Electronic Preprint for Journal of Information Processing Vol.29

Regular Paper

Hierarchical Latent Words Language Models for Automatic Speech Recognition

Ryo Masumura[1,a)] Taichi Asami[1] Takanobu Oba[1] Sumitaka Sakauchi[1]

Received: August 15, 2020, Accepted: January 12, 2021

Abstract: This paper presents hierarchical latent words language models (h-LWLMs) for improving automatic speech recognition (ASR) performance in out-of-domain tasks. Language models called h-LWLM are an advanced form of LWLM that are one one hopeful approach to domain robust language modeling. The key strength of the LWLMs is having a latent word space that helps to efficiently capture linguistic phenomena not present in a training data set. However, standard LWLMs cannot consider that the function and meaning of words are essentially hierarchical. Therefore, h-LWLMs employ a multiple latent word space with hierarchical structure by estimating a latent word of a latent word recursively. The hierarchical latent word space helps us to flexibly calculate generative probability for unseen words. This paper provides a definition of h-LWLM as well as a training method. In addition, we present two implementation methods that enable us to introduce the h-LWLMs into ASR tasks. Our experiments on a perplexity evaluation and an ASR evaluation show the effectiveness of h-LWLMs in out-of-domain tasks.

Keywords: hierarchical latent words language models, automatic speech recognition, domain robust language modeling

1. Introduction

In recent practical automatic speech recognition (ASR) systems, language models (LMs) that estimate generative probability of words are an essential component along with acoustic models. It is widely known that the performance of LMs strongly depends on the quality and quantity of their training data sets [1], [2], [3]. Superior performance is usually obtained by using enormous domain-matched training data sets to construct LMs. Unfortunately, in practical ASR tasks, large amounts of domain-matched data sets are not available, so LMs are often required to robustly predict the probability of unobserved linguistic phenomena. In this paper, we focus on a method that aims to improve the robustness of LMs and make them more flexible in dealing with out-of-domain tasks.

For domain robust language modeling, several technologies have been proposed. Fundamental techniques are smoothing [4] and clustering [5]. Other solutions are Bayesian modeling [6] and ensemble modeling [7], [8]. Moreover, continuous representation of words in neural network LMs including feed-forward neural network LMs, recurrent neural network (RNN) LMs, long short-term memory LMs and transformer LMs can also support robust modeling [9], [10], [11], [12], [13], [14]. However, most previous studies are focused on maximizing performance in the same domain as that of the training data. In other words, whether or not these technologies can robustly support out-of-domain tasks is still uncertain.

In contrast, latent words LMs (LWLMs) that are generative models with a latent word space are known as an effective way of improving out-of-domain tasks [15]. The latent word space can flexibly take into account relationships between words and the modeling helps to efficiently increase the robustness to out-of-domain tasks. In addition, domain robust mixture modeling can be achieved by using multiple LWLMs [16], [17], In fact, the LWLMs can be applied to ASR using two kinds of methods. One method is n-gram approximation that converts LWLMs into smoothed n-gram structure [18]. The other method is the Viterbi approximation that only considers one latent word assignment for computing the generative probability of words [19]. Previous studies reported that LWLMs was significantly superior in out-of-domain ASR tasks while the performance was comparable to conventional LMs in domain-matched tasks.

However, standard LWLMs merely represent the latent word space as n-gram modeling of latent words. It is considered that the function and meaning of words are thought to have an essentially hierarchical structure. The hierarchical structure can take into account the process of abstracting the function and meaning of words. Examples of the abstraction process include the use of “apple” as a typical word in referring to fruits, while “food” is a typical word in referring to foods including “apple” and other fruits. Conventional LWLMs do not model the hierarchy while the latent words are used to represent the function and meaning of words. This hierarchical stuture will prove useful in increasing the robustness to out-of-domain tasks.

In this paper, we present LWLMs with multiple latent word spaces that are hierarchically structured. We refer to these as hierarchical LWLM or h-LWLM. The h-LWLMs assume that there is a latent word behind a latent word. The h-LWLMs are related to

[1] The authors are with NTT Media Intelligence Laboratories, NTT Corporation, Yokosuka, Kanagawa 239–0847, Japan
[a)] ryou.masumura.ba@hco.ntt.co.jp

generative modeling methods with a hierarchical latent variable structure. Examples of such methods are the hierarchical hidden Markov models (h-HMMs) [20], [21] and the hierarchical latent Dirichlet allocation (h-LDA) [22]. The h-HMMs and h-LDA are generalizations of standard HMMs and standard LDA, respectively. Similarly, the h-LWLMs can be regarded as a generalized form of the standard LWLMs. Thus, standard LWLMs correspond to h-LWLM with only one layer. Unlike original LWLMs, these h-LWLM with multiple latent word spaces will allow flexibly calculating the generative probability of unseen words. In addition, the h-LWLMs are related to other extended modeling of LWLMs. One related modeling is latent words RNN LMs (LWRNNLMs) [23], [24] that use the RNN modeling for latent variable modeling instead of n-gram modeling. The h-LWLMs differ from these extended models by taking into account the hierarchical structure of latent words.

In order to create a hierarchical latent word structure from training data sets, this paper introduces a layer-wise inference method. The key idea for modeling the hierarchy in the latent word space is estimating a latent word of a latent word recursively. The initial idea for this inference method is a deep Boltzmann machine [25] that stacks up restricted Boltzmann machines [26]. The inference can be achieved using the Gibbs sampling. In addition, this paper presents two implementation methods for ASR, n-gram approximation and the Viterbi approximation, as well as the standard LWLM since it is impractical to directly apply the h-LWLM to the ASR decoding process. In experiments, the effectiveness of the proposed method is shown by perplexity and speech recognition evaluation.

Note that this paper is an extended study of our previous work [27]. Main differences are as follows.

- This paper provides detailed definitions of h-LWLMs and their training methods.
- This paper introduces not only n-gram approximation but also the Viterbi approximation to implement h-LWLMs into ASR.
- This paper examines perplexity evaluation using PennTreebank corpus that is the most representative evaluation set.

This paper is organized as follows. Section 2 explains LWLMs that are the conventional method in this work. Section 3 provides a definition of h-LWLMs. In addition, a training method and two ASR implementation methods for h-LWLM are introduced in detail. Sections 4 and 5 describe a perplexity evaluation and an ASR evaluation. Section 6 concludes this paper.

2. Latent Words Language Models

This section briefly describes definition and a training method of latent words LMs (LWLMs).

2.1 Definition

LWLMs are generative models that have a latent variable for every observed word. A graphic representation of LWLM is shown in **Fig. 1**. In this figure, relationships between latent words were modeled by 3-gram modeling. The gray circles denote observed words and the white circles denote latent variables.

In the generative process of LWLM, a latent variable called

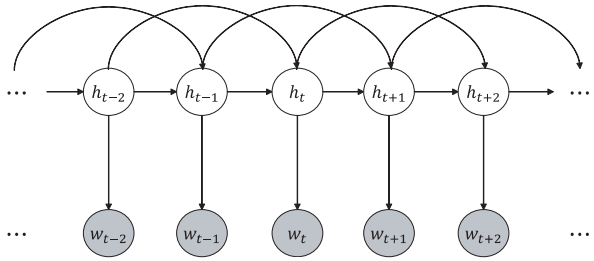

Fig. 1 Model structure of LW-LMs.

latent word h_t, is generated depending on the transition probability distribution given context $\boldsymbol{l}_t = \{h_{t-n+1}, ..., h_{t-1}\}$, where n is an n-gram order. Next, an 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. Here, $P(h_t|\boldsymbol{l}_t, \boldsymbol{\Theta}_{\mathtt{lw}})$ 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.

LWLMs have an important property in which 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 a latent word.

In the Bayesian approach, the generative probability of observed words $\boldsymbol{w} = \{w_1, \cdots, w_T\}$ is defined as:

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

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

$$\begin{aligned} P(\boldsymbol{w}) &\simeq \frac{1}{M} \sum_{m=1}^{M} \sum_{\boldsymbol{h}} P(\boldsymbol{w}|\boldsymbol{h}, \boldsymbol{\Theta}_{\mathtt{lw}}^m) P(\boldsymbol{h}|\boldsymbol{\Theta}_{\mathtt{lw}}^m) \\ &\simeq \frac{1}{M} \sum_{m=1}^{M} \prod_{t=1}^{T} \sum_{h_t} \sum_{\boldsymbol{l}_t} P(w_t|h_t, \boldsymbol{\Theta}_{\mathtt{lw}}^m) P(h_t|\boldsymbol{l}_t, \boldsymbol{\Theta}_{\mathtt{lw}}^m), \end{aligned} \tag{4}$$

where $\boldsymbol{\Theta}_{\mathtt{lw}}^m$ means the m-th point estimated model parameter. The generative probability can be approximated using M instances of $\boldsymbol{\Theta}_{\mathtt{lw}}^m$. In fact, an ensemble of several models ($M > 1$) is effective for LMs such as random class based LMs [8] and random forest LMs [7].

LWLM has a similar structure to the standard class based n-gram model. The latent word approximately corresponds to the class of the standard class based n-gram model [5]. 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 a vast class space, about as large as the vocabulary, while the number of classes in the standard class based n-gram model is often defined as several hundreds or thousands of classes.

2.2 Training

LWLMs are trained from a training data set $\mathcal{W}$. In LWLM training, the latent word assignment $\mathcal{H}$ behind $\mathcal{W}$ has to be inferred. In fact, multiple latent word assignments $\{\mathcal{H}_1, \cdots, \mathcal{H}_M\}$ are estimated for the Bayesian modeling. Once a latent word assignment $\mathcal{H}_m$ is defined, $P(w_t|h_t, \boldsymbol{\Theta}^m_{\mathtt{lw}})$ and $P(h_t|\boldsymbol{l}_t, \boldsymbol{\Theta}^m_{\mathtt{lw}})$ 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 $\mathcal{H}$. The conditional probability distribution of possible values for latent word h_t is given by:

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

where $\mathcal{H}_{-t}$ 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 $\mathcal{W}$ and $\mathcal{H}_{-t}$.

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 prior hierarchical Pitman-Yor [28]. $P(h_t|\boldsymbol{l}_t, \boldsymbol{\Theta}_{\mathtt{lw}})$ is given as:

$$\begin{aligned} P(h_t|\boldsymbol{l}_t, \boldsymbol{\Theta}_{\mathtt{lw}}) &= P(h_t|\boldsymbol{l}_t, \mathcal{H}), \\ &= \frac{c(h_t, \boldsymbol{l}_t) - d_{|\boldsymbol{l}_t|}s(h_t, \boldsymbol{l}_t)}{\theta_{|\boldsymbol{l}_t|} + c(\boldsymbol{l}_t)} \\ &\quad + \frac{\theta + d_{|\boldsymbol{l}_t|}s(\boldsymbol{l}_t)}{\theta_{|\boldsymbol{l}_t|} + c(\boldsymbol{l}_t)} P(h_t|\pi(\boldsymbol{l}_t), \mathcal{H}), \end{aligned} \tag{6}$$

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 $\mathcal{H}$. $s(h_t, \boldsymbol{l}_t)$ and $s(\boldsymbol{l}_t)$ are calculated from a seating arrangement defined by the Chinese restaurant franchise representation of the Pitman-Yor process [28]. $d_{|\boldsymbol{l}_t|}$ and $\theta_{|\boldsymbol{l}_t|}$ are discount and strength parameters of the Pitman-Yor process, respectively. Moreover, a Dirichlet prior is used for the emission probability distribution [29]. $P(w_t|h_t, \boldsymbol{\Theta}_{\mathtt{lw}})$ is given as:

$$\begin{aligned} P(w_t|h_t, \boldsymbol{\Theta}_{\mathtt{lw}}) &= P(w_t|h_t, \mathcal{W}, \mathcal{H}), \\ &= \frac{c(w_t, h_t) + \alpha P(w_t)}{c(h_t) + \alpha}, \end{aligned} \tag{7}$$

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

3. Hierarchical Latent Words Language Models

This section details hierarchical latent words language models (h-LWLMs). First, we describe the definition of h-LWLMs. Next, we present a training method of h-LWLMs. In addition, we show two methods, n-gram approximation and Viterbi approximation, for implementing the h-LSLMs into ASR since it is impractical to directly apply the h-LWLM to ASR.

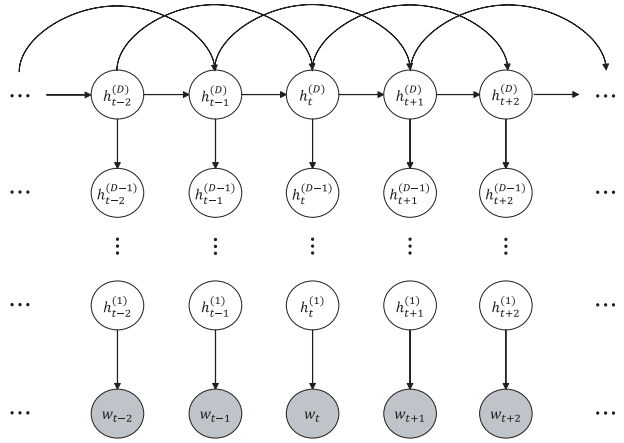

Fig. 2 Model structure of h-LWLMs.

3.1 Definition

The h-LWLMs have multiple latent word spaces in a hierarchical structure. Thus, the definition assumes that there is a latent word behind a latent word. The h-LWLMs can be regarded as a generalized form of the standard LWLMs. Thus, standard LWLMs correspond to h-LWLM with only one layer. As well as the standard LWLMs, the latent words in all layers are represented as a specific word that is selected from the entire vocabulary. A graphic representation of h-LWLM is shown in **Fig. 2**. In this figure, relationships between latent words were modeled by 3-gram modeling. Gray circles denote observed words and white circles denote latent words.

In a generative process of the h-LWLM, a latent word in the highest layer $h^{(D)}_t$ is first generated depending on its context latent words $\boldsymbol{l}^{(D)}_t = \{h^{(D)}_{t-n+1}, ..., h^{(D)}_{t-1}\}$ where n is an n-gram order. Next, a latent word in a lower layer $h^{(d-1)}_t$ is recursively generated depending on the latent word in the upper layer $h^{(d-1)}_t$. Finally, an observed word w_t is generated depending on the latent word in the lowest layer $h^{(1)}_t$. The generative process is formulated as:

$$h^{(D)}_t \sim P(h^{(D)}_t|\boldsymbol{l}^{(D)}_t, \boldsymbol{\Theta}_{\mathtt{hlw}}), \tag{8}$$

$$h^{(d-1)}_t \sim P(h^{(d-1)}_t|h^{(d)}_t, \boldsymbol{\Theta}_{\mathtt{hlw}}), \tag{9}$$

$$h^{(1)}_t \sim P(h^{(1)}_t|h^{(2)}_t, \boldsymbol{\Theta}_{\mathtt{hlw}}), \tag{10}$$

$$w_t \sim P(w_t|h^{(1)}_t, \boldsymbol{\Theta}_{\mathtt{hlw}}), \tag{11}$$

where $\boldsymbol{\Theta}_{\mathtt{hlw}}$ is a model parameter of h-LWLM and D is the number of layers. $P(h^{(D)}_t|\boldsymbol{l}^{(D)}_t, \boldsymbol{\Theta}_{\mathtt{hlw}})$ represents the transition probability that is expressed by the n-gram structure for latent words in the highest layer. $P(h^{(d)}_t|h^{(d+1)}_t, \boldsymbol{\Theta}_{\mathtt{hlw}})$, $P(h^{(1)}_t|h^{(2)}_t, \boldsymbol{\Theta}_{\mathtt{hlw}})$ and $P(w_t|h^{(1)}_t, \boldsymbol{\Theta}_{\mathtt{hlw}})$ represent the emission probabilities that respectively model the dependency between latent words in two layers and the dependency between the observed word and the latent word in the lowest layer.

In the Bayesian h-LWLMs, the generative probability for observed words $\boldsymbol{w} = \{w_1, \cdots, w_T\}$ is defined as:

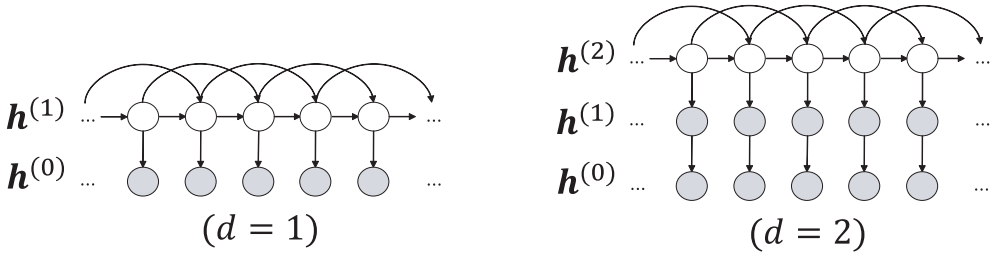

Fig. 3 Layer-wise inference procedure of h-LWLMs.

$$P(\boldsymbol{w}) = \int \sum_{\boldsymbol{h}^{(1)}} \cdots \sum_{\boldsymbol{h}^{(D)}} P(\boldsymbol{w}|\boldsymbol{h}^{(1)}, \boldsymbol{\Theta}_{\mathtt{hlw}}) \cdots P(\boldsymbol{h}^{(D-1)}|\boldsymbol{h}^{(D)}, \boldsymbol{\Theta}_{\mathtt{hlw}}) P(\boldsymbol{h}^{(D)}|\boldsymbol{\Theta}_{\mathtt{hlw}}) P(\boldsymbol{\Theta}_{\mathtt{hlw}}) d\boldsymbol{\Theta}_{\mathtt{hlw}}, \quad (12)$$

where $\boldsymbol{h}^{(d)} = \{h_1^{(d)}, \cdots, h_T^{(d)}\}$ denotes a latent word sequence in the d-th layer. $P(\boldsymbol{w})$ can be formulated as:

$$P(\boldsymbol{w}) = \int \prod_{t=1}^{T} \sum_{h_t^{(1)}} \cdots \sum_{h_t^{(D)}} \sum_{\boldsymbol{l}_t^{(D)}} P(w_t|h_t^{(1)}, \boldsymbol{\Theta}_{\mathtt{hlw}}) \cdots P(h_t^{(D-1)}|h_t^{(D)}, \boldsymbol{\Theta}_{\mathtt{hlw}}) P(h_t^{(D)}|\boldsymbol{l}_t^{(D)}, \boldsymbol{\Theta}_{\mathtt{hlw}}) P(\boldsymbol{\Theta}_{\mathtt{hlw}}) d\boldsymbol{\Theta}_{\mathtt{hlw}}, \quad (13)$$

As the integral with respect to $\boldsymbol{\Theta}_{\mathtt{hlw}}$ is analytically intractable, we approximate the generative probability as:

$$P(\boldsymbol{w}) = \frac{1}{M} \sum_{m=1}^{M} \prod_{t=1}^{T} \sum_{h_t^{(1)}} \cdots \sum_{h_t^{(D)}} \sum_{\boldsymbol{l}_t^{(D)}} P(w_t|h_t^{(1)}, \boldsymbol{\Theta}_{\mathtt{hlw}}^m) \cdots P(h_t^{(D-1)}|h_t^{(D)}, \boldsymbol{\Theta}_{\mathtt{hlw}}^m) P(h_t^{(D)}|\boldsymbol{l}_t^{(D)}, \boldsymbol{\Theta}_{\mathtt{hlw}}^m), \quad (14)$$

where M is the number of instances of point estimated parameters for approximating the Bayesian h-LWLMs. $\boldsymbol{\Theta}_{\mathtt{hlw}}^m$ indicates the m-th point estimated parameter.

The proposed h-LWLMs are modeled so that every word in a vocabulary can be more or less a latent word, which is probabilistically determined from the relationship between the latent word's neighboring contexts and latent words in the lower layers. This prompts the h-LWLMs to give a higher probability to more generalized words in the upper layers. Therefore, the hierarchical structure can take into account the process of abstracting the function and meaning of words.

3.2 Training

H-LWLMs are constructed from a training data set $\mathcal{W} = \{w_1, \cdots, w_T\}$ using a layer-wise inference procedure. **Figure 3** shows an image representation of the procedure that increases along with a greater number of layers.

In the procedure, LWLM structure is recursively accumulated by estimating a latent word sequence in an upper layer from a latent word sequence in the lower layer. Thus, latent word assignments of each layer $\mathcal{H}^{(1,\cdots,D)} = \{\mathcal{H}^{(1)}, \cdots, \mathcal{H}^{(D)}\}$ behind $\mathcal{W}$ are inferred where $\mathcal{H}^{(d)} = \{h_1^{(d)}, \cdots, h_T^{(d)}\}$ is a latent word assignment in the d-th latent word space. In fact, multiple latent word assignments $\{\mathcal{H}_1^{(1,\cdots,D)}, \cdots, \mathcal{H}_M^{(1,\cdots,D)}\}$ have to be inferred for the Bayesian inference. Once latent word assignments of each layer $\mathcal{H}^{(1,\cdots,D)}$ are defined, $P(h_t^{(D)}|\boldsymbol{l}_t^{(D)}, \boldsymbol{\Theta}_{\mathtt{hlw}})$, $P(h_t^{(d)}|h_t^{(d+1)}, \boldsymbol{\Theta}_{\mathtt{hlw}})$, and $P(w_t|h_t^{(1)}, \boldsymbol{\Theta}_{\mathtt{hlw}})$ can be calculated. The detailed procedure to sample latent word assignments $\{\mathcal{H}_1^{(1,\cdots,D)}, \cdots, \mathcal{H}_M^{(1,\cdots,D)}\}$ is shown in Algorithm 1.

Algorithm 1 Inference procedure for h-LWLM.

Input: Training data set $\mathcal{W}$,
number of instances M, number of layers D
Output: $\mathcal{H}_1^{(1,\cdots,D)}, \cdots, \mathcal{H}_M^{(1,\cdots,D)}$

1: **for** $m = 1$ to M **do**
2: $\quad\mathcal{H}^{(0)} = \mathcal{W}$
3: $\quad$**for** $d = 1$ to D **do**
4: $\quad\quad\mathcal{H}^{(d)} \sim P(\mathcal{H}^{(d)}|\mathcal{H}^{(d-1)})$
5: $\quad$**end for**
6: $\quad\mathcal{H}_m^{(1,\cdots,D)} = \mathcal{H}^{(1)}, \cdots, \mathcal{H}^{(D)}$
7: **end for**
8: **return** $\mathcal{H}_1^{(1,\cdots,D)}, \cdots, \mathcal{H}_M^{(1,\cdots,D)}$

Line 4 in Algorithm 1 denotes the key procedure for estimating a latent word sequence in an upper layer from a latent word sequence in the lower layer. For the inference of $\mathcal{H}^{(d)}$ from $\mathcal{H}^{(d-1)}$, Gibbs sampling is suitable [30], [31], [32]. Gibbs sampling picks a new value for $h_t^{(d)}$ according to its probability distribution, which is estimated from both $\mathcal{H}_{-t}^{(d)}$ and $\mathcal{H}^{(d-1)}$. $\mathcal{H}_{-t}^{(d)}$ represents all latent words in the d-th layer except for $h_t^{(d)}$. The probability distribution is given by:

$$P(h_t^{(d)}|\mathcal{H}_{-t}^{(d)}, \mathcal{H}^{(d-1)}) \propto P(h_t^{(d-1)}|h_t^{(d)}, \boldsymbol{\Theta}_{\mathtt{hlw}}) \prod_{j=t}^{t+n-1} P(h_j^{(d)}|\boldsymbol{l}_j^{(d)}, \boldsymbol{\Theta}_{\mathtt{hlw}}). \quad (15)$$

where $P(h_t^{(d)}|\boldsymbol{l}_t^{(d)}, \boldsymbol{\Theta}_{\mathtt{hlw}})$ and $P(h_t^{(d-1)}|h_t^{(d)}, \boldsymbol{\Theta}_{\mathtt{hlw}})$ can be calculated from $\mathcal{H}^{(d-1)}$ and $\mathcal{H}_{-t}^{(d)}$.

For the inference, the prior distribution is necessary for each probability distribution. Besides standard LWLMs, a hierarchical Pitman-Yor prior [28] is used for each transition probability and a Dirichlet prior [29] is used for each emission probability. The transition probability distribution $P(h_t^{(d)}|\boldsymbol{l}_t^{(d)}, \boldsymbol{\Theta}_{\mathtt{hlw}})$ is given as:

$$\begin{aligned} P(h_t^{(d)}|\boldsymbol{l}_t^{(d)}, \boldsymbol{\Theta}_{\mathtt{hlw}}) &= P(h_t^{(d)}|\boldsymbol{l}_t^{(d)}, \mathcal{H}^{(d)}) \\ &= \frac{c(h_t^{(d)}, \boldsymbol{l}_t^{(d)}) - d_{|\boldsymbol{l}_t^{(d)}|} s(h_t^{(d)}, \boldsymbol{l}_t^{(d)})}{\theta_{|\boldsymbol{l}_t^{(d)}|} + c(\boldsymbol{l}_t^{(d)})} \\ &\quad + \frac{\theta_{|\boldsymbol{l}_t^{(d)}|} + d_{|\boldsymbol{l}_t^{(d)}|} s(\boldsymbol{l}_t^{(d)})}{\theta_{|\boldsymbol{l}_t^{(d)}|} + c(\boldsymbol{l}_t^{(d)})} P(h_t^{(d)}|\pi(\boldsymbol{l}_t^{(d)}), \mathcal{H}^{(d)}), \end{aligned} \quad (16)$$

where $\pi(\boldsymbol{l}_t^{(d)})$ is the shortened context obtained by removing the earliest word from $\boldsymbol{l}_t^{(d)}$. $c(h_t^{(d)}, \boldsymbol{l}_t^{(d)})$ and $c(\boldsymbol{l}_t^{(d)})$ are counts calculated from a latent word assignment $\mathcal{H}^{(d)}$. $s(h_t^{(d)}, \boldsymbol{l}_t^{(d)})$ and $s(\boldsymbol{l}_t^{(d)})$ are calculated from a seating arrangement defined by the

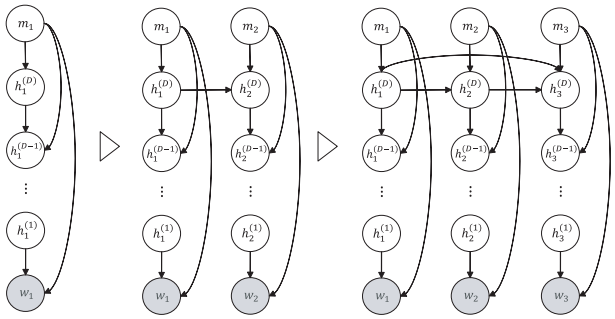

Fig. 4 Random sampling based on h-LWLM.

Chinese restaurant franchise representation of the Pitman-Yor process [28]. $d_{|\boldsymbol{l}_t^{(d)}|}$ and $\theta_{|\boldsymbol{l}_t^{(d)}|}$ are discount and strength parameters of the Pitman-Yor process, respectively.

In addition, the emission probability distributions $P(h_t^{(d-1)}|h_t^{(d)}, \boldsymbol{\Theta}_{\texttt{hlw}})$ and $P(w_t|h_t^{(1)}, \boldsymbol{\Theta}_{\texttt{hlw}})$ are given as:

$$P(h_t^{(d-1)}|h_t^{(d)}, \boldsymbol{\Theta}_{\texttt{hlw}}) = P(h_t^{(d-1)}|h_t^{(d)}, \mathcal{H}^{(d-1)}, \mathcal{H}^{(d)}) = \frac{c(h_t^{(d-1)}, h_t^{(d)}) + \alpha P(h_t^{(d-1)})}{c(h_t^{(d)}) + \alpha}, \tag{17}$$

$$P(w_t|h_t^{(1)}, \boldsymbol{\Theta}_{\texttt{hlw}}) = P(w_t|h_t^{(1)}, \mathcal{W}, \mathcal{H}^{(1)}) = \frac{c(w_t, h_t^{(1)}) + \alpha P(w_t)}{c(h_t^{(1)}) + \alpha}, \tag{18}$$

where $P(w_t)$ is the maximum likelihood estimation value of unigram probability in $\mathcal{W}$, and $P(h_t^{(d-1)})$ is the maximum likelihood estimation value of unigram probability in $\mathcal{H}^{(d)}$. $c(w_t, h_t^{(1)})$ and $c(h_t^{(1)})$ are counts calculated from $\mathcal{W}$ and latent word assignment $\mathcal{H}^{(1)}$. α is a hyper parameter of the Dirichlet prior.

In the inference procedure, we have to compute the probability distributions against all possible latent words. In practice, the computation cost is not expensive since each term can be calculated using count look-up tables.

3.3 N-gram Approximation

One implementation method is the n-gram approximation that converts h-LWLMs into the back-off n-gram structure. A basic concept is to construct smoothed n-gram LM that can generate similar words to those generated from h-LWLM. Thus, the approximated h-LWLM $P(\boldsymbol{w}|\boldsymbol{\Theta}_{\texttt{hlwng}})$ has the following properties:

$$\boldsymbol{w}_{\texttt{hlw}} \sim P(\boldsymbol{w}|\boldsymbol{\Theta}^1_{\texttt{hlw}}, \cdots, \boldsymbol{\Theta}^M_{\texttt{hlw}}), \tag{19}$$

$$\boldsymbol{w}_{\texttt{hlwng}} \sim P(\boldsymbol{w}|\boldsymbol{\Theta}_{\texttt{hlwng}}), \tag{20}$$

$$\boldsymbol{w}_{\texttt{hlw}} \simeq \boldsymbol{w}_{\texttt{hlwng}}, \tag{21}$$

where $\boldsymbol{w}_{\texttt{hlw}}$ is an observed word sequence generated from the h-LWLM, and $\boldsymbol{w}_{\texttt{hlwng}}$ is an observed word sequence generated from the approximated h-LWLM with back-off n-gram structure. The approximated LWLM can be constructed from words generated from the h-LWLM.

Since h-LWLM is a generative model, it can generate latent words and observed words based on random sampling. **Figure 4** shows a random sampling procedure based on h-LWLM. As shown in the figure, an instance index $m_t \in \{1, \cdots, M\}$ for model parameters, latent words for each layer $\{h_t^{(D)}, \cdots, h_t^{(D)}\} \in \mathcal{V}$, and an observed word $w_t \in \mathcal{V}$ are recursively generated. The random sampling is based on Algorithm 2.

Algorithm 2 Random sampling based on h-LWLM.

Input: Model parameters $\boldsymbol{\Theta}^1_{\texttt{hlw}}, \cdots, \boldsymbol{\Theta}^M_{\texttt{hlw}}$, number of sampled words T
Output: Sampled data $\boldsymbol{w}$

1: $\boldsymbol{l}_1^{(D)}$ = <s>
2: **for** $t = 1$ to N **do**
3: $\quad m_t \sim P(m_t) = \frac{1}{\mathrm{M}}$
4: $\quad h_t \sim P(h_t^{(D)}|\boldsymbol{l}_t^{(D)}, \boldsymbol{\Theta}^{m_t}_{\texttt{hlw}})$
5: $\quad$ **for** $d = D-1$ to 1 **do**
6: $\quad\quad h_t^{(d)} \sim P(h_t^{(d)}|h_t^{(d+1)}, \boldsymbol{\Theta}^{m_t}_{\texttt{hlw}})$
7: $\quad$ **end for**
8: $\quad w_t \sim P(w_t|h_t^{(1)}, \boldsymbol{\Theta}^{m_t}_{\texttt{hlw}})$
9: **end for**
10: **return** $\boldsymbol{w} = w_1, \cdots, w_T$

In line 1 of Algorithm 2, $\boldsymbol{l}_1^{(D)}$ is initialized as a sentence head symbol <s>. Through iterations of lines 3–8 in Algorithm 2, a large number of word sequences can be obtained. With T iterations, T latent words, and T observed words are generated. The N observed words are used only for back-off n-gram model estimation.

3.4 Viterbi Approximation

The other implementation method is a Viterbi approximation. It is known that the Viterbi algorithm is a formal technique to compute the joint probability of an observed word sequence and its optimal latent variable sequence. However, there are innumerable combinations of the recognition hypothesis and its latent word assignment in LWLMs. Therefore, this paper implements the Viterbi approximation as a two-pass process using Gibbs sampling as well as our previous work [19]. The Viterbi approximation of h-LWLMs uses the joint probability of a word sequence $\boldsymbol{w} = \{w_1, \cdots, w_T\}$ and its optimal latent word assignment in each layer $\bar{\boldsymbol{h}}^{(1,\cdots,D)} = \{\bar{\boldsymbol{h}}^{(1)}, \cdots, \bar{\boldsymbol{h}}^{(D)}\}$ where $\bar{\boldsymbol{h}}^{(d)} = \{\bar{h}_1^{(d)}, \cdots, \bar{h}_T^{(d)}\}$. The joint probability, i.e., a Viterbi probability, is defined as:

$$P(\boldsymbol{w}, \bar{\boldsymbol{h}}^{(1)}, \cdots, \bar{\boldsymbol{h}}^{(D)}) = \frac{1}{M}\sum_{m=1}^{M} P(\boldsymbol{w}|\bar{\boldsymbol{h}}^{(1)}, \boldsymbol{\Theta}^m_{\texttt{hlw}}) P(\bar{\boldsymbol{h}}^{(d)}|\boldsymbol{\Theta}^m_{\texttt{hlw}}) \prod_{d=2}^{D} P(\bar{\boldsymbol{h}}^{(d-1)}|\bar{\boldsymbol{h}}^{(d)}, \boldsymbol{\Theta}^m_{\texttt{hlw}}). \tag{22}$$

In order to calculate the Viterbi probability, the optimal latent word assignment $\bar{\boldsymbol{h}}^{(1,\cdots,D)}$ has to be estimated. The optimal latent word assignment in each layer is recursively estimated by:

$$\bar{\boldsymbol{h}}^{(1)} = \arg\max_{\boldsymbol{h}^{(1)}} P(\boldsymbol{h}^{(1)}|\boldsymbol{w}) = \arg\max_{\boldsymbol{h}^{(1)}} \frac{1}{M}\sum_{m=1}^{M} P(\boldsymbol{w}|\boldsymbol{h}^{(1)}, \boldsymbol{\Theta}^m_{\texttt{hlw}}) P(\boldsymbol{h}^{(1)}|\boldsymbol{\Theta}^m_{\texttt{hlw}}) \tag{23}$$

$$\bar{\boldsymbol{h}}^{(d)} = \arg\max_{\boldsymbol{h}^{(d)}} P(\boldsymbol{h}^{(d)}|\bar{\boldsymbol{h}}^{(d-1)}) = \arg\max_{\boldsymbol{h}^{(d)}} \frac{1}{M}\sum_{m=1}^{M} P(\bar{\boldsymbol{h}}^{(d-1)}|\boldsymbol{h}^{(d)}, \boldsymbol{\Theta}^m_{\texttt{hlw}}) P(\boldsymbol{h}^{(d)}|\boldsymbol{\Theta}^m_{\texttt{hlw}}) \tag{24}$$

Gibbs sampling can be utilized for the estimation. A conditional probability distribution of the possible values for latent word $h_t^{(d)}$ is defined as:

$$P(h_t^{(d)}|\bar{\boldsymbol{h}}^{(d-1)}, \boldsymbol{h}_{-t}^{(d)}) \propto \sum_{m=1}^{M}\left\{P(\bar{h}_t^{(d-1)}|h_t^{(d)}, \boldsymbol{\Theta}_{\texttt{hlw}}^m)\prod_{j=t}^{t+n-1}P(h_j^{(d)}|\boldsymbol{l}_j^{(d)}, \boldsymbol{\Theta}_{\texttt{hlw}}^m)\right\}, \quad (25)$$

where $\boldsymbol{h}_{-t}^{(d)}$ is a latent word assignment in d-th layer except for $h_t^{(d)}$. A conditional probability distribution of the possible values for latent word $h_t^{(1)}$ is defined as

$$P(h_t^{(1)}|\boldsymbol{w}, \boldsymbol{h}_{-t}^{(1)}) \propto \sum_{m=1}^{M}\left\{P(w_t|h_t^{(1)}, \boldsymbol{\Theta}_{\texttt{hlw}}^m)\prod_{j=t}^{t+n-1}P(h_j^{(1)}|\boldsymbol{l}_j^{(1)}, \boldsymbol{\Theta}_{\texttt{hlw}}^m)\right\}. \quad (26)$$

Note that the Viterbi perplexity degrades because the number of layers increases due to the number of latent word assignment.

4. Experiment 1: Perplexity Evaluation

4.1 Datasets

The first experiments used the Penn Treebank corpus in Ref. [33]. Sections 0–20 were used as a training data set (Train), sections 21 and 22 were used as a validation data set (Valid), and sections 23 and 24 were used as a test data set (Test A). This selection matches those of many previous works. In addition, a human-human discussion text data set (Test B) was prepared for evaluations in a domain different from the training data set. Each vocabulary was limited to 10K words and there were no out-of-vocabulary words. **Table 1** shows details.

Table 1 Data sets for perplexity evaluation.

	Domain	Number 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

Table 2 PPL results of Viterbi approximation on each data set.

	Number of layers	Valid	Test A	Test B
MKN5	-	148.0	141.2	238.6
HPY5	-	145.1	139.3	232.7
RNN	-	134.4	128.9	212.9
LR-NA	-	148.6	140.6	212.4
HLW-NA	1	**138.7**	**131.7**	205.5
HLW-NA	2	140.8	132.9	202.3
HLW-NA	3	142.5	134.8	200.2
HLW-NA	4	144.2	136.6	199.7
HLW-NA	5	145.4	137.2	**199.6**
HLW-VA	1	148.4	142.9	224.7
HLW-VA	2	182.1	175.9	266.7
HLW-VA	3	182.1	175.9	266.7
HLW-VA	4	182.1	175.9	266.7
HLW-VA	5	207.6	198.5	298.5

4.2 Setups

In this evaluation, the following LMs were prepared.

- MKN5: A word-based 5-gram LM with modified Kneser-Ney smoothing constructed from the training data set [4].
- HPY5: A word-based 5-gram hierarchical Pitman-Yor LM (HPYLM) constructed from the training data set. For the training, 200 iterations were used for burn-in, and 10 instances were collected [6].
- RNN: A word-based recurrent neural network LM (RNNLM) [10]. The hidden layer size was set to 200 by referring to a preliminary experiment.
- LR-NA: A word-based 5-gram HPYLM constructed from data generated on the basis of latent words RNNLM (LWRNNLM) constructed from the training data set [23], [24]. LWRNNLM is generative models that combine RNNLM and LWLM. The models have a soft class structure based on a latent word space as does LWLM and the latent word space is modeled using an RNNLM. The hidden unit size was set to 400. The generated data size was one billion words. We applied entropy-based pruning to n-gram entries to match the computation complexity of HPY5 [34].
- HLW-NA: A word-based 5-gram HPYLM constructed from data generated on the basis of 5-gram h-LWLM (HLW) constructed from training data set. HLW with 1 layer represents a standard LWLM, and HLW with 2–5 layers represents the proposed h-LWLM. For their training, 500 iterations were used for burn-in and 10 samples were collected. The generated data size was set to one billion words. We applied entropy based pruning to n-gram entries to match the computation complexity of HPY5 [34].
- HLW-VA: Viterbi approximation of HLW. To calculate the Viterbi probability, 100 samples of latent words assignments were obtained using Gibbs sampling.

In addition, several mixed models constructed by linearly interpolating the above LMs were employed. Hyper parameters and the interpolation weights were optimized using a validation data set.

4.3 Results

Table 2 shows perplexity (PPL) results of HLW-NA and HLW-VA when the number of layers was varied.

First, the results of HLW-NA were investigated. In the validation data set and the test data set A, HLW-NA with 1 layer was superior to that with 3 or 5 layers. On the other hand, in the test data set B, HLW-NA with 5 layers outperformed that with 1 layer. Also, HLW-NA with 5 layers was superior to MKN5, HPY5. Furthermore, HLW-NA outperformed LR-NA, which is n-gram approximation of conventional extended modeling of LWLM. The results indicate the hierarchical latent word space is effective for taking into account unseen words although the PPL deteriorates slightly in the in-domain tasks.

Next, the results of HLW-VA were investigated. PPL performance for each test data set was deteriorated as the number of layers increased. This is because the PPL was calculated using the Viterbi probability which is the joint probability of observed word sequence and latent word sequences in each layer.

5. Experiment 2: ASR Evaluation

5.1 Datasets

The second experiment used the Corpus of Spontaneous Japanese (CSJ) [35]. The CSJ was divided into a training data set (Train), a small validation data set (Valid), and a test data set (Test A). For evaluation in out-of-domain environments, a contact center dialog task (Test B) and a voice mail task (Test C) were prepared. The vocabulary size of the training data set was 83,536. For each data set, the number of words and out-of-vocabulary

Table 3 Data sets for ASR evaluation.

	Domain	Number of words	OOV rate (%)
Train	Lecture	7,317,392	-
Valid	Lecture	28,046	0.72
Test A	Lecture	27,907	0.51
Test B	Contact center	24,665	3.66
Test C	Voice mail	21,044	4.41

(OOV) rate are detailed in **Table 3**.

5.2 Setups

For ASR evaluation, an acoustic model based on hidden Markov models with deep neural networks (DNN-HMM) was prepared [36]. The DNN-HMM had 8 hidden layers with 2,048 nodes. The speech recognizer was a weighted finite state transducer (WFST) decoder [37], [38].

In this evaluation, the following LMs were constructed.

- `MKN3`: A word-based 3-gram LM with modified Kneser-Ney smoothing constructed from training data set [4].
- `HPY3`: A word-based 3-gram HPYLM constructed from the training data set [6]. For the training, 200 iterations were used for burn-in, and 10 samples were collected.
- `RNN`: A class-based RNNLM with 500 hidden nodes and 500 classes [10].
- `LR-NA`: A word-based 3-gram HPYLM constructed from data generated on the basis of class-based LWRNNLM constructed from the training data set [23], [24]. Its latent word space was modeled by an RNN structure. The hidden unit size was set to 500 and the class size was set to 500. We generated 1,000 M words for the n-gram approximation.
- `HLW-NA`: A word-based 3-gram HPYLM constructed from data generated on the basis of 3-gram h-LWLM (`HLW`) constructed from training data set. `HLW` with 1 layer represents a standard LWLM, and `HLW` with 2–5 layers represents the proposed h-LWLM. For their training, 500 iterations were used for burn-in and 10 samples were collected.
- `HLW-VA`: Viterbi approximation of `HLW`. To calculate the Viterbi probability, 100 samples of latent words assignments were obtained using Gibbs sampling.

In addition, several mixed models constructed from the above LMs by linear interpolation were examined. The mixture weights were optimized using the validation set and the EM algorithm. Other hyper parameters were also optimized using the validation set. `MKN3`, `HPY3`, `LR-NA` and `HLW-NA` were converted into the WFST to perform one-pass decoding. `RNN` and `HLW-VA` can be used for a rescoring. For the rescoring, 1000-best lists were generated in the decoding pass.

5.3 Results

For the n-gram approximation of each h-LWLM, the generated data size is related to the performance of an approximation. Relationships between the generated data size and perplexity (PPL) reduction were investigated for the validation set and each test set. The results are shown in **Figs. 5**, **6**, **7**, where the horizontal axis is in log-scale. In the figures, `HLW-NA` with 1 layer means a standard LWLM, and `HLW-NA` with 3 or 5 layers means the proposed h-LWLM. The results show that the PPL of each model

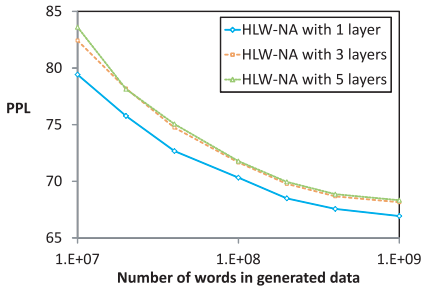

Fig. 5 PPL reduction results of n-gram approximation on Test A.

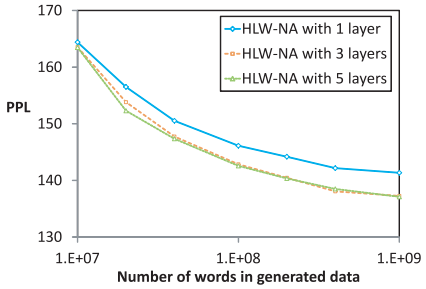

Fig. 6 PPL reduction results of n-gram approximation on Test B.

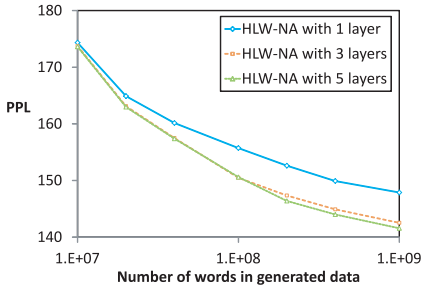

Fig. 7 PPL reduction results of n-gram approximation on Test C.

was reduced as the generated data increased in both the validation set and test sets. In in-domain task, i.e., test A, `HLW-NA` with 1 layer was superior to that with 3 or 5 layers. On the other hand, in out-of-domain tasks, i.e., test B and C, `HLW-NA` with 5 layers displayed the best performance. This confirms that h-LWLMs are effective for improving performance in out-of-domain tasks.

In addition, **Table 5** shows PPL results and speech recognition results in terms of word error rate (WER) for each condition. While `HPY3+RNN` was implemented by 2-pass decoding based on n-best rescoring, `LW-NA` and `LR-NA` can be directly used by converting it into WFST. Therefore, `LW-NA` and `LR-NA` outperformed `HPY3+RNN` in terms of WER. In in-domain tasks, PPL was not improved by the hierarchical structure in `HLW-NA`. `HLW-NA` is comparable to `MKN3` and `HPY3`, and inferior to `RNN` in terms of PPL. On the other hand, in out-of-domain tasks, PPL improved with an increase in the number of layers in `HLW-NA`. Additionally, `HPY3+HLW-NA` outperformed `HPY3+LR-NA` in both in-domain and out-of-domain tasks. This indicates that hierarchically-structured modeling is an effective way to improve ASR performance compared with large-context modeling, i.e., RNN structure. `HLW-NA` with 5 layers was superior to 1 layer in terms of PPL and WER. In terms of WER, statistically significant performance improvements ($p < 0.01$) were achieved by `HLW-NA` with 5 layers compared to `MKN3`, `HPY3`, `HPY3+RNN` and `HLW-NA` with 1 layer in each out-of-domain task. In terms of `HLW-VA`, the results show PPL

Table 4 PPL results and WER results [%].

	Number of layers	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
MKN3	-	81.38	19.98	69.36	24.79	167.61	38.67	189.93	32.00
HPY3	-	79.32	19.74	67.50	24.67	158.13	38.29	175.63	31.69
RNN	-	69.49	-	60.78	-	145.05	-	158.57	-
HPY3+RNN	-	**64.01**	18.53	**55.84**	23.45	122.52	37.45	142.62	30.89
LR-NA	-	90.17	19.89	75.17	25.30	140.72	36.64	145.09	29.75
HPY3+LR-NA	-	72.86	18.65	62.05	23.58	134.65	35.99	141.23	28.74
HLW-NA	1	79.57	19.61	66.93	24.54	141.34	36.93	147.87	30.42
HLW-NA	3	80.42	19.77	68.15	24.68	137.25	37.56	142.52	29.62
HLW-NA	5	80.92	19.86	68.33	24.75	137.10	36.49	141.56	29.57
HLW-VA	1	86.84	-	74.50	-	142.49	-	133.97	-
HLW-VA	3	96.21	-	83.58	-	160.59	-	149.45	-
HLW-VA	5	101.84	-	88.22	-	169.90	-	159.49	-
HPY3+HLW-NA	1	72.86	18.65	62.05	23.58	134.65	35.99	141.23	28.74
HPY3+HLW-NA	3	73.16	18.68	62.71	23.51	130.63	35.67	137.54	28.37
HPY3+HLW-NA	5	73.31	18.63	62.67	23.45	130.32	35.57	136.83	28.32
HPY3+HLW-NA+HLW-VA	1	65.72	18.32	56.05	23.30	102.21	35.65	100.36	28.47
HPY3+HLW-NA+HLW-VA	3	66.05	18.36	57.26	23.36	97.85	35.40	98.38	28.04
HPY3+HLW-NA+HLW-VA	5	67.63	**18.24**	57.80	**23.21**	**95.28**	**35.33**	**97.44**	**27.96**

Table 5 Number of n-gram entries of HLW-NA.

	Number of layers	Data size	# of 2-gram	# of 3-gram
HPY3	-	7.3M	951,124	2,675,189
HLW-NA	1	10M	1,605,191	4,369,301
	1	100M	8,176,468	30,451,404
	1	1,000M	38,837,590	197,267,846
HLW-NA	3	10M	1,656,948	4,465,860
	3	100M	8,541,809	31,470,719
	3	1,000M	40,857,199	206,245,905
HLW-NA	5	10M	1,675,970	4,497,688
	5	100M	8,668,745	31,825,523
	5	1,000M	41,585,099	209,402,061

Table 6 Examples of Japanese transcriptions for reference text and ASR hypotheses.

	Number of layers	Transcriptions	WER (%)
Reference	-	川崎の川崎支店だと思うんですけども	-
HPY3	-	川崎の川崎市展だと思うんですけども	20.0
HPY3+HLW-NA+HLW-VA	1	川崎の川崎市展だと思うんですけども	20.0
HPY3+HLW-NA+HLW-VA	5	川崎の川崎支店だと思うんですけども	0.0

performance for each test data set deteriorated as the number of layers increased.

This is because the PPL was calculated using the Viterbi probability. Among the n-gram language modeling, HPY3+HLW-NA with 5 layers showed the lowest WER. The best results were attained by HPY3+HLW-NA+HLW-VA with 5 layers although the WER differences between HPY3+HLW-NA+HLW-VA with 5 layers and HPY3+HLW-NA+HLW-VA with 1 layer were not statistically significant ($p > 0.05$) in each out-of-domain task. These results show that h-LWLM with multiple layers offers robust performance not possible with other LMs although its performance in the in-domain tasks was not improved. The results also confirm that combining the n-gram approximation and the Viterbi approximation is effective for improving ASR performance of both in-domain tasks and out-of-domain tasks.

The properties of each of the approximated LWLMs were investigated. Table 5 shows the number of 2- and 3-gram entries in each model; the generated data sizes of each model were set to 10M, 100M and 1,000M. The results show that random sampling based on h-LWLM with multiple layers can generate a greater variety of linguistic phenomena than the standard LWLM. This shows that, unlike non-hierarchical LWLM, h-LWLM can generate unseen words. This also indicates that the "abstraction process" is acheived by introducing a hierarchically structured latent word space. In addition, **Table 6** demonstrates examples of Japanese transcriptions for reference text and ASR hypotheses. While ASR errors were caused by HPY3 and HPY3+HLW-NA with 1 latent layer, our proposed method could generate reference transcriptions. In fact, the ASR errors were homonyms for reference words that were not included in the training data set. Thus, the results indicate that our proposed method is robust against words that do not appear in the training data set, i.e., out-of-domain tasks.

6. Conclusions

This paper presents the h-LWLM for improving automatic speech recognition (ASR) performance in out-of-domain tasks. The h-LWLM has a hierarchical latent word space and can flexibly handle linguistic phenomena not present in the training data set. The hierarchical structure enables us to increase the robustness to out-of-domain tasks. Experiments showed that h-LWLM offers improved robustness for out-of-domain tasks. An n-gram approximation of h-LWLM is also superior to standard LWLM in terms of PPL and WER. Furthermore, the proposed

approach is significantly superior to the smoothed n-gram LMs or the RNNLMs in out-of-domain tasks.

A future direction is to develop LMs that perform well in both in-domain and out-of-domain tasks. To this end, we will combine recent neural modeling including long short-term memory LMs [12], [13] and transformer LMs [14] with hierarchically-structured latent variable modeling.

References

[1] Rosenfeld, R.: Two decades of statistical language modeling: Where do we go from here?, *Proc. IEEE*, Vol.88, pp.1270–1278 (2000).

[2] Goodman, J.T.: A bit of progress in language modeling, *Computer Speech and Language*, Vol.15, pp.403–434 (2001).

[3] Masumura, R., Asami, T., Oba, T., Masataki, H., Sakauchi, S. and Ito, A.: Investigation of Combining Various Major Language Model Technologies including Data Expansion and Adaptation, *IEICE Trans. Information and Systems*, Vol.E99-D, No.10, pp.2452–2461 (2016).

[4] Chen, S.F. and Goodman, J.: An Empirical Study of Smoothing techniques for language modeling, *Computer Speech and Language*, Vol.13, pp.359–383 (1999).

[5] Brown, P.F., deSouza, P.V., Mercer, R.L., Pietra, V.J.D. and Lai, J.C.: Class-based n-gram models of natural language, *Computational Linguistics*, Vol.18, pp.467–479 (1992).

[6] Teh, Y.W.: A Hierarchical Bayesian Language Model based on Pitman-Yor Processes, *Proc. International Conference on Computational Linguistics and Annual Meeting of the Association for Computational Linguistics* (*COLING-ACL*), pp.985–992 (2006).

[7] Xu, P. and Jelinek, F.: Random forests in language modeling, *Proc. Conference on Empirical Methods in Natural Language Processing* (*EMNLP*), pp.325–332 (2004).

[8] Emami, A. and Jelinek, F.: Random clusterings for language modeling, *Proc. International Conference on Acoustics, Speech, and Signal Processing* (*ICASSP*), Vol.1, pp.581–584 (2005).

[9] Mikolov, T., Karafiat, M., Burget, L., Cernocky, J. and Khudanpur, S.: Recurrent Neural Network based Language Model, *Proc. Annual Conference of the International Speech Communication Association* (*INTERSPEECH*), pp.1045–1048 (2010).

[10] Mikolov, T., Stefan, S.K., Burget, L., Cernocky, J. and Khudanpur, S.: Extensions of Recurrent Neural Network Language Model, *Proc. International Conference on Acoustics, Speech, and Signal Processing* (*ICASSP*), pp.5528–5531 (2011).

[11] Bengio, Y., Ducharme, R., Vincent, P. and Jauvin, C.: A neural probabilistic language model, *Journal of Machine Learning Research*, Vol.3, pp.1137–1155 (2003).

[12] Sundermeyer, M., Schluter, R. and Ney, H.: LSTM Neural Networks for Language Modeling, *Proc. Annual Conference of the International Speech Communication Association* (*INTERSPEECH*), pp.194–197 (2012).

[13] Sundermeyer, M., Ney, H. and Schluter, R.: From Feedforward to Recurrent LSTM Neural Networks for Language Modeling, *IEEE/ACM Trans. Audio, Speech, and Language Processing*, Vol.23, pp.517–529 (2015).

[14] Irie, K., Zeyer, A., Schluter, R. and Ney, H.: Language Modeling with Deep Transformers, *Proc. Annual Conference of the International Speech Communication Association* (*INTERSPEECH*), pp.3905–3909 (2019).

[15] Deschacht, K., Belder, J.D. and Moens, M.-F.: The latent words language model, *Computer Speech and Language*, Vol.26, pp.384–409 (2012).

[16] Masumura, R., Asami, T., Oba, T., Masataki, H. and Sakauchi, S.: Mixture of Latent Words Language Models for Domain Adaptation, *Proc. Annual Conference of the International Speech Communication Association* (*INTERSPEECH*), pp.1425–1429 (2014).

[17] Masumura, R., Asami, T., Oba, T., Masataki, H., Sakauchi, S. and Ito, A.: Domain Adaptation based on Mixture of Latent Words Language Models for Automatic Speech Recognition, *IEICE Trans. Information and Systems*, Vol.E101-D, No.6, pp.1581–1590 (2018).

[18] Masumura, R., Adami, T., Oba, T., Masataki, H., Sakauchi, S. and Takahashi, S.: N-gram Approximation of Latent Words Language Models for Domain Robust Automatic Speech Recognition, *IEICE Trans. Information and Systems*, Vol.E99-D, No.10, pp.2462–2470 (2016).

[19] Masumura, R., Asami, T., Oba, T., Masataki, H., Sakauchi, S. and Ito, A.: Viterbi Approximation of Latent Words Language Models for Automatic Speech Recognition, *Journal of Information Processing*, Vol.27, No.2, pp.168–176 (2019).

[20] Fine, S., Singer, Y. and Tishby, N.: The Hierarchical Hidden Markov Model: Analysis and Applications, *Machine Learnning*, Vol.32, pp.41–62 (1998).

[21] Murphy, K.P. and Paskin, M.A.: Linear time inference in hierarchical HMMs, *Proc. Advances in Neural Information Processing Systems* (*NIPS*), Vol.2, pp.833–840 (2002).

[22] Blei, D.M., Griffiths, T.L., Jordan, M.I. and Tenenbaum, J.B.: Hierarchical topic models and the nested Chinese restaurant process, *Proc. advances in Neural Information Processing Systems* (*NIPS*), Vol.16, p.17 (2004).

[23] Masumura, R., Asami, T., Oba, T., Masataki, H., Sakauchi, S. and Ito, A.: Latent Words Recurrent Neural Network Language Models, *Proc. Annual Conference of the International Speech Communication Association* (*INTERSPEECH*), pp.2380–2384 (2015).

[24] Masumura, R., Asami, T., Oba, T., Sakauchi, S. and Ito, A.: Latent Words Recurrent Neural Network Language Models for Automatic Speech Recognition, *IEICE Trans. Information and Systems*, Vol.E102-D, No.12, pp.2257–2267 (2019).

[25] Salakhutdinov, R. and Hinton, G.: Deep Boltzmann Machines, *Proc. International Conference on Artificial Intelligence and Statistics*, Vol.5, pp.448–455 (2009).

[26] Hinton, G.E., Osindero, S. and Teh, Y.-W.: A fast Learning Algorithm for Deep Bilief Nets, *Neural Computation*, Vol.18, pp.1527–1554 (2006).

[27] Masumura, R., Asami, T., Oba, T., Masataki, H., Sakauchi, S. and Ito, A.: Hierarchical Latent Words Language Models for Robust Modeling to Out-Of Domain Tasks, *Proc. Conference on Empirical Methods in Natural Language Processing* (*EMNLP*), pp.1896–1901 (2015).

[28] Teh, Y.W., Jordan, M.I., Beal, M.J. and Blei, D.M.: Hierarchical Dirichlet processes, *Journal of the American Statistical Association*, Vol.101, pp.1566–1581 (2006).

[29] MacKay, D.J.C. and Peto, L.C.: A hierarchical Dirichlet language model, *Natural Language Engineering*, Vol.1, pp.289–308 (1994).

[30] Casella, G. and George, E.I.: Explaining the Gibbs sampler, *The American Statistician*, Vol.46, pp.167–174 (1992).

[31] Robert, C.P., Celeux, G. and Diebolt, J.: Bayesian Estimation of Hidden Markov Chains: A Stochastic Implementation, *Statistics and Probability Letters*, Vol.16, pp.77–83 (1993).

[32] Scott, S.L.: Bayesian methods for hidden Markov models: Recursive computing in the 21st century, *Journal of the American Statistical Association*, Vol.97, pp.337–351 (2002).

[33] Marcus, M.P., Marcinkiewicz, M.A. and Santorini, B.: Building a Large Annotated Corpus of English: The Penn Treebank, *Computational Linguistics*, Vol.19, pp.313–330 (1993).

[34] Stolcke, A.: Entropy-based pruning of backoff language models, *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, pp.270–274 (1998).

[35] Maekawa, K., Koiso, H., Furui, S. and Isahara, H.: Spontaneous speech corpus of Japanese, *Proc. International Conference on Language Resources and Evaluation* (*LREC*), pp.947–952 (2000).

[36] Hinton, G., Deng, L., Yu, D., Dahl, G., rahman Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. and Kingsbury, B.: Deep Neural Networks for Acoustic Modeling in Speech Recognition, *Signal Processing Magazine*, pp.1–27 (2012).

[37] Mohri, M., Pereira, F. and Riley, M.: Weighted finite-state transducers in speech recognition, *Computer Speech and Language*, Vol.16, pp.69–88 (2002).

[38] Hori, T., Hori, C., Minami, Y. and Nakamura, A.: Efficient WFST-based one-pass decoding with on-the-fly hypothesis rescoring in extremely large vocabulary continuous speech recognition, *IEEE Trans. Audio, Speech and Language Processing*, Vol.15, No.4, pp.1352–1365 (2007).

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 and the SIG-NL Excellent paper award from the Information Processing Society of Japan (IPSJ) in 2014 and 2018, the Young Researcher Award and the Paper Award from the Association for Natural Language Processing (NLP) in 2015 and 2020, the ISS Young Researcher's Award in Speech Field and the ISS Excellent Paper Award from the Institute of Electronic, Information and Communication Engineers (IEICE) in 2015 and 2018. 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 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.

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 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.