed approach improves both the monophone system's and the triphone system's performances. The margin on TIMIT, a corpus with a small amount of training data, is more remarkable. Albeit the limited improvements in the already highly-optimized systems, it reduces the parameters of those systems by 15%. It is safe to conclude that this light-weight HMM structure possesses considerable potentials in various realistic situations, e.g., the keyword spotting task in the always-on and battery-powered application scenarios for smart devices, with severe constraints on hardware resources and power consumption; the task of low-resource speech recognition; the classification task on portable devices. Therefore, our future work will concentrate on the proper employment of the NFBG-based HMM in realistic situations.

REFERENCES

[1] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *Ann. Math. Statist.*, vol. 37, no. 6, pp. 1554–1563, Dec. 1966.

[2] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. Audio, Speech, Language Process.*, vol. 20, no. 1, pp. 30–42, Jan. 2012.

[3] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "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.

[4] T. N. Sainath, A.-R. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 8614–8618.

[5] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 6645–6649.

[6] 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.*, 2006, pp. 369–376.

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

[8] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 577–585.

[9] 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.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[11] L. Bin, S. Nie, S. Liang, W. Liu, M. Yu, and L. Chen, "Jointly adversarial enhancement training for robust end-to-end speech recognition," in *Proc. ISCA*, 2019, pp. 491–495.

[12] C. Lüscher, E. Beck, K. Irie, M. Kitza, W. Michel, A. Zeyer, R. Schlüter, and H. Ney, "RWTH ASR systems for LibriSpeech: Hybrid vs attention–w/o data augmentation," 2019, *arXiv:1905.03072*. [Online]. Available: http://arxiv.org/abs/1905.03072

[13] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 5206–5210.

[14] Y. Wang, A. Mohamed, D. Le, C. Liu, A. Xiao, J. Mahadeokar, H. Huang, A. Tjandra, X. Zhang, F. Zhang, C. Fuegen, G. Zweig, and M. L. Seltzer, "Transformer-based acoustic modeling for hybrid speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 6874–6878.

[15] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Proc. Interspeech*, 2015, pp. 3586–3589.

[16] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proc. Interspeech*, 2015, pp. 3214–3218.

[17] W. Chan and I. Lane, "Deep recurrent neural networks for acoustic modelling," 2015, *arXiv:1504.01482*. [Online]. Available: http://arxiv.org/abs/1504.01482

[18] G. Saon, H.-K. J. Kuo, S. Rennie, and M. Picheny, "The IBM 2015 english conversational telephone speech recognition system," 2015, *arXiv:1505.05899*. [Online]. Available: http://arxiv.org/abs/1505.05899

[19] G. Saon, T. Sercu, S. Rennie, and H.-K.-J. Kuo, "The IBM 2016 english conversational telephone speech recognition system," in *Proc. Interspeech*, 2016, pp. 7–11.

[20] S. Hazmoune, F. Bougamouza, S. Mazouzi, and M. Benmohammed, "A new hybrid framework based on hidden Markov models and K-nearest neighbors for speech recognition," *Int. J. Speech Technol.*, vol. 21, no. 3, pp. 689–704, Sep. 2018.

[21] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohammadi, and S. Khudanpur, "Semi-orthogonal low-rank matrix factorization for deep neural networks," in *Proc. Interspeech*, 2018, pp. 3743–3747.

[22] D. Le, X. Zhang, W. Zheng, C. Fugen, G. Zweig, and M. L. Seltzer, "From senones to chenones: Tied context-dependent graphemes for hybrid speech recognition," in *Proc. IEEE Autom. Speech Recognit. Understand. Workshop (ASRU)*, Dec. 2019, pp. 457–464.

[23] W. Zhou, W. Michel, K. Irie, M. Kitza, R. Schluter, and H. Ney, "The RWTH ASR system for TED-LIUM release 2: Improving hybrid HMM with specaugment," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 7839–7843.

[24] R. Bakis, "Continuous speech recognition via centisecond acoustic states," *J. Acoust. Soc. Amer.*, vol. 59, no. S1, p. S97, Apr. 1976.

[25] L. Rabiner and S. Levinson, "A speaker-independent, syntax-directed, connected word recognition system based on hidden Markov models and level building," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 33, no. 3, pp. 561–573, Jun. 1985.

[26] A. Biem, "A model selection criterion for classification: Application to HMM topology optimization," in *Proc. 7th Int. Conf. Document Anal. Recognit.*, Aug. 2003, pp. 104–108.

[27] J. Geiger, J. Schenk, F. Wallhoff, and G. Rigoll, "Optimizing the number of states for HMM-based on-line handwritten whiteboard recognition," in *Proc. 12th Int. Conf. Frontiers Handwriting Recognit.*, Nov. 2010, pp. 107–112.

[28] M. Zimmermann and H. Bunke, "Hidden Markov model length optimization for handwriting recognition systems," in *Proc. 8th Int. Workshop Frontiers Handwriting Recognit.*, Aug. 2002, pp. 369–374.

[29] M.-P. Schambach, "Model length adaptation of an HMM based cursive word recognition system," in *Proc. 7th Int. Conf. Document Anal. Recognit.*, vol. 3, Aug. 2003, p. 109.

[30] Z. Jiang, X. Ding, L. Peng, and C. Liu, "Analyzing the information entropy of states to optimize the number of states in an HMM-based off-line handwritten Arabic word recognizer," in *Proc. 21st Int. Conf. Pattern Recognit. (ICPR)*, Nov. 2012, pp. 697–700.

[31] K. Ait-Mohand, T. Paquet, and N. Ragot, "Combining structure and parameter adaptation of HMMs for printed text recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 9, pp. 1716–1732, Sep. 2014.

[32] N. Cirera, A. Fornes, and J. Llados, "Hidden Markov model topology optimization for handwriting recognition," in *Proc. 13th Int. Conf. Document Anal. Recognit. (ICDAR)*, Aug. 2015, pp. 626–630.

[33] R. K. Moore and L. Skidmore, "On the use/misuse of the term 'Phoneme,'" 2019, *arXiv:1907.11640*. [Online]. Available: http://arxiv.org/abs/1907.11640

[34] L. R. Bahl, P. F. Brown, P. V. de Souza, and M. A. Picheny, "Acoustic Markov models used in the tangora speech recognition system," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Jan. 1988, pp. 497–498.

[35] L. R. Bahl, P. F. Brown, P. V. de Souza, R. L. Mercer, and M. A. Picheny, "A method for the construction of acoustic Markov models for words," *IEEE Trans. Speech Audio Process.*, vol. 1, no. 4, pp. 443–452, Oct. 1993.

[36] G. Rigoll, C. Neukirchen, and J. Rottland, "A new hybrid system based on MMI-neural networks for the RM speech recognition task," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. Conf.*, vol. 2, May 1996, pp. 865–868.

[37] T. Watzel, L. Li, L. Kürzinger, and G. Rigoll, "Deep neural network quantizers outperforming continuous speech recognition systems," in *Proc. Int. Conf. Speech Comput.* Cham, Switzerland: Springer, 2019, pp. 530–539.

[38] K. Veselý, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *Proc. Interspeech*, 2013, pp. 2345–2349.

[39] A. Rousseau, P. Deléglise, and Y. Esteve, "Enhancing the TED-LIUM corpus with selected data for language modeling and more TED talks," in *Proc. LREC*, 2014, pp. 3935–3939.

[40] J. S. Garofolo, "Timit acoustic phonetic continuous speech corpus," in *Linguistic Data Consortium*. San Diego, CA, USA: Abacus Data Network, 1993.

[41] T. K. Vintsyuk, "Element-wise recognition of continuous speech composed of words from a specified dictionary," *Cybernetics*, vol. 7, no. 2, pp. 361–372, 1971.

[42] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: http://arxiv.org/abs/1502.03167

[43] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[45] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, and M. Hannemann, "The Kaldi speech recognition toolkit," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand.*, Dec. 2011, pp. 1–4.

[46] K. T. Abou-Moustafa, M. Cheriet, and C. Y. Suen, "On the structure of hidden Markov models," *Pattern Recognit. Lett.*, vol. 25, no. 8, pp. 923–931, Jun. 2004.

[47] M. Ravanelli and Y. Bengio, "Speech and speaker recognition from raw waveform with SincNet," 2018, *arXiv:1812.05920*. [Online]. Available: http://arxiv.org/abs/1812.05920

[48] J. Vaněk, J. Zelinka, D. Soutner, and J. Psutka, "A regularization post layer: An additional way how to make deep neural networks robust," in *Proc. Int. Conf. Stat. Lang. Speech Process.* Cham, Switzerland: Springer, 2017, pp. 204–214.

[49] S. Nayak, S. Bhati, and K. S. R. Murty, "An investigation into instantaneous frequency estimation methods for improved speech recognition features," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Nov. 2017, pp. 363–367.

[50] 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

[51] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

LUJUN LI is currently pursuing the Ph.D. degree with the Chair of Human-Machine Communication, Department of Electrical and Computer Engineering, Technical University of Munich, Munich, Germany. Her research interests include pattern recognition, machine learning, and deep learning, covering areas, such as computational auditory scene analysis, robust speech recognition leveraging reinforcement learning, jointly-trained system of speech enhancement and speech recognition using generative adversarial networks.

TOBIAS WATZEL was born in Mainz, Germany, in 1991. He received the B.Eng. degree in electrical and computer engineering from the Frankfurt University of Applied Science, in 2014, and the M.Sc. degree from the Technical University of Munich (TUM), in 2017, where he is currently pursuing the Ph.D. degree with the Chair of Human-Machine Communication. His research interests include machine learning, speech recognition, and computer vision.

LUDWIG KÜRZINGER received the Dipl.-Ing. (Univ.) degree in electrical and computer engineering from the Technical University of Munich, Germany, in 2014, where he is currently pursuing the Ph.D. degree. From 2015 to 2018, he worked with the Fraunhofer Institute of Integrated and Applied Security (AISEC), Munich, in the field of hardware security with a focus on physically unclonable functions. He joined the Speech Recognition Group, Chair of Human-Machine-Communication, Technical University of Munich, in 2018. His current research interests include sequence classification and speech recognition.

GERHARD RIGOLL (Fellow, IEEE) received the Dipl.-Ing. degree from Stuttgart University, Germany, in 1982, the Dr.-Ing. degree in automatic speech recognition from the Department of Advanced Information and Communication Technologies, Fraunhofer-Institute (IAO), Stuttgart, in 1986, and the Dr.-Ing. Habil. degree from Stuttgart University, in 1991, with a thesis on speech synthesis. He joined the Department of Advanced Information and Communication Technologies, Fraunhofer-Institute (IAO), as a Researcher. From 1986 to 1988, he worked as a Postdoctoral Fellow with the IBM T. J. Watson Research Center, Yorktown Heights/USA on acoustic modeling and speaker adaptation for the IBM Tangora speech recognition system. From 1991 to 1993, he worked as a Guest Researcher in the framework of the EC Scientific Training Program in Japan with NTT Human Interface Laboratories, Tokyo, Japan, in the area of neural networks and hybrid speech recognition systems. He was appointed to a Full Professor of computer science with Gerhard-Mercator-University, Duisburg, Germany, in 1993. He joined the Technical University of Munich (TUM), in 2002, where he is currently heading the Institute for Human-Machine Communication. He has been involved in international research and teaching activities as a Visiting Professor with NAIST, Nara, Japan, in 2005. Since 2011, he has been a Lecturer with TUM-Asia, Singapore. Since 2017, he has been a Coordinator with the Electrical Engineering Faculty, Chinese-German College for Postgraduate Studies (CDHK), Tongji University, Shanghai, China. He has been active as a project reviewer and a proposal evaluator in a variety of national and international projects, sponsored by the European Commission, the German National Science Foundation (DFG), the German Ministry for Research and Education (BMBF), and other research foundations in U.K., The Netherlands, Finland, and Switzerland. He is the author or coauthor of more than 550 articles in the field of pattern recognition, covering the above mentioned application areas. His research interests include human-machine communication and multimedia information processing, covering areas, such as speech and handwriting recognition, gesture recognition, face detection and identification, action and emotion recognition, and interactive computer graphics. He is an IEEE Fellow (for contributions to multimodal human-machine communication). He serves as a Reviewer for many scientific journals. He has been a Session Chairman and a member of the program committee for numerous international conferences.

• • •