A COMPARATIVE STUDY ON SYSTEM COMBINATION SCHEMES FOR LVCSR

Chengyuan Ma²*, Hong-Kwang Jeff Kuo¹, Hagen Soltau¹, Xiaodong Cui¹, Upendra Chaudhari¹, Lidia Mangu¹, and Chin-Hui Lee²

¹IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA
²School of ECE, Georgia Institute of Technology, Atlanta, GA 30332, USA
cyma@ece.gatech.edu, hkuo@us.ibm.com

ABSTRACT

We present a comparative study on combination schemes for large vocabulary continuous speech recognition by incorporating long-span class posterior probability features into conventional short-time cepstral features. System combination can improve the overall speech recognition performance when multiple systems exhibit different error patterns and multiple knowledge sources encode complementary information. A variety of combination approaches are investigated in this paper, e.g., feature concatenation single stream system, model combination multi-stream system, lattice rescoring and ROVER. These techniques work at different levels of a LVCSR system and have different computational cost. We compared their performance and analyzed their advantages and disadvantages on large vocabulary English broadcast news transcription tasks. Experimental results showed that model combination with independent tree consistently outperforms ROVER, feature concatenation and lattice rescoring. In addition, the phoneme posterior probability features do provide complementary information to short-time cepstral features.

Index Terms— system combination, feature concatenation, multi-stream, model combination, lattice rescoring, ROVER

1. INTRODUCTION

Short-time cepstral features like MFCC and PLP are the most widely used acoustic features in automatic speech recognition (ASR), which are based on local spectral envelop estimation within a window of 25 or 30 ms. However, the fine temporal structure of speech signal is lost in short-time spectral analysis. The temporal context is informative and critical for large vocabulary continuous speech recognition (LVCSR). Therefore, dynamic features (first and second order derivatives of the static cepstral features) are usually employed to represent the time evolution trajectories of the spectrum [1]. Some alternatives have been proposed to capture more context information and improve the discriminative capability of cepstral features. The left and right context frames of a central frame are stacked together first and then linearly transformed to a new feature space by principal component analysis (PCA), linear discriminant analysis (LDA) or heteroscedastic linear discriminant analysis (HLDA) [2]. By contrast, tandem acoustic modeling [3] and TempoRAI Pattern (TRAP) [4] feature use multi-layer perceptrons (MLP) to discriminatively transform raw feature vectors into class (phonemes or articulatory attributes) posterior probability vectors by a nonlinear mapping. Tandem modeling uses 9 consecutive cepstral feature vectors, which roughly cover 100 ms speech signal, while TRAP features are computed from a much longer window (300ms - 1s), which represents the long-span log energy trajectories of each critical band.

The MLP features have some advantages over conventional short-time cepstral features: (1) they capture more context information; (2) they are discriminative features because of the discriminative modeling nature of a MLP. However, these MLP features alone never consistently outperform conventional cepstral feature on LVCSR tasks [5]. They have been shown to improve performance when used in conjunction with cepstral features [5] [6]. So the intention of this paper is to investigate and compare several typical combination schemes that integrate short-time cepstral features and long-span MLP posterior probability features. A variety of combination schemes exist in the literature of speech recognition, e.g., recognizer output voting error reduction (ROVER) [7] [8], feature concatenation [6], multi-stream model combination [9] [10] and lattice rescoring [11]. These techniques are usually used at different levels of a LVCSR system. Previous research reported performance improvements using one or two of these schemes on different tasks [5] [6] [11].

In this paper, we conducted a comparative evaluation of all these typical system combination schemes on a large vocabulary broadcast news transcription task. The effectiveness, advantages and computational cost of each of these approaches have been investigated and analyzed. This paper is not intended to cover all system combination techniques that can be found in literature. We can draw some conclusions from our experiments: (1) Even though the MLP feature alone is 5.9% worse than PLP feature, they do encode complementary information to PLP features; (2) Model combination with independent tree is an effective combination scheme, which consistently outperforms ROVER, feature concatenation and lattice rescoring.

2. COMBINATION SCHEMES

Four typical combination schemes are briefly described in the following sections. The feature streams investigated in this paper are denoted by \( \phi_L \) (PLP features), \( \phi_{MFCC} \) (MFCC features) and \( \phi_{MLP} \) (MLP-based phoneme posterior probability features) respectively for time instance \( t \).

2.1. Feature concatenation and single-stream system

A simple and straightforward way of system combination is to augment the conventional PLP features with the MLP posterior features at the input to a HMM system, i.e., \( \phi^t = [\phi^t_L, \phi^t_{MLP}] \), which is a time-synchronous early fusion scheme. Then a single stream HMM-based ASR system is built in the conventional way and all feature streams are modeled jointly. The output probability distribution of
state \( j \) at time \( t \) is usually represented by a Gaussian mixture model (GMM) as shown in Eq. (1):

\[
b_j(\mathbf{o}^t) = \sum_{m=1}^{M_j} c_{jm} \mathcal{N}(\mathbf{o}^t; \mathbf{\mu}_{jm}, \mathbf{\Sigma}_{jm}),
\]

(1)

where \( M_j \) is the number of mixture components of state \( j \), \( c_{jm} \) is the weight of the \( m \)th component of state \( j \) and \( \mathcal{N}(\cdot; \mathbf{\mu}, \mathbf{\Sigma}) \) is a multivariate Gaussian with mean vector \( \mathbf{\mu} \) and covariance matrix \( \mathbf{\Sigma} \).

Feature concatenation usually results in a high-dimensional feature vector \( \mathbf{o}^t \). Sometimes it is too large to model all feature streams efficiently. When covariance matrices \( \mathbf{\Sigma} \) are diagonal matrices (and they usually are), each state \( j \) has \( M_j (2 \text{dim}(\mathbf{o}^t) + 1) \) parameters to be estimated, where \( \text{dim}(\mathbf{o}^t) \) is the dimension of vector \( \mathbf{o}^t \).

2.2. Model combination and multi-stream system

Model level combination is generally formulated as a multi-stream system, which is more flexible and enables separate modeling of multiple information sources, permits different number of states and different number of Gaussian components in each stream. Likelihood scores from all feature streams are combined by weighted linear functions. A basic assumption on multi-stream systems is that given the state \( q^t \) at any time \( t \), the feature vectors from each of these streams are statistically independent from all other streams. So in the decoding stage, the state output probability distribution is factorized as follows:

\[
b_j(\mathbf{o}^t) = \prod_{s=1}^{S} \left[ \sum_{m=1}^{M_j} c_{jms} \mathcal{N}(\mathbf{o}^t; \mathbf{\mu}_{jms}, \mathbf{\Sigma}_{jms}) \right]^{w_{js}},
\]

(2)

where \( S \) is the number of streams in a multi-stream system and the exponent \( w_{js} \) is a state-dependent weight for state \( j \) of stream \( s \) or simply a shared state-independent stream weight \( w_s \). It is used to control the contribution from each of these feature streams and indicate our confidence on each feature stream.

When all covariance matrices are diagonal matrices, each state \( j \) has \( \sum_{s=1}^{S} M_j s (2 \text{dim}(\mathbf{o}^t) + 1) \) parameters to be estimated and \( b_j(\mathbf{o}^t) \) can be equivalently represented as a single-stream system with \( \prod_{s=1}^{S} M_j s (2 \text{dim}(\mathbf{o}^t) + 1) \) parameters. Therefore, multi-stream system can model the feature streams more accurately than single-stream system with similar number of parameters.

Several approaches have been proposed to estimate the parameters of multi-stream systems [9], which consists of two parts: estimation of HMM parameters for each stream and estimation of appropriate stream exponents. For instance, parameters of each stream can be estimated independently such that we have separate models for each feature stream, which means different streams will have independent decision trees and HMM model sets. For the optimal stream weight estimation, there are some techniques discussed in [9] [12] and it is beyond the intention of this paper.

A computationally efficient model combination scheme [9] is employed in this paper for LVCSR tasks, where a primary feature stream (e.g., PLP features) is chosen to build the phonetic decision trees and the initial maximum likelihood models. Then a single-pass retraining is carried out to estimate the HMM model parameters for each stream based on the state occupation probabilities accumulated from the primary feature stream. One advantage of this approach is that only one decision tree and one decoding graph are needed. The stream weights are set to equal for all feature streams in this paper, which had been demonstrated to be as good as state-dependent weights in [9].

2.3. Lattice rescoring

A lattice is a compact representation of the competing hypotheses generated by a decoder. It can be expanded by additional acoustic and language model scores. Usually the lattice oracle word error rate is much smaller than the best path word error rate, which means that it is possible to improve recognition accuracy by re-ranking these competing hypotheses using complementary knowledge sources.

The MLP phoneme (or articulatory attribute) posterior probability was used to rescore word lattices [11]. We followed the exact lattice rescoring scheme proposed in [11]. The phoneme posterior probability is added to the conventional acoustic score by a weighted linear function for each hypothesized word, which corresponds to an arc in the lattice. The rescoring formula used in [11] can be reformulated as Eq. (3):

\[
b_j(\mathbf{o}^t) = \left[ b_j(\mathbf{o}_{\text{MLP}}^t) \right]^{w_1} \left[ f_j(\mathbf{o}_{\text{MLP}}^t) \right]^{w_2},
\]

(3)

where \( f_j(\mathbf{o}_{\text{MLP}}^t) \) simply outputs the phoneme posterior probability corresponding to the hypothesized phoneme at time \( t \) based on the aligned state information.

By this reformulation, the difference between lattice rescoring and model combination multi-stream system is clear: in lattice rescoring, the phoneme posterior probability vector MLP is directly used in score combination, while in model combination system, it is integrated through a probability distribution.

2.4. ROVER

Recognizer output voting error reduction (ROVER) [7] is a post-processing scheme for multiple ASR system combination. The rationale behind ROVER is that multiple ASR systems usually exhibit different error patterns, which means that simple majority voting can achieve a lower word error rate than any of the individual systems. The outputs of multiple ASR systems are aligned into a word transition network (WTN) by dynamic programming and then a majority voting is performed for each correspondence set. In contrast to other combination schemes, ROVER works at the output of multiple ASR systems and no acoustic and language models are involved at this stage.

3. MLP TRAINING AND MLP FEATURE EXTRACTION

A MLP is trained on a set of 3-hour broadcast news speech (2.5-hour for training and 0.5-hour for cross-validation), which is separate from the data sets used in LVCSR experiments. The raw feature used as input of MLP is the TRAP feature, which is the temporal trajectory of log energy from 31 consecutive frames (±15 frames of a central frame) for each critical band. Band-wise mean and variance normalization was also conducted. Then discrete cosine transformation (DCT) is applied to each band to perform dimension reduction and de-correlation. Only the first 10 DCT coefficients and the log-energy of each band are preserved in our experiments. There are 23 critical bands used for 16k Hz speech data. TRAP features were extracted using the trapper software1.

The topology of the MLP has 253 input units, 800 hidden units and 44 output units that corresponds to 44 phonemes used in LVCSR experiments. The total number of the parameters of the MLP is about 240k. The MLP was trained using the ICSI’s QuickNet toolkit2.

1http://speech.fit.vutbr.cz/files/software/trapper.html
2http://www.icsi.berkeley.edu/Speech/qn.html
Table 1. WERs of feature concatenation.

<table>
<thead>
<tr>
<th>Feature Stream</th>
<th>WER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLP</td>
<td>30.8</td>
</tr>
<tr>
<td>MFCC</td>
<td>31.0</td>
</tr>
<tr>
<td>TRAPMLP</td>
<td>36.7</td>
</tr>
<tr>
<td>PLP-TRAPMLP</td>
<td>32.2</td>
</tr>
</tbody>
</table>

Table 1, RAPMLP indicates the TRAP MLP feature streams and PLP-TRAPMLP indicates the concatenated feature stream from PLP and TRAPMLP streams. We observed that the best single system is with PLP features and MFCC can achieve comparable results. TRAPMLP system is worse in WER by 5.9%, which is consistent with previous research [5] [6]. The MLP used in this paper was trained with very limited data (2.5-hour), so performance is expected to improve when trained on more data. PLP-TRAPMLP is worse than PLP by 1.4% in WER but better than TRAPMLP by 4.5%.

4.2. Model combination with independent tree

This experiment is to evaluate the model combination scheme, where each stream has its independent decision tree and HMM models as in section 4.1. The WERs for each feature stream is shown in Table 1. Table 2 summarizes the WERs for both model combination and ROVER, where “+” means model combination. We observed that multi-stream decoding is better than ROVER and always get improvements over the PLP baseline system. In addition, the best combination (PLP + MFCC + PLP-TRAPMLP) is 1.9% better in WER than best single system (PLP).

Table 2. WER of independent tree model combination.

<table>
<thead>
<tr>
<th>Feature Stream Combination</th>
<th>WER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLP + TRAPMLP</td>
<td>30.7</td>
</tr>
<tr>
<td>PLP + MFCC + TRAPMLP</td>
<td>29.8</td>
</tr>
<tr>
<td>ROVER</td>
<td>30.1</td>
</tr>
<tr>
<td>PLP + MFCC + PLP-TRAPMLP</td>
<td>28.9</td>
</tr>
<tr>
<td>ROVER</td>
<td>29.7</td>
</tr>
</tbody>
</table>

4.3. Model combination with shared TRAPMLP tree

In this experiment, the decision tree of TRAPMLP was shared by all feature streams. The HMM models for each stream is obtained by a single pass retraining procedure [9]. From Table 3, we observed that combining three models with shared TRAPMLP tree (32.2%), we get a nice improvement (1.8%) ER over the best single component model (PLP 34.0%). However, with this tree, the PLP model is sub-optimal: 3.2% worse than a decision tree trained using PLP (30.8%) in Table 1.

Table 3. WER of shared MLP tree.

<table>
<thead>
<tr>
<th>Feature Stream Combination</th>
<th>WER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRAPMLP</td>
<td>36.7</td>
</tr>
<tr>
<td>PLP</td>
<td>34.0</td>
</tr>
<tr>
<td>MFCC</td>
<td>34.2</td>
</tr>
<tr>
<td>PLP + MFCC</td>
<td>34.0</td>
</tr>
<tr>
<td>TRAPMLP + PLP + MFCC</td>
<td>32.2</td>
</tr>
</tbody>
</table>
4.4. Model combination with shared PLP tree

In this experiment, the decision tree of PLP was shared by all feature streams [9]. It is clear that the MFCC, TRAPMLP and PLP-TRAPMLP systems are worse than that with their independent trees. In addition, combining three models (PLP + MFCC + TRAPMLP) with shared PLP tree didn’t get any improvement over the PLP baseline system. However, combining three models (PLP + MFCC + PLP-TRAPMLP) achieved an improvement about 1.3%.

Table 4. WER of shared PLP tree.

<table>
<thead>
<tr>
<th>Combination</th>
<th>WER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLP</td>
<td>30.8</td>
</tr>
<tr>
<td>MFCC</td>
<td>31.7</td>
</tr>
<tr>
<td>TRAPMLP</td>
<td>38.6</td>
</tr>
<tr>
<td>PLP-TRAPMLP</td>
<td>32.8</td>
</tr>
<tr>
<td>PLP + MFCC + TRAPMLP</td>
<td>31.2</td>
</tr>
<tr>
<td>ROVER</td>
<td>30.7</td>
</tr>
<tr>
<td>PLP + MFCC + PLP-TRAPMLP</td>
<td>29.5</td>
</tr>
<tr>
<td>ROVER</td>
<td>30.2</td>
</tr>
</tbody>
</table>

4.5. Lattice rescoring

Lattice rescoring experiments were conducted on speaker adapted systems. We evaluated the combined systems with several configurations, which correspond to some pairs of $w_1$ and $w_2$. The experiments results are shown in Table 5. The baseline system has a WER of 24.2% when $w_1=0.055$ (which is the acoustic score scaling factor used in LVCSR experiments) and $w_2=0$. When both $w_1$ and $w_2$ are 0, only the language model score was used to find the best path and can achieve a WER of 39.2%. When only the MLP posterior score was used to find the best path ($w_1=0,w_2=0.05$), WER is about 34.8%. When both $w_1$ and $w_2$ are set appropriately, we observed a very slight (0.2%) improvement over the PLP baseline system.

Table 5. WER of lattice rescoring.

<table>
<thead>
<tr>
<th>$w_1$</th>
<th>$w_2$</th>
<th>WER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>39.2</td>
</tr>
<tr>
<td>0.0</td>
<td>0.05</td>
<td>34.8</td>
</tr>
<tr>
<td>0.05</td>
<td>0.00</td>
<td>24.2</td>
</tr>
<tr>
<td>0.05</td>
<td>0.005</td>
<td>24.0</td>
</tr>
</tbody>
</table>

Some observations from lattice rescoring experiments are: (1) phoneme recognition accuracy could be critical to the effectiveness of lattice rescoring. In our experiments, the MLP based phone recognizer achieved a frame-wise accuracy about 59% on broadcast news; (2) Because the MLP features were obtained from a long-span TRAP features, the phoneme boundaries on the posteriogram as shown in Fig. 1 are not sharp, which cause some errors in lattice rescoring.

5. CONCLUSION

This paper described a comparative study on system combination schemes for LVCSR tasks. Long-span MLP posterior probability features and conventional short-term cepstral feature are combined using 4 typical combination schemes. Even though the MLP features by themselves are 5.9% worse than PLP, when they are employed in conjunction with conventional cepstral features using model combination with independent tree, we can get a 1.9% improvement. Simple feature concatenation scheme doesn’t work in our experiments and lattice rescoring can achieve a very slight improvement. We observed that multi-stream decoding is better than ROVER. The data set used for MLP training is relative small in our experiments, performance is expected to improve when trained on more data. It would also be interesting to incorporate the MLP features with discriminatively trained speaker adapted systems in the future.

6. ACKNOWLEDGEMENT

The authors would like to thank Vaibhava Goel and Brian Kingsbury for their help in this study and the first author is grateful to Sabato Marco Siniscalchi for his help on lattice rescoring. This work was partially funded by DARPA under Grant HR0011-06-2-0001. The views, opinions, and/or findings contained in this article/presentation are those of the author/presenter and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.

7. REFERENCES