

PAPER

Domain Adaptation Based on Mixture of Latent Words Language Models for Automatic Speech Recognition

Ryo MASUMURA^{†a)}, Taichi ASAMI[†], Takanobu OBA^{†*}, Hirokazu MASATAKI[†], Sumitaka SAKAUCHI[†],
and Akinori ITO^{††}, *Members*

SUMMARY This paper proposes a novel domain adaptation method that can utilize out-of-domain text resources and partially domain matched text resources in language modeling. A major problem in domain adaptation is that it is hard to obtain adequate adaptation effects from out-of-domain text resources. To tackle the problem, our idea is to carry out model merger in a latent variable space created from latent words language models (LWLMs). The latent variables in the LWLMs are represented as specific words selected from the observed word space, so LWLMs can share a common latent variable space. It enables us to perform flexible mixture modeling with consideration of the latent variable space. This paper presents two types of mixture modeling, i.e., LWLM mixture models and LWLM cross-mixture models. The LWLM mixture models can perform a latent word space mixture modeling to mitigate domain mismatch problem. Furthermore, in the LWLM cross-mixture models, LMs which individually constructed from partially matched text resources are split into two element models, each of which can be subjected to mixture modeling. For the approaches, this paper also describes methods to optimize mixture weights using a validation data set. Experiments show that the mixture in latent word space can achieve performance improvements for both target domain and out-of-domain compared with that in observed word space.

key words: domain adaptation, mixture modeling, latent words language models, latent variable space, automatic speech recognition

1. Introduction

Language models (LMs) are invaluable for natural language processing tasks such as automatic speech recognition (ASR) and statistical machine translation [1], [2]. LM performance strongly depends on the quantity and quality of the training data sets. Superior performance is usually obtained by using enormous domain-matched data sets to construct LMs [3]. Unfortunately, in practical ASR tasks, large amounts of domain-matched data sets are not available.

Therefore, LMs demand domain adaptation techniques to allow the use of multiple out-of-domain text resources [4], [5]. In language modeling, one of the most popular approaches to domain adaptation is based on mixture modeling [6], [7]. An adapted model can be constructed by combining LMs that are individually constructed from out-of-domain text resources with mixture weighting [8]. The mix-

ture weights are optimized using a small amount of target domain text.

Previously, observed word space mixture modeling, i.e., n-gram mixture modeling, has been used in various cases [9]–[11]. Also, mixture modeling of recurrent neural network LMs (RNNLMs) was performed in the observed word space [12]–[14]. However, mixtures in the observed word space do not support flexible domain adaptation if domain-related data sets are hardly obtained. In the observed word space, a word directly represents a state in a mixture. It can be considered that effective state sharing is not available by merging LMs individually constructed from out-of-domain text resources since words are not overlapped.

In order to conduct flexible domain adaptation using the LMs constructed from out-of-domain text resources, this paper develops methods in which model merging is conducted in a latent variable space. In the latent variable space, a word is mapped into a latent variable space, so it can be expected to perform more flexible state sharing than is possible in the observed word space. To this end, this paper introduces latent words language models (LWLMs) to the mixture modeling [15]–[18]. The latent variables in usual class based n-gram LMs are only model-dependent indices, so each model has a different latent variable space [19], [20]. Therefore, conventional class-based n-gram mixture modeling have to be performed in the observed word space [21], [22]. On the other hand, latent variables in LWLMs are represented as specific latent word, multiple LWLMs can share the common latent variable space.

In addition, this paper also focuses on the fact that any LWLM can be split into two elements, a transition probability model and an emission probability model, and each of which can be mixed independently. This concept of mixture modeling yields flexibility in that both elements are the intersections of different data sources. It is assumed that each element model has a different role, i.e., the transition probability model captures the sentence pattern in the latent variable space, while the emission probability model captures the lexical pattern in the observed word space. In fact, most available out-of-domain text resources in practical ASR tasks will partially match either the sentence pattern or the lexical pattern. It can be expected that a domain matched model will become available by optimizing both elements independently.

In this paper, two types of mixture modeling meth-

Manuscript received July 3, 2017.

Manuscript revised January 5, 2018.

Manuscript publicized February 26, 2018.

[†]The authors are with NTT Media Intelligence Laboratories, NTT Corporation, Yokosuka-shi, 239–0847 Japan.

^{††}The author is with the Graduate School of Engineering, Tohoku University, Sendai-shi, 980–8579 Japan.

*Presently, with NTT Docomo Corporation, Yokosuka-shi, 239–8536 Japan.

a) E-mail: masumura.ryo@lab.ntt.co.jp

DOI: 10.1587/transinf.2017EDP7210

ods using multiple LWLMs are proposed. One is LWLM mixture models that can merge multiple LWLM in the latent word space with mixture weights. The other is LWLM cross-mixture models in which two elements in LWLMs are independently combined with mixture weights. Although the proposed models have complex model structure, they can be implemented into ASR decoder using n-gram approximation method, which randomly generates a lot of text data according to a stochastic process and a simple n-gram model is constructed from the generated data [16], [18].

For domain adaptation, this paper also presents their optimization method using a validation data set. In the observed word space mixture, the maximum likelihood (ML) criterion can be used because generative probabilities of each word of the validation data set can be directly calculated [23]. Unfortunately, this advantage is offset by the fact the latent word sequence of the validation data set cannot be determined uniquely. In order to estimate optimal mixture weights of the LWLM mixture models and the LWLM cross-mixture models, we introduce Bayesian criterion. The Bayesian criterion can be flexibly applied to various model structures, and sampling techniques can be used. In this paper, Gibbs sampling is introduced for estimating the latent word sequence and model index sequence underlying the validation data set [24].

In fact, this paper is an extended study of our previous work in which LWLM mixture models were only presented [25]. In this paper, we additionally formulate the LWLM cross-mixture modeling and its optimization method, and clarify relationships to each mixture model. Our evaluation examines two kinds of setups. The first experiment employs in-domain training data set and out-of-domain training data set for constructing a target domain LM, and shows effectiveness of the LWLM-mixture models. The second experiment employs two types of partially matching training data sets on the assumption of a practical spontaneous speech recognition task, and shows the LWLM-cross mixture model yields additional adaptation effects which cannot be obtained by the LWLM-mixture model.

The rest of this paper is organized as follows. Section 2 overviews LWLMs and n-gram mixture models. Section 3 describes definitions of LWLM mixture models and LWLM cross-mixture models. In addition, optimization methods for domain adaptation and implementation methods for ASR tasks are detailed. Sections 4 and 5 present automatic speech recognition experiments. Section 6 concludes this paper with a summary of key points.

2. Previous Work

2.1 Latent Words Language Models

LWLMs are generative models that employ a latent variable called latent word [15]. An LWLM has a soft clustering structure, and a latent word is a specific word that can be selected from the entire vocabulary. Thus, the number of la-

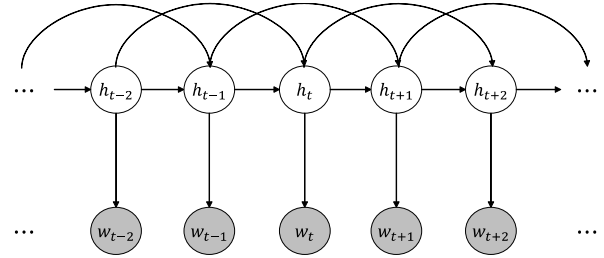


Fig. 1 Model structure of LWLMs.

tent words equals the number of observed words, and multiple LWLMs can share a common latent variable space.

In the generative process of an LWLM, a latent word h_t is generated on the basis of a transition probability model and its context $\mathbf{l}_t = h_{t-n+1}, \dots, h_{t-1}$. An observed word w_t is generated on the basis of an emission probability model and a latent word h_t . A graphic rendering of LWLM is shown in Fig. 1. The gray circles denote observed words and the white circles denote latent variables. The LWLM produces the generative probability of the observed word sequence $\mathbf{w} = w_1, \dots, w_T$. The probability is approximately calculated by the following point estimation:

$$P(\mathbf{w}) \approx \prod_{t=1}^T \sum_{h_t \in \mathcal{V}} P(w_t|h_t, \Theta_{1w})P(h_t|\mathbf{l}_t, \Theta_{1w}), \quad (1)$$

where Θ_{1w} indicates a model parameter of the LWLM, and \mathcal{V} is the vocabulary. The transition probability model $P(h_t|\mathbf{l}_t, \Theta_{1w})$ is expressed as an n-gram model for latent words; it can capture the sentence pattern on the basis of a latent variable sequence. The emission probability model $P(w_t|h_t, \Theta_{1w})$ is expressed as a unigram model for each latent word and can capture the lexical pattern. Usually a hierarchical Pitman-Yor prior is used as the transition probability model, and a Dirichlet prior is used as the emission probability model. More details are provided in previous studies [15]–[18].

2.2 N-gram Mixture Models

N-gram mixture model is constructed by combining several n-gram LMs trained using different sources. A graphic rendering of an n-gram mixture model is shown in Fig. 2; the model index is represented as $z_t \in \{1, \dots, Z\}$. Each n-gram LM calculates the generative probability of word w_t given context information \mathbf{u}_t using $n-1$ words behind w_t . As shown in Fig. 2, the observed word sequence $\mathbf{w} = w_1, \dots, w_T$ is generated dependent on model index sequence $\mathbf{z} = z_1, \dots, z_T$. The generative probability of the observed word sequence \mathbf{w} is defined as:

$$P(\mathbf{w}) = \prod_{t=1}^T \sum_{z_t \in \{1, \dots, Z\}} P(z_t)P(w_t|\mathbf{u}_t, \Theta_{ng}^{z_t}), \quad (2)$$

where $P(z_t)$ is the mixture weight for the z_t -th n-gram model.

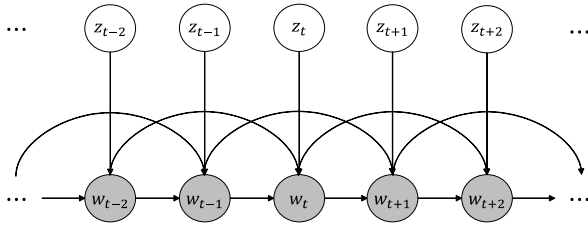


Fig. 2 Model structure of n-gram mixture models.

In practice, direct implementation of the n-gram mixture model to ASR is not ideal because it does not have a back-off n-gram structure. Actually, the n-gram mixture model can be approximately represented as a single back-off n-gram structure [26].

For domain adaptation of n-gram mixture models, mixture weights are optimized using a validation data set. The expectation maximization algorithm, which is based on maximum likelihood (ML) criterion, can be used for the optimization [23]. Given a validation data set $\mathcal{W} = w_1, \dots, w_{|\mathcal{W}|}$, the optimized mixture weight $\hat{P}(z)$ is estimated in an iterative manner as:

$$\hat{P}(z) = \frac{1}{|\mathcal{W}|} \sum_{t=1}^{|\mathcal{W}|} \frac{P(w_t | \mathbf{u}_t, \Theta_{ng}^z) P(z)}{\sum_{z' \in \{1, \dots, Z\}} P(w_t | \mathbf{u}_t, \Theta_{ng}^{z'}) P(z')}. \quad (3)$$

After iterations, the optimized weight $\hat{P}(z)$ is used in Eq. (2).

3. Proposed Method

3.1 LWLM Mixture Models

This paper details LWLM mixture models. A graphic rendering of LWLM mixture models is shown in Fig. 3. As shown, LWLM mixture modeling can be considered to be the union of Fig. 1 and Fig. 2. The gray circles denote observed words and the white circles denote latent variables and model indices.

The generative process starts with model index $z_t \in \{1, \dots, Z\}$, which corresponds to each LWLM index. Then, latent word h_t and observed word w_t are generated based on the basis of the selected LWLM's stochastic process. In LWLM mixture models, the generative probability of w is defined as:

$$P(w) = \prod_{t=1}^T \sum_{h_t \in \mathcal{V}} \sum_{z_t \in \{1, \dots, Z\}} P(w_t | h_t, \Theta_{1w}^{z_t}) P(h_t | \mathbf{l}_t, \Theta_{1w}^{z_t}) P(z_t), \quad (4)$$

where $\Theta_{1w}^{z_t}$ is the z_t -th model parameter of the pre-trained LWLM, and $P(z_t)$ indicates the mixture weight for the z_t -th model. In this equation, $P(z_t)$ can be estimated from a validation data set. This equation is based on the characteristics that LWLMs share a common latent variable space.

3.2 LWLM Cross-Mixture Models

This paper also proposes LWLM cross-mixture models. In

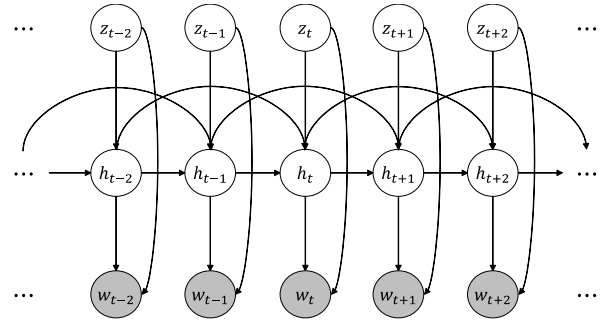


Fig. 3 Model structure of LWLM mixture models.

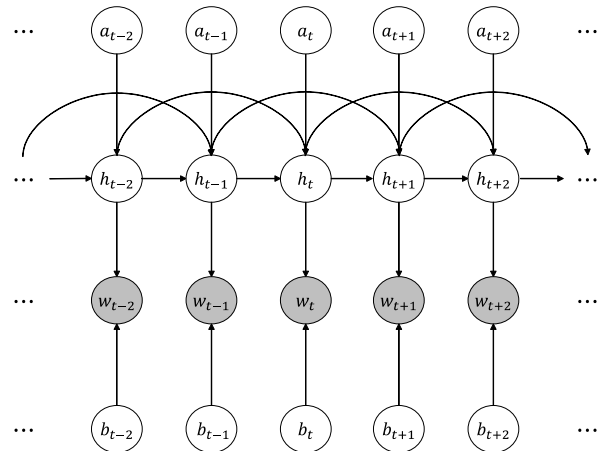


Fig. 4 Model structure of LWLM cross-mixture models.

fact, LWLMs are divided into two components, i.e., a transition probability model and an emission probability model. Therefore, each component can be independently mixed. The mixture of transition probability models is performed in the latent variable space, and the mixture of emission probability models is performed in the observed word space. To this end, two model indices are introduced with respect to each component model. A graphic rendering of the LWLM cross-mixture models is shown in Fig. 4.

Its generative process starts when the transition probability model index $a_t \in \{1, \dots, Z\}$ and the emission probability model index $b_t \in \{1, \dots, Z\}$ are generated independently. Then, latent word h_t is generated on the basis of the selected transition probability model and its context \mathbf{l}_t . Observed word w_t is generated on the basis of the selected emission probability model and latent word h_t . In a standard LWLM mixture model, h_t and w_t are generated using the same LWLM, whereas they are generated using different LWLMs in an LWLM cross-mixture model. In LWLM cross-mixture models, the generative probability of w is defined as:

$$P(w) = \prod_{t=1}^T \sum_{h_t \in \mathcal{V}} \sum_{a_t \in \{1, \dots, Z\}} \sum_{b_t \in \{1, \dots, Z\}} P(w_t | h_t, \Theta_{1w}^{b_t}) P(h_t | \mathbf{l}_t, \Theta_{1w}^{a_t}) P(a_t) P(b_t), \quad (5)$$

where $\Theta_{LW}^{a_t}$ and $\Theta_{LW}^{b_t}$ are respectively the a_t -th parameter of the transition probability model and the b_t -th parameter of the emission probability model in the pre-trained LWLMs. $P(a_t)$ and $P(b_t)$ are mixture weights and can be optimized using a validation data set.

3.3 Optimization for Domain Adaptation

To optimize mixture weights using a validation data set for domain adaptation, the ML criterion cannot be employed because the latent word sequence is an underspecified variable. If the ML criterion is used, all possible latent word assignments have to be considered since LWLM has a soft clustering structure. It is computationally and analytically intractable to calculate the expectation value. Therefore, this paper employs the Bayesian criterion and a sampling based procedure that is compatible with LWLM training. In the sampling based procedure, model index sequences of the validation data set are estimated for determining the mixture weights.

3.3.1 Optimization of LWLM Mixture Models

In order to optimize the LWLM mixture models, the optimized mixture weight $\hat{P}(z)$ is estimated using a validation data set $\mathcal{W} = w_1, \dots, w_{|\mathcal{W}|}$. In the Bayesian criterion, a model index sequence of the validation data set \mathcal{Z} and a latent word sequence \mathcal{H} are estimated using the Gibbs sampling. A conditional probability of possible values for latent word $h_t \in \mathcal{V}$ is given by:

$$P(h_t|\mathcal{W}, \mathcal{H}_{-t}, \mathcal{Z}) \sim P(w_t|h_t, \Theta_{LW}^{z_t}) \prod_{j=t}^{t+n-1} P(h_j|l_j, \Theta_{LW}^{z_t}), \quad (6)$$

where \mathcal{H}_{-t} represents all latent words except for h_t . In a similar way, a conditional probability of possible values for model index $z_t \in \{1, \dots, Z\}$ is given by:

$$P(z_t|\mathcal{W}, \mathcal{H}, \mathcal{Z}_{-t}) \sim P(w_t|h_t, \Theta_{LW}^{z_t})P(h_t|l_t, \Theta_{LW}^{z_t})P(z_t|\mathcal{Z}_{-t}), \quad (7)$$

where \mathcal{Z}_{-t} represents model index sequence except for z_t . Gibbs sampling can be used to sample new values for the model index and the latent variable according to these two distributions and place them at position t .

Once model index sequence is concluded, $P(z_t|\mathcal{Z})$ can be calculated as:

$$P(z_t|\mathcal{Z}) = \frac{c(z_t) + \beta}{\sum_{z' \in \{1, \dots, Z\}} c(z') + \beta Z}, \quad (8)$$

where $c(z_t)$ denotes the count of model index z_t in \mathcal{Z} . β is a hyper parameter for Dirichlet distribution.

In a Bayesian criterion, optimized value $\hat{P}(z)$ is estimated by Monte Carlo integration. Multiple model index sequences sampled after the burn-in period are defined as $\mathcal{Z}^1, \dots, \mathcal{Z}^S$. $\hat{P}(z)$ is estimated as:

$$\hat{P}(z) = \frac{1}{S} \sum_{s=1}^S P(z|\mathcal{Z}^s). \quad (9)$$

If β approaches 0, the Bayesian criterion is equivalent to the ML criterion.

3.3.2 Optimization of LWLM Cross-Mixture Models

In LWLM cross-mixture models, optimized mixture weights $\hat{P}(a)$ and $\hat{P}(b)$ are simultaneously estimated by defining both model index sequences of a validation data set \mathcal{W} . Gibbs sampling can also be used to assign latent word sequence \mathcal{H} , transition probability model index sequence \mathcal{A} , and emission probability model index sequence \mathcal{B} to the validation data set \mathcal{W} . The conditional probability of possible values for latent word $h_t \in \mathcal{V}$ is given by:

$$P(h_t|\mathcal{W}, \mathcal{H}_{-t}, \mathcal{A}, \mathcal{B}) \sim P(w_t|h_t, \Theta_{LW}^{b_t}) \prod_{j=t}^{t+n-1} P(h_j|l_j, \Theta_{LW}^{a_t}). \quad (10)$$

In a similar way, the conditional probabilities of possible values for model indices $a_t \in \{1, \dots, Z\}$ and $b_t \in \{1, \dots, Z\}$ are given by:

$$P(a_t|\mathcal{W}, \mathcal{H}, \mathcal{A}_{-t}, \mathcal{B}) \sim P(h_t|l_t, \Theta_{LW}^{a_t})P(a_t|\mathcal{A}_{-t}), \quad (11)$$

$$P(b_t|\mathcal{W}, \mathcal{H}, \mathcal{A}, \mathcal{B}_{-t}) \sim P(w_t|h_t, \Theta_{LW}^{b_t})P(b_t|\mathcal{B}_{-t}), \quad (12)$$

where \mathcal{A}_{-t} and \mathcal{B}_{-t} represent model index sequences except for a_t and b_t . $P(a_t|\mathcal{A}_{-t})$ and $P(b_t|\mathcal{B}_{-t})$ are respectively estimated from \mathcal{A}_{-t} and \mathcal{B}_{-t} . This sampling procedure is iterated until convergence is achieved.

Once both assignments are defined, each probability can be calculated.

$$P(a_t|\mathcal{A}) = \frac{c(a_t) + \beta}{\sum_{a' \in \{1, \dots, Z\}} c(a') + \beta Z}, \quad (13)$$

$$P(b_t|\mathcal{B}) = \frac{c(b_t) + \beta}{\sum_{b' \in \{1, \dots, Z\}} c(b') + \beta Z}, \quad (14)$$

where $c(a_t)$ denotes the count of model index a_t in \mathcal{A} , and $c(b_t)$ denotes the count of model index b_t in \mathcal{B} , β is a hyper parameter for Dirichlet distribution.

In a Bayesian criterion, optimized values $\hat{P}(a)$ and $\hat{P}(b)$ are also estimated by Monte Carlo integration. Multiple model index sequences sampled after the burn-in period are defined as $\mathcal{A}^1, \dots, \mathcal{A}^S$ and $\mathcal{B}^1, \dots, \mathcal{B}^S$. $\hat{P}(a)$ and $\hat{P}(b)$ are estimated as:

$$\hat{P}(a) = \frac{1}{S} \sum_{s=1}^S P(a|\mathcal{A}^s), \quad (15)$$

$$\hat{P}(b) = \frac{1}{S} \sum_{s=1}^S P(b|\mathcal{B}^s). \quad (16)$$

3.4 Implementation for ASR

In order to implement the LWLM mixture model and the

Algorithm 1 Random sampling based on LWLM mixture model.**Input:** Model parameters $\Theta_{1w}^1, \dots, \Theta_{1w}^M$, number of sampled words T **Output:** Sampled words w

```

1:  $l_1 = \langle s \rangle$ 
2: for  $t = 1$  to  $T$  do
3:    $z_t \sim P(z_t)$ 
4:    $h_t \sim P(h_t | l_t, \Theta_{1w}^{z_t})$ 
5:    $w_t \sim P(w_t | h_t, \Theta_{1w}^{z_t})$ 
6: end for
7: return  $w = w_1, \dots, w_T$ 

```

LWLM cross-mixture model to ASR, a special technique is needed as well as a standard LWLM. Therefore, this paper introduces an n-gram approximation technique for both the LWLM mixture model and the LWLM cross-mixture model. The n-gram approximation is a method that approximates target LM as a simple back-off n-gram structure, and offers one-pass ASR decoding. The n-gram approximation of LWLM mixture model has the following properties:

$$w_{1wm} \sim P(w | \Theta_{1wm}), \quad (17)$$

$$w_{1wmng} \sim P(w | \Theta_{1wmng}), \quad (18)$$

$$w_{1wm} \approx w_{1wmng}, \quad (19)$$

where w_{1wm} is an observed word sequence generated from the LWLM mixture model, and w_{1wmng} is an observed word sequence generated from the approximated model with back-off n-gram structure.

In a similar way, the n-gram approximation of LWLM cross-mixture model has the following properties:

$$w_{1wcm} \sim P(w | \Theta_{1wcm}), \quad (20)$$

$$w_{1wcmng} \sim P(w | \Theta_{1wcmng}), \quad (21)$$

$$w_{1wcm} \approx w_{1wcmng}, \quad (22)$$

where w_{1wcm} is an observed word sequence generated from the LWLM cross-mixture model, and w_{1wcmng} is an observed word sequence generated from approximated model with back-off n-gram structure.

The random sampling of LWLM mixture model is based on Algorithm 1. In addition, the random sampling of LWLM cross-mixture model is based on Algorithm 2. In line 1, l_1 is initialized as a sentence head symbol $\langle s \rangle$. With T iterations, T latent words, and T observed words are generated. The T observed words are used only for back-off n-gram model estimation.

4. Experiment 1

4.1 Setups

In the first experiment, a target domain data set and an out-of-domain data set were prepared for constructing an LM. In the experiment, the Corpus of Spontaneous Japanese (CSJ) was divided into academic lectures and extemporaneous lectures [27]. Target domain was set to the academic lectures;

Algorithm 2 Random sampling based on LWLM cross-mixture model.**Input:** Model parameters $\Theta_{1w}^1, \dots, \Theta_{1w}^M$, number of sampled words T **Output:** Sampled words w

```

1:  $l_1 = \langle s \rangle$ 
2: for  $t = 1$  to  $T$  do
3:    $a_t \sim P(a_t)$ 
4:    $b_t \sim P(b_t)$ 
5:    $h_t \sim P(h_t | l_t, \Theta_{1w}^{a_t})$ 
6:    $w_t \sim P(w_t | h_t, \Theta_{1w}^{b_t})$ 
7: end for
8: return  $w = w_1, \dots, w_T$ 

```

Table 1 Experimental data set in Experiment 1.

	Domain	# of words
Train A	Academic lecture	3,468,133
Train B	Extemporaneous lecture	3,847,816
Valid	Academic lecture	28,046
Test A	Academic lecture	27,907
Test B	Extemporaneous lecture	18,251

a validation data set (Valid) was prepared for the target domain. Training data sets (Train A and B) and test data sets (Test A and B) were prepared for both domains. Vocabulary size for Train A was 40,725 and that for Train B was 64,543. Details of the experimental data set are shown in Table 1.

For ASR evaluation, an acoustic model on the basis of hidden Markov models with deep neural networks (DNN-HMM) was prepared [28]. The DNN-HMM had eight hidden layers with 2048 nodes and was trained using the CSJ. The speech recognition decoder was VoiceRex, a WFST-based decoder [29], [30]. JTAG was used as the morpheme analyzer to split sentences into words [31].

In the evaluation, we aimed to compare following two settings. One setting was that single training data set is only available. Another setting was that multiple training data sets are available. For the former setting, the following base LMs were individually constructed from each training data set.

1. HPY3: Word-based 3-gram hierarchical Pitman-Yor LM (HPYLM) constructed from a training data set [32]. For the training, 200 iterations were used for burn-in, and collected 10 samples. HPY3 constructed from the training data set A is denoted as 1-A, and HPY3 constructed from the training data set B is denoted as 1-B
2. LW3: Word-based 3-gram HPYLM constructed from data generated on the basis of 3-gram LWLM. The generated data size was one billion words which was determined in consideration of previous work [18]. We pruned n-gram entries as to be comparable computation complexity to HPY3 using entropy based pruning [33]. The LWLM was constructed from a training data set. For the training of LWLM, 500 iterations were used for burn-in and collected a sample. LW3 constructed from the training data set A is denoted as 2-A, and LW3 constructed from the training data set B is de-

Table 3 Perplexity and word error rate [%] results in Experiment 1.

			Valid (Target domain)		Test A (Target domain)		Test B (Out-of-domain)	
			PPL	WER	PPL	WER	PPL	WER
1-A.	Base model A	HPY3	70.57	20.80	62.85	21.98	183.38	32.51
2-A.	(Target domain training set)	LW3	70.02	20.72	62.34	21.85	165.87	31.43
3-A.		HPY3+LW3	65.30	19.27	58.25	21.09	156.45	30.26
1-B.	Base model B	HPY3	180.02	30.76	127.26	32.24	88.48	24.22
2-B.	(Out-of-domain training set)	LW3	174.84	30.25	122.44	31.45	90.71	24.30
3-B.		HPY3+LW3	161.60	29.20	115.57	30.68	83.20	23.02
4.	Adapted model	HPYM3	71.68	18.78	64.19	20.34	178.71	26.56
5.		ALWM3	72.83	18.56	64.57	20.22	178.48	26.38
6.		LWM3	72.72	18.45	64.39	20.10	162.87	25.29
7.		HPYM3+ALWM3	67.52	17.88	60.45	19.62	178.53	26.25
8.		HPYM3+LWM3	67.38	17.64	60.19	19.36	164.46	25.34

Table 2 Out-of-vocabulary rate [%] in Experiment 1.

Vocabulary size	OOV rate			
	Valid	Test A	Test B	
Base model A	40,725	0.89	0.90	3.65
Base model B	64,543	3.59	3.56	1.11
Adapted model	81,856	0.67	0.50	0.94

noted as 2-B.

3. HPY3+LW3: Mixed model which combined HPY3 and LW3 with a mixture weight. The mixture weight was set as 0.5 that was an optimal value for the validation set. The mixed model trained from training data set A is denoted as 3-A, and the mixed model trained from training data set B is denoted as 3-B.

Besides, for the latter setting, five adapted LMs which used not only the training data set A but also the training data set B were prepared.

4. HPYM3: HPYLM mixture model consisting of HPYLMs (HPY3) individually constructed from each training data.
5. ALWM3: HPYLM mixture model consisting of n-gram approximated LWLMs (LW3) individually constructed from each training data.
6. LWM3: Word-based 3-gram HPYLM constructed from data generated on the basis of an LWLM mixture model. The LWLM mixture model was constructed from an LWLM trained from the training data set A and an LWLM trained from the training data set B. For the n-gram approximation, one billion words were generated as with LW3. We pruned n-gram entries as to be comparable computation complexity to HPYM3 using entropy based pruning.
7. HPYM3+ALWM3: Mixed model of HPYM3 and ALWM3.
8. HPYM3+LWM3: Mixed model of HPYM3 and LWM3.

The vocabulary size for each adapted model was 81,558. The mixture weights in the adapted models were optimized for the validation data set. For the Monte Carlo integration, S was set to 10. Other hyper parameters were also optimized using the validation data set. Table 2 shows out-of-vocabulary (OOV) rate for both base LMs and adapted LMs.

4.2 Results

Table 3 shows the perplexity (PPL) and word error rate (WER) results for each condition. The difference of PPL in base models and in adapted models cannot be compared since each vocabulary size differs.

Lines 1-A to 3-A show the results which only used the training data set A, and lines 1-B to 3-B show the results which only used the training data set B. LW3 provides results comparable to HPY3 in a same domain, and performs robustly in out-of-domain. The WER difference between LW3 and HPY3 in test set B was statistically significant ($p < 0.05$). The highest performance was obtained by HPY3+LW3. The WER differences between HPY3 and HPY3+LW3 in each test set were statistically significant ($p < 0.05$). It can be considered that the performance was improved because LW3 and HPY3 had different attributes, which observed words are generated depending on latent words in LW3 while they are generated depending on last observed words in HPY3.

Next, Lines 4-8 show the results of adapted LMs which used both the training data set A and the training data set B. The results show performance improvements were obtained by using the out-of-domain training data compared with only using the target domain training data. About the target domain, ALWM3 and LWM3 provided comparable performance to HPYM3. In addition, both of LWLM-based adapted models acquired the improvement by combining mixture of HPYLMs. It is considered to originate in having the character in which HPYM3 differ from LWM3. The highest performance was obtained by HPYM3+LWM3. In terms of WER, statistically significant performance improvements ($p < 0.01$) were achieved by HPYM3+LWM3 compared to HPYM3.

On the other hand, about the out-of-domain, LWM3 achieved higher performance than HPYM3 and ALWM3. The WER difference between ALWM3 and LWM3 in test set B was statistically significant ($p < 0.01$). This result shows that mixture modeling in the latent variable space can perform more flexible adaptation than that in the observed word space. Actually, in mixture modeling on a latent variable space, the mixture weight for base model B is comparatively high compared with that in an observed word space. The mixture weight for base model B in ALWM3 was 0.09 while

that in LWM3 was 0.13. In terms of WER, statistically significant performance improvements ($p < 0.01$) were also achieved by HPYM3+LWM3 compared to HPYM3. It turned out that LWM3 can achieve improvement for both the target domain and the out-of-domain compared with HPYM3.

5. Experiment 2

5.1 Setups

In the second experiment, two types of partially matched training data sets were prepared for constructing an LM. The target domain was set to academic lecture speech; its style is spontaneous speech and the topic is related to acoustics. A validation data set (Valid) and a test data set (Test) for the target domain were prepared from CSJ [27]. Each data set had about 30K words.

Training data set A (Train A) consisted of transcriptions of simulated lecture speeches that are included in CSJ. The data size was about 4M words and the style matched that of the target domain but the topic was not related to the target domain. The vocabulary size was 64,761. On the other hand, training data set B (Train B) consisted of Web documents collected using the validation data set based on relevant document retrieval techniques [34]. The data size was about 11M words and the topic was related to the acoustics but the style was written text. The vocabulary size was 64,152. These setups seem to be reasonable for practical spontaneous speech recognition tasks. Details of the experimental data set are summarized in Table 4.

For evaluating ASR performance, a DNN-HMM acoustic model was prepared [28]. The DNN-HMM had 8 hidden layers with 2048 nodes and 3072 outputs. The speech recognition decoder was a WFST-based decoder [29].

Our experimental settings aimed to compare following two settings. One setting was that single partially matched training data set is only available. Another setting was that multiple training data sets which complement each other are available. For the former setting, four types of base LMs were individually constructed from each training data set.

1. HPY3: Word-based 3-gram HPYLM constructed from a training data set [32]. For the training, 200 iterations were used for burn-in, and collected 10 samples. HPY3 constructed from the training data set A is denoted as 1-A, and HPY3 constructed from the training data set B is denoted as 1-B.
2. RNN: Class-based RNNLM with 500 hidden nodes and 500 classes constructed from a training data set [12].

RNN constructed from the training data set A is denoted as 2-A, and RNN constructed from the training data set B is denoted as 2-B.

3. LW3: Word-based 3-gram HPYLM constructed from data generated on the basis of 3-gram LWLM. The generated data size was one billion words which was determined in consideration of our previous work [18]. We pruned n-gram entries as to be comparable computation complexity to HPY3 using entropy based pruning [33]. The LWLM was constructed from a training data set. For the training of LWLM, 500 iterations were used for burn-in and collected 10 samples. LW3 constructed from the training data set A is denoted as 3-A, and LW3 constructed from the training data set B is denoted as 3-B.
4. HPY3+LW3: Mixed model which combined HPY3 and LW3 with a mixture weight. The mixture weight was set as 0.5 that was an optimal value for the validation set. The mixed model trained from training data set A is denoted as 4-A, and the mixed model trained from training data set B is denoted as 4-B.

Next, for the latter setting, following adapted LMs were constructed using the trained base LMs.

5. HPYM3: HPYLM mixture model constructed from HPY3 trained from the training data set A and HPY3 trained from the training data set B. The mixture weights were optimized using the validation data set. It was converted into a back-off n-gram structure and implemented in a WFST-based one-pass decoder.
6. RNNM: RNN mixture model constructed from RNN trained from the training data set A and RNN trained from the training data set B. The mixture weights were optimized using the validation data set. RNNM cannot be converted into WFST format, so single use of RNNM was only tested in perplexity evaluation. 1000-best rescoring was used when RNNM was combined with other LM.
7. LWM3: Word-based 3-gram HPYLM constructed from data generated on the basis of an LWLM mixture model. The LWLM mixture model was constructed from an LWLM trained from the training data set A and an LWLM trained from the training data set B. For the n-gram approximation, one billion words were generated as with LW3. We pruned n-gram entries as to be comparable computation complexity to HPYM3 using entropy based pruning. The mixture weights were optimized using the validation data set.
8. LWM3: Word-based 3-gram HPYLM constructed from data generated on the basis of an LWLM cross-mixture model. The LWLM cross-mixture model was constructed from an LWLM trained from the training data set A and an LWLM trained from the training data set B. For the n-gram approximation, one billion words were generated as with LW3 and LWM3. We pruned n-gram entries as to be comparable computation complexity to HPYM3 using entropy based pruning. The mixture weights were optimized using the validation

Table 4 Experimental data set in Experiment 2.

	Style	Topic	# of words
Train A	Spontaneous	Various topics	3,833,883
Train B	Written	Acoustics	10,541,945
Valid	Spontaneous	Acoustics	28,547
Test	Spontaneous	Acoustics	28,504

Table 6 Perplexity and word error rate [%] results in Experiment 2.

			Valid (Target domain)		Test (Target domain)	
			PPL	WER	PPL	WER
1-A.	Base model A	HPY3	247.73	31.42	186.11	35.94
2-A.	(Training set A)	RNN	244.73	-	184.16	-
3-A.		LW3	239.91	31.09	179.83	35.45
4-A.		HPY3+LW3	223.68	29.86	169.93	34.16
1-B.	Base model B	HPY3	235.91	30.68	273.09	37.33
2-B.	(Training set B)	RNN	275.23	-	326.31	-
3-B.		LW3	207.91	30.05	240.08	36.47
4-B.		HPY3+LW3	200.90	28.82	232.93	34.88
5.	Adapted model	HPYM3	130.30	25.24	119.22	30.33
6.		RNNM	126.35	-	118.45	-
7.		LWM3	121.60	24.47	113.76	29.84
8.		LWCM3	133.85	25.50	123.55	30.48
9.		HPYM3+RNNM3	114.15	24.32	106.49	29.40
10.		LWM3+LWCM3	116.54	24.06	108.44	29.37
11.		HPYM3+LWM3	115.72	24.27	109.26	29.55
12.		HPYM3+LWM3+LWCM3	111.90	23.88	105.74	29.20
13.		HPYM3+LWM3+LWCM3+RNNM	105.81	23.42	100.82	28.64

Table 5 Out-of-vocabulary rate [%] in Experiment 2.

	Vocabulary size	OOV rate (%)	
		Valid	Test
Base model A	64,761	3.59	3.55
Base model B	64,152	1.77	1.61
Adapted model	100,677	0.83	0.68

data set.

The vocabulary size for each adapted model was 100,677. Furthermore, combined models of the adapted models were also constructed. For the Monte Carlo integration, S was set to 10. In these settings, other hyper parameters were optimized using the validation data set. Table 5 demonstrates OOV rate for both base LMs and adapted LMs.

5.2 Results

Table 6 shows the PPL and WER results for each condition. The difference of PPL in base models and in adapted models cannot be compared since each vocabulary size differs.

Base LMs constructed from training data set A are shown in lines 1-A to 4-A, and those constructed from training data set B are shown in lines 1-B to 4-B. The validation set was linguistically more complicated than the test set so low perplexity could be achieved in the test set. On the other hand, the test set was acoustically more complicated than the validation set, so WERs on the test set were higher than those of the validation set. In addition, training data set B was collected using the validation data set, so perplexity results for the validation set were relatively low compared to the test set. Among the base LMs, LW3 provided better results than RNN and HPY3, and the highest result was achieved by LW3+HPY3 for both the base model A and the base model B. These results are in agreement with experiment 1 and previous papers that state that LWLMs offer robust performance in multiple domains [16]–[18]. In both validation set and test set, the WER differences between LW3+HPY3 and HPY3 for both training data sets were statistically significant

($p < 0.01$).

Adapted LMs constructed from the base LMs are shown in lines 5 to 13. They show that each adapted LM was superior to the base LMs in terms of WER, so domain adaptation based on mixture modeling seem to be effective. LWCM3 was relatively weaker than LWM3 although LWM3 did achieve some improvement over HPYM3. It can be considered that the cross-mixture structure, which makes component parameters partially exchangeable between the base LMs, adversely impacts mixture modeling. In fact, individual LWLMs were trained by a sampling technique, so latent word space were not universal between LWLMs. It can be expected that LWCM3 is well constructed by increasing number of samples in LWLM training. On the other hand, the highest performance was achieved by LWM3+LWCM3 that combines an LWLM mixture model and an LWLM cross-mixture model although the WER differences between LWM3+LWCM3 to LWM3 were statistically not significant ($p > 0.05$). This is because an LWLM cross-mixture model has different characteristics than those of a standard LWLM mixture model. Thus, it seems that an LWLM cross-mixture model can mitigate domain mismatching between the target domain and each training data set. In both validation set and test set, the WER differences between LWM3+LWCM3 and HPYM3 were statistically significant ($p < 0.05$). LWM3+LWCM3 demonstrated higher performance than state-of-the-art RNNM in terms of PPL.

In addition, HPYM3+LWM3 outperformed HPYM3 and LWM3. It is considered to originate in having different characters in which observed words are generated depending on latent words in LWM3 while they are generated depending on last observed words in HPYM3. Among WFST-based one-pass decoding results, the highest performance was obtained by HPYM3+LWM3+LWCM3. The WER differences between HPYM3+LWM3+LWCM3 to HPYM3 were statistically significant ($p < 0.01$). In all results, HPYM3+LWM3+LWCM3+RNNM that fused HPYM3+LWM3+LWCM3 with RNNM in two-pass decoding presented the best performance. In terms of WER, statis-

tically significant performance improvements ($p < 0.05$) were achieved by HPYM3+LWM3+LWCM3+RNNM compared to HPYM3+RNNM in validation set and test set. This indicates that LWM3+LWCM3 could improve the state-of-the-art domain adapted systems that combines n-gram language modeling and RNN language modeling.

6. Conclusions

In this paper, LWLM mixture models and LWLM cross-mixture models were reported to enhance domain adaptation using out-of-domain text resources. Latent variables in LWLMs are represented as specific words that can be selected from the observed word space, so we can realize mixture modeling with consideration of the latent variable space. The LWLM mixture models can perform latent word space mixture that can mitigate a domain mismatch between a target domain and training data sets. Besides, the LWLM cross mixture models that construct a mixture model for each component in LWLMs can utilize partially matched text resources. The proposed models can be optimized using a small amount of target domain data as well as n-gram mixture modeling. Detailed experiments showed that LWLM mixture modeling outperformed n-gram mixture modeling. In addition, combination of the LWLM cross-mixture model and the LWLM mixture model yielded performance improvements, while using an LWLM cross-mixture model by itself offers little benefit.

References

- [1] R. Rosenfeld, "Two decades of statistical language modeling: Where do we go from here?," *Proceedings of the IEEE*, vol.88, pp.1270–1278, 2000.
- [2] J.T. Goodman, "A bit of progress in language modeling," *Computer Speech & Language*, vol.15, no.4, pp.403–434, 2001.
- [3] T. Brants, A.C. Papat, P. Xu, F.J. Och, and J. Dean, "Large language models in machine translation," *Proc. ACL*, pp.858–867, 2007.
- [4] J.R. Bellegarda, "Statistical language model adaptation: Review and perspectives," *Speech Communication*, vol.42, no.1, pp.93–108, 2004.
- [5] P. Koehn and J. Schroeder, "Experiments in domain adaptation for statistical machine translation," *Proc. Second Workshop on Statistical Machine Translation*, pp.224–227, 2007.
- [6] S. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE Transactions on Audio, Speech and Language Processing*, vol.35, no.3, pp.400–401, 1987.
- [7] R.M. Iyer and M. Ostendorf, "Modeling long distance dependence in language: topic mixtures versus dynamic cache models," *IEEE Transactions on Speech and Audio Processing*, vol.7, no.1, pp.30–39, 1999.
- [8] R. Iyer, M. Ostendorf, and H. Gish, "Using out-of-domain data to improve in-domain language models," *IEEE Signal Process. Lett.*, vol.4, no.8, pp.221–223, 1997.
- [9] G. Foster and R. Kuhn, "Mixture-model adaptation for SMT," *Proc. Second Workshop on Statistical Machine Translation*, pp.128–135, 2007.
- [10] B.-J. Hsu, "Generalized linear interpolation of language models," *Proc. ASRU*, pp.136–140, 2007.
- [11] X. Liu, M.J.F. Gales, and P.C. Woodland, "Context dependent language model adaptation," *Proc. INTERSPEECH*, pp.837–840, 2008.
- [12] T. Mikolov, S. Kombrink, L. Burget, J. Cernocky, and S. Khudanpur, "Extensions of recurrent neural network language model," *Proc. ICASSP*, pp.5528–5531, 2011.
- [13] Y. Shi, M. Larson, and C.M. Jonker, "K-component recurrent neural network language models using curriculum learning," *Proc. ASRU*, pp.1–6, 2013.
- [14] Y. Shi, M. Larson, and C.M. Jonker, "Recurrent neural network language model adaptation with curriculum learning," *Computer Speech & Language*, vol.33, no.1, pp.136–154, 2015.
- [15] K. Deschacht, J.D. Belder, and M.-F. Moens, "The latent words language model," *Computer Speech & Language*, vol.26, no.5, pp.384–409, 2012.
- [16] R. Masumura, H. Masataki, T. Oba, O. Yoshioka, and S. Takahashi, "Use of latent words language models in ASR: a sampling-based implementation," *Proc. ICASSP*, pp.8445–8449, 2013.
- [17] R. Masumura, T. Oba, H. Masataki, O. Yoshioka, and S. Takahashi, "Viterbi decoding for latent words language models using Gibbs sampling," *Proc. INTERSPEECH*, pp.3429–3433, 2013.
- [18] R. Masumura, T. Adami, T. Oba, H. Masataki, S. Sakauchi, and S. Takahashi, "N-gram approximation of latent words language models for domain robust automatic speech recognition," *IEICE Transaction. on Information and Systems*, vol.E99-D, no.10, pp.2462–2470, 2016.
- [19] S. Goldwater and T. Griffiths, "A fully Bayesian approach to unsupervised part-of-speech tagging," *Proc. ACL*, pp.744–751, 2007.
- [20] P. Blunsom and T. Cohn, "A hierarchical Pitman-Yor process HMM for unsupervised part of speech induction," *Proc. ACL*, pp.865–874, 1996.
- [21] T.R. Niesler and P.C. Woodland, "Combination word-based and category-based language models," *Proc. ICSLP*, vol.1, pp.220–223, 1996.
- [22] R.C. Moore and W. Lewis, "Intelligent selection of language model training data," *Proc. ACL*, pp.220–224, 2010.
- [23] F. Jelinek and R.L. Mercer, "Interpolated estimation of Markov source parameters from sparse data," *Pattern Recognition in Practice*, pp.381–397, 1980.
- [24] G. Casella and E.I. George, "Explaining the Gibbs sampler," *The American Statistician*, vol.46, no.3, pp.167–174, 1992.
- [25] R. Masumura, T. Asami, T. Oba, H. Masataki, and S. Sakauchi, "Mixture of latent words language models for domain adaptation," *Proc. INTERSPEECH*, pp.1425–1429, 2014.
- [26] A. Stolcke, "SRILM – an extensible language modeling toolkit," In *Proc. ICSLP*, vol.2, pp.901–904, 2002.
- [27] K. Maekawa, H. Koiso, S. Furui, and H. Isahara, "Spontaneous speech corpus of Japanese," *Proc. LREC*, pp.947–952, 2000.
- [28] G. Hinton, L. Deng, D. Yu, G.E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.N. Sainath, and B. Kingsbury, "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups," *Signal Processing Magazine*, vol.29, no.6, pp.82–97, 2012.
- [29] T. Hori, C. Hori, Y. Minami, and A. Nakamura, "Efficient WFST-based one-pass decoding with on-the-fly hypothesis rescore in extremely large vocabulary continuous speech recognition," *IEEE transactions on Audio, Speech and Language Processing*, vol.15, no.4, pp.1352–1365, 2007.
- [30] H. Masataki, D. Shibata, Y. Nakazawa, S. Kobashikawa, A. Ogawa, and K. Ohtsuki, "VoiceRex spontaneous speech recognition technology for contact-center conversations," *NTT Technical Review*, vol.5, no.1, pp.22–27, 2007.
- [31] T. Fuchi and S. Takagi, "Japanese morphological analyzer using word co-occurrence: JTAG," *Proc. COLING/ACL*, pp.409–413, 1998.
- [32] S. Huang and S. Renals, "Hierarchical Pitman-Yor language models for ASR in meetings," *Proc. ASRU*, pp.124–129, 2007.
- [33] A. Stolcke, "Entropy-based pruning of backoff language models," In *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, pp.270–274, 1998.

- [34] R. Masumura, S. Hahm, and A. Ito, "Language model expansion using webdata for spoken document retrieval," Proc. INTERSPEECH, pp.2133–2136, 2011.



Ryo Masumura received B.E., M.E., and Ph.D. degrees in engineering from Tohoku University, Sendai, Japan, in 2009, 2011, 2016, respectively. Since joining Nippon Telegraph and Telephone Corporation (NTT) in 2011, he has been engaged in research on speech recognition, spoken language processing, and natural language processing. He received the Student Award and the Awaya Kiyoshi Science Promotion Award from the Acoustic Society of Japan (ASJ) in 2011 and 2013, respectively, the Sendai

Section Student Awards The Best Paper Prize from the Institute of Electrical and Electronics Engineers (IEEE) in 2011, the Yamashita SIG Research Award from the Information Processing Society of Japan (IPSJ) in 2014, the Young Researcher Award from the Association for Natural Language Processing (NLP) in 2015, and the ISS Young Researcher's Award in Speech Field from the Institute of Electronic, Information and Communication Engineers (IEICE) in 2015. He is a member of the ASJ, the IPSJ, the NLP, the IEEE, and the International Speech Communication Association (ISCA).



Taichi Asami received B.E. and M.E. degrees in computer science from Tokyo Institute of Technology, Tokyo, Japan, in 2004 and 2006, respectively. Since joining Nippon Telegraph and Telephone Corporation (NTT) in 2006, he has been engaged in research on speech recognition and spoken language processing. He received the Awaya Kiyoshi Science Promotion Award and the Sato Prize Paper Award from the Acoustic Society of Japan (ASJ) in 2012 and 2014, respectively. He is a member of the ASJ,

the Institute of Electronics, Information and Communication Engineers (IEICE), Institute of Electrical and Electronics Engineers (IEEE), and the International Speech Communication Association (ISCA).



Takanobu Oba received B.E. and M.E. degrees from Tohoku University, Sendai, Japan, in 2002 and 2004, respectively. In 2004, he joined Nippon Telegraph and Telephone Corporation (NTT), where he was engaged in the research and development of spoken language processing technologies including speech recognition at the NTT Communication Science Laboratories, Kyoto, Japan. In 2012, he started the research and development of spoken applications at the NTT Media Intelligence Labora-

tories, Yokosuka, Japan. Since 2015, he has been engaged in development of spoken dialogue services at the NTT Docomo Corporation, Yokosuka, Japan. He received the Awaya Kiyoshi Science Promotion Award from the Acoustical Society of Japan (ASJ) in 2007. He received Ph. D. (Eng.) degree from Tohoku University in 2011. He is a member of the Institute of Electrical and Electronics Engineers (IEEE), the Institute of Electronics, Information, and Communication Engineers (IEICE) and the ASJ.



Hirokazu Masataki received B.E., M.E., and Ph.D. degrees from Kyoto University in 1989, 1991, and 1999, respectively. From 1995 to 1998, he worked with ATR Interpreted Telecommunications Research Laboratories, where specialized in statistical language modeling for large vocabulary continuous speech recognition. He joined Nippon Telegraph and Telephone Corporation (NTT) in 2004 and has been engaged in the practical use of speech recognition. He received the Maejima

Hisoka Award from the Tsushin-bunko Association in 2013, and the 54-th Sato Prize Paper Award from the Acoustic Society of Japan (ASJ) in 2014. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) and the ASJ.



Sumitaka Sakauchi received M.S. degree from Tohoku University in 1995 and Ph.D. degree from Tsukuba University in 2005. Since joining Nippon Telegraph and Telephone Corporation (NTT) in 1995, he has been engaged in research on acoustics, speech and signal processing. He is now Senior Manager in the Research and Development Planning Department of NTT. He received the Paper Award from the Institute of Electronics, Information and Communication Engineers (IEICE) in 2001, and

Awaya Kiyoshi Science Promotion Award from the Acoustic Society of Japan (ASJ) in 2003. He is a member of the IEICE and the ASJ.



Akinori Ito received B.E., M.E., and Ph.D. degrees from Tohoku University, Sendai, Japan. Since 1992, he has worked with Research Center for Information Sciences and Education Center for Information Processing, Tohoku University. He was with the Faculty of Engineering, Yamagata University, from 1995 to 2002. From 1998 to 1999, he worked with the College of Engineering, Boston University, MA, USA, as a Visiting Scholar. He is now a Professor of the Graduate School of Engineering, Tohoku University.

He is engaged in spoken language processing, statistical text processing, and audio signal processing. He is a member of the Acoustic Society of Japan, the Information Processing Society of Japan, and the IEEE.