ABSTRACT
Speaker variability has a significant impact to the state-of-the-art speech recognition systems. Traditionally speaker clustering is performed without considering individual or class phonetic similarities across different speakers. In fact, clustered speaker groups may have very different degrees of variations for different phonetic classes. In this paper, speaker clustering is performed at subword level or subphonetic level. With one or more instances derived from clustering for each subword or subphonetic unit, we model speaker variation explicitly across different subword or subphonetic instances. In addition, we select from massive possible combinations of speaker-clustered subword models to form our initial model for speaker adaptation. Experiments show that subword-dependent speaker clustering is more effective than the traditional speaker clustering.

1. INTRODUCTION
It is well known that the performance difference between a speaker-independent system and a speaker-dependent system is significant. Speaker adaptation [1] is effective to bridge the performance gap. However, speaker adaptation requires a fair amount of speaker-dependent data and a decent speaker independent model as an initial model. Speaker-clustered models offer improved speaker-independent performance since we can find a clustered model that is similar to the test speaker, which also results in improved speaker adaptation performance in terms of both accuracy and adaptation rate because of better initial models.

Speaker clustering has been studied considerably in the past. Gender-dependent models can be considered as primitive yet very successful speaker-clustered models. More sophisticated pre-clustering of training speakers [2,3]. Cluster Adaptive Training [4] and Eigenvoice [5] are examples related to speaker clustering. They all show that by providing a better starting point for adaptation, improved adaptation can be achieved.

Traditionally speaker clustering is performed without considering individual phonetic similarities across different speakers. Namely, one generally divides speakers into a few clusters based on some overall phonetic similarity measures on the speaker basis. For each group of speakers, one model is trained, albeit some speakers may have dramatically different acoustic realizations for certain phones. For example, Gao et al [2] used such an approach to build clustered models. For a test speaker, a subset of the clusters are identified and transformed to the test speaker’s acoustic space.

In this paper, we propose to apply the speaker clustering principle to more detailed levels. We cluster speakers together only if these speakers have a similar acoustic realization of the phonetic events. Thus, clustering is performed at the level of subword level or subphonetic level, which is of great importance to speech recognition systems. This is analogous to the argument of senone models in comparison to generalized triphone models [6].

Using multiple instances for each subword or subphonetic unit, we have more accurate representations of acoustic space for all speakers in the clustered models. Consequently, we can select from massive possible combinations of speaker-clustered subword instances to form a new initial model that is acoustically closer to the test speaker. Figure 1 is an illustration of initial model selection for a test speaker. The unit could be generalized triphone (subword) or senone (subphonetic), each unit has one or more instances depending on inter-speaker variation. For a given test speaker, for each unit, the closest-matched instance will be selected based on a little adaptation data from the speaker. All selected instances form a new set of model to be used as the initial model for speaker adaptation.

Figure 1. Illustration of model selection for a test speaker.
2. SUBWORD-DEPENDENT SPEAKER CLUSTERING

By training a separate model for a group of acoustically similar speakers, the distributions of the model can be sharper than the speaker independent models. A typical example is the widely used gender-dependent model. However, speaker clustering also creates potential problems. The following issues need to be addressed:

- **Data Fragmentation**: Partitioning speakers into clusters results in data fragmentation. Smoothing is usually needed.
- **Model Selection**: As we create multiple clusters we need to decide whether to use a cluster model for a test speaker.

For subword-dependent speaker clustering, the number of clusters for each subword unit can vary, based on the speaker variations among the training data. In fact, we can allocate more parameters to those units that have larger cross-speaker variations.

2.1. Parameter Allocation

For each subword unit, there may be multiple instances depending on the inter-speaker variation in the training data. If the inter-speaker variation is relatively small, we need a small number of parameters only to capture the inherent speaker variation. To decide how many instances we need for each unit, we have experimented with two methods.

**Bottom-Up Clustering using Likelihood Loss**

The bottom-up clustering has been widely used for tasks such as generalized triphone clustering or speaker clustering. It starts from a large number of instances for each unit, and merges a pair of instances that produce the least loss of likelihood over all the training data until the desired total number of instances is reached.

Assume we have two Gaussian density functions $G_1 = N(\mu_1, \Sigma_1)$ and $G_2 = N(\mu_2, \Sigma_2)$ with counts $c_1$ and $c_2$ respectively. When we merge these two Gaussian density functions together, the new Gaussian density has the following mean and variance:

$$c = c_1 + c_2$$

$$\mu = \frac{c_1 \mu_1 + c_2 \mu_2}{c}$$

$$\Sigma = \frac{c_1 \Sigma_1 + (\mu_1 - \mu)(\mu_1 - \mu)^T}{c} + \frac{c_2 \Sigma_2 + (\mu_2 - \mu)(\mu_2 - \mu)^T}{c}$$

The likelihood loss over the data resulted in by merging $G_1$ and $G_2$ is:

$$\Delta_{12} = \frac{c_1 \log |\Sigma_1| - c_1 \log |\Sigma| - c_2 \log |\Sigma_2|}{2}$$

Notice that we need to start from context-dependent (CD) subword unit models for clustering. There might not be enough data to train a set of CD model for each speaker. There are two possible ways to get around the problem:

1. Adapt the speaker-independent (SI) model to each speaker using MLLR [7].
2. Do traditional speaker clustering on context-independent (CI) models and group similar speakers together such that each group has just enough data to train a set of CD model.

**Scatter Matrix Measurement**

Alternatively, we can use scatter matrix [8] to measure the inter-speaker variation for each subword unit to determine how we should allocate the parameters. The motivation is that likelihood loss might be hardly comparable across different phonemes as it does not contain discriminative information.

The **within-class scatter matrix** and **between-class scatter matrix** for $L$ classes are defined as

$$S_w = \sum_{i=1}^{L} p_i \Sigma_i$$

$$S_b = \sum_{i=1}^{L} p_i (\mu_i - \mu_w)(\mu_i - \mu_w)^T$$

where $\mu_w$ is given by

$$\mu_w = \sum_{i=1}^{L} p_i \mu_i$$

The separability of classes (inter-speaker variation) is measured by

$$J = \text{tr}(S_w^{-1}S_b)$$

Once we determine the number of instances for each subword unit, we still need to use bottom-up clustering to partition speakers for that subword unit.

2.2. Model Smoothing

Due to data fragmentation, typically there is insufficient amount of data for each cluster. Thus, smoothing is very important. Two different smoothing can be used:

- **Variance tying**: Variance of different instances for same subword unit can be tied together. Therefore variances can be well trained.
- **MAP [9]** based smoothing. MAP can be used to smooth each cluster-dependent model with a speaker-independent model.

2.3. Clustered-Model Decoding

Speaker variations are explicitly modeled in the subword-dependent speaker-clustered models. We can directly use it to replace the general SI model for improved performance. All the instances are treated as parallel states. During decoding, for each subword unit, we need to evaluate all the instances and choose the instance that has the maximum likelihood.
2.4. Model Selection

With very little adaptation data from a test speaker, most context-dependent subword units are not observed. Therefore, with subword-dependent speaker clustering, selecting acoustically matched instance for a test speaker for each subword unit is critical.

For the subword units that appear in the test speaker’s data, it is straightforward to score the match. We can first decode the data using the speaker-clustered model (or speaker-independent model). Subsequently, based on the decoding transcription, segmentation can be obtained by Viterbi alignment. The likelihood or distance between the instance models and the acoustic data, conditioned on the Viterbi segmentation, can be calculated, and the instances can be ranked in the order of the score.

For the unseen subword units, our solution is to make use of co-occurrence information. We assume if two subword instances are observed together (co-occurring) or sharing a lot of common speakers, they are likely to be selected together as well. The definition of co-occurrence matrix is

\[ p_{ij} = p_{i} + p_{j} \]

where \( p_{ij} \) is co-occurrence probability for subword unit instance \( i \) and \( j \), \( p_{i} \) is the probability that instances \( i \) and \( j \) are both observed for the same speaker.

The co-occurrence matrix can be estimated by aligning all the training data using clustered-models. For each subword unit, the instance with highest likelihood is chosen as the winner. For each utterance, we can sort all the observations of the same subword unit and pick the overall winner instance for that unit. Then co-occurrence counts are calculated based on the sequence of all instance winners in the utterance.

As an alternative, the co-occurrence matrix can also be approximated by computing the overlap between the training speakers at different subword instances. Further assumption is made in this case that the more training speaker overlap it has, the stronger chance that the instances are observed together. This method does not require aligning any training data and therefore it is very computationally efficient.

\[ p_{ij} = \frac{2 \times \text{Spk}_i \times \text{Spk}_j}{\text{Spk}_i + \text{Spk}_j} \]

where \( \text{Spk}_i \) and \( \text{Spk}_j \) are the number of speakers in subword instance \( i \) and \( j \) respectively, \( \text{Spk}_i \) is the number of speakers appearing in both instance \( i \) and \( j \).

With co-occurrence matrix, for every subword instance, we can combine the scores derived from co-occurrence and from observed samples if any. A weight can be used to balance the information from observation and from pre-computed correlation.

\[ \text{Scr} = \text{Scr}_{\text{obs}} + \beta \times \text{Scr}_{\text{co-occurrence}} \]

\[ \beta \]

can be empirically determined. For each test speaker, we align the adaptation data using speaker-clustered models and accumulate scores for each subword instance based on above procedure. After going through all the data, we rank the scores of all instances for each subword unit and select the top one or more instances of each unit for the test speaker.

2.5. Adaptation

MLLR [7] can transform the model parameters toward test speaker’s acoustic space even with a very small amount of data. With subword-dependent speaker-clustered models, there are a few possibilities on applying MLLR transformations.

- Select one instance for each subword unit and form a new set of model. The size of the model will be the same as the SI model. Then apply MLLR.
- Select one or more instances for each subword unit based on a threshold. It is actually pruning the model based on test speaker’s data. The size of the model will be larger than SI model. Apply MLLR to all the instances of every subword unit in the model and keep all of them.
- Assume all instances of a subword unit share the same variance. Prune the speaker-clustered model and apply MLLR to all the means of instances. After that, interpolate the transformed means based on EM counts. The size of adapted model will be the same as SI model.

3. EXPERIMENTAL RESULTS

We focus experiments on speaker clustering at senone (shared HMM state) level but the technique can be generalized to other levels (e.g. phonetic level) easily.

Microsoft’s WHISPER speech recognition system [10] is used in our experiments. It processes 16kHz PCM data using a MEL-scale cepstrum along with its dynamics into a multi-dimensional feature vector. The acoustic model we used here is a simplified version – a set of HMMs with continuous-density output probabilities consisting of 3000 senones. A mixture of 10 Gaussian densities with diagonal covariances is used for each senone. The phonetic modeling in the system consists of position and context dependent within-word and crossword triphones.

We used Wall Street Journal database as our training corpus, which contains 284 speakers. Test set has about 420 sentences on business news from 20 speakers. The task is 60,000-word vocabulary continuous speech recognition with trigram trained on North American Business News (NAB) corpus.

3.1. Baseline and Speaker-Clustered Models

We first trained a baseline speaker-independent model using data from all the speakers. Each senone has a mixture of 10 Gaussians. The error rate is 7.76%. In addition, we built a traditional speaker-clustered model that partitioned speakers into six groups based on the overall distance from all phones. One model is trained for each group of speakers with MAP based smoothing. During testing, all models are decoded in parallel and one model is selected for each speaker based on likelihood. The error rate is 7.18%.
We trained our senone-dependent speaker-clustered (SDSC) models for various configurations. We found that scatter matrix based parameter allocation is better than likelihood loss based one. We suspect that it is partly due to the fact that we use single Gaussian model for clustering. With 6 instances per senone, the error rate is 6.61%, better than the baseline traditional speaker-clustering models.

3.2. Adapted Models

For simplicity, we used the co-occurrence matrix derived from speaker overlap for model selection. With about 2 minutes’ speech for each speaker, we experimented with pruning clustered model down to 1 and 2 instances per senone. In Table 1&2 you can find the results before and after MLLR adaptation.

Table 1. Results before and after adaptation with model selection, all final models have 1 instance per senone

<table>
<thead>
<tr>
<th>MODEL</th>
<th>#INSTANCE</th>
<th>WER</th>
<th>#INSTANCE</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>SI</td>
<td>1</td>
<td>7.76</td>
<td>1</td>
<td>6.76</td>
</tr>
<tr>
<td>SDSC</td>
<td>4</td>
<td>7.04</td>
<td>1</td>
<td>6.07</td>
</tr>
<tr>
<td>SDSC</td>
<td>6</td>
<td>6.93</td>
<td>1</td>
<td>6.10</td>
</tr>
<tr>
<td>SDSC</td>
<td>8</td>
<td>7.05</td>
<td>1</td>
<td>6.26</td>
</tr>
</tbody>
</table>

Table 2. Results with 2 instances per senone kept after selection.

<table>
<thead>
<tr>
<th>MODEL</th>
<th>#INSTANCE</th>
<th>WER</th>
<th>#INSTANCE</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDSC</td>
<td>4</td>
<td>6.89</td>
<td>2</td>
<td>5.87</td>
</tr>
<tr>
<td>SDSC</td>
<td>6</td>
<td>6.88</td>
<td>2</td>
<td>5.72</td>
</tr>
<tr>
<td>SDSC</td>
<td>8</td>
<td>6.80</td>
<td>2</td>
<td>6.05</td>
</tr>
</tbody>
</table>

We also compared the performance on adaptation starting from our senone-dependent speaker-clustered models with starting from the traditional speaker-clustered models. From Table 3 you can see that senone-dependent models provide better adaptation performance.

Table 3. Comparison of speaker-adaptation starting from different speaker-clustered models.

<table>
<thead>
<tr>
<th>MODEL</th>
<th>#INSTANCE</th>
<th>WER</th>
<th>#INSTANCE</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDSC</td>
<td>6</td>
<td>6.93</td>
<td>1</td>
<td>6.10</td>
</tr>
<tr>
<td>Baseline</td>
<td>6</td>
<td>7.24</td>
<td>1</td>
<td>6.36</td>
</tr>
</tbody>
</table>

4. SUMMARY

Subword-dependent speaker-clustered models provides more flexibility at customizing a new model for a test speaker due to its massive number of possible combinations of subword instance models. The experiments show that we can reduce the size of a model through model selection and achieve better before-adaptation performance because the model is customized toward test speaker’s acoustic space.

Experiments also show that the gain brought by a better initial model typically holds after adaptation. It means a good starting point for adaptation can boost the effectiveness of speaker adaptation. The best adaptation performance starting from senone-dependent speaker-clustered model is 10% better than starting from a SI model (6.76%) with the same parameter size in the adapted model. In addition, our senone-dependent speaker clustering slightly outperforms the traditional speaker clustering.

Data fragmentation is the biggest problem for speaker clustering. We believe when we have very large amount of data, speaker clustering will show more potential. Further work is needed on how to customize it toward test speaker’s acoustic space more quickly and accurately.

5. REFERENCES