Speaker Adaptation Using i-Vector Based Clustering

Minsoo Kim[1], Gil-Jin Jang[1*], Ji-Hwan Kim[2], and Minho Lee[1]
[1] School of Electronics Engineering, Kyungpook National University
80 Daehakro, Bukgu, Daegu, 41566, Korean
[e-mail: minsookim@knu.ac.kr, gjang@knu.ac.kr, mholee@gmail.com]
[2] Department of Computer Science and Engineering, Sogang University
35 Baekbeom-ro, Mapo-gu, Seoul, 04107, Republic of Korea
[e-mail: kimjihwan@sogang.ac.kr]
*Corresponding author: Gil-Jin Jang

Received November 29, 2019; revised February 3, 2020; accepted May 8, 2020; published July 31, 2020

Abstract

We propose a novel speaker adaptation method using acoustic model clustering. The similarity of different speakers is defined by the cosine distance between their *i*-vectors (intermediate vectors), and various efficient clustering algorithms are applied to obtain a number of speaker subsets with different characteristics. The speaker-independent model is then retrained with the training data of the individual speaker subsets grouped by the clustering results, and an unknown speech is recognized by the retrained model of the closest cluster. The proposed method is applied to a large-scale speech recognition system implemented by a hybrid hidden Markov model and deep neural network framework. An experiment was conducted to evaluate the word error rates using Resource Management database. When the proposed speaker adaptation method using *i*-vector based clustering was applied, the performance, as compared to that of the conventional speaker-independent speech recognition model, was improved relatively by as much as 12.2% for the conventional fully neural network, and by as much as 10.5% for the bidirectional long short-term memory.

Keywords: Speaker adaptation, speech recognition, i-vector, clustering, hybrid HMM-DNN

A preliminary version of this paper appeared in ICEIC 2018, January 24-27, Honolulu, USA. This version includes a concrete analysis and improved speaker clustering method with extended experimental results on resource management (RM) dataset. Therefore, more than 30% substantial new contributions are added to the methods, experiments, and analysis of the results. This work was supported by Institute for Information and communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. R7124-16-0004, Development of Intelligent Interaction Technology Based on Context Awareness and Human Intention Understanding, 50%) and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. NRF-2017M3C1B6071400, 50%).

http://doi.org/10.3837/tiis.2020.07.003 **ISSN : 1976-7277**

1. Introduction

The performance of a speech recognition system is largely dependent on the choice training dataset. If the speaking styles of a target speaker including speaking rates and pronunciation variations are different to those in the training dataset, the speech recognition model is not suited to the target speaker and the performance drops proportionally to this difference. In the case that the entire dataset is composed of speakers with too many different speaking styles, the unified speaker-independent (SI) model may not be trained reliably because of too many intra-unit variabilities.

To solve the problem of this mismatch between training and testing, many speaker adaptation methods have been proposed, including maximum likelihood linear regression (MLLR) [1] and eigenvoice projection [2]. These methods are based on the assumptions that speaker variations can be correctly modeled by a linear transformation. A more general approach has been proposed using maximum a posteriori (MAP) parameter estimation [3]. However, all of the aforementioned adaptation methods are tightly coupled with multivariate Gaussian probability density functions, so they can be applied only to hidden Markov models (HMM) with their observation probabilities modeled by a mixture of Gaussians. Therefore, these methods cannot be applied to other observation probability models such as artificial neural networks.

This study proposes a novel method for speaker adaptation. The proposed method performs clustering of speakers in the training dataset based on *i*-vector similarities [4, 5] given speaker-specific training data. It then generates cluster-specific, adapted models by retraining the unified SI model using training data of the corresponding cluster. The SI model is obtained by a hybrid hidden Markov model and deep neural network (HMM-DNN) [6-9] with the training data of all the speakers. In addition, the number of speaker group-dependent models equal to the number of clusters. We also propose a means of choosing the best model using *i*-vector similarity.

Section 2 presents related work on *i*-vector extraction method, and HMM-DNN speech recognition system. Section 3 describes *i*-vector based similarity and speaker clustering procedure that are based on it. The section also describes the method for recognizing an unknown sentence using the clustered models. Section 4 presents experimental results of the large-scale speech recognition system based on a resource management (RM) database [10] and describes the performance improvement derived from using the proposed method.

2. Related Work

Speaker recognition is a task of identifying different characteristics of various human speakers and applying them in such a manner that two or more speakers can be distinguished. To model effectively the characteristics of speech signals spoken by the given speakers, Gaussian mixture models (GMM) based on the universal background model (UBM) [11] have been shown to be effective. Joint factor analysis (JFA) [12] finds two subspaces that represent the speaker and channel variabilities, respectively. Based on the subspace assumption, the GMM supervector **M** can be modeled as follows:

$$\mathbf{M} = \mathbf{m} + \mathbf{T}\mathbf{w}\,, \tag{1}$$

where **m** denotes a speaker and channel independent supervector often constructed by a UBM model parameters, **T** is a low rank matrix that is the basis of the reduced total variability space, and **w** represents total factors distributed as a standard normal random variable [13, 14]. Most of the speaker information is assumed to be present in the total factor vector **w**. Therefore, it is referred to as an *intermediate* vector (*i*-vector). This speaker information represented by the *i*-vector can be used to measure the similarity between speakers.

Several steps as shown in **Fig. 1** extracts the *i*-vector. In the first step, sixty-dimensional multi-taper mel-frequency cepstral coefficient (MFCC) features [15, 16] are extracted from the training data. The multi-taper features are obtained by splicing together 3 frames on each side of the center frame of 13-dimensional MFCCs, resulting in $7 \times 13 = 91$ dimensions, and they are projected down to 40 using linear discriminant analysis (LDA). On the LDA-projected MFCCs, a single semi-tied covariance (STC) transform [17] is performed, and additional 20 features are added. The combined, 60-dimensional features are referred to as LDA+STC [15] and used as an input to the *i*-vector extractor. The GMM-UBM using full-covariance GMMs with 512 components is trained using Baum-Welch statistics extraction [18]. All the parameters of the trained GMM-UBM are converted into a single supervector, and reduced to 100 dimensional *i*-vectors using the *i*-vector extractor (the total variability matrix **T**).

In addition, we used the HMM-DNN hybrid speech recognition model as a baseline. This model is developed using an open-source Kaldi toolkit [19], and trained using frame-based cross-entropy and different sequence-discriminative criteria. The Kaldi toolkit supports stochastic gradient descent learning using restricted Boltzmann machine (RBM) prerequisite [6, 9, 20] and NVidia graphics processing unit.

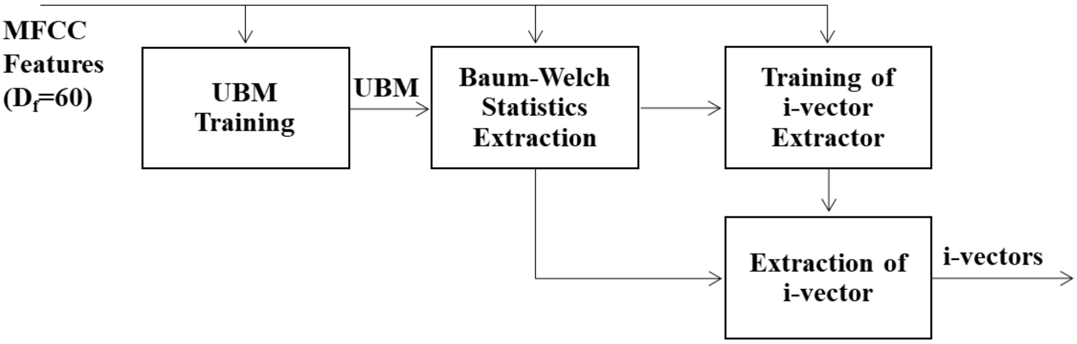

Fig. 1. GMM-UBM based *i*-vector extractor.

3. Proposed Methods

3.1 Speaker Similarity Measure

The similarity of a pair of speakers is defined by the proximity of the subspaces spanned by two *i*-vectors [4]. Mathematically, it is computed by cosine similarity between the *i*-vectors of the given speakers as follows:

$$S(\mathbf{w}_1, \mathbf{w}_2) = \cos \angle_{\mathbf{w}_1,\mathbf{w}_2} = \frac{\mathbf{w}_1 \cdot \mathbf{w}_2}{\|\mathbf{w}_1\|\|\mathbf{w}_2\|} = \frac{\sum_{k=1}^{d} w_{1,k} w_{2,k}}{\sqrt{\sum_{k=1}^{d} w_{1,k}^2}\sqrt{\sum_{k=1}^{d} w_{2,k}^2}}, \qquad (2)$$

where $\mathbf{w}_1$ and $\mathbf{w}_2$ are the *i*-vectors of two speakers, d is their shared dimension, and $w_{1,k}$ and $w_{2,k}$ are the k^{th} component of the vectors $\mathbf{w}_1$ and $\mathbf{w}_2$, respectively. The cosine similarity becomes 1.0 when the direction of the two vectors match perfectly, and 0.0 for perpendicular vectors. This measure is useful when the scale does not affect the similarity.

3.2 Speaker Clustering Using Group Average

Once a similarity metric is defined, a set of speakers can be grouped into a number of subsets using agglomerative hierarchical clustering [21, 22], which reduces the number of clusters by merging the closest pair of speakers one by one until a desired number of clusters is achieved. When two clusters are merged into a new cluster, the *i*-vector of the new one is approximated by the average of the *i*-vectors before the merge. The proposed clustering algorithm using group averaging is as follows:

Algorithm 1: *i*-vector-based speaker clustering using group average method

– **Input:**

1) *i*-vector extractor
2) N: total number of speakers
3) C: desired number of clusters, $C \leq N$

– **Output:** C clusters of speakers

I. Extract *i*-vectors of all speakers from the UBM of speaker-specific training data.

II. Construct initial clusters of single speakers as many as the total number of speakers, and set the current number of clusters, c, as the total number of speakers:

$$\{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_c\}, \quad c = N\,.$$

III. Compute cosine similarities between all possible pairs of clusters by Equation (2):

$$\mathbf{S} = \begin{bmatrix} S_{11} & S_{12} & \ldots & S_{1c} \\ S_{21} & S_{22} & \ldots & S_{1c} \\ \vdots & \ddots & \ddots & \vdots \\ S_{c1} & S_{c2} & \ldots & S_{cc} \end{bmatrix}, \quad S_{ij} = S(\mathbf{w}_i, \mathbf{w}_j) \tag{3}$$

IV. Choose a pair with the highest cosine similarity:

$$(i^*, j^*) = \arg\max_{i<j} S_{ij}\,.$$

V. Merge the two clusters into one, assign a new *i*-vector for the new cluster by the average of the merged clusters, and decrease the current number of clusters by 1,

$$\mathbf{w}_{i*} \Leftarrow \frac{1}{2}\left(\mathbf{w}_{i^*} + \mathbf{w}_{j^*}\right), \quad c \Leftarrow c - 1, \quad \text{delete cluster } j^*\,.$$

VI. Repeat steps III, IV, and V until the current number of clusters is less than or equal to the desired number of clusters: $c \leq C$.

First, extract *i*-vectors of all the speakers using the *i*-vector extractor trained by the UBM of speaker-specific training data. Second, construct initial set of clusters from the individual speakers, with the number of clusters equaling the number of speakers. Third, compute cosine similarities of all cluster pairs using Equation (2). The matrix **S** in step II is of size $c \times c$, but only upper half above the diagonal elements are needed because $S_{ij} = S_{ji}$ (symmetric) and the diagonal elements $S_{ii} = 1$ for $1 \leq i, j \leq c$. For the same reason, only $i < j$ pair indices are considered in the subsequent steps. Fourth, choose a pair with the highest cosine similarity. Fifth, merge the two selected clusters, and assign the new cluster *i*-vector by their average. Finally, repeat the aforementioned steps until the desired number of clusters is achieved.

3.3 Speaker Clustering using Ward Linkage Method

The problem of the simple group-averaging scheme for clustering is that the numbers of speakers in the different clusters may become too different if some of the speakers are close in the *i*-vector space. To obtain balanced numbers of speakers, an agglomerative hierarchical clustering using Ward linkage method [20, 21] is applied. The clustering is redesigned using Ward linkage scheme as follows:

Algorithm 2: *i*-vector-based speaker clustering using Ward linkage method

– **Input:** *i*-vector extractor; *N*: total number of speakers; *C*: desired number of clusters, $C \leq N$

– **Output:** *C* clusters of speakers

I. Extract *i*-vectors of all speakers from the UBM of speaker-specific training data.

II. Construct initial clusters of single speakers as many as the total number of speakers, and set the current number of clusters, *c*, as that of the total speakers.

$$\{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_c\}, \quad c = N\,.$$

III. Compute cosine similarity of every pair of clusters using Equation (2),

$$S_{ij} = S(\mathbf{w}_i, \mathbf{w}_j)\,, \quad 1 \leq i, j \leq c,\ i < j\,.$$

IV. Compute merging factor α of every pair of clusters using Ward linkage method:

$$\alpha_{ij} = \frac{n_i + n_j}{n_i n_j}\,, \tag{4}$$

where n_i and n_j are the numberrs of speakers in clusters *i* and *j*, respectively.

V. Choose a pair with the highest cosine similarity weighted by the merging factor:

$$(i^*, j^*) = \arg\max_{i<j} \alpha_{ij} S_{ij}\,.$$

VI. Merge the two clusters into one, assign its *i*-vector by the weighted average of the merged clusters, and decrease the current number of clusters by 1:

$$\mathbf{w}_{i*} \Leftarrow \frac{n_i \mathbf{w}_{i^*} + n_j \mathbf{w}_{j^*}}{n_i + n_j}\,, \quad c \Leftarrow c - 1\,, \quad \text{delete cluster } j^*\,.$$

VII. Repeat steps III–VI until the current number of clusters is less than or equal to the desired number of clusters: $c \leq C$.

The inputs and outputs, and all procedures except steps IV–VI are the same as Algorithm 1. As shown in step IV, merging factor of every pair is computed by Ward linkage criteria [21]. In order to choose a pair to merge, we select a pair with the highest cosine similarity weighted by the merging factor as shown in step V. The merging factor α_{ij} in Equation (4) is maximized when $n_i = n_j$ for constant $n_i + n_j$, so weighting by α_{ij} guides Algorithm 2 to choose a pair of clusters with similar number of speakers, also resulting in similar numbers of speakers in the final set of clusters [20, 21].

3.4 Speech Recognition Model Retraining and Cluster Selection

Once the clustering is completed, cluster dependent acoustic models can be generated by adapting the speaker-independent (SI) model with the data that belong to the individual clusters. As shown in **Fig. 2**, *i*-vectors are extracted from training data, and used in grouping similar speakers into clusters. According to the speaker clustering results, total training dataset are split into non-overlapping subsets, and they are used in obtaining cluster-dependent (CD) models. However, because the amount of training data for each cluster reduces to $1/N$ on average, acoustic unit training is less reliable due to the reduced amount of training data. Therefore, we train an SI acoustic model by using the data from all speakers, and retrain the SI model with the training data of the cluster-specific speakers to obtain the individual CD models. The proposed algorithm for efficient generation of CD models as follows:

```
Algorithm 3: Cluster-dependent model adaptation
– Input: training data; i-vector extractor; C clusters of speakers.
– Output: C cluster models.
 I.   Obtain a speaker-idenpendent HMM-DNN model using training data of all speakers
      (denoted simply as “SI”).
 II.  Use Algorithms in Sections 3.2 and 3.3 to obtain C speaker clusters.
 III. Split the whole training data into cluster-specific data by the speakers of the individual
      clusters.
 IV.  For c from 1 to C,
       A. Retrain SI using training data of cluster c
       B. To avoid overfitting, start learning rate from half of the value used in training SI
       C. Store the individual models
```

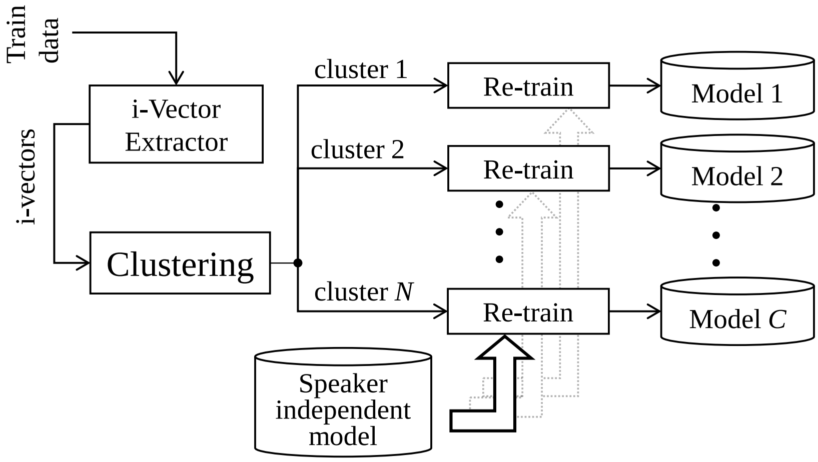

Fig. 2. Cluster selection and retraining from speaker independent model.

To recognize unknown input speech, the best model is chosen among the clustered models. **Fig. 3** illustrates the recognition procedure. Using the *i*-vector extractor obtained by SI UBM, the *i*-vector of unknown speech is extracted. The cosine similarities between the *i*-vectors of the input and cluster center are computed, and the best cluster is chosen whose pairwise similarity is the highest. Speech recognition result is generated by using the chosen CD model. Using the following Algorithm 4, best-matched model is chosen according to the cluster membership so that improved performance should be expected over using the SI model.

Algorithm 4: Cluster-dependent model selection for unknown inputs

```
– Input: speech signal; i-vector extractor; i-vectors of C clusters; C cluster models
– Output: Speech recognition result (text)
I.   Extract features for i-vector from input speech signal.
II.  Extract i-vector using pretrained i-vector extractor from the input features.
III. Calculate cosine similarities between the cluster representative i-vectors (average of
     i-vectors of clustered speakers) and input i-vector.
IV.  Select cluster-dependent model with the highest similarity value.
V.   Perform speech recognition with the selected model.
```

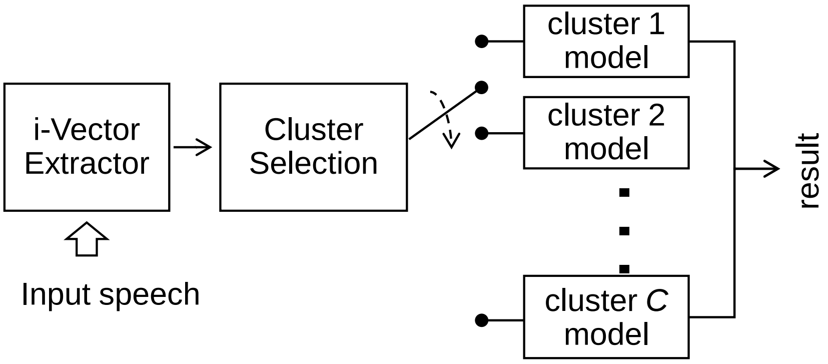

Fig. 3. Speech recognition using clustered models.

4. Performance Evaluation

To show the effectiveness of the proposed method, we have conducted speech recognition experiments on the Resource Management (RM) database [10]. The language is English. Its training set contains 109 speakers and 3,990 sentences, and the test set includes 59 speakers and 1,460 sentences that are entirely different from the training set. The speakers in the test set are not present in the training set as well. Speech recognition and *i*-vector extraction modules are constructed according the recipes in Kaldi speech recognition toolkit [19]. The simulation platfom is composed of NVIDIA GTX Titan X (1.076 GHz) GPU with GDDR5 12GB memory, on a desktop computer with Intel i3-6100 (3.7 GHz) CPU and DDR4 64GB main memory. The operating system and GPU driver versions are Ubuntu Linux 18.04 and CUDA version 10.1. The Kaldi toolkit for speech recognition is compiled from the distributed source code to avoid any mismatch between OS and the toolkits.

4.1 Speech Feature and i-Vector Extraction

The input features for *i*-vector extraction is 60-dimensional multi-taper MFCC [15] from the time-domain speech signal. Using the *i*-vector extraction module described in Section 2, 100-dimensional *i*-vectors are extracted. This model configuration fits the 2010 NIST Speaker

Recognition Evaluation (SRE10) speech database [15]. We modified this model to be suited to RM datasets. The GMM-UBM used to train the *i*-vector extractor was itself trained on RM datasets as well [26].

4.2 Speech Recognition Models

Baseline HMM-DNN speech recognition models are constructed using two different types of DNNs. The first model uses a fully connected neural network (FCN). The input features are 40-dimensional, log mel-filterbank energies at every 10 ms shift length. To model context information in time, delta and acceleration vectors are extracted over 5 frames before and after the current frame. The dimension of the resultant vector is therefore 40×3×(5+1+5) = 1,320. In the HMM-DNN model shown in **Fig. 4**, the extracted input feature vector is passed through five hidden layers, each of which contains 1024 output units, and softmax layer on the top predicts the HMM state label of the input frame. To obtain a reliable initial FCN weights, the hidden layers are pre-trained in an unsupervised manner using restricted Boltzmann machine (RBM) training algorithm without state labels [27]. With the decoded HMM state labels, FCN weights are fine-tuned with per-frame cross-entropy loss and ReLU (rectified linear unit) activation functions at the output units [15, 29].

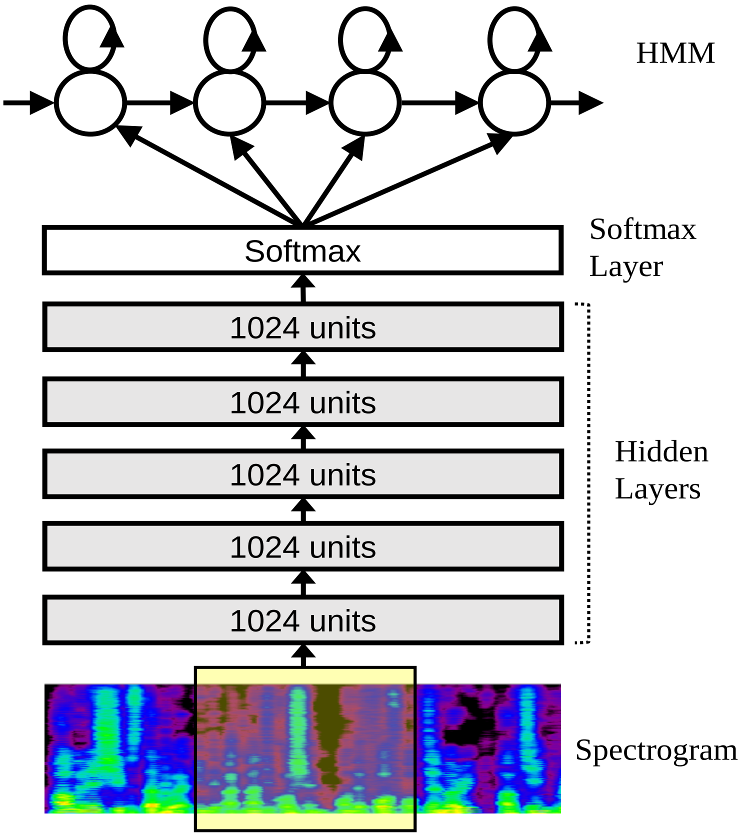

Fig. 4. HMM-DNN speech recognition models using fully-connected network (FCN). The input is a supervector of 11 frames of 40 log mel-filterbank energies, and it is passed through 5 hidden layers and 1 softmax layer to predict HMM state labels.

The second one uses bidirectional long short-term memory (BLSTM). The input feature vector is 40 log mel-filterbank energies at every 10 ms, with their delta and acceleration vectors only because the temporal trajectories of the features are modeled by forward and backward recurrent paths as shown in **Fig. 5**. The total input vector dimension is therefore 120 (40×3), which is much smaller than that of the FCN. The forward and backward layers are given 320 hidden units, and they are projected to 200 state output units [15, 29]. Initial learning rate for retraining cluster-dependent model is set to be half of the learning rate of the SI model.

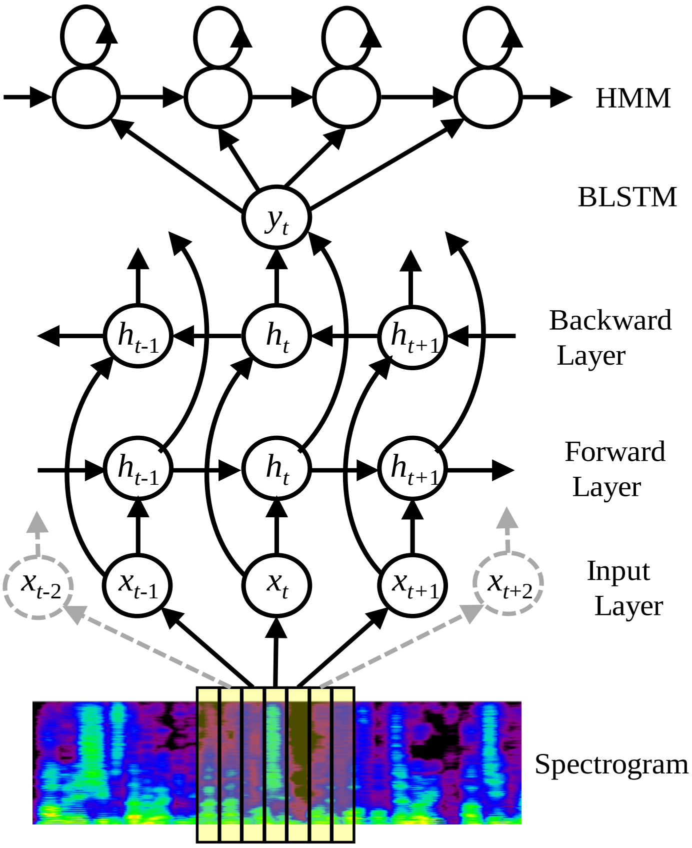

Fig. 5. HMM-DNN speech recognition models using BLSTM. The input is 40 log mel-filterbank energies. Forward and backward layers of 320 hidden nodes models the temporal behavior of the input features, and a fully-connected, single layer is built upon those hidden node outputs to generate 200 state outputs, which is then used to predict HMM state labels.

4.3 Preliminary Analysis of the Clusters

To verify that the proposed clustering methods in Sections 3.2 and 3.3 can split the given set of speakers into appropriate subsets, we analyzed the results of clustering by the number of speakers in the subsets. **Table 1** shows the number of selected speakers for five cluster models using group average in Algorithm 1 and Ward linkage in Algorithm 2. The total number of speakers is 109, so the mean number of speakers per cluster is 21.8 in both cases. The computed standard deviations are 6.5 and 5.4 for group average and Ward linkage methods, respectively, so Ward linkage method generated speaker subsets of more balanced sizes.

We further analyzed the effect of clustering on the performance of speech recognition in terms of word error rates (WERs). Table 2 shows the performance variations according to matched (same cluster numbers) and unmatched (different cluster numbers) conditions. Columns represent test set division by the output of the *i*-vector clustering. Rows are the cluster models used in recognition. The used methods are group average clustering and FCN speech recognition model. “SI (baseline)” is the universal, speaker-independent model trained by the data of all speakers. The row names “Cluster 1”, “Cluster 2”, …, “Cluster 5” are models trained by cluster 1-5 subsets, respectively. In each column, if the column number is the same as the row number, the speaker is optimally classified into the cluster with the least *i*-vector distance. The diagonal ones are represented by boldface fonts, and the lowest WERs are indicated by asterisks. For the test inputs which selected cluster models 1 and 3, the chosen models were optimal in WER values. For the test inputs of cluster 2, although their selected models were not the best but showed at least the second best WER. When SI results were compared to the cluster-matched results, the models for clusters 1, 2, and 3 outperformed the SI model by 0.05%–0.30% WER. For clusters 4 and 5, the *i*-vector-matched models were not the best ones for the inputs, and even failed to improve the baseline SI model. However, this had little effect on the overall performance improvements. The overall performance improved by 0.13% on average, which means the performance relatively improved by 6.8%.

Table 1. Number of speakers assigned to each cluster, when the desired number of clusters is set to 5. Column names “C1”-“C5” indicate the clusters of the speakers.

	Number of speakers of the clusters					mean	**Standard deviation**
	C1	C2	C3	C4	C5		
Group average	30	13	24	18	30	21.8	6.5
Ward linkage	19	29	17	18	26	21.8	5.4

Table 2. WER variabilities by test data matched and unmatched to cluster models using FCN. The second row is the WER by the SI model (single cluster), and rows named “Cluster 1”-“Cluster 5” are the WERs by cluster 1-5 models. The matched condition by *i*-vector clustering is represented by boldface fonts, and the best WER in a column is marked by asterisks (*).

FCN model	Selected cluster numbers of test inputs				
	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
SI (baseline)	2.47	1.51	1.10	2.02	1.75
Cluster 1	**2.17***	2.17	1.13	2.12	2.18
Cluster 2	2.67	**1.46**	1.00	2.01	1.75
Cluster 3	2.63	1.24*	**0.91***	1.85*	1.75
Cluster 4	2.55	1.67	1.18	**2.17**	1.57*
Cluster 5	2.67	1.64	0.96	2.33	**1.83**

Table 3. WER variabilities by test data matched and unmatched to cluster models using BLSTM..

FCN model	Selected cluster numbers of test inputs				
	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
SI (baseline)	2.09	1.73	1.27	1.75	1.66
Cluster 1	**1.91***	1.68	1.22	1.91	1.75
Cluster 2	2.19	**1.64**	1.22	1.64*	1.75
Cluster 3	2.21	1.59*	**1.18***	1.75	1.75
Cluster 4	2.03	1.59*	1.27	**1.75**	1.48*
Cluster 5	2.29	1.68	1.22	2.01	**1.57**

In the case of speech recognition models using BLSTM, we also analyzed the results of five cluster models. **Table 3** shows the performance variations according to the matched and the unmatched conditions. Similarly to the FCN results in **Table 2**, for the test inputs which selected cluster models 1 and 3, the chosen models were the optimal in WER values. For the test inputs of cluster 2, 4, and 5, the selected models were the second best in WER numbers. When SI results were compared with cluster-matched results, all cluster models outperformed the SI model by 0.09%–0.18%, except for cluster 4 which produced the same result as that of the SI model. Therefore, the overall performance improved by 0.13% on average, which means that the performance improved relatively by 7.2%.

4.4 Speech Recognition Performance Evaluation

Speech recognition experiments on RM datasets were carried out to assess the performance variations according to the changes in clustering methods, number of clusters, and speech recognition models. To assess the variations in performance according to the number of clusters, we tried 5, 7, 10, 20, and 40 clusters with 2 different types of clustering methods: group average and Ward linkage methods, respectively. **Table 4** shows comparison results of the baseline SI and cluster-dependent speech recognition models with FCN. The performances were compared based on WERs, and the relative WER improvements over the baseline system. Both methods with various numbers of clusters outperformed the baseline WER by 0.04–0.24%, and their relatively improvements were 2.1–12.2%. Among the various cluster numbers, 20 clusters model was the best for group average method, and 40 clusters were the best for Ward linkage clustering method. Generally larger the number of clusters was, better the speech recognition performance was. This is because, as the number of models is increased, it becomes more likely that better matched models for the input utterances can be used for various speakers, so improved performance can be expected. The downsides are the lack of adaptation data for the cluster models, the computational overhead in finding the best model (shown in **Fig. 3**.), and storage overhead in maintaining large number of models.

Table 4. Performance comparison of the SI baseline model and the proposed models using FCN in terms of WERs and their relative improvements (rel. imp.) over the baseline. All the numbers are in percentages. The relative improvement is not available for the baseline result.

FCN model	Group average	Ward linkage
	WER (rel. imp.)	
SI (baseline)	1.91 (-)	
5 clusters	1.78 (6.8%)	1.87 (2.1%)
7 clusters	1.76 (7.9%)	1.84 (3.7%)
10 clusters	1.81 (5.2%)	1.82 (4.7%)
20 clusters	1.67 (12.2%)	1.73 (9.3%)
40 clusters	1.77 (7.2%)	1.69 (11.4%)
Average	1.75 (8.0%)	1.79 (6.3%)

We also tried various numbers of clusters with the proposed methods for BLSTM as well. **Table 5** shows the comparison of the baseline speaker-independent and cluster-dependent BLSTM speech recognition models. Both of the proposed clustering methods with all the numbers of clusters outperformed the baseline WER by 0.1–0.19%, the relative WER improvements for which were 5.6%–10.5%. When compared to FCN, the baseline WER was 0.11% lower, and 0.03%–0.18% lower WERs with cluster models. Because BLSTM can model the temporal variation of the input features more precisely, the performances were better and reliable with the change in the number of clusters. Comparing clustering methods, Ward linkage exhibited consistent improvement over group average method, by showing continuous improvements with the increase of the number of clusters.

Table 5. Performance comparison of the SI baseline and proposed models using BLSTM in terms of WERs and relative improvements in WER. Similar to FCN, the WERs were kept being improved as the number of clusters increased in most cases.

FCN model	Group average	Ward linkage
	WER (rel. imp.)	
SI (baseline)	1.80 (-)	
5 clusters	1.67 (7.2%)	1.70 (5.6%)
7 clusters	1.69 (6.1%)	1.66 (7.8%)
10 clusters	1.69 (6.1%)	1.64 (8.9%)
20 clusters	1.70 (5.6%)	1.64 (8.9%)
40 clusters	1.66 (7.8%)	1.61 (10.5%)
Average	1.68 (6.6%)	1.65 (8.3%)

5. Conclusion

In this study, the performance of a large-scale speech recognition system was improved by clustering training data into similar speakers. Novel speaker clustering methods were proposed using the *i*-vector similarity metric and bottom-up hierarchical clustering, and retrained cluster-dependent models were used to improve overall speech recognition performance. Experimental results on the RM database showed that, Ward linkage clustering generated speaker subsets of balanced sizes, and showed more consistent word error rate improvements with the increase of the number of clusters. The set of deep neural network models (FCN and BLSTM) using the proposed clustering methods produced 12.2% and 10.5% relative performance improvements in terms of word error rates, respectively, over the baseline speaker independent model.

Acknowledgement

This work was supported by the Institute for Information and communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. R7124-16-0004, Development of Intelligent Interaction Technology Based on Context Awareness and Human Intention Understanding, 50%) and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. NRF-2017M3C1B6071400, 50%).

References

[1] C. J. Leggetter and P. C. Woodland, “Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models,” *Computer speech & language*, vol. 9, no. 4, pp. 171–185, 1995. Article (CrossRef Link)

[2] R. Kuhn, J.-C. Junqua, P. Nguyen, and N. Niedzielski, “Rapid speaker adaptation in eigenvoice space,” *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 6, pp. 695–707, 2000. Article (CrossRef Link)

[3] J.-L. Gauvain and C.-H. Lee, “Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains,” *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 2, pp. 291–298, 1994. Article (CrossRef Link)

[4] M. Senoussaoui, P. Kenny, N. Dehak, and P. Dumouchel, “An i-vector extractor suitable for speaker recognition with both microphone and telephone speech,” in *Proc. of the Odyssey*, 2010. Article (CrossRef Link)

[5] X. Fang, N. Dehak, and J. Glass, “Bayesian distance metric learning on i-vector for speaker verification,” in *Proc. of the INTERSPEECH*, pp. 2514–2518, August 2013. Article (CrossRef Link)

[6] G. E. Dahl, D. Yu, L. Deng, and A. Acero, “Large vocabulary continuous speech recognition with context-dependent DBN-HMMs,” in *Proc. of the IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, pp. 4688–4691, 2011. Article (CrossRef Link)

[7] D. Lee, M. Lim, H. Park, Y. Kang, J.-S. Park, G.-J. Jang, and J.-H. Kim, “Long short-term memory recurrent neural network-based acoustic model using connectionist temporal classification on a large-scale training corpus,” *China Communications*, vol. 14, Issue 9, pp. 23–31, September 2017. Article (CrossRef Link)

[8] M. Lim, D. Lee, H. Park, Y. Kang, J. Oh, J.-S. Park, G.-J. Jang, and J.-H. Kim, “Convolutional neural network based audio event classification,” *KSII Transactions on Internet and Information Systems*, vol. 12, no. 6, pp. 2748–2760, June 2018. Article (CrossRef Link)

[9] G. E. Dahl, D. Yu, L. Deng, and A. Acero, “Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, November 2012. Article (CrossRef Link)

[10] P. Price, W. Fisher, J. Bernstein, and D. Pallett, “Resource Management RM2 2.0,” *Philadelphia: Linguistic Data Consortium*, LDC93S3C, 1993. Article (CrossRef Link)

[11] M. Liu, B. Dai, Y. Xie, and Z. Yao, “Improved GMM-UBM/SVM for speaker verification,” in *Proc. of the IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, pp. 925–928, 2006. Article (CrossRef Link)

[12] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, “A study of inter-speaker variability in speaker verification,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 5, pp. 980–988, 2008. Article (CrossRef Link)

[13] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, “Front-end factor analysis for speaker verification,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, May 2011. Article (CrossRef Link)

[14] N. Dehak, R. Dehak, P. J. Kenny, N. Brummer, P. Dumouchel, and P. Ouellet, “Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification,” in *Proc. of the INTERSPEECH*, pp. 1559–1562, September 2009. Article (CrossRef Link)

[15] K. Veselý, A. Ghoshal, L. Burget, and D. Povey, “Sequence-discriminative training of deep neural networks,” in *Proc. of the INTERSPEECH*, 2013. Article (CrossRef Link)

[16] A. L. Caterini and D.-E. Chang, *Deep neural networks in a mathematical framework*, Springer, ISBN 978-3-319-75303-4, 2018. Article (CrossRef Link)

[17] M. J. F. Gales, “Semi-tied covariance matrices for hidden Markov models,” *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 272–281, May 1999. Article (CrossRef Link)

[18] M. J. Alam, V. Gupta, P. Kenny, and P. Dumouchel, “Speech recognition in reverberant and noisy environments employing multiple feature extractors and i-vector speaker adaptation,” *EURASIP Journal on Advances in Signal Processing*, vol. 2015, no. 50, pp. 1–13, 2015. Article (CrossRef Link)

[19] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, “The Kaldi speech recognition toolkit,” in *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, December 2011. Article (CrossRef Link)

[20] F. Seide, G. Li, and D. Yu, “Conversational speech transcription using context-dependent deep neural networks,” in *Proc. of the INTERSPEECH*, pp. 437–440, August 2011. Article (CrossRef Link)

[21] J. H. Ward Jr., “Hierarchical grouping to optimize an objective function,” *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 236–244, 1963. Article (CrossRef Link)

[22] L. Rokach and O. Maimon, “Clustering Methods,” *Data Mining and Knowledge Discovery Handbook, Springer, Boston, MA*, pp. 321-352, 2005. Article (CrossRef Link)

[23] Jungyu Ahn and Ju-Hong Lee, “Clustering algorithm for time series with similar shapes,” *KSII Transactions on Internet and Information Systems*, vol. 12, no. 7, pp. 3112–3127, 2018. Article (CrossRef Link)

[24] Keonsoo Lee, Chanki Moon and Yunyoung Nam, “Diagnosing vocal disorders using Cobweb clustering of the jitter, shimmer, and harmonics-to-noise ratio,” *KSII Transactions on Internet and Information Systems*, vol. 12, no. 11, pp. 5541–5554, 2018. Article (CrossRef Link)

[25] A. F. Martin and C. S. Greenberg, “The NIST 2010 speaker recognition evaluation,” in *Proc. of the INTERSPEECH*, pp. 2726–2729, September 2010. Article (CrossRef Link)

[26] D. Snyder, D. Garcia-Romero, and D. Povey, Kaldi SRE10 recipe, 2017. Article (CrossRef Link)

[27] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006. Article (CrossRef Link)

[28] D. Lee, J.-H. Park, K.-H. Kim, J.-S. Park, J.-H. Kim, G.-J. Jang, and U. Park, “Maximum likelihood-based automatic lexicon generation for AI assistant-based interaction with mobile devices,” *KSII Transactions on Internet and Information Systems*, vol. 11, no. 9, pp. 4264–4279, September 30, 2017. Article (CrossRef Link)

[29] K. Veselý and D. Povey, Kaldi RM recipe, 2019. Article (CrossRef Link)

Minsoo Kim is currently a Ph.D. student at the School of Electronics Engineering, Kyungpook National University, South Korea. He received his B.S. and M.S. degrees from the current affiliation, August 2016 and August 2018, respectively. He participated many government-funded projects including Development of Intelligent Interaction Technology Based on Context Awareness and Human Intention Understanding. His main role in that project is development of speaker diarization system. His research interests include speech recognition, speaker recognition, and many other diverse machine learning applications.

Gil-Jin Jang is an associate professor at Kyungpook National University, South Korea. He received his B.S. and M.S. degree in computer science from Korea Advanced Institute of Science and Technology (KAIST), Daejon, South Korea in 1997 and 1999 respectively. He also received his Ph.D. degree in the same department in February 2004. From 2004 to 2006 he was a research staff at Samsung Advanced Institute of Technology and from 2006 to 2007 he worked as a research engineer at Softmax, Inc. in San Diego. From 2008 to 2009 he joined Shiley Eye Center at University of California, San Diego as a postdoctoral scholar. From November 2009 to February 2014 he was an assistant professor at School of Electrical and Computer Engineering, Ulsan National Institute of Science and Technology (UNIST). Dr. Jang's research interests include deep learning and machine learning theories, acoustic signal processing, speech recognition and enhancement, computer vision, multimedia data analysis, and biomedical signal engineering.

Ji-Hwan Kim received the B.E. and M.E. degrees in Computer Science from Korea Advanced Institute of Science and Technology (KAIST) in 1996 and 1998 respectively and Ph.D. degree in Engineering from the University of Cambridge in 2001. From 2001 to 2007, he was a chief research engineer and a senior research engineer in LG Electronics Institute of Technology, where he was engaged in development of speech recognizers for mobile devices. In 2004, he was a visiting scientist in MIT Media Lab. Since 2007, he has been a faculty member in the Department of Computer Science and Engineering, Sogang University. Currently, he is a full professor. His research interests include spoken multimedia content search, speech recognition for embedded systems and dialogue understanding.

Minho Lee received the Ph.D. from Korea Advanced Institute of Science and Technology (KAIST) in 1995, and is currently a professor of School of Electronics Engineering and directors for AI Institute of Technology and KNU-LG Electronics Convergence Research Center, Kyungpook National University, Taegu, Korea. He established Mobile Technology Commercial Center at Daegu, and worked for Education \& Training Department as a director from 2005 to 2006. Also, he was a visiting scholar for Dept. of Brain and Cognitive Science at MIT from 2006 to 2007. He was president for Asia-Pacific Neural Network Assembly (APNNA) at 2013, and now he is a vice president and governing board member for Asia-Pacific Neural Network Society (APNNS) and International Neural Network Society (INNS) from 2017, respectively. He received several awards such as APNNA Excellent Service Award (2014) and Best Industry-Academic Cooperation Award (2014), and best paper awards at international conferences including AEARU (2015), ICONIP (2007 and 2009), IDEAL(2008), ICAISC(2006) and so on. He has been served for several international journals (Neural Networks and Neural Processing Letters, etc.) as an associate editor and for international conference as general chairs for ICONIP2013 and HAI2015 and program chairs for ICONIP2009, ICONIP2016 and ICONIP2019. His research interests include deep neural networks, brain-neuroinformatics, biologically inspired vision systems, human augmented cognition, selective attention, brain-machine interaction and natural language processing (home page: http://abr.knu.ac.kr).