IEICE TRANS. INF. & SYST., VOL.E99–D, NO.10 OCTOBER 2016

PAPER Special Section on Recent Advances in Machine Learning for Spoken Language Processing

N-gram Approximation of Latent Words Language Models for Domain Robust Automatic Speech Recognition

Ryo MASUMURA[†a)], **Taichi ASAMI**[†], **Takanobu OBA**[†*], **Hirokazu MASATAKI**[†], **Sumitaka SAKAUCHI**[†], ***and*** **Satoshi TAKAHASHI**[†], ***Members***

SUMMARY This paper aims to improve the domain robustness of language modeling for automatic speech recognition (ASR). To this end, we focus on applying the latent words language model (LWLM) to ASR. LWLMs are generative models whose structure is based on Bayesian soft class-based modeling with vast latent variable space. Their flexible attributes help us to efficiently realize the effects of smoothing and dimensionality reduction and so address the data sparseness problem; LWLMs constructed from limited domain data are expected to robustly cover unknown multiple domains in ASR. However, the attribute flexibility seriously increases computation complexity. If we rigorously compute the generative probability for an observed word sequence, we must consider the huge quantities of all possible latent word assignments. Since this is computationally impractical, some approximation is inevitable for ASR implementation. To solve the problem and apply this approach to ASR, this paper presents an n-gram approximation of LWLM. The n-gram approximation is a method that approximates LWLM as a simple back-off n-gram structure, and offers LWLM-based robust one-pass ASR decoding. Our experiments verify the effectiveness of our approach by evaluating perplexity and ASR performance in not only in-domain data sets but also out-of-domain data sets.

key words: *language models, domain robustness, latent words language models, n-gram approximation, automatic speech recognition*

1. Introduction

Language models (LMs) are necessary for modern automatic speech recognition (ASR) systems. One of main goals of language modeling research is domain robustness [1]. For example, academic lectures, call center recordings and meeting domains have different linguistic properties. In fact, LM performance strongly depends on the quantity and quality of the training data. In practical ASR systems, LMs are often required to robustly predict the probability of unobserved linguistic phenomena even though the target training data is limited. Also, LMs constructed from out-of-domain data are required to robustly work for unknown domains since ideal training data is seldom available. This paper, therefore, aims to improve the domain robustness of language modeling for ASR.

For domain robust language modeling, it is necessary to tackle the data sparseness problem for which there are two representative approaches; smoothing and dimensionality reduction. Smoothing is a fundamental technique to mitigate the data sparseness problem in n-gram modeling [2]. Various smoothing methods have been proposed and Kneser-Ney smoothing is known to be one of the most accurate methods [3]. The hierarchal Pitman-Yor LMs (HPYLMs), whose smoothing is based on the Pitman-Yor process, can slightly outperform the Kneser-Ney method in ASR [4], [5]. The other solution to the data sparseness problem is based on dimensionality reduction. Instances include class-based n-gram modeling [6]. Similar ideas have been employed in decision tree LMs [7] and random forest LMs [8], in which context information is clustered into some groups. Also, neural network LMs and recurrent neural network LMs (RNNLMs) can reduce dimensionality on the basis of learning the distributed representation of words [9]–[11].

To further advance towards domain robust ASR, this paper focuses on the latent words LMs (LWLMs) recently proposed in the machine learning area [12]. LWLMs are generative models that have latent variables called latent words. LMLMs can employ a smoothing effect based on Bayesian modeling as well as HPYLMs. In addition, LWLMs share a soft clustering structure with Bayesian hidden Markov models (HMMs) [13], [14] and the Bayesian class-based LMs [15], [16]. However, in contrast to those models, LWLMs have vast latent variable space about as large as the vocabulary of the training data. Thus, LWLMs are trained by taking into account the latent words and it is this advance that allows LWLMs to tackle the data sparseness problem. These flexible attributes help us to efficiently realize smoothing and dimensionality reduction simultaneously, so LWLMs are expected to robustly cover multiple domains in ASR.

However, an LWLM is difficult to directly use for ASR because of its soft clustering structure and vast latent variable space. In the case of a hard clustering structure such as standard class-based n-gram models [6], class assignment can be identified uniquely. The use of the soft clustering structure, however, forces us to consider all possible class assignments. In fact, all words can be generated from all latent variables in the LWLM approach. Additionally, the possible class assignments are innumerable because the number of latent variables corresponds to vocabulary size. Thus, if the length of an observed word sequence is L and the number of latent words is $|\mathcal{V}|$, the number of possible class assignments is $|\mathcal{V}|^L$. It is impractical for modern ASR systems

Manuscript received February 3, 2016.
Manuscript revised May 21, 2016.
Manuscript publicized July 14, 2016.
[†]The authors are with NTT Media Intelligence Laboratories, NTT Corporation, Yokosuka-shi, 239–0847 Japan.
[*]Presently, with NTT Docomo Corporation.
a) E-mail: masumura.ryo@lab.ntt.co.jp
DOI: 10.1587/transinf.2016SLP0014

to rigorously compute the generative probability of a word sequence. Also, one-pass ASR decoding is impossible even though the computation is possible.

To overcome these problems, this paper presents an n-gram approximation of LWLMs. Our idea is to use a simple back-off n-gram structure to approximate an LWLM and to use the approximated model for realizing one-pass ASR decoding. As LWLMs are generative models, a lot of observed word sequences can be generated on the basis of the generative process without calculating definitive generative probabilities. The generated word sequences include multiple phrases that are not included in the original training data. Therefore the n-gram model trained from them would be able to accurately perform over multiple domains, while that is the approximation model. Furthermore, the criterion for training an LWLM greatly differs from that for a standard word n-grams model, so interpolating both the approximated n-gram model and the standard n-gram model would effectively improve ASR performance.

Our approach is related to technologies that recast LMs, whose structure is complex, as back-off n-gram variants [17]–[23]. Complex LMs are difficult to use in ASR due to their computation complexity and poor compatibility with ASR decoding. Recent ASR decoders are based on the weighted finite state transducer (WFST). Although back-off n-gram models can be converted into WFST, most LMs are too complex to suit WFST decoding. The existing solution is to use them for rescoring recognition hypotheses (n-best lists or confusion networks) generated in WFST-based decoding. However, rescoring is sensitive to the performance of the generated recognition hypotheses. Therefore, to implement WFST-based decoding, techniques are needed that can convert complex LMs into back-off n-gram structures. The previous studies report that ASR performance can be improved by applying the converted models to WFST-based decoding although the converted models are inferior to the original models. This paper also uses the n-gram approximation approach since LWLMs cannot even calculate definitive generative probabilities.

In fact, this paper is an extended study of our previous work [24], which showed preliminary results of our approach. In this paper, we extend our evaluation in which n-gram approximation approaches based on RNNLMs and HPYLMs are also examined [21]–[23]. In addition, we also reveal detailed properties for investigating the number of n-gram entries and the n-gram hit rates of each approximated model. Additionally, we can use the entropy pruning technique to reduce model size of the approximated model although the model size becomes excessive as many words are sampled to realize an adequate approximation. Therefore, this paper also investigates the relationship between model size and the performance of the pruned model variants.

This paper is organized as follows. First, Sect. 2 briefly describes LWLMs. Section 3 explains our approach. Sections 4 and 5 describe a perplexity evaluation and an ASR evaluation, respectively. Section 6 concludes this paper.

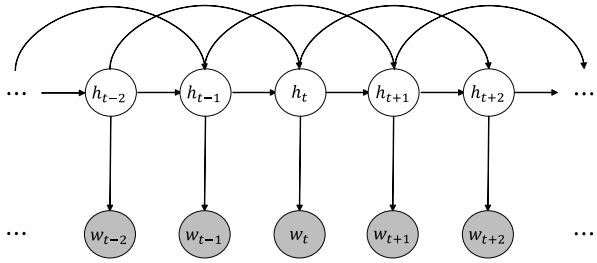


Fig. 1 Model structure of LWLMs.

2. Latent Words Language Models

2.1 Definition

LWLMs are generative models that set a latent variable for every observed word. A graphic rendering of LWLM is shown in Fig. 1. The gray circles denote observed words and the white circles denote latent variables. In the generative process of LWLM, a latent variable, called latent word h_t, is generated depending on the transition probability distribution given context $\boldsymbol{l}_t = h_{t-n+1}, \ldots, h_{t-1}$ where n is an n-gram order. Next, observed word w_t is generated depending on the emission probability distribution given latent word h_t, i.e.,

$$h_t \sim P(h_t|\boldsymbol{l}_t, \boldsymbol{\Theta}_{\mathtt{lw}}), \tag{1}$$

$$w_t \sim P(w_t|h_t, \boldsymbol{\Theta}_{\mathtt{lw}}), \tag{2}$$

where $\boldsymbol{\Theta}_{\mathtt{lw}}$ is a model parameter of LWLM. $P(h_t|\boldsymbol{l}_t, \boldsymbol{\Theta}_{\mathtt{lw}})$ is expressed as an n-gram model for latent words, and $P(w_t|h_t, \boldsymbol{\Theta}_{\mathtt{lw}})$ models the dependency between the observed word and the latent word.

An important property of LWLMs is that the latent word is expressed as a specific word that can be selected from an entire vocabulary $\mathcal{V}$. Thus, the number of latent words is the same as the vocabulary size $|\mathcal{V}|$. This is the reason the latent variable is called as a latent word.

2.2 Bayesian LWLMs

In the Bayesian approach, LWLM produces the following generative probability for observed words $\boldsymbol{w} = w_1, \cdots, w_N$:

$$\begin{aligned} P(\boldsymbol{w}) &= \int \sum_{\boldsymbol{h}} P(\boldsymbol{w}|\boldsymbol{h}, \boldsymbol{\Theta}_{\mathtt{lw}}) P(\boldsymbol{h}|\boldsymbol{\Theta}_{\mathtt{lw}}) P(\boldsymbol{\Theta}_{\mathtt{lw}}) d\boldsymbol{\Theta}_{\mathtt{lw}}, \\ &= \int \prod_{t=1}^{N} \sum_{h_t \in \mathcal{V}} P(w_t|h_t, \boldsymbol{\Theta}_{\mathtt{lw}}) P(h_t|\boldsymbol{l}_t, \boldsymbol{\Theta}_{\mathtt{lw}}) P(\boldsymbol{\Theta}_{\mathtt{lw}}) d\boldsymbol{\Theta}_{\mathtt{lw}}, \end{aligned} \tag{3}$$

where $\boldsymbol{h} = h_1, \cdots, h_N$ is a latent word assignment. The Bayesian approach takes account of all possible model parameters. As the integral with respect to $\boldsymbol{\Theta}_{\mathtt{lw}}$ is analytically intractable, a sampling technique is used as a feasible approximation. Eq. (3) is approximated as:

$$P(\boldsymbol{w}) \simeq \frac{1}{M}\sum_{m=1}^{M} P(\boldsymbol{w}|\boldsymbol{\Theta}_{\mathtt{lw}}^{m}) = \frac{1}{M}\sum_{m=1}^{M}\prod_{t=1}^{N}\sum_{h_t\in\mathcal{V}} P(w_t|h_t,\boldsymbol{\Theta}_{\mathtt{lw}}^{m})P(h_t|\boldsymbol{l}_t,\boldsymbol{\Theta}_{\mathtt{lw}}^{m}), \quad (4)$$

where $\boldsymbol{\Theta}_{\mathtt{lw}}^{m}$ means m-th point estimated model parameter. The generative probability can be approximated using M instances of $\boldsymbol{\Theta}_{\mathtt{lw}}^{m}$. Although we can only use one instance for the approximation, we conduct ensemble modeling ($M > 1$). In fact, the ensemble of several models is effective for LMs such as random class-based LMs [25] and random forest LMs [8].

LWLM has a similar structure to the standard class-based n-gram model. The latent word corresponds, approximately, to the class of the standard class-based n-gram model [6]. LWLM has a soft word clustering structure that differs from a simple hard word clustering structure in the standard class-based n-gram model. In the hard word clustering structure, one word belongs to only one class. In the soft word clustering structure, on the other hand, one word belongs to multiple classes. Strictly speaking, each word belongs to all classes in LWLM. In addition, LWLM has vast class space about as large as the vocabulary while the number of class in the standard class-based n-gram model is often defined as several hundreds or thousands.

2.3 Training

LWLM is trained using word sequence $\boldsymbol{w} = w_1, \cdots, w_N$. In LWLM training, we have to infer the latent word assignment $\boldsymbol{h} = h_1, \cdots, h_N$ behind $\boldsymbol{w}$. In fact, we infer latent word assignments $\boldsymbol{h}_1, \cdots, \boldsymbol{h}_M$. Once a latent word assignment $\boldsymbol{h}_m$ is defined, $P(w_t|h_t,\boldsymbol{\Theta}_{\mathtt{lw}}^{m})$ and $P(h_t|\boldsymbol{l}_t,\boldsymbol{\Theta}_{\mathtt{lw}}^{m})$ can be calculated.

To estimate the latent word assignments, Gibbs sampling is suitable. Gibbs sampling samples a new value for the latent word in accordance with its distribution and places it at position k in $\boldsymbol{h}$. The conditional probability distribution of possible values for latent word h_t is given by:

$$P(h_t|\boldsymbol{w},\boldsymbol{h}_{-t}) \propto P(w_t|h_t,\boldsymbol{\Theta}_{\mathtt{lw},-t})\prod_{j=t}^{t+n-1} P(h_j|\boldsymbol{l}_j,\boldsymbol{\Theta}_{\mathtt{lw},-t}), \quad (5)$$

where $\boldsymbol{h}_{-t}$ represents all latent words except for h_t. In the sampling procedure, $P(h_t|\boldsymbol{l}_t,\boldsymbol{\Theta}_{\mathtt{lw},-t})$ and $P(w_t|h_t,\boldsymbol{\Theta}_{\mathtt{lw},-t})$ can be calculated from $\boldsymbol{w}$ and $\boldsymbol{h}_{-t}$.

The transition probability distribution and the emission probability distribution are calculated on the basis of their prior distributions. For the transition probability distribution, this paper uses a hierarchical Pitman-Yor prior. $P(h_t|\boldsymbol{l}_t,\boldsymbol{\Theta}_{\mathtt{lw}})$ is given as:

$$P(h_t|\boldsymbol{l}_t,\boldsymbol{\Theta}_{\mathtt{lw}}) = P(h_t|\boldsymbol{l}_t,\boldsymbol{h}), \quad (6)$$

$$P(h_t|\boldsymbol{l}_t,\boldsymbol{h}) = \frac{c(h_t,\boldsymbol{l}_t) - d_{|\boldsymbol{l}_t|}s(h_t,\boldsymbol{l}_t)}{\theta_{|\boldsymbol{l}_t|}+c(\boldsymbol{l}_t)} + \frac{\theta_{|\boldsymbol{l}_t|}+d_{|\boldsymbol{l}_t|}s(\boldsymbol{l}_t)}{\theta_{|\boldsymbol{l}_t|}+c(\boldsymbol{l}_t)}P(h_t|\pi(\boldsymbol{l}_t),\boldsymbol{h}), \quad (7)$$

Algorithm 1 : Random sampling based on trained LWLM.

Input: Model parameters $\boldsymbol{\Theta}_{\mathtt{lw}}^{1}, \cdots, \boldsymbol{\Theta}_{\mathtt{lw}}^{M}$, number of sampled words N
Output: Sampled words $\boldsymbol{w}$
1: $\boldsymbol{l}_1$ = <s>
2: **for** $t = 1$ to N **do**
3: $\quad m \sim P(m) = \frac{1}{\mathrm{M}}$
4: $\quad h_t \sim P(h_t|\boldsymbol{l}_t,\boldsymbol{\Theta}_{\mathtt{lw}}^{m})$
5: $\quad w_t \sim P(w_t|h_t,\boldsymbol{\Theta}_{\mathtt{lw}}^{m})$
6: **end for**
7: **return** $\boldsymbol{w} = w_1, \cdots, w_N$

where $\pi(\boldsymbol{l}_t)$ is the shortened context obtained by removing the earliest word from $\boldsymbol{l}_t$. $c(h_t,\boldsymbol{l}_t)$ and $c(\boldsymbol{l}_t)$ are counts calculated from a latent word assignment $\boldsymbol{h}$. $s(h_t,\boldsymbol{l}_t)$ and $s(\boldsymbol{l}_t)$ are calculated from a seating arrangement defined by the Chinese restaurant franchise representation of the Pitman-Yor process [26]. $d_{|\boldsymbol{l}_t|}$ and $\theta_{|\boldsymbol{l}_t|}$ are discount and strength parameters of the Pitman-Yor process, respectively. Moreover, we use a Dirichlet prior for the emission probability distribution [27]. $P(w_t|h_t,\boldsymbol{\Theta}_{\mathtt{lw}})$ is given as:

$$P(w_t|h_t,\boldsymbol{\Theta}_{\mathtt{lw}}) = P(w_t|h_t,\boldsymbol{w},\boldsymbol{h}), \quad (8)$$

$$P(w_t|h_t,\boldsymbol{w},\boldsymbol{h}) = \frac{c(w_t,h_t)+\alpha P(w_t)}{c(h_t)+\alpha}, \quad (9)$$

where $P(w_t)$ is the maximum likelihood estimation value of unigram probability in the training data $\boldsymbol{w}$. $c(w_t,h_t)$ and $c(h_t)$ are counts calculated from $\boldsymbol{w}$ and latent word assignment $\boldsymbol{h}$. A hyper parameter α can be optimized via a validation set.

3. N-gram Approximation of LWLMs

3.1 Sampling Based Approximation

Our idea is to convert trained LWLMs into the back-off n-gram structure. The n-gram approximation of a trained LWLM is based on random sampling of observed words. As LWLM is a generative model, it can generate latent words and observed words. Therefore, we can easily sample a lot of observed words and construct a back-off n-gram model from them. The random sampling is based on Algorithm 1.

In line 1, $\boldsymbol{l}_1$ is initialized as sentence head symbol <s>. Through iterations of lines 2-6, we can obtain a large number of word sequences. With N iterations, we can generate N latent words, and N observed words. The N observed words are used only for back-off n-gram model estimation. We define the probability distribution of the approximated model as $P(w_t|\boldsymbol{u}_t,\boldsymbol{\Theta}_{\mathtt{lwng}})$ where $\boldsymbol{u}_t$ means context information $w_{t-n+1}, \ldots, w_{t-1}$, n is n-gram order, and $\boldsymbol{\Theta}_{\mathtt{lwng}}$ represents the model parameter. In fact, any back-off n-gram structure, including HPYLMs, can be used for the approximation.

3.2 Linear Interpolation

It can be considered that the approximated model has properties that differ from the equivalent n-gram model directly

constructed from training data because the sampled data derived from the trained LWLM includes various linguistic phenomena that are not present in the original training data. Therefore, we can expect that interpolating both LMs will effectively improve ASR performance. The probability distribution of interpolated n-gram model $P(w_t|\boldsymbol{u}_t, \boldsymbol{\Theta}_{\texttt{mix}})$ is defined as:

$$P(w_t|\boldsymbol{u}_t, \boldsymbol{\Theta}_{\texttt{mix}}) = \lambda P(w_t|\boldsymbol{u}_t, \boldsymbol{\Theta}_{\texttt{ng}}) + (1-\lambda)P(w_t|\boldsymbol{u}_t, \boldsymbol{\Theta}_{\texttt{lwng}}), \quad (10)$$

where $P(w_t|\boldsymbol{u}_t, \boldsymbol{\Theta}_{\texttt{ng}})$ means the probability distribution of the n-gram model constructed from the original training data. Interpolation weight λ can be optimized by using a validation data set. In fact, the interpolated model can be approximately represented as a single back-off n-gram structure [28]. The calculation can be conducted by adding both smoothed n-gram probabilities with mixture weights and recomputing back-off probabilities.

3.3 Entropy Pruning for Reducing Model Size

Our approach has a weakness in that the approximated model size becomes excessive as many words are sampled to realize an adequate approximation. To reduce model size, the entropy pruning technique can be used [29]. The entropy pruning can efficiently reduce the n-gram entries in the back-off n-gram structure. The decision as to whether the n-gram entry $(\boldsymbol{u}_t, w_t)$ should be retained or pruned is based on the relative entropy between non-pruned model $P(w_t|\boldsymbol{u}_t, \boldsymbol{\Theta})$ and pruned model $(w_t|\boldsymbol{u}_t, \boldsymbol{\Theta}')$. Relative entropy $D(\boldsymbol{\Theta}||\boldsymbol{\Theta}')$ is calculated by:

$$D(\boldsymbol{\Theta}||\boldsymbol{\Theta}') = P(\boldsymbol{u}_t) \sum_{w_t} P(w_t|\boldsymbol{u}_t, \boldsymbol{\Theta}) \log \frac{P(w_t|\boldsymbol{u}_t, \boldsymbol{\Theta}')}{P(w_t|\boldsymbol{u}_t, \boldsymbol{\Theta})}, \quad (11)$$

where $P(\boldsymbol{u}_t)$ is computed using the chain rule and lower order n-gram model. A threshold for the relative entropy should be defined in accordance with the actual use case.

4. Experiment 1: Perplexity Evaluation

4.1 Setups

First, we conducted experiments using the Penn Treebank corpus, one of the most widely used sources for evaluating LMs [30]. Sections 0-20 were used as a training set (Train), sections 21 and 22 were used as a validation set (Valid), and Sect. s 23 and 24 were used as a test set (Test A). This selection matches those of many previous works. In addition, we prepared human-human discussion text data (Test B) for evaluations in a domain different from the training set. Each vocabulary was limited to 10K words and there were no out-of-vocabulary words. Table 1 shows details.

In this evaluation, we constructed the following LMs.

- `MKN5`: Word-based 5-gram LM with interpolated Kneser-Ney smoothing constructed from training set [3].
- `HPY5`: Word-based 5-gram HPYLM constructed from the training set [5]. For the training, we used 200 iterations for burn-in, and collected 10 sets of samples.
- `C-HPY5`: Hard class-based 5-gram HPYLM constructed from training set. Brown clustering was used for deciding word class [6]. The class size was 1K.
- `RNN`: Word-based RNNLM [10]. The hidden layer size was set as 200 by referring to a preliminary experiment.
- `A-HPY5`: Word-based 5-gram HPYLM constructed from data generated on the basis of `HPY5`.
- `RNN5`: Word-based 5-gram HPYLM constructed from data generated on the basis of `RNN` [21].
- `LW5`: Word-based 5-gram HPYLM constructed from data generated on the basis of 5-gram LWLM constructed from training set. For training LWLM, we used 500 iterations for burn-in, and collected 10 samples.

Table 1 Data sets in Experiment 1.

	Domain	# of words
Train	Penn Treebank	929,589
Valid	Penn Treebank	70,390
Test A	Penn Treebank	78,669
Test B	Human-Human Discussion	50,507

In addition, we employed several mixed models constructed by linearly interpolating the above LMs. For example, `HPY5+LW5` is a mixed model constructed from `HPY5` and `LW5`. The mixture weights were optimized using a validation set on the basis of the EM algorithm. Other hyper parameters were also optimized using the validation set.

4.2 Results

For the n-gram approximation approaches (`A-HPY5`, `RNN5`, `LW5`), the generated data size is related to the performance of the approximated models. Therefore, we investigated the relationship between the data size generated by random sampling and perplexity (PPL) reduction in the validation set and each test set. We constructed each approximated model, and mixed models (`RNN5+HPY5`, `LW5+HPY5`) by varying the generated data size and computed the corresponding PPL. We plot the results along with the results of `HPY5` and `RNN` in Fig. 2, where the horizontal axis is in log-scale.

Figure 2 shows that the PPL of each model was reduced as the generated data increased. In `A-HPY5`, PPL converged with just a small amount of generated data because HPYLM has a simple model structure. `A-HPY5` matched the performance of `HPY5` in each data set when a lot of data was generated. On the other hand, in `RNN5` and `LW5`, more generated data was necessary for PPL convergence than in `A-HPY5`. `LW5` outperformed `A-HPY5` and `RNN5` when a lot of data was generated. `RNN5` could not match the performance of the original `RNN`. This is because the back-off n-gram structure cannot take into account long-range context although RNNLM can consider the long-range context dis-

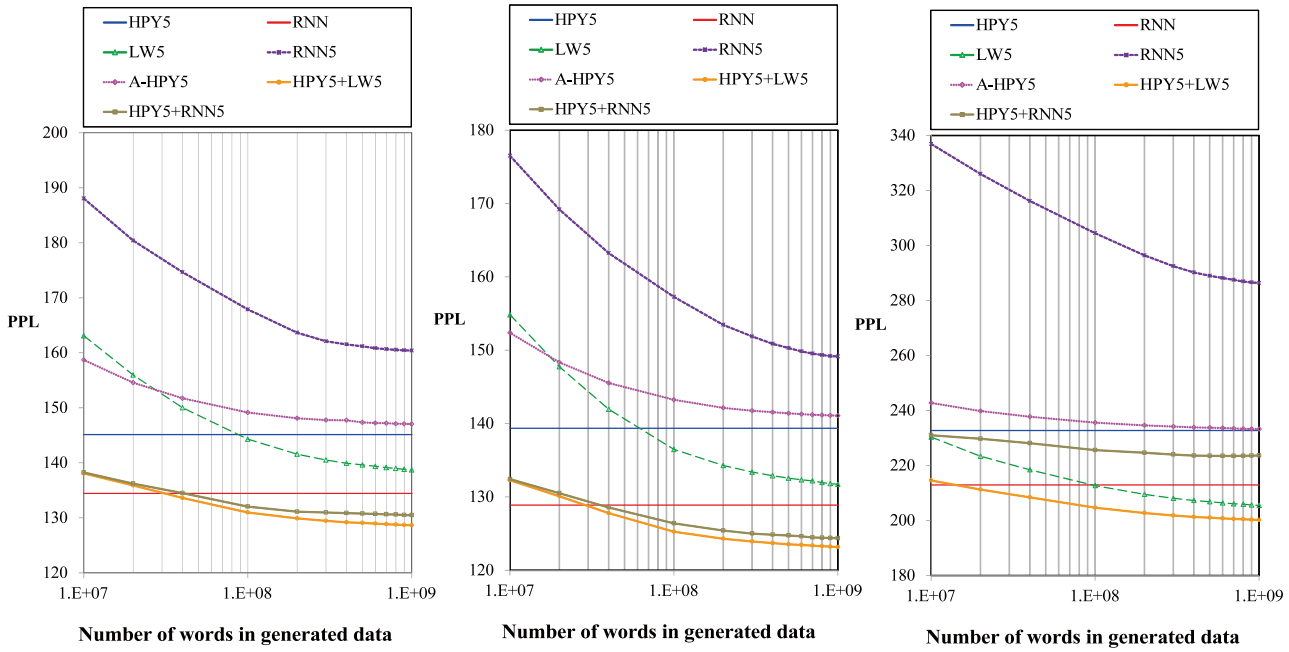


Fig. 2 Relations between data size generated by random sampling and perplexity in Experiment 1.

Table 2 Perplexity results in Experiment 1.

		Valid	Test A	Test B
1.	MKN5	148.0	141.2	238.6
2.	HPY5	145.1	139.3	232.7
3.	C-HPY5	150.8	142.2	237.0
4.	A-HPY5	147.1	141.1	233.3
5.	RNN5	160.4	150.4	286.4
6.	LW5	**138.7**	**131.7**	**205.5**
7.	HPY5+RNN5	130.5	124.5	226.7
8.	HPY5+LW5	**128.6**	**123.1**	**200.8**
9.	RNN	134.4	128.9	212.9
10.	HPY5+RNN	111.4	107.9	180.6
11.	HPY5+RNN5+RNN	113.4	109.2	186.8
12.	HPY5+LW5+RNN	**109.8**	**105.4**	**175.8**

tributed representation of words.

In addition, both mixed models also experienced improvements in PPL performance as the generated data size increased. Although the isolated use of RNN5 was not effective, its combination with HPY5 yielded improved PPL performance. Also, LW5 performance was improved through combination with HPY5. This combination was effective because the data generated on the basis of RNN or LW have different attributes from the original training data. The combination of HPY5 and A-HPY5 did not improve performance because they were almost the same (the results are not shown in Fig. 2).

Next, we investigated PPL performance for all LMs; the generated data size was set to 1 giga (1.E+09) words for A-HPY5, RNN5 and LW5. The results are shown in Table 2.

Lines 1-6 show the results for the back-off n-gram structure. In each data set, LW5 achieved the best results. Note that C-HPY5 could not achieve better results than LW5. Thus, the simple hard clustering structure does not improve PPL performance for either in-domain data or out-of-domain data, and LWLM structure (soft clustering with vast class size) seems to be effective for domain robustness.

Lines 7 and 8 show the results for the mixed n-gram models that can also be expressed as simple back-off n-gram structures. The combination of HPY5 and RNN5 or LW5 could improve PPL performance more than their isolated use. In each data set, HPY5+LW5 was superior to HPY5+RNN5. It can be considered that the performance was improved because LW5 had different attributes from HPY5.

Lines 9-12 show the results for RNN and its combination with the back-off n-gram structure. RNN outperformed other isolated models (lines 1-6) for the validation set and test set A. On the other hand, LW5 was superior to RNN for test set B although LW5 has a back-off n-gram structure. The combinations of RNN with the models with back-off n-gram structure were effective. In each data set, the best results were obtained by HPY5+LW5+RNN. This shows that performance can be improved by n-gram approximation of LWLM even if RNNLM is also used.

Additionally, we investigated the properties of each approximated model to reveal that LW5 was more effective than A-HPY5 and RNN5. Table 3 shows the number of 3- and 5-gram entries in each model and n-gram hit rate for the validation set and each test set. The hit rate represents the percentage of n-gram entries in the reference data that are explicitly listed in the LMs. We calculated 3-gram hit rate ($N \geq 3$), which includes the high order (4-gram and 5-gram) hit rate and 5-gram hit rate ($N = 5$); the generated data sizes of each model were set to 10M, 100M and 1000M.

Table 3 shows that LW5 had a lot more n-gram entries than A-HPY5 and RNN5 for the same generated data size. This means that random sampling based on LWLM can gen-

Table 3 Number of n-gram entries and n-gram hit rate [%] results in Experiment 1.

	Data size	# of 3-gram	# of 5-gram	Valid		Test A		Test B	
				N ≥ 3	N = 5	N ≥ 3	N = 5	N ≥ 3	N = 5
HPY5	930K	586558	737952	39.58	8.62	40.26	7.43	29.01	0.81
A-HPY5	10M	5663883	8848534	49.33	9.80	50.51	8.77	44.62	1.84
	100M	40649386	83207920	62.53	14.40	64.28	13.71	61.06	4.91
	1000M	274813830	760971607	74.83	19.83	76.61	20.68	75.61	10.54
RNN5	10M	4905245	7768561	47.75	8.75	48.86	7.92	37.49	1.07
	100M	33458232	72434311	62.30	13.54	64.02	12.92	55.24	3.18
	1000M	219833882	665415678	74.77	19.98	76.75	19.83	71.62	7.62
LW5	10M	6608797	9409511	50.40	9.33	51.62	8.27	46.22	1.69
	100M	48464818	90136805	65.28	14.16	67.45	13.58	64.77	4.74
	1000M	319956811	841974035	**77.45**	**20.98**	**79.41**	**21.14**	**78.84**	**10.82**

erate a greater variety of linguistic phenomena than HPYLM or RNNLM. In addition, the 3-gram hit rate and 5-gram hit rate of LW5 were superior to those of A-HPY5 and RNN5 for each data set. This means that random sampling based on LWLM can generate words that are actually in the data set. These results show that an approximated model based on effective random sampling can perform robustly in multiple domains.

5. Experiment 2: ASR Evaluation

5.1 Setups

The second experiments used the Corpus of Spontaneous Japanese (CSJ) for ASR evaluation [31]. We divided CSJ into a training set (Train), a small validation set (Valid), and a test set (Test A). Vocabulary size of the training set was 83,536. The validation set was used in optimizing the hyper parameters of the LMs. In addition, a contact center dialog task (Test B) and a voice mail task (Test C) were prepared for evaluation in out-of-domain environments. In the contact center dialogue task, two speakers, an operator and a customer, talked to each other as in call center dialogs. Twenty four phone calls (24 operator channels and 24 customer channels) were used in the evaluation. In the voice mail task, a person left small voice messages using a smart phone, and 237 messages were used in the evaluation. Table 4 shows details.

For speech recognition evaluation, we prepared an acoustic model on the basis of hidden Markov models with deep neural networks (DNN-HMM) [32]. The DNN-HMM had eight hidden layers with 2048 nodes and was trained using the CSJ training set. The speech recognition decoder was VoiceRex, a WFST-based decoder [33], [34]. JTAG was used as the morpheme analyzer to split sentences into words [35].

In this evaluation, we constructed the following LMs.

- MKN3: Word-based 3-gram LM with interpolated Kneser-Ney smoothing constructed from training set [3].
- HPY3: Word-based 3-gram HPYLM constructed from the training set [5]. For the training, we used 200 iterations for burn-in, and collected 10 samples.

Table 4 Data sets in Experiment 2.

	Domain	# of words
Train	Lecture	7,317,392
Valid	Lecture	28,046
Test A	Lecture	27,907
Test B	Contact center	24,665
Test C	Voice mail	21,044

- C-HPY3: Hard class-based 3-gram HPYLM constructed from training set. Brown clustering was used for deciding word class. The class size was 5K [6].
- RNN: Class-based RNNLM with 500 hidden nodes and 500 classes [11].
- A-HPY3: Word-based 3-gram HPYLM constructed from 1000M data generated on the basis of HPY3.
- RNN3: Word-based 3-gram HPYLM constructed from 1000M data generated on the basis of RNN.
- LW3: Word-based 3-gram HPYLM constructed from data generated on the basis of 3-gram LWLM constructed from training set. For the training of LWLM, we used 500 iterations for burn-in and collected 10 samples.

In addition, we examined several mixed models constructed from the above LMs by linear interpolation. The mixture weights were optimized using the validation set and the EM algorithm. Other hyper parameters were also optimized using the validation set. These LMs, except for RNN, can be represented using ARPA format, a standard back-off n-gram format, and they can be directly introduced to WFST decoders. RNN can be used as a rescoring model that is introduced after decoding. For rescoring, we generated 1000-best lists in the decoding pass. For example, HPY3+LW3 was used for decoding to generate the recognition hypotheses when we examined HPY3+LW3+RNN.

5.2 Results

Table 5 shows the PPL and word error rate (WER) results for each condition. PPL was only evaluated in RNN since RNNLMs cannot be applied to ASR directly.

Lines 1-6 show the results for the back-off n-gram structure. HPY3 outperformed MKN3 and C-HPY3 in terms of PPL and WER. Although C-HPY3 yielded dimensionality

Table 5 Perplexity and word error rate [%] results in Experiment 2.

		Valid (In-Domain)		Test A (In-Domain)		Test B (Out-Of-Domain)		Test C (Out-Of-Domain)	
		PPL	WER	PPL	WER	PPL	WER	PPL	WER
1.	MKN3	81.38	19.98	69.36	24.79	167.61	38.67	189.93	32.00
2.	HPY3	**79.32**	19.74	67.50	24.67	158.13	38.29	175.63	31.69
3.	C-HPY3	82.97	19.91	69.36	24.59	158.92	38.39	180.89	32.17
4.	A-HPY3	82.69	20.20	70.43	25.23	161.56	38.82	177.61	32.04
5.	RNN3	98.65	21.63	82.23	26.24	153.89	39.32	163.99	31.96
6.	LW3	79.57	**19.61**	**66.93**	**24.54**	**141.34**	**36.93**	**147.87**	**30.42**
7.	HPY3+RNN3	77.96	19.53	66.09	24.26	143.88	37.60	149.81	30.18
8.	HPY3+LW3	**72.86**	**18.65**	**62.05**	**23.58**	**134.65**	**35.99**	**141.23**	**28.74**
9.	RNN	69.49	-	60.78	-	145.05	-	158.57	-
10.	HPY3+RNN	64.01	18.53	55.84	23.45	122.52	37.45	142.62	30.89
11.	HPY3+RNN3+RNN	63.76	18.41	55.77	23.12	119.00	36.85	138.54	29.24
12.	HPY3+LW3+RNN	**61.56**	**17.85**	**53.36**	**22.68**	**114.71**	**35.36**	**133.09**	**28.06**

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

	Size	Valid (In-Domain)		Test A (In-Domain)		Test B (Out-Of-Domain)		Test C (Out-Of-Domain)	
		PPL	WER	PPL	WER	PPL	WER	PPL	WER
HPY3	322 M	79.32	19.74	67.50	24.67	158.13	38.29	175.63	31.69
HPY3+LW3	22.0 G	**72.86**	18.65	**62.05**	23.58	**134.65**	**35.99**	**141.23**	28.74
	9.8 G	73.39	18.70	62.86	**23.56**	135.02	36.01	143.05	**28.63**
	3.9 G	73.74	**18.53**	63.21	23.60	135.47	36.05	143.91	28.76
	2.2 G	74.09	18.58	63.51	23.67	136.02	36.13	144.44	28.81
	563 M	75.41	18.75	64.69	23.87	137.86	36.50	147.66	28.91
	353 M	76.39	18.81	65.46	23.78	138.95	36.49	149.08	28.78

reduction and smoothing, C-HPY3 performed comparably or inferiorly to HPY3. Among approximated models, LW3 performed the best in terms of PPL and WER. For the validation set and test set A, LW3 was comparable to HPY3. On the other hand, for test sets B and C, LW3 performed remarkably better than HPY3. This result shows that LWLM robustly handles speech domains different from that of the training data. It seems that the learning criteria, which identify related words, are effective in expanding the application range of LMs. These results correspond to those in Experiment 1.

Lines 7 and 8 show the results of mixed n-gram models that can be also used for WFST-based one-pass decoding. HPY3+LW3 was superior to HPY3+RNN3 in all data sets. We obtained better WER reduction from HPY3+LW3 than HPY3 or LW3. Although the mixture weight of HPY3+LW3 was optimized for the validation data, the mixed model performed robustly in out-of-domain data sets.

Lines 9-12 show the results of RNN and the combination with back-off n-gram structure. RNN was superior to HPY3 and LW3 in the validation set and test set A. On the other hand, in test sets B and C, RNN was inferior to LW3. LW3 seems to be robust at supporting multiple domains. Lines 10-12 compare rescored results using RNN after WFST-based decoding with back-off n-gram structure in terms of ASR performance. For example, in line 12, decoding was based on HPY3+LW3 and 1000-best rescoring was based on HPY3+LW3+RNN. Combining RNN with the back-off n-gram structure improved PPL and WER. The highest result was obtained by HPY3+LW3+RNN in all data sets. These results suggest that WFST-based decoding performance must be improved for utilizing an intelligent rescoring model such as RNNLM.

Next, we applied entropy pruning to HPY3+LW3, the combination with the highest performance among the back-off n-gram structures [29]. We investigated the relationship between model size and the performance of the pruned model variants. Model size is taken to be ARPA file size with ASCII format.

The results in Table 6 show that entropy pruning could reduce model size efficiently. Even if we reduced the model size of HPY3+LW3 such that it was comparable to that of HPY3, HPY3+LW3 was superior to HPY3 in terms of PPL and WER. Especially, in out-of-domain data, the pruned models outperformed HPY3. These results show that entropy pruning is suitable for introducing our approach to practical ASR systems.

6. Conclusions

In this paper, we proposed an n-gram approximation of LWLM for improving ASR performance in multiple domains. Our approach allows LWLM to support one-pass ASR decoding by converting it into the back-off n-gram structure. We revealed that random sampling based on LWLM can generate various linguistic phenomena and that the back-off n-gram model constructed from the generated data performs robustly in not only in-domain data but also out-of-domain data. We also showed that the interpolation of approximated model and standard n-gram model effectively improves ASR performance. Moreover, we revealed

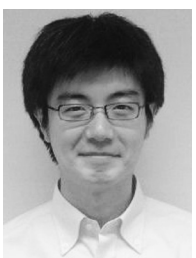
that entropy pruning is useful in reducing constructed model size even though a lot of data is needed to adequately approximate LWLM.

References

[1] J.T. Goodman, "A bit of progress in language modeling," Computer Speech & Language, vol.15, pp.403–434, 2001.
[2] S.F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," Computer Speech & Language, vol.13, pp.359–383, 1999.
[3] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," Proc. ICASSP, vol.1, pp.181–184, 1995.
[4] Y.W. Teh, "A hierarchical Bayesian language model based on Pitman-Yor processes," Proc. COLING/ACL, pp.985–992, 2006.
[5] S. Huang and M. Yor, "Hierarchical Pitman-Yor language models for ASR in meetings," Proc. ASRU, pp.124–129, 2007.
[6] P.F. Brown, P.V. deSouza, R.L. Mercer, V.J.D. Pietra, and J.C. Lai, "Class-based n-gram models of natural language," Computational Linguistics, vol.18, pp.467–479, 1992.
[7] G. Potamianos and F. Jelinek, "A study of n-gram and decision tree letter language modeling methods," Speech Communication, vol.24, no.3, pp.171–192, 1998.
[8] P. Xu and F. Jelinek, "Random forests in language modeling," Proc. EMNLP, pp.325–332, 2004.
[9] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," J. Mach. Learn. Res., vol.3, pp.1137–1155, 2003.
[10] T. Mikolov, M. Karafiat, L. Burget, J. Cernocky, and S. Khudanpur, "Recurrent neural network based language model," Proc. INTERSPEECH, pp.1045–1048, 2010.
[11] T. Mikolov, S. Kombrink, L. Burget, J. Cernocky, and S. Khudanpur, "Extensions of recurrent neural network language model," Proc. ICASSP, pp.5528–5531, 2011.
[12] K. Deschacht, J.D. Belder, and M.-F. Moens, "The latent words language model," Computer Speech & Language, vol.26, pp.384–409, 2012.
[13] S. Goldwater and T. Griffiths, "A fully Bayesian approach to unsupervised part-of-speech tagging," Proc. ACL, pp.744–751, 2007.
[14] P. Blunsom and T. Cohn, "A hierarchical Pitman-Yor process HMM for unsupervised part of speech induction," Proc. ACL, pp.865–874, 2011.
[15] Y. Su, "Bayesian class-based language models," Proc. ICASSP, pp.5564–5567, 2011.
[16] J.-T. Chien and C.H. Chueh, "Dirichlet class language models for speech recognition," IEEE Transactions on Audio, Speech and Language Processing, vol.19, pp.482–495, 2011.
[17] W. Wang, A. Stolcke, and M.P. Harper, "The use of a linguistically motivated language model in conversational speech recognition," Proc. ICASSP, pp.261–264, 2004.
[18] R. Wang, M. Utiyama, I. Goto, E. Sumita, H. Zhao, and B.L. Lu, "Converting continuous-space language models into n-gram language models for statistical machine translation," Proc. EMNLP, pp.845–850, 2013.
[19] E. Arisoy, S.F. Chen, B. Ramabhadran, and A. Sethy, "Converting neural network language models into back-off language models for efficient decoding in automatic speech recognition," Proc. ICASSP, pp.8242–8246, 2013.
[20] E. Arisoy, S.F. Chen, B. Ramabhadran, and A. Sethy, "Converting neural network language models into back-off language models for efficient decoding in automatic speech recognition," IEEE/ACM Trans. Audio, Speech, Language Process., vol.22, pp.2329–9290, 2014.
[21] A. Deoras, T. Mikolov, S. Kombrink, M. Karafiat, and S. Khudanpur, "Variational approximation of long-span language models in LVCSR," Proc. ICASSP, pp.5532–5535, 2011.
[22] A. Deoras, T. Mikolov, S. Kombrink, and K. Church, "Approximate inference: A sampling based modeling technique to capture complex dependencies in a language model," Speech Communication, vol.55, no.1, pp.162–177, 2013.
[23] H. Adel, K. Kirchhoff, N.T. Vu, D. Telaar, and T. Schultz, "Comparing approaches to convert recurrent neural networks into back-off language models for efficient decoding," Proc. INTERSPEECH, pp.651–655, 2014.
[24] 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.
[25] A. Emami and F. Jelinek, "Random clusterings for language modeling," Proc. ICASSP, vol.1, pp.581–584, 2005.
[26] Y.W. Teh, M.I. Jordan, M.J. Beal, and D.M. Blei, "Hierarchical Dirichlet processes," Journal of the American Statistical Association, vol.101, no.476, pp.1566–1581, 2006.
[27] D.J.C. MacKay and L.C. Peto, "A hierarchical Dirichlet language model," Natural Language Engineering, vol.1, pp.289–308, 1994.
[28] A. Stolcke, "SRILM – an extensible language modeling toolkit," Proc. ICSLP, vol.2, pp.901–904, 2002.
[29] A. Stolcke, "Entropy-based pruning of backoff language models," Proc. DARPA Broadcast News Transcription and Understanding Workshop, pp.270–274, 1998.
[30] M.P. Marcus, M.A. Marcinkiewicz, and B. Santorini, "Building a large annotated corpus of English: The penn treebank," Computational Linguistics, vol.19, pp.313–330, 1993.
[31] K. Maekawa, H. Koiso, S. Furui, and H. Isahara, "Spontaneous speech corpus of Japanese," Proc. LREC, pp.947–952, 2000.
[32] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," IEEE Signal Process. Mag., vol.29, no.6, pp.82–97, 2012.
[33] T. Hori, C. Hori, Y. Minami, and A. Nakamura, "Efficient WFST-based one-pass decoding with on-the-fly hypothesis rescoring in extremely large vocabulary continuous speech recognition," IEEE transactions on Audio, Speech and Language Processing, vol.15, no.4, pp.1352–1365, 2007.
[34] 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.
[35] T. Fuchi and S. Takagi, "Japanese morphological analyzer using word co-occurrence: JTAG," Proc. COLING/ACL, pp.409–413, 1998.

Ryo Masumura received B.E. and M.E. degrees in engineering from Tohoku University, Sendai, Japan, in 2009 and 2011, 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 IEICE, 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 Laboratories, 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.

Satoshi Takahashi received B.E., M.E., and Ph.D. degrees in information science from Waseda University, Tokyo, in 1987, 1989, and 2002, respectively. Since joining Nippon Telegraph and Telephone (NTT) Corporation in 1989, he has been engaged in speech recognition, spoken dialog system, and pattern recognition. He received the Awaya Kiyoshi Science Promotion Award and the Sato Prize Paper Award from the Acoustic Society of Japan (ASJ) in 1992 and 2014, respectively, and the Takayanagi Kenjiro Achievement Prize form the Takayanagi Memorial Foundation for Electronics Science and Technology in 2014. He is now Vice President in NTT Media Intelligence Laboratories. He is a member of the Acoustic Society of Japan (ASJ), the Institute of Electronics, Information and Communication Engineers (IEICE), the Information Processing Society of Japan (IPSJ) and Institute of Electrical and Electronics Engineers (IEEE).