INCORPORATING DISCRIMINATIVE DPGMM POSTERIORGRAMS FOR LOW-RESOURCE ASR

Bin Wu[1], *Sakriani Sakti*[1,2], *and Satoshi Nakamura*[1,2]

[1]Nara Institute of Science and Technology, Japan
[2]RIKEN, Center for Advanced Intelligence Project AIP, Japan

ABSTRACT

The first step in building an ASR system is to extract proper speech features. The ideal speech features for ASR must also have high discriminabilities between linguistic units and be robust to such non-linguistic factors as gender, age, emotions, or noise. The discriminabilities of various features have been compared in several Zerospeech challenges to discover linguistic units without any transcriptions, in which the posteriorgrams of DPGMM clustering show strong discriminability and get several top results of ABX discrimination scores between phonemes. This paper appends DPGMM posteriorgrams to increase the discriminability of acoustic features to enhance ASR systems. To the best of our knowledge, DPGMM features, which are usually applied to such tasks as spoken term detection and zero resources tasks, have not been applied to large vocabulary continuous speech recognition (LVCSR) before. DPGMM clustering can dynamically change the number of Gaussians until each one fits one segmental pattern of the whole speech corpus with the highest probability such that the linguistic units of different segmental patterns are clearly discriminated. Our experimental results on the WSJ corpora show our proposal stably improves ASR systems and provides even more improvement for smaller datasets with fewer resources.

Index Terms— ASR, DPGMM, zerospeech, discrimination

1. INTRODUCTION

ASR seeks a sequence of linguistic units such as phonemes and words for each speech utterance. One critical issue that affects its performance is speech feature extraction. Such acoustic features as Mel-Frequency Cepstrum Coefficients (MFCC) [1] and Perceptual Linear Prediction (PLP) [2] extract smooth formant envelopes, mimic non-linear auditory properties, and work well in ASR systems [3].

MFCC, which is widely used as the default feature for ASR systems [4, 5], accurately grasps the temporal spectral properties of each phoneme. However, MFCC relatively weakly discriminates phonemes [6, 7] because the MFCC of one phoneme does not consider to further contrast itself with the phonemes of other categories with similar acoustics (e.g., /p/ in contrast with /b/ and /d/) in different utterances.

The ideal speech features for ASR also require sufficient ability to discriminate between such linguistic units as phonemes or words and should be robust to such non-linguistic factors as gender, emotions, and noise.

The quantification of discriminability between linguistic units was recently developed [8]. Various features [9, 10, 11] were compared in several Zerospeech challenges to discover linguistic units without transcriptions and evaluated by the ABX discrimination test [8], which measures ability to discriminate the phonemes.

The proposed features in Zerospeech include such acoustic features as MFCC or PLP [8], transformed features from neural representation learning by autoencoder [12, 13, 14], neural discriminative training by ABnet [15], neural discretized learning by VQ-VAE [16], traditional clustering by GMM [16] or k-means [17, 16], and nonparametric clustering by the Dirichlet Process Gaussian Mixture Model (DPGMM) trained with Gibbs sampling [18] or variational inference [19, 20]. Among them, DPGMM achieved the top performance in the ABX discrimination test at the Zerospeech challenges of 2015, 2017, and 2019 [6, 7, 21].

DPGMM clustering can discriminate phonemes well because it dynamically changes the number of Gaussians until each one fits one segmental pattern of the whole speech corpus with the highest probability such that the linguistic units of different segmental patterns are clearly discriminated.

In the field of speech processing, DPGMM was initially applied in spoken term detection [18] and later in Zerospeech challenges [7], suggesting its ability to discriminate linguistic units. To the best of our knowledge, DPGMM features have not been applied to large vocabulary continuous speech recognition (LVCSR) before. Inspired by DPGMM's relatively strong discriminability, we applied it to an LVCSR system by concatenating acoustic features with DPGMM posteriorgrams such that the concatenated features combine the power of both to enhance the ASR system.

Thanks to JSPS KAKENHI Grant Numbers JP17H06101 and JP17K00237 for funding.

2. RELATED WORKS

Not only the features themselves but also their transformations are used to improve ASR. Some transformations append deltas [22] to features by taking the orders of derivatives, and others reconstruct features by applying self-supervised learning [23]. They grasp the temporal structure to improve the performance. Other transformations combine different features, such as an MFCC concatenated with an PLP [24] and an MFCC concatenated with a posteriorgram [25, 26] from neural networks, which are often used for combining the merits of different features.

A scheme that resembles our proposal in enhancing feature discriminability for ASR is the tandem system [26], which uses posteriorgrams obtained from neural networks targeted at phonemes or states with supervised learning. The tandem approach needs the alignments of phonemes or states and a large amount of data for training neural networks. Such accurate alignments or rich data resources are often unavailable. DPGMM gets posteriorgrams with unsupervised clustering, which is robust in a small amount of data, implying the promise of our proposal for a low-resource ASR.

Our work is also different from transforming features by appending deltas [22] or reconstructing features by applying self-supervised learning [23], both of which model the absolute or statistical local temporal structure. DPGMM clustering globally searches for distinct segment patterns over all the acoustic features of the whole speech corpus.

3. PROPOSED APPROACH

3.1. DPGMM Clustering

We can view each frame of a speech feature as one sample generated by a Gaussian Mixture Model (GMM) for the following reasons. Theoretically, a GMM has the power to model any distribution, especially spherical or elliptical ones with multiple local modes; practically, the GMM can fit the spectrum feature, as done in HMM-GMM speech recognition systems.

The Dirichlet Process Gaussian Mixture Model (DPGMM) is a nonparametric Bayesian version of GMM. The number of clusters K is learned from data nonparametrically (the number of parameters can grow with the data size). In the Bayesian world, our parameters are no longer unknown constants; they are random variables with certain distributions.

We can treat DPGMM as an infinite GMM with density function $p(x_i) = \sum_{k=1}^{\infty} \pi_k p(x_i|\mu_k, \Sigma_k)$ (alternatively, $p(x_i) = \sum_{k=1}^{\infty} p(Z_i = k)p(x_i|Z_i = k)$).

This generative model (Fig. 1) is defined by the following procedures. It samples mixture weights $\{\pi_k\}_{k=1}^{\infty}$ from the stick-breaking process [27] (with concentration parameter α) and the means and variances $\{\mu_k, \Sigma_k\}_{k=1}^{\infty}$ from the normal-inverse-Wishart (NIW) distribution (with the belief of mean μ_0, the belief of variance Σ_0, the belief-strength of mean λ, and the belief-strength of variance ν). It also samples Gaussian cluster indicator hidden variable Z_i by mixture weights and each data point X_i by the Gaussian cluster indicated by Z_i. We summarize this sampling procedure by describing the dependency relation of the random variables of the joint distribution of model $\mathrm{DPGMM}(\alpha, \mathrm{NIW}(\mu_0, \lambda, \Sigma_0, \nu))$ in Fig. 1.

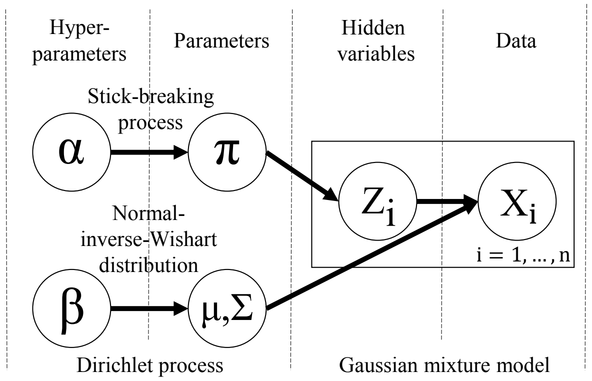

Fig. 1: Graphical model of Dirichlet Process Gaussian Mixture Model (DPGMM): We generated parameters of weights ($\pi = \pi_1, \ldots, \pi_k, \ldots$), means, and variances ($(\mu, \Sigma) = (\mu_1, \Sigma_1), \ldots, (\mu_k, \Sigma_k), \ldots$) for Gaussians from stick-breaking process (with concentration parameter α) and normal-inverse-Wishart distribution (with parameter $\beta = (\mu_0, \lambda, \Sigma_0, \nu)$) respectively. We then generated each frame of speech feature X_i of data $X = X_1, \ldots, X_n$ by first sampling one Gaussian with mean μ_k and variance Σ_k indicated by hidden variable $Z_i = k$ according to weights and sampling X_i from that Gaussian cluster. The box, with (Z_i, X_i) inside, is a simplified notation of all n data points (features) with their indicator hidden variables $((Z_1, X_1), \ldots, (Z_i, X_i), \ldots, (Z_n, X_n))$.

3.2. DPGMM Clustering to Generate Posteriorgrams

Given the model definition and data $\{x_i\}_{i=1}^{n}$, we infer from Gibbs sampling (Algorithm 1) to get posteriorgram $p(z_i|x_i)$.

First, we update the weights by sampling from a Dirichlet distribution:

$$\pi_1, \cdots, \pi_K, \pi_{K+1}^{*}|z, \alpha \sim \mathrm{Dir}(n_1, n_2, \cdots, n_K, \alpha)\,, \quad (1)$$

where K is the number of the clusters of the currently observed data, $\pi_{K+1}^{*} = \sum_{k=K+1}^{\infty} \pi_k$ is the sum of the weights for the future possible clusters, and $n_k = \sum_{i=1}^{n} \delta(z_i = k)$ is the number of data points in cluster k, counted by hidden indicator variables $z = z_1, \ldots, z_n$.

Second, we update the mean and the variance for each Gaussian cluster k by sampling a normal-inverse-Wishart dis-

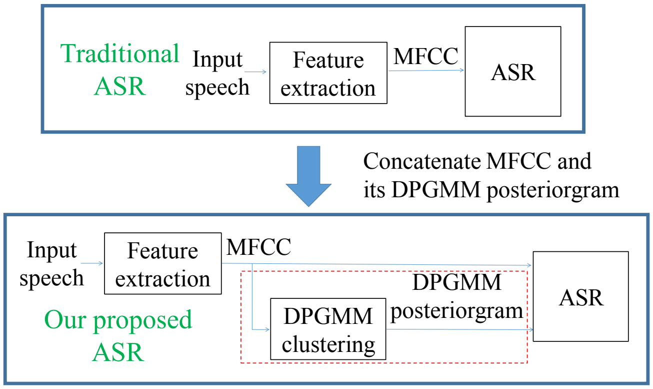

Fig. 2: Construction of our proposed ASR system from traditional ASR: First, we do DPGMM clustering on the acoustic feature (MFCC) to get a DPGMM posteriorgram, highlighted by a red rectangle; then we replace the MFCC feature of the traditional ASR system with the concatenation of the MFCC and its posteriorgram. Concatenated features are expected to improve ASR systems.

Algorithm 1 Gibbs sampling for DPGMM (Fig. 1) given hyperparameters α and β and observed data x

```
Randomly initialize cluster indicator $z = z_1, ..., z_n$
for Iteration $iter = 1, 2, \ldots$ do
    Sample $\pi' \sim p(\pi|z, \alpha)$ by Eq. (1),
      $\pi_1, \cdots, \pi_K, \pi^*_{K+1}|z, \alpha \sim \mathrm{Dir}(n_1, n_2, \cdots, n_K, \alpha)$
    Sample $\mu', \Sigma' \sim p(\mu, \Sigma|z, \beta, x)$ by Eq. (2),
      $\mu_k, \Sigma_k|z, \beta, x \sim \mathrm{NIW}(\mu_0^{(k)}, \lambda^{(k)}, \Sigma_0^{(k)}, \nu^{(k)})$
    Sample $z'_i \sim p(z_i|\pi', \mu', \Sigma', x_i)$ by Eq. (3),
      $z_i|\pi, \mu, \Sigma, x_i \sim \pi_k p(x_i|\mu_k, \Sigma_k)/p(x_i)$
    Update $z = (z'_1, ..., z'_n)$.
end for
```

tribution [28] after observing data x:

$$\mu_k, \Sigma_k|z, \beta, x \sim \mathrm{NIW}(\mu_0^{(k)}, \lambda^{(k)}, \Sigma_0^{(k)}, \nu^{(k)})\,, \tag{2}$$

where $\mu_0^{(k)}, \lambda^{(k)}, \Sigma_0^{(k)}, and \nu^{(k)}$ are the updated parameters for the k-th cluster after seeing the data [28].

Third, we update the hidden variables by sampling the posterior distribution:

$$p(z_i = k|\pi, \mu, \Sigma, x_i) = \frac{\pi_k p(x_i|\mu_k, \Sigma_k)}{p(x_i)} \propto \pi_k p(x_i|\mu_k, \Sigma_k)\,. \tag{3}$$

3.3. Concatenating DPGMM Posteriorgrams with MFCC Features

Compared with a traditional ASR system which directly extracts such acoustic features as MFCC for recognition tasks, our proposal applies the DPGMM clustering algorithm on the acoustic features, gets the unsupervised DPGMM posteriorgrams and concatenates the DPGMM posteriorgrams with the MFCC features as enhanced features for the ASR system (Fig. 2).

Table 1: Hyperparameters for encoder-decoder ASR and DPGMM: Notion D is number of dimensions of MFCC features.

Model	Parameters	Value
ASR	Dropout probability	0.25
	Label-smoothing ratio	0.05
	Learning rate	0.001
	Beam size	10
DPGMM	Concentration parameter	1
	Belief-strength of mean	1
	Belief-strength of variance	$D+2$
	Belief of mean	Feature mean
	Belief of variance	Feature variance
	Number of iterations	1500

Before concatenation, we applied Cepstral Mean and Variance Normalization (CMVN) to the MFCC features to reduce the feature distortion by noise contamination, which lowers the number of DPGMM clusters. Finally, we got 99 clusters with 99-dimensional DPGMM posteriorgrams in our experiment. Although the dimensions of the DPGMM posteriorgrams are relatively high, the probabilities are usually concentrated on one or two dimensions for each frame, and most of the other dimensions are zeros. MFCC is full of acoustic details in all the dimensions, but the DPGMM posteriorgram is discriminative with few dimensions; they complement each other in feature combinations.

Since the DPGMM posteriorgram satisfactorily discriminates the phonemes evaluated by the ABX discrimination test [7, 6], we show that combining an MFCC feature and its DPGMM posteriorgram improves the ASR performance.

4. DATASET AND EXPERIMENT SETUP

4.1. Dataset

We analyzed the MFCC features and their DPGMM posteriorgrams of an example utterance from TIMIT [29], which is an English corpus of read speech. TIMIT is suitable for analysis because it includes reliable and detailed phoneme annotations.

We checked whether our proposal that concatenates acoustic features with their posteriorgrams can improve the ASR on the WSJ corpus [30], a commonly used English corpus for ASR tasks on spontaneous speech. All experiments followed the same division of training, development, and test sets of the ASR examples of the TIMIT and WSJ corpora of

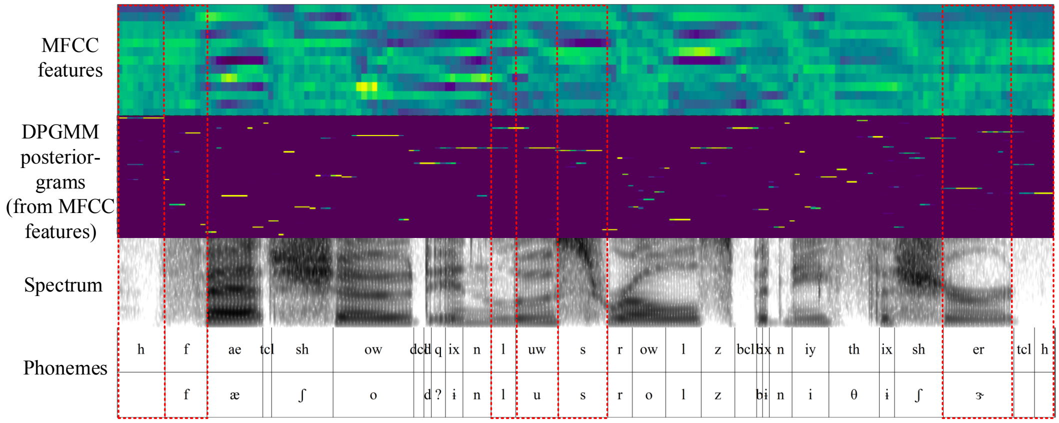

Fig. 3: Utterance 'Fat showed in loose rolls beneath the shirt' with id FADG0_SI1909 from TIMIT test set to show how DPGMM posteriorgrams complement MFCC features in identifying phonemes. Top layer is MFCC features of 13 dimensions, followed by layers of DPGMM posteriorgrams of 99 dimensions, spectrograms, and phonemes. DPGMM posteriorgrams are generated from MFCC features. Light green represents continuous values of MFCC features and DPGMM posteriorgrams. Posteriorgrams are only concentrated in a few dimensions (green or yellow); other dimensions have low probabilities (close to black). Red rectangles show some phonemes are clearly classified by DPGMM posteriorgrams, although not obviously identified by MFCC features.

the Kaldi [4] toolkit.

4.2. Feature Extraction

We also followed the Kaldi toolkit using a 39-dimensional MFCC+Δ+$\Delta\Delta$ (25-ms frame size and 10-ms frame shift) with mean and variance normalization (CMVN) as acoustic features for TIMIT. We used a 40-dimensional MFCC with high resolution followed by CMVN as the features for WSJ.

These MFCC acoustic features extracted from the TIMIT and WSJ training sets were trained to get the parameters of the DPGMM models, such as the weight, mean, and variance of each Gaussian. These trained parameters were fixed and used to generate the DPGMM posteriorgrams of the training, development, and test sets of each corpus. Note here that since we assumed we did not know the test set until the evaluation, we did not directly apply DPGMM clustering on the features of the test set. Instead, we used the parameters of the DPGMM model of the training set to extract the posteriorgrams of the test set during the evaluation. We compared the character error rate (CER) of our attentional encoder-decoder system with the acoustic features of MFCC and the features of the concatenation of MFCC and its DPGMM posteriorgram.

4.3. System Details and Experimental Setup

We used pytorch to implement an ASR system of an attentional encoder-decoder model [32] that consisted of a three-layer pyramid bidirectional LSTM encoder [32] that had 256 hidden units at each direction and dropped half of the frames to reduce the time resolution by a factor of 2 at each layer, a decoder [31] that contains a single-layer LSTM with 512 hidden units, and MLP attention [31].

MLP attention scheme generated the expected contextual vector by a probability vector output from a fully connected layer (MLP) fed with the concatenation of the current decoder hidden state and the encoder output (contextual vector). Table 2 shows that the decoder [31], at each time step, was fed with the concatenated feature of the output from the embedding layer and output from the previous decoding step, which was further processed by the LSTM and dropout layers. The output that was concatenated with the expected contextual vector from the attention was fed into a fully connected layer of 256 hidden units, followed by a tanh activation function. For the encoder with three layers, we dropped half of frames at each layer such that the 3-layer encoder output has a length that is $1/8$ of the number of frames of the current utterance features, which decreased the number of frames and captured the contexts across successive frames.

In the encoding stage, we fed speech features into a fully connected layer of 512 hidden units, followed by a ReLU activation function and a dropout layer with probability 0.25 before the pyramid BiLSTM. On the decoding stage, we put each character into an embedding layer of 256 hidden units, followed by a dropout layer before the decoder, whose output was converted into a probability vector by a softmax layer. For the MLP attention, we used one hidden layer of 256 units, followed by a tanh activation function. We used weight tying [33] between the input and output embeddings and label smoothing [34] with a ratio 0.05 in the decoder. We used the weight normalization in the attention.

Table 2: Architecture of attentional encoder-decoder ASR system: A → B denotes next layer of layer A is layer B. pBiLSTM denotes a pyramid bidirectional LSTM; FC stands for a full-connected layer; EMBED denotes an embedding layer. Module-N means the module with N hidden units (e.g., FC-512 denotes fully connected layer with 512 hidden units). Contextual FC-256 is a fully connected layer fed with the current embedding concatenated with expected contextual vector from attention. At each time step, the decoder, proposed by Luong [31], is fed with a concatenated feature of the output of the decoder pre-net and the output of decoder from the previous step. The encoder input are an acoustic features; input of decoder pre-net are characters. The pBiLSTM uses dropout regularization at each layer.

Module	**Cascaded layers of module**
Encoder	FC-512 → ReLU → Dropout → 3-layer pBiLSTM-256 (reduce half of the frames per layer)
Decoder pre-net	EMBED-256 → Dropout
Decoder [31]	(Pre-net output + Prev. decoder output) Single-layer LSTM-512 → Dropout → Contextual FC-256 → Tanh
Decoder post-net	Softmax
MLP attention	FC-256 → Tanh

When we trained the ASR system, we set the batch size to 32 and used the Adam optimizer [35] with an initial learning rate of 0.001, which decreased by a half whenever the loss successively increased for more than three epochs. Our ASR systems usually converged between 30 and 70 epochs after the learning rate dropped below 1e-5. We used a gradient norm clipping strategy [36] when training each batch to deal with the problems of exploding and vanishing gradients.

We evaluated our ASR system with a beam search where the beam size was 10 and the expand size [37] (which denotes as the maximum candidates per node to introduce more diversity into the search) was 5. We also increased the penalty [38] for long sentences with coefficient 1.

We used python to implement the DPGMM model, whose training process used the same parameter setting as previous works [7, 39]. We set the concentration parameter to 1 and the mean and variance of the prior to the global mean and the global variance of the MFCC features with belief-strengths 1 and $D + 2$, where D is the number of dimensions of the MFCC features. We obtained cluster labels after 1500 sampling iterations.

The ASR and DPGMM hyperparameters are summarized in Table 1 and the structure of our attentional encoder-decoder ASR system is summarized in Table 2.

Table 3: We compared the attentional encoder-decoder ASR systems with or without feature extension of the DPGMM posteriorgrams, along with two baselines [40, 30], by the character error rates (CERs) on the WSJ speech corpus [41]. No systems used pronunciation dictionaries or language models in the decoding process. We divided the WSJ corpus into the following datasets based on the Kaldi recipe [4]: training datasets of "train_si84" (about 15 hours) or "train_si284" (about 80 hours); an identical development dataset of "dev93" and an identical evaluation dataset of "eval92" for all systems.

Systems on WSJ train_si84 dataset (15 hrs)	**CER (%)**
Att Enc-Dec (Baseline ASR1) [30]	17.01
Att Enc-Dec (Baseline ASR2) [40]	17.35
Att Enc-Dec (Our ASR with MFCC)	16.61
Att Enc-Dec (Our ASR with MFCC + DPGMM)	14.86
Systems on WSJ train_si284 dataset (80 hrs)	**CER (%)**
Att Enc-Dec (Baseline ASR1) [30]	8.17
Att Enc-Dec (Baseline ASR2) [40]	7.12
Att Enc-Dec (Our ASR with MFCC)	6.57
Att Enc-Dec (Our ASR with MFCC + DPGMM)	5.67

5. RESULT

5.1. Analysis of Features of MFCC and DPGMM Posteriorgrams

To show that DPGMM posteriorgrams can complement MFCC features in identifying the phoneme sequence that underlies the utterance, we did DPGMM clustering on the TIMIT corpus [29] with the same parameter initialization as we did on the WSJ corpus [41]. We obtained the weight, mean, and variance parameters of the DPGMM model of each Gaussian of the MFCC feature on the TIMIT training set, froze them and applied them to the acoustic features on the test set. Fig. 3 shows that the word "loose" in the utterance, indicated by red rectangles, lacks clear phoneme categories within its MFCC representation, although it is relatively clearly classified by the DPGMM posteriorgram. The segmentations between the silences and the phonemes at the beginning and the end of the utterance, indicated by red rectangles, are not clearly observed in the MFCC features; but they are clearly segmented by the posteriorgram. In our preliminary experiment on TIMIT with the complete test set, phoneme error rate (PER) was lower in DPGMM-posteriorgram enhanced encoder-decoder attentional system than MFCC based system (PER of 22.74% vs. 23.92%).

5.2. ASR Results on Different Features

We verified the effectiveness and stability of our proposed method with the spontaneous speech recognition task on the WSJ corpus [41] with two tasks of different amounts of data.

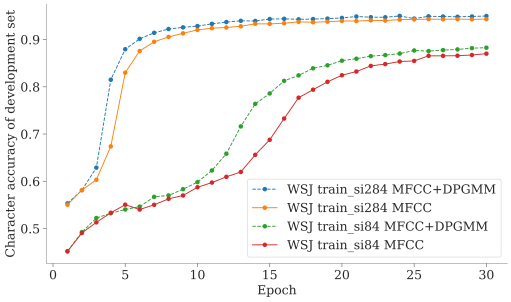

Fig. 4: Comparison between ASR systems with acoustic MFCC features (solid lines) and concatenated MFCC features and their DPGMM posteriorgrams (dashed lines) by character accuracy, which is the average ratio of correctly recognized characters per utterance, on the development set "dev93" of WSJ corpus, trained by "train_si284" or "train_si84" datasets.

One task was trained on the 15-hour "train_si84" dataset and another on the 80-hour "train_si284"; both tasks used the identical development dataset of "dev93" and the identical evaluation dataset of "eval92". Table 3 shows that on both tasks with identical ASR system settings, we observed a more constant decrease of CER with the feature with extension (MFCC + DPGMM) than in the original feature (MFCC). The WERs, consistent with CERs, of attentional ASR with MFCC and MFCC+DPGMM features on WSJ SI-284 set were 16.96% and 15.25%, compared with 18.2% in [30].

We analyzed the performance of the ASR systems during the entire training process. Fig. 4 shows that the ASR systems with feature extension by the DPGMM posteriorgram converged faster and retained the improvement compared to that without feature extension on the character accuracy of the development set ("dev93"). It improves more obviously on the system trained on the small dataset ("train_si84") than on the large dataset ("train_si284").

We also compared proposed features with the tandem bottleneck features [26], which we extracted from Kaldi [4] following its default settings except changing the bottleneck dimension same as that of DPGMM posteriorgrams. Our method had better performance on TIMIT corpus that the PERs of ASR with MFCC, MFCC+bottleneck, and MFCC+DPGMM were 23.92%, 23.22%, and 22.74%; with a bigger dataset of WSJ SI-284, tandem system got slightly better performance where CERs with MFCC, MFCC+bottleneck, and MFCC+DPGMM were 6.57%, 5.12%, and 5.67%.

6. DISCUSSION

For years some Zerospeech challenges [6, 7] have constantly comfirming the power of DPGMM features to discriminate phonemes [8] across different speakers, different languages under such harsh conditions as interviews with randomly interrupted disfluent speech [9], and wild or noisy recording environments [9, 10]. We are inspired by the discriminability of DPGMM features and combine them with acoustics features to improve the LVCSR system.

Our results shows that these acoustic features, which are concatenated with their DPGMM posteriorgrams, stably decrease the CERs of the ASR systems for spontaneous speech (Table 3). Compared with the original acoustic features, although the ASR performance is similar for the first few epochs, soon the speech recognition accuracy of the DPGMM-concatenated enhanced features improves (Fig. 4). The ASR systems with a small amount of data seem to improve more than those with a relatively large amount (Table 3, Fig. 4), which suggests the potential of our proposal for low-resource tasks. Unlike the tandem system [26], which combines the posteriorgrams of phonemes or states from data-hungry supervised learning, such unsupervised clustering as DPGMM needs less data to get robust posteriorgrams.

If we use DPGMM posteriorgrams alone without concatenating the original acoustic features to feed into ASR systems, they do not work as well as systems that just use acoustic features. Though DPGMM satisfactorily discriminates the phonemes, it does not directly work well for the ASRs: the CERs of the WSJ "train_si284" are 12.35% and the WSJ "train_si84" is 35.5% with the DPGMM posteriorgram alone. This is because the DPGMM model is weak at contextual modeling and its joint likelihood does not depend on the order of the observed data if they are infinite [42] and mainly captures acoustic information at the frame level. Framewise contextual modeling sometimes creates tempo-rally unsmoothed and fragmented DPGMM posteriorgrams, and the corresponding spectrum has smoothed and clear formants (for example, the word 'roll' in Fig. 3). The DPGMM posteriorgram sometimes struggles to model the contexts across several phonemes, which causes insertion or deletion errors in the ASR. One of our future works will enhance the weak contextual modeling of DPGMM features to get a better representation and further improve our ASR system.

7. CONCLUSION

The discriminability of DPGMM posteriorgrams has proved in several Zerospeech challenges. We combined an acoustic feature with its DPGMM posteriorgram to enhance the discriminability. Our result shows this proposal stably improved the performance of an attentional encoder-decoder system on spontaneous ASR tasks, especially with fewer resources.

Acknowledgements

Part of this work was supported by JSPS KAKENHI Grant Number JP17H06101.

8. REFERENCES

[1] Steven Davis and Paul Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[2] Hynek Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *the Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.

[3] Lawrence Rabiner, "Fundamentals of speech recognition," *Fundamentals of Speech Recognition*, 1993.

[4] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The Kaldi speech recognition toolkit," in *2011 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE Signal Processing Society, 2011, number EPFL-CONF-192584.

[5] Stephen J Young, PC Woodland, and WJ Byrne, "HTK: Hidden Markov model toolkit v1. 5," *User Manual, U.K., Cambridge*, 1993.

[6] Michael Heck, Sakriani Sakti, and Satoshi Nakamura, "Feature optimized DPGMM clustering for unsupervised subword modeling: A contribution to Zerospeech 2017," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 740–746.

[7] Hongjie Chen, Cheung-Chi Leung, Lei Xie, Bin Ma, and Haizhou Li, "Parallel inference of Dirichlet process Gaussian mixture models for unsupervised acoustic modeling: A feasibility study," in *INTERSPEECH*, 2015.

[8] Thomas Schatz, Vijayaditya Peddinti, Francis Bach, Aren Jansen, Hynek Hermansky, and Emmanuel Dupoux, "Evaluating speech features with the minimal-pair ABX task: Analysis of the classical MFC/PLP pipeline," in *INTERSPEECH*, 2013, pp. 1–5.

[9] Maarten Versteegh, Roland Thiolliere, Thomas Schatz, Xuan Nga Cao, Xavier Anguera, Aren Jansen, and Emmanuel Dupoux, "The zero resource speech challenge 2015," in *INTERSPEECH*, 2015.

[10] Ewan Dunbar, Xuan Nga Cao, Juan Benjumea, Julien Karadayi, Mathieu Bernard, Laurent Besacier, Xavier Anguera, and Emmanuel Dupoux, "The zero resource speech challenge 2017," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 323–330.

[11] Ewan Dunbar, Robin Algayres, Julien Karadayi, Mathieu Bernard, Juan Benjumea, Xuan-Nga Cao, Lucie Miskic, Charlotte Dugrain, Lucas Ondel, Alan W Black, et al., "The zero resource speech challenge 2019: TTS without T," *arXiv preprint arXiv:1904.11469*, 2019.

[12] Daniel Renshaw, Herman Kamper, Aren Jansen, and Sharon Goldwater, "A comparison of neural network methods for unsupervised representation learning on the zero resource speech challenge," in *INTERSPEECH*, 2015.

[13] Leonardo Badino, Claudia Canevari, Luciano Fadiga, and Giorgio Metta, "An Auto-encoder based approach to unsupervised learning of subword units," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 7634–7638.

[14] Leonardo Badino, Alessio Mereta, and Lorenzo Rosasco, "Discovering discrete subword units with binarized Autoencoders and Hidden-Markov-Model encoders," in *INTERSPEECH*, 2015.

[15] Roland Thiolliere, Ewan Dunbar, Gabriel Synnaeve, Maarten Versteegh, and Emmanuel Dupoux, "A hybrid dynamic time warping-deep neural network architecture for unsupervised acoustic modeling," in *INTERSPEECH*, 2015.

[16] Andros Tjandra, Berrak Sisman, Mingyang Zhang, Sakriani Sakti, Haizhou Li, and Satoshi Nakamura, "VQVAE unsupervised unit discovery and multi-scale Code2Spec inverter for Zerospeech challenge 2019," *arXiv preprint arXiv:1905.11449*, 2019.

[17] Céline Manenti, Thomas Pellegrini, and Julien Pinquier, "Unsupervised speech unit discovery using k-means and neural networks," in *International Conference on Statistical Language and Speech Processing*. Springer, 2017, pp. 169–180.

[18] Chia-ying Lee and James Glass, "A nonparametric Bayesian approach to acoustic model discovery," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, 2012, pp. 40–49.

[19] Lucas Ondel, Lukáš Burget, and Jan Černocký, "Variational inference for acoustic unit discovery," *Procedia Computer Science*, vol. 81, pp. 80–86, 2016.

[20] Janek Ebbers, Jahn Heymann, Lukas Drude, Thomas Glarner, Reinhold Haeb-Umbach, and Bhiksha Raj, "Hidden Markov model variational Autoencoder for acoustic unit discovery.," in *INTERSPEECH*, 2017, pp. 488–492.

[21] Siyuan Feng, Tan Lee, and Zhiyuan Peng, "Combining adversarial training and disentangled speech representation for robust zero resource subword modeling," *arXiv preprint arXiv:1906.07234*, 2019.

[22] Sadaoki Furui, "Speaker-independent isolated word recognition based on emphasized spectral dynamics," in *1986 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1986, vol. 11, pp. 1991–1994.

[23] Mirco Ravanelli, Jianyuan Zhong, Santiago Pascual, Pawel Swietojanski, Joao Monteiro, Jan Trmal, and Yoshua Bengio, "Multi-task self-supervised learning for robust speech recognition," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6989–6993.

[24] András Zolnay, Ralf Schluter, and Hermann Ney, "Acoustic feature combination for robust speech recognition," in *2005 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2005, vol. 1, pp. I–457.

[25] Fabio Valente, Mathew Magimai-Doss, and Wen Wang, "Analysis and comparison of recent mlp features for lvcsr systems," in *INTERSPEECH*, 2011.

[26] Hynek Hermansky, Daniel PW Ellis, and Sangita Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *2000 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2000, vol. 3, pp. 1635–1638.

[27] Jayaram Sethuraman, "A constructive definition of dirichlet priors," *Statistica Sinica*, pp. 639–650, 1994.

[28] Kevin P Murphy, "Conjugate Bayesian analysis of the Gaussian distribution," *Technical Report, University of British Columbia*, 2007.

[29] John S Garofolo, Lori F Lamel, William M Fisher, Jonathan G Fiscus, and David S Pallett, "DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1," *NASA STI Technical Report*, vol. 93, 1993.

[30] Suyoun Kim, Takaaki Hori, and Shinji Watanabe, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 4835–4839.

[31] Minh-Thang Luong, Hieu Pham, and Christopher D Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[32] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.

[33] Ofir Press and Lior Wolf, "Using the output embedding to improve language models," *arXiv preprint arXiv:1608.05859*, 2016.

[34] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[35] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[36] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio, "On the difficulty of training recurrent neural networks," in *International Conference on Machine Learning*, 2013, pp. 1310–1318.

[37] Markus Freitag and Yaser Al-Onaizan, "Beam search strategies for neural machine translation," *arXiv preprint arXiv:1702.01806*, 2017.

[38] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al., "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[39] Michael Heck, Sakriani Sakti, and Satoshi Nakamura, "Unsupervised linear discriminant analysis for supporting DPGMM clustering in the zero resource scenario," *Procedia Computer Science*, vol. 81, pp. 73–79, 2016.

[40] Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura, "Machine speech chain," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (IEEE TASLP)*, vol. 28, pp. 976–989, 2020.

[41] Douglas B Paul and Janet Baker, "The design for the Wall Street Journal-based CSR corpus," in *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*, 1992.

[42] Yee Whye Teh, "Dirichlet process," *Encyclopedia of Machine Learning*, pp. 280–287, 2010.