Nonparametric Bayesian Double Articulation Analyzer for Direct Language Acquisition From Continuous Speech Signals

Tadahiro Taniguchi, Shogo Nagasaka, and Ryo Nakashima

***Abstract*—Human infants can discover words directly from unsegmented speech signals without any explicitly labeled data. Current machine learning methods cannot efficiently estimate language model (LM) and acoustic model (AM) and discover words directly from continuous human speech signals in an unsupervised manner. To solve this problem, we propose an integrative generative model that combines an LM and an AM into a single generative model called the hierarchical Dirichlet process hidden LM (HDP-HLM). The HDP-HLM is obtained by extending the hierarchical Dirichlet process hidden semi-Markov model (HDP-HSMM) proposed by Johnson *et al.* An inference procedure for the HDP-HLM is derived using the blocked Gibbs sampler originally proposed for the HDP-HSMM. This procedure enables the simultaneous and direct inference of LM and AM from continuous speech signals. Based on the HDP-HLM and its inference procedure, we develop a novel machine learning method called nonparametric Bayesian double articulation analyzer (NPB-DAA) that can directly acquire LM and AM from observed continuous speech signals. By assuming HDP-HLM as a generative model of observed time series data, and by inferring latent variables of the model, the method can analyze latent double articulation structure, i.e., hierarchically organized latent words and phonemes, of the data in an unsupervised manner. We also carried out two evaluation experiments using synthetic data and actual human continuous speech signals representing Japanese vowel sequences. In the word acquisition and phoneme categorization tasks, the NPB-DAA outperformed a conventional double articulation analyzer and baseline automatic speech recognition system whose AM was trained in a supervised manner. The main contributions of this paper are as follows: 1) we develop a probabilistic generative model that integrates LM and AM, i.e., HDP-HLM; 2) we derive an inference method for this, and propose the NPB-DAA; and 3) we show that the NPB-DAA can discover words directly from continuous human speech signals in an unsupervised manner.**

***Index Terms*—Bayesian nonparametrics, child development, language acquisition, latent variable model.**

Manuscript received May 27, 2015; revised January 15, 2016; accepted March 24, 2016. Date of publication April 21, 2016; date of current version September 7, 2016. This work was supported by the Grant-in-Aid for Young Scientists (B) 2012–2014 under Grant 24700233 through the Ministry of Education, Culture, Sports, Science, and Technology, Japan.

T. Taniguchi is with the College of Information Science and Engineering, Ritsumeikan University, Kusatsu 525-8577, Japan (e-mail: taniguchi@em.ci.ritsumei.ac.jp).

S. Nagasaka and R. Nakashima are with the Graduate School of Information Science and Engineering, Ritsumeikan University, Kusatsu 525-8577, Japan (e-mail: s.nagasaka@em.ci.ritsumei.ac.jp; nakashima@em.ci.ritsumei.ac.jp).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCDS.2016.2550591

I. Introduction

INFANTS must solve the word segmentation problem in order to acquire language from continuous speech signals to which they are exposed. The word segmentation problem is that of identifying word boundaries in continuous speech. If the speech signals are given to infants as isolated words, the task is easy for them. However, it has been known that a relatively small number of infant-directed utterances consist of an isolated word [1]. If infants had knowledge about words and phonemes innately, the problem could be solved relatively easily. On the contrary, the fact that each language has different lists of phonemes and words clearly shows that infants have to acquire them through developmental processes.

From the viewpoint of statistical learning, the learning problem, i.e., direct language acquisition from continuous speech signals, is very difficult because infants do not have access to the truth labels of speech recognition results. In other words, the language acquisition process must be completely unsupervised. The main problem of this paper is to develop a computational model that can estimate language model (LM) and acoustic model (AM), and discover words directly from continuous human speech signals.

Most modern automatic speech recognition (ASR) systems have an LM that represents knowledge about words and their distributional probabilities as well as an acoustic model that represents knowledge about phonemes and their acoustic features (see [2], [3]). Both are usually trained using large transcribed speech datasets and linguistic corpora through supervised learning. However, infants do not have access to such explicitly labeled datasets. They have to acquire both LM and AM from raw acoustic speech signals in an unsupervised manner.

The question about what kind of cues human infants utilize to discover words from continuous speech signals arises. Saffran *et al.* [4] listed three types of cues for word segmentation: 1) prosodic; 2) distributional; and 3) co-occurrence:

1) prosodic cues rely on acoustic information, such as post-utterance pauses, stressed syllables, and acoustically distinctive final syllables;
2) distributional cues represent the statistical relationships between pairs of neighboring speech sounds;
3) co-occurrence cues are used by children to learn words by detecting sounds that co-occur with certain entities in the environment.

Although many researchers had considered the distributional cues to be too complex for infants to use, Saffran *et al.* [5] reported that word segmentation from fluent speech can be accomplished by eight-month-old infants based on solely on distributional cues. It is also reported that the distributional cues seem to be used by infants by the age of seven months, which is earlier than most other cues [6]. These results imply that infants have a fundamental mechanism that can estimate word segments using distributional cues. In addition to this fundamental segmentation mechanism using distributional cues, the prosodic and co-occurrence cues are believed to help the word segmentation task only as supplemental cues [4]. From the viewpoint of phonemic category acquisition, distributional patterns of sounds have been considered to provide infants with clues about the phonemic structure of a language as well [7].

Based on these findings, in this paper, we focus on distributional cues. We explore the fundamental computational mechanism that can discover words from speech signals using only distributional cues, and develop an unsupervised machine learning method which can discover phonemes and words directly from unsegmented speech signals

In this paper, we propose an unsupervised learning method called the nonparametric Bayesian double articulation analyzer (NPB-DAA) that can automatically estimate double articulation structures, i.e., hierarchically organized latent words and phonemes, embedded in speech signals. We propose this as a computationally valid explanation for the simultaneous acquisition of LM and AM. To develop the NPB-DAA, we introduce a probabilistic generative model called the hierarchical Dirichlet process hidden LM (HDP-HLM) as well as its inference algorithm.

The remainder of this paper is organized as follows. Section II describes the background of the proposed method. Section III presents the HDP-HLM by extending hierarchical Dirichlet process-hidden semi-Markov model (HDP-HSMM) proposed by Johnson and Willsky [8]. The HDP-HLM is an probabilistic generative model that integrates acoustic and LMs for continuous speech signals. Section IV describes the inference procedure of HDP-HLM, and our proposed NPB-DAA. Sections V and VI evaluate the effectiveness of the proposed method using synthetic data and actual sequential vowel speech signals. Section VII concludes this paper.

II. Background

A. Word Segmentation Using Distributional Cues in Transcribed Data

With respect to statistical computational models, many kinds of unsupervised machine learning methods for word segmentation have been proposed in the last two decades [9]–[17]. Brent [9] proposed model-based dynamic programming 1 (MBDP-1) for recovering deleted word boundaries in a natural-language text. The MBDP-1 presumes that there is an information source generating the text explicitly and segments the target text so as to maximize the text's probability. Venkataraman [10] proposed a statistical model for segmentation and word discovery from phoneme sequences by improving Brent's [9] algorithm.

Recently, Bayesian nonparametrics, including the hierarchical Dirichlet process and hierarchical Pitman–Yor process, have enabled more sophisticated methods for word segmentation. These models have fully Bayesian generative models and make it possible to calculate the appropriately smoothed n-gram probability for a word that has a long context. Theoretically, they can treat an infinite number of possible words. Goldwater *et al.* [11], [12] proposed an HDP-based word segmentation method and showed that taking context into account is important for statistical word segmentation. Mochihashi *et al.* [13] proposed a nested Pitman–Yor LM (NPYLM), in which a letter n-gram model based on a hierarchical Pitman–Yor LM is embedded in the word n-gram model. They also developed the forward filtering backward sampling procedure to achieve efficient blocked Gibbs sampling and hence infer word boundaries.

However, all of the above mentioned word segmentation methods presume that transcribed phoneme sequences or text data without any recognition errors can be obtained by the learning system. In practice, before acquiring an LM containing an inventory of words, a learning system, i.e., an infant, has to recognize speech signals without any knowledge of words, only with the knowledge of phonemes and/or syllables in an AM. In such a recognition task, the phoneme recognition error rate inevitably becomes high. To overcome this problem, several researchers have proposed word discovery methods utilizing co-occurrence cues.

B. Lexical Acquisition Using Co-Occurrence Cues

Roy and Pentland [18] ambitiously implemented a computational model that enables a robot to autonomously discover words from raw multimodal sensory input. Their results were imperfect compared with recent state-of-art results. However, their results showed it was possible to develop cognitive models that can process raw sensor data and acquire a lexicon without the need for human transcription or labeling.

Iwahashi [19] implemented an interactive learning method for a robot to acquire spoken words through human–robot interaction using audio-visual interfaces. Their learning process was carried out on-line, incrementally, actively, and in an unsupervised manner. Iwahashi [20] also proposed a method that enables a robot to learn linguistic knowledge through human–robot communication in an unsupervised manner. The model combines speech, visual, and behavioral information in a probabilistic framework. Though its performance was still limited, the model is considered to be a more sophisticated model than that proposed in Roy and Pentland's [18] previous study from the viewpoint of statistical machine learning. On the basis of this paper, Iwahashi *et al.* [21] developed an integrated online machine learning system combining speech, visual, and tactile information obtained through interaction. It enabled robots to learn beliefs regarding speech units, words, the concepts of objects, motions, grammar, and pragmatic and communicative capabilities. They called the system LCore.

Araki *et al.* [22] built a robot that formed object categories and acquired their names by combining a multimodal latent Dirichlet allocation (MLDA) and the NPYLM. They showed that the iterative learning of MLDA and NPYLM increases word segmentation performance by using distributional cues and co-occurrence cues simultaneously, but they reported that the prediction accuracy decreases as the phoneme recognition error rate increases. To overcome this problem, Nakamura *et al.* [23] integrated statistical models for word segmentation and multimodal categorization. They showed that a robot can autonomously form object categories and related words from continuous speech signals and continuous visual, auditory, and haptic information by updating its language and categorization models iteratively.

Not only object information, but also place information can be used as co-occurrence cues. Taguchi *et al.* [24] proposed a method for the unsupervised learning of place-names from information pairs that consist of spoken utterances and the mobile robot's estimated current location without any prior linguistic knowledge other than a phoneme AM. They optimized a word list using a model selection method based on description length criterion.

C. Word Segmentation Using Distributional Cues in Noisy Input

As described above, it becomes clear that using co-occurrence cues can mitigate the ill effects of phoneme recognition errors in a word discovery task. However, whether or not the word discovery task can be achieved solely from raw speech signals is still an open question. Neubig *et al.* [25] extended the unsupervised morphological analyzer proposed by Mochihashi *et al.* [13] and enabled it to analyze phoneme lattices. Heymann *et al.* [26] modified Neubig *et al.*'s [25] algorithm and proposed a suboptimal two-stage algorithm. Heymann *et al.* [26] reported that their proposed method outperformed the original method in an experiment that used lattice input generated artificially from text input. In addition, they used the discovered LM for phoneme recognition in an iterative manner and reported that recognition performance was improved [27]. Elsner *et al.* [28] proposed a computational model that jointly performs word segmentation and learns an explicit model of phonetic variation. However, they did not start with acoustic sound, but with dictated noisy text, i.e., recognized phoneme sequences with errors. Their model does not include AM learning.

They showed that the ill effect of phoneme recognition errors can be mitigated to some extent by using distributional information more appropriately. However, all of these methods, except for Iwahashi's [19], used an AM previously trained in a supervised manner. Therefore, these models are insufficient as a constructive model for language acquisition from raw speech signals. Hence, the unsupervised learning of an AM is also an important problem.

D. Unsupervised Learning of Acoustic Model

In contrast with the word segmentation task, the acquisition of an AM is basically a categorization task of the feature vectors transformed from continuous speech signals. Mixture models, including hidden Markov models (HMMs) and Gaussian mixture models, have been used to model phoneme category acquisition. For example, Lake *et al.* [29] used an online mixture estimation model for vowel category learning. The model was originally proposed by Vallabha *et al.* [30]. However, the phoneme acquisition has proven to be complex categorization task in a feature space. The distribution of the feature vectors of each phoneme overlap with each other, and the actual sound of the phoneme depends on its context. Feldman *et al.* [31] pointed out that feedback information from segmented words is important for phonetic category acquisition. They demonstrated this effect through simulations using Bayesian models.

Lee and Glass [32] proposed a hierarchical Bayesian model that can discover a proper set of subword units and an acoustic model in an unsupervised manner. However, their model did not estimate the LM. Lee *et al.* [33] also proposed a hierarchical Bayesian model simultaneously discovering the phonetic inventory and the letter-to-sound mapping rules on the basis of transcribed data only. The method is not a completely unsupervised learning method from raw speech signals, but does automatically determine relations between sounds and transcribed alphabets and forms an AM in an unsupervised manner.

There have been several studies about the simultaneous unsupervised learning of acoustic and LMs. However, a very small number of statistical learning methods that can simultaneously acquire integrated acoustic and LMs have been proposed. Brandl *et al.* [34] attempted to develop an unsupervised learning method that enables a robot to simultaneously obtain phonemes, syllables, and words from acoustic speech. They did not successfully build such a system, but reported their preliminary results. Walter *et al.* [35] proposed a word discovery method that uses an HMM-based method for finding acoustic unit descriptors in parallel with a dynamic time warping technique for finding word segments. However, their model is still heuristic from the viewpoint of probabilistic computational models. As Feldman *et al.* [31] pointed out, word segmentation and phonetic category acquisition are undoubtedly mutually dependent. Therefore, a theoretically integrated probabilistic generative model for the simultaneous acquisition of LM and AM is desirable. Very recently, Kamper *et al.* [36] and Lee *et al.* [37] proposed probabilistic computational models that achieved unsupervised direct word discovery from continuous speech signals. However, they did not provide an explicit, integrated probabilistic generative model for unsupervised simultaneous learning of LM and AM. To develop such an integrated theoretical model, the authors introduced the general concept of double articulation analysis.

E. Double Articulation Analysis

From a general point of view, unsupervised word discovery from raw speech signals is regarded as a double articulation analysis of the time series data representing a speech signal. The double articulation structure is a well-known two-layer

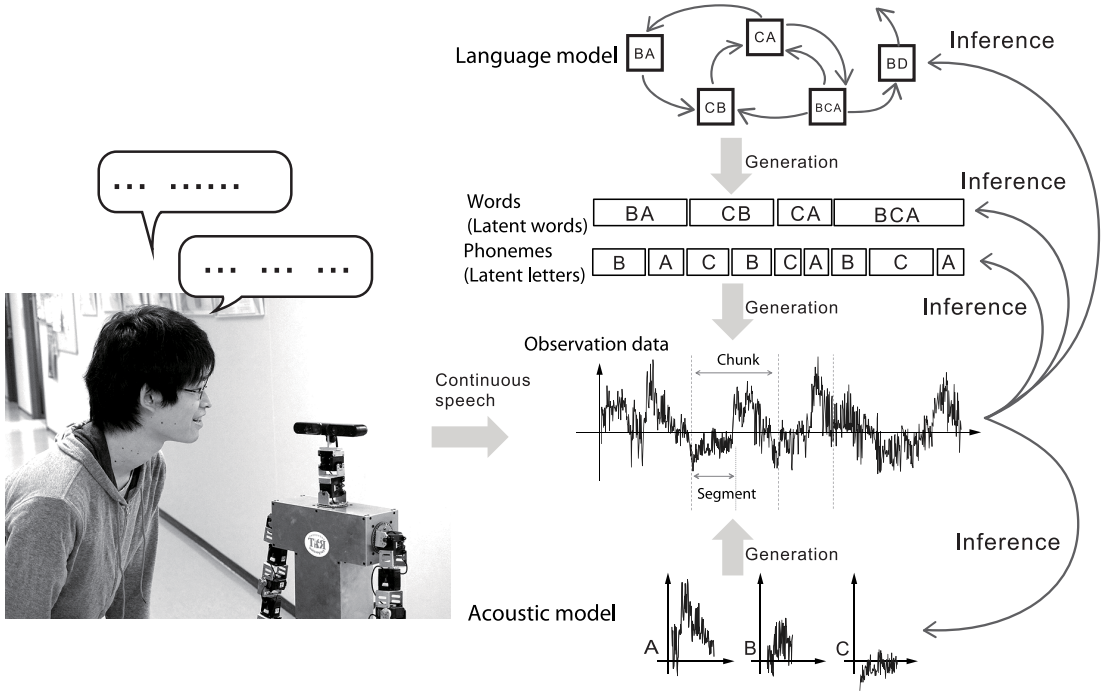


Fig. 1. Overview of unsupervised learning of LM and AM through human–robot interaction, and the generative process of speech signal assumed in the DAA.

hierarchical structure, i.e., a word sequence is generated from an LM, a word is a sequence of phonemes, and each phoneme outputs observation data during the period it persists. The word discovery problem becomes a general problem about analyzing the time series data that potentially have a double articulation structure by estimating the latent AM as well as the latent LM.

Taniguchi and Nagasaka [38] proposed a double articulation analyzer (DAA) by combining the sticky HDP-HMM and the NPYLM. The sticky HDP-HMM proposed by Fox *et al.* [39] is an nonparametric Bayesian extension of HMM. They applied the DAA to human motion data to extract unit motion from unsegmented human motion data. However, they simply used the two nonparametric Bayesian methods sequentially. They did not integrate the two models into a single generative model. Therefore, if there are many recognition or categorization errors in the result of the first latent letter recognition process, i.e., segmentation process by the sticky HDP-HMM, the performance of the subsequent process, i.e., unsupervised chunking by the NPYLM, deteriorates. In the terminology of a DAA, a latent letter and a latent word basically correspond to a phoneme and a word in speech signals, respectively. In this paper, we call this method "conventional DAA" in order to differentiate it from the DAA newly proposed in this paper, i.e., NPB-DAA. Conventional DAA has been successfully applied to human motion data and driving behavior data, which were also considered to potentially have a double articulation structure. Conventional DAA has been used for various purposes, e.g., segmentation [40], prediction [41], [42], data mining [43], topic modeling [44], [45], and video summarization [46]. Conventional DAA owes its successful result with respect to driving behavior data to the fact that driving behavior data were continuous and smooth compared with raw speech signals. For a driving letter, which corresponds to a phoneme in continuous speech signals, the recognition error rate was still low. However, it is expected that a straightforward application of the conventional DAA to raw speech signals will inevitably turn out badly.

Therefore, based on the background mentioned above, in this paper, we propose an integrated probabilistic generative model, HDP-HLM, representing a latent double articulation structure that contains both an LM and an AM. By assuming HDP-HLM as a generative model of observed time series data, and by inferring latent variables of the model, we can analyze latent double articulation structure of the data in an unsupervised manner. A novel DAA is developed on the basis of the HDP-HLM and its inference algorithm. This HDP-HLM-based double articulation analysis method is called NPB-DAA.

III. Generative Model

In this section, we propose a novel generative model, the HDP-HLM, for time series data that potentially has a double articulation structure, by extending HDP-HSMM [8]. As indicated in its name, HDP-HLM latently contains an LM. In contrast with the conventional case where a latent state transits to the next state on the basis of a Markov process in the HDP-HMM, a latent word in the HDP-HLM transits to the next latent word on the basis of an LM. An illustrative overview of the proposed method and the target task are shown in Fig. 1. We can naturally derive an inference procedure for the HDP-HLM based on the blocked Gibbs sampler. First, we briefly describe the HDP-HSMM. We then describe the HDP-HLM.

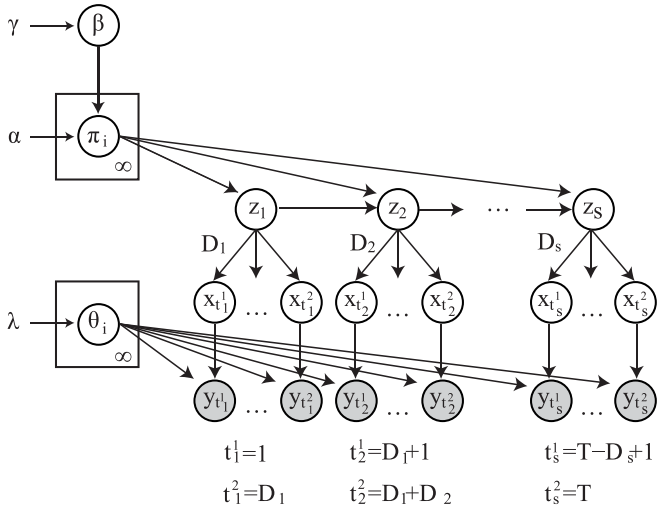


Fig. 2. Model of the HDP-HSMM [8].

A. HDP-HSMM

HDP-HSMM is a nonparametric Bayesian extension of the conventional HSMM [8], [47]. Unlike HDP-HMM, which is an nonparametric Bayesian extension of conventional HMM [39], [48], the HDP-HSMM explicitly models the duration time of a hidden state. A graphical model of the HDP-HSMM is shown in Fig. 2. The generative process of the HDP-HSMM is described as follows:

$$\beta \sim \mathrm{GEM}(\gamma) \tag{1}$$

$$\pi_i \sim \mathrm{DP}(\alpha, \beta) \quad i = 1, 2, \ldots, \infty \tag{2}$$

$$(\theta_i, \omega_i) \sim H \times G \quad i = 1, 2, \ldots, \infty \tag{3}$$

$$z_s \sim \pi_{z_{s-1}} \quad s = 1, 2, \ldots, S \tag{4}$$

$$D_s \sim g(\omega_{z_s}) \tag{5}$$

$$x_t = z_s \quad t = t_s^1, t_s^1 + 1, \ldots, t_s^2 \tag{6}$$

$$y_t = h(\theta_{x_t}) \tag{7}$$

$$t_s^1 = \sum_{s'<s} D_{s'} \tag{8}$$

$$t_s^2 = t_s^1 + D_s - 1 \tag{9}$$

where GEM and DP represent the stick breaking process and Dirichlet process, respectively [48], [49]. The parameters γ and α are hyperparameters of the DP, β is a global transition probability that becomes the base measure of the transition probability distributions, and π_i is a transition probability distribution related to the ith super state. Variable z_s is the sth super state in the sequence of super states, D_s is the frame duration of z_s, and the variables x_t and y_t are a hidden state and an observation at time frame t, respectively. Parameters of an emission distribution and a duration distribution for the ith super state are described as θ_i and ω_i. Additionally, H and G are base measures for emission distribution and duration distribution. The functions h and g represent emission and duration distributions, respectively. The time frames t_s^1 and t_s^2 are frames corresponding to a start point and a end point of a segment corresponding to z_s.

In contrast with the case where HMM assumes that a hidden state x_t transits to the next hidden state x_{t+1} according to a Markov process, the HSMM assumes that a hidden super state z_s transits to next hidden super state z_{s+1} after a probabilistically determined duration time D_s, which is sampled from a duration distribution $g(\omega_{z_s})$. The super state z_s is sampled from a categorical distribution $\pi_{z_{s-1}}$ related to the previous super state z_{s-1}. When the super state z_s and duration time D_s are sampled, a sequence of hidden states $\{x_t \mid 1 + \sum_{s'=1}^{s-1} D_{s'} \le t \le \sum_{s'=1}^{s} D_{s'}\}$ are determined to be z_s.

An observation datum y_t at time t is assumed to be drawn from an emission distribution h whose parameter is θ_{x_t}. Observation data y_t are generated by $h(\theta_{x_t})$ for D_s steps.

An efficient sampling inference procedure based on the backward filtering forward sampling technique was proposed for constructing a blocked Gibbs sampler [8]. A similar algorithm was proposed for HDP-HMM by Fox *et al.* [39]. The algorithm is derived from a weak-limit approximation of the number of hidden super states. The computational cost of the message passing algorithm can be reduced to $O(Td_{\max}N^2)$, where T is the length of the observed data, N is the state cardinality, and $d_{\max}$ is the maximal duration of a super state for truncation. The order is almost the same as that of the backward filtering forward sampling algorithm for the HDP-HMM, except for the constant factor $d_{\max}$.

B. HDP-HLM

The generative model for time series data that potentially have a double articulation structure can be obtained by extending the HDP-HSMM. A graphical model of the proposed HDP-HLM is shown in Fig. 3. In the generative model of HDP-HLM, the super state z_s corresponds to a word in spoken language, which is the fundamental idea of the extension. The ith super state $z_s = i$ has a phoneme sequence $w_i = (w_{i1}, \ldots, w_{ik}, \ldots, w_{iL_i})$, where L_i is the length of the ith word w_i. The generative process of the HDP-HLM is described as follows:

$$\beta^{\mathrm{LM}} \sim \mathrm{GEM}(\gamma^{\mathrm{LM}}) \tag{10}$$

$$\pi_i^{\mathrm{LM}} \sim \mathrm{DP}(\alpha^{\mathrm{LM}}, \beta^{\mathrm{LM}}) \quad i = 1, 2, \ldots, \infty \tag{11}$$

$$\beta^{\mathrm{WM}} \sim \mathrm{GEM}(\gamma^{\mathrm{WM}}) \tag{12}$$

$$\pi_j^{\mathrm{WM}} \sim \mathrm{DP}(\alpha^{\mathrm{WM}}, \beta^{\mathrm{WM}}) \quad j = 1, 2, \ldots, \infty \tag{13}$$

$$w_{ik} \sim \pi_{w_{ik-1}}^{\mathrm{WM}} \quad i = 1, 2, \ldots, \infty, k = 1, 2, \ldots, L_i \tag{14}$$

$$(\theta_j, \omega_j) \sim H \times G \quad j = 1, 2, \ldots, \infty \tag{15}$$

$$z_s \sim \pi_{z_{s-1}}^{\mathrm{LM}} \quad s = 1, 2, \ldots, S \tag{16}$$

$$l_{sk} = w_{z_s k} \quad s = 1, 2, \ldots, S \tag{17}$$

$$k = 1, 2, \ldots, L_{z_s} \tag{18}$$

$$D_{sk} \sim g(\omega_{l_{sk}}) \quad s = 1, 2, \ldots, S \tag{19}$$

$$k = 1, 2, \ldots, L_{z_s} \tag{20}$$

$$x_t = l_{sk} \quad t = t_{sk}^1, \ldots, t_{sk}^2 \tag{21}$$

$$t_{sk}^1 = \sum_{s'<s} D_{s'} + \sum_{k'<k} D_{sk'} + 1 \tag{22}$$

$$t_{sk}^2 = t_{sk}^1 + D_{sk} - 1 \tag{23}$$

$$y_t = h(\theta_{x_t}) \quad t = 1, 2, \ldots, T \tag{24}$$

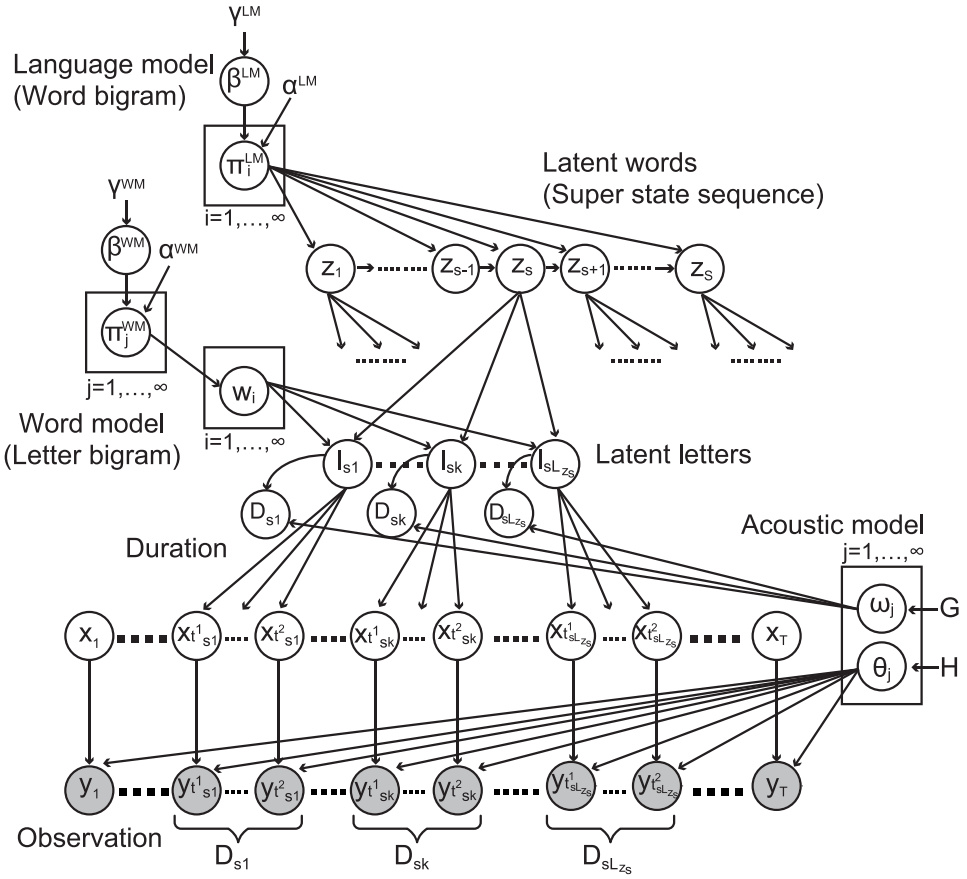


Fig. 3. Model of the proposed HDP-HLM.

where β^{WM} is the base measure and α^{WM} and γ^{WM} are hyperparameters of a word model (WM), which generates words, i.e., latent letter sequences. Furthermore, DP(α^{WM}, β^{WM}) outputs π_j^{WM}, representing the transition probability from latent letter j to the next latent letter. By contrast, β^{LM} is the base measure, α^{LM} and γ^{LM} are hyperparameters of the LM, and DP(α^{LM}, β^{LM}) outputs π_i^{LM}, representing the transition probability from latent word i to the next latent word. The superscripts LM and WM indicate language model or word model, respectively. The latent letters contained in the ith latent word w_i are sequentially sampled from $\pi_{w_{ik-1}}^{\mathrm{WM}}$. The kth latent letter of the ith latent word is represented by w_{ik}. The emission distribution h and the duration distribution g have parameters θ_j and ω_j for the jth latent letter, respectively. The base measures H and G generate θ_j and w_j, respectively. Variable z_s is the sth latent word in the sequence of latent words, and corresponds to the super state in HDP-HSMM, D_s is the frame duration of z_s, $l_{sk} = w_{z_s k}$ is the kth latent letter of the sth latent word, and D_{sk} is the frame duration of l_{sk}. The variable x_t and y_t are a hidden state and an observation at time frame t, respectively. The time frames t_{sk}^1 and t_{sk}^2 are frames corresponding to a start point and a end point of a segment corresponding to l_{sk}, respectively.

In contrast with HMMs, the duration distribution is explicitly determined for each latent letter l_{sk} in the HDP-HLM. The HDP-HLM inherits this property from the HDP-HSMM [8]. The duration time D_{sk} of latent letter l_{sk}, which is the kth latent letter of the sth latent word z_s in a sampled word sequence, is drawn from the duration distribution $g(\omega_{l_{sk}})$, where $\omega_{l_{sk}}$ is the duration parameter for latent letter l_{sk}. The duration of a latent word w_{z_s} becomes $D_s = \sum_{k=1}^{L_{z_s}} D_{sk}$. If we assume that g is a Poisson distribution, the duration distribution of a latent word z_s also follows a Poisson distribution. In this case, the Poisson parameter of the duration distribution becomes $\sum_{k=1}^{L_{z_s}} \omega_{l_{sk}}$. This relation owes to the reproductive property of Poisson distributions.

In the HDP-HLM, latent word z_s determines a latent letter sequence $l_{sk} = w_{z_s k}$ $(k = 1, 2, \ldots, L_{z_s})$. Based on the determined sequence w_{z_s}, duration D_{sk} of l_{sk} is drawn, and observations y_t are drawn from an emission distribution $h(\theta_{x_t})$ corresponding to $x_t = l_{s(t)k(t)}$. The maps $s(t)$ and $k(t)$ represent the indices of words and letters, respectively, in a latent word sequence at time t. Using this generative model, a continuous time series data with a latent double articulation structure can be generated. In this paper, we assume that observed time series data y_t represents a feature vector of the speech signal at time t and is generated in this way. Generally, the HDP-HLM can be applied to any kind of time series data that has a double articulation structure.

From the viewpoint of language acquisition, we review the generative model. In the conventional DAA [38], a DAA is composed of two separated machine learning methods, i.e., sticky HDP-HMM for encoding observation data to letter sequences and NPYLM for chunking letter sequences into

word sequences. On the one hand, the transition probabilities π_i^{LM} and π_i^{WM} correspond to the word bigram and letter bigram models in the NPYLM, respectively. Therefore, $(\pi^{\mathrm{LM}}, \pi^{\mathrm{WM}})$ contains information regarding an LM. On the other hand, $\{\omega_j, \theta_j\}_{j=1,2,\ldots,\infty}$ contains information regarding an AM, which corresponds to a sticky HDP-HMM in conventional DAA.

The HDP-HLM assumes that the LM consists of a word bigram model. Mochihashi *et al.* [13] compared the bigram and trigram LMs and showed that the trigram assumption hardly improved the word segmentation performance although computational cost and complexity increased. Therefore, the bigram assumption must be appropriate for a word segmentation and word discovery task.

If we derive an efficient inference procedure for this two-layer hierarchical generative model, the inference procedure can infer the AM and LM simultaneously.

IV. Inference Algorithm

In this section, we derive an approximated blocked Gibbs sampler for the HDP-HLM. The sampler can simultaneously infer latent letters, latent words, an LM, and an AM. Concurrently, the inference procedure can estimate the overall double articulation structure from continuous time series data. Therefore, we propose the unsupervised machine learning method NPB-DAA. The overall inference procedure is shown in Algorithm 1.

A. Inference of Latent Words: z_s

In the HDP-HSMM, a backward filtering forward sampling procedure is adopted instead of the direct assignment procedure. When each latent state strongly depends on other neighboring latent states, the direct assignment procedure, which is a naive implementation of the Gibbs sampler, results in a poor mixing rate [8]. Johnson and Willsky [8] showed that a blocked Gibbs sampler using a backward filtering forward sampling procedure that can simultaneously sample all hidden states of an observed sequence outperforms a direct-assignment Gibbs sampler. By extending the backward filtering forward-sampling procedure and making it applicable to HDP-HLM, we can obtain an inference procedure for HDP-HLM.

The calculation of the backward messages for super states i in HDP-HSMM is as follows:

$$B_t(i) = P\big(y_{t+1:T} \mid z_{s(t)} = i, F_t = 1\big) \tag{25}$$

$$= \sum_j B_t^*(j) P\big(z_{s(t+1)} = j \mid z_{s(t)} = i\big) \tag{26}$$

$$B_t^*(i) = P\big(y_{t+1:T} \mid z_{s(t+1)} = i, F_t = 1\big) \tag{27}$$

$$= \sum_{d=1}^{T-t} B_{t+d}(i) P\big(D_{t+1} = d \mid z_{s(t+1)} = i\big) \times P\big(y_{t+1:t+d} \mid z_{s(t+1)} = i, D_{t+1} = d\big) \tag{28}$$

$$B_T(i) = 1 \tag{29}$$

where F_t is a variable indicating that t is the boundary of the super state. If $F_t = 1$, $z_{s(t)} \neq z_{s(t+1)}$. The variable $B_t(i)$ in (25) represents the probability that the latent super state $z_{s(t)} = i$ and that it transitions into a different super state at the next time step. Probability $B_t(i)$ is obtained by marginalizing over all super states j at time step $t+1$. Variable $B_t^*(j)$ in (27) represents the probability that the latent super state becomes j from time step $t+1$. This probability can be obtained by marginalizing over the duration variable in (28). Probability $P(y_{t+1\,:\,t+d} \mid z_{s(t+1)} = i, D_{t+1} = d)$ in (28) shows the emission probability of observed data $y_{t+1\,:\,t+d}$ given the condition that the duration D_{t+1} of $z_{s(t+1)}$ is d. In the HDP-HSMM, all time steps with the same super state z share the same emission distribution. Therefore, the likelihood of a super state $z_{s(t+1)}$, i.e., $P(y_{t+1:t+d} \mid z_{t+1}, D_{t+1} = d)$, can be calculated easily.

Surprisingly, in HDP-HLM, the exact same procedure of calculating backward messages as that of HDP-HSMM can be used. We obtain a message passing algorithm for HDP-HLM

Algorithm 1 Blocked Gibbs Sampler for HDP-HLM

Initialize all parameters.
Observe M time series data $\{y^m_{1:T_m}\}_{m\in\{1,2,\ldots,M\}}$.
repeat
  for $m = 1$ to M **do**
    // Backward filtering procedure
    For each $i \in \{1, 2, \ldots, N\}$, initialize messages $B_T(i) = 1$.
    for $t = T$ to 1 **do**
      For each $i \in \{1, 2, \ldots, N\}$, compute backward messages $B_{t-1}(i)$ and $B^*_{t-1}(i)$ using (25)–(28).
    end for
    // Forward sampling procedure
    Initialize $s = 1$ and $D_s^{\mathrm{sum}} = 0$
    while $D^{\mathrm{sum}_s} < T_m$ **do**
      // Sampling a super state representing a latent word
      $z_s \sim p(z_s \mid y^m_{1:T_m}, z_{s-1}, F_{D_s^{\mathrm{sum}}} = 1)$
      // Sampling duration of the super state
      $D_s \sim p(D_s | z_s, F_{D_s^{\mathrm{sum}}} = 1)$
      $D_{s+1}^{\mathrm{sum}} \leftarrow D_s^{\mathrm{sum}} + D_s$
      $s \leftarrow s + 1$
    end while
    $S^m \leftarrow s - 1$
    // Sampling a tentative latent letter sequences
    for $s = 1$ to S^m **do**
      $\bar{w}^m_s \sim P(w | y^m_{D_{s-1}^{\mathrm{sum}}+1:D_s^{\mathrm{sum}}}, \{\pi_j^{\mathrm{WM}}, \omega_j, \theta_j\}_{j=1,2,\ldots,J})$
    end for
  end for
  // Update model parameters
  Sample acoustic model parameters $\{\omega_j, \theta_j\}$ on the basis of tentatively sampled latent letter sequences $\{\bar{w}^m_s\}$.
  Sample language model parameter $\{\pi_i^{\mathrm{LM}}\}, \beta^{\mathrm{LM}}$ on the basis of sampled super states, i.e., latent words.
  Sample a word inventory $\{w_i\}_{i=1,2,\ldots,N}$ using SIR procedure (see (37)).
  Sample a word model $\{\pi_i^{\mathrm{WM}}\}, \beta^{\mathrm{WM}}$ on the basis of sampled word inventory $\{w_i\}_{i=1,2,\ldots,N}$.
until a predetermined exit condition is satisfied.

by replacing a super state z_s in HDP-HSMM with latent word z_s in HDP-HLM. Only the likelihood of the latent word w_s, i.e., $P(y_{t+1:t+d} \mid z_{s(t+1)} = i, D_{t+1} = d)$, is different between the two message passing algorithms. The likelihood of the occurrence of latent word $z_{s(t+1)} = i$ then becomes

$$P\left(y_{t+1:t+d} \middle| z_{s(t+1)} = i, D_{t+1} = d\right) = \sum_{r \in R^{(L_i,d)}} \prod_{k=1}^{L_i} P\left(r_k \mid \omega_{w_{ik}}\right) \times \prod_{m=1}^{r_k} P\left(y_{t+m+\sum_{k'=1}^{k-1} r_{k'}} \middle| \theta_{w_{ik}}\right) \quad (30)$$

$$R^{(L_i,d)} = \left\{ r \mid |r| = L_i, \sum_{k=1}^{|r|} r_k = d \right\} \quad (31)$$

where $|x|$ indicates the number of elements in vector x, and $r = (r_i, r_2, \ldots, r_{L_i})$ is an L_i-partition of duration d. By substituting (30) into (28), we can obtain a formula to calculate the backward message of HDP-HLM.

The calculation of (30) looks complicated at first glance. However, the value of (30) can be efficiently calculated using dynamic programming. If we define forward message $\alpha_t(k)$ as the probability that the kth latent letter in the relevant latent word w_i transits to the next latent letter at time t after emitting observations, forward message $\alpha_t(k)$ can be recursively calculated as follows:

$$\alpha_t(k) = \sum_{d'=1}^{t-k+1} \alpha_{t-d'}(k-1) P\left(d' \mid \omega_{w_{ik}}\right) \prod_{t'=0}^{d'-1} P\left(y_{t-t'} \mid \theta_{w_{ik}}\right) \quad (32)$$

$$\alpha_0(0) = 1. \quad (33)$$

As a result, $P(y_{t+1:t+d} \mid z_{s(t+1)} = i, D_{t+1} = d) = \alpha_d(L_i)$. By applying the calculation formula shown above, backward messages $B_t(i)$ and $B_t^*(i)$ can be calculated. Using the calculation procedure for backward messages, the forward sampling procedure proposed in the HDP-HSMM can be employed. The backward filtering forward sampling procedure enables the blocked Gibbs sampler to directly sample latent words from observation data without explicitly sampling latent letters in HDP-HLM.

In the forward sampling procedure, super state z_s and its duration D_s are sampled iteratively using backward messages as follows:

$$P\left(z_s = i | y_{1:T}, z_{s-1} = j, F_{D_s^{\text{sum}}} = 1\right) = P(z_s = i | z_{s-1} = j) B_{D_s^{\text{sum}}}(i) P\left(y_{D_s^{\text{sum}}} | z_s = i\right) \quad (34)$$

$$P\left(D_s = d | y_{1:T}, z_s = i, F_{D_s^{\text{sum}}} = 1\right) = P(D_s = d) \times \frac{P\left(y_{D_s^{\text{sum}}+1:D_s^{\text{sum}}+d} | D_s = d, z_s = i, F_{D_s^{\text{sum}}} = 1\right) B_{D_s^{\text{sum}}+d}(i)}{B^*_{D_s^{\text{sum}}}(i)} \quad (35)$$

where $D_s^{\text{sum}} = \sum_{s'<s} D_{s'}$. For further details, please refer to the original paper, in which the HDP-HSMM was introduced [8].

B. Sampling Letter Sequence for Latent Word: w_i

The sampled z_s is only an index of a latent word. Concrete letter sequences w_i for each latent word i should be sampled according to the correspondence of each subsequence of time series data $\boldsymbol{y}^k = (y_1^k, y_2^k, \ldots, y_{T^k}^k)$ to each latent word. When a latent word z_s is given, the generative model of the observation in the range of a latent word z_s can be regarded as an HDP-HSMM whose super states correspond to latent letters. Therefore, in the proposed model, each subsequence of observation data corresponding to a latent word can be considered an observed sequence generated by an HDP-HSMM. If only a single subsequence of observations corresponds to a latent word, a latent letter sequence could be sampled using an ordinal sampling procedure in the HDP-HSMM. However, observations containing the same latent word have to share the same latent letter sequence w. Therefore, latent letter sequences for observations with the same latent word are simultaneously sampled, given that they have the same latent letter sequence. We employ an approximate sampling procedure based on sampling importance resampling (SIR) [50].

If we define the observations sharing the same latent word as $\boldsymbol{y}^{1:k} = \{\boldsymbol{y}^1, \boldsymbol{y}^2, \ldots, \boldsymbol{y}^k\}$ and the shared latent letter sequence as w, the posterior probability $P(w \mid \boldsymbol{y}^{1:k})$ becomes

$$P\left(w \mid \boldsymbol{y}^{1:k}\right) \propto P(w) P\left(\boldsymbol{y}^{1:k} \mid w\right) \quad (36)$$

$$= \underbrace{P\left(w \mid \boldsymbol{y}^j\right)}_{\text{sampling}} \underbrace{P\left(\boldsymbol{y}^j\right) \prod_{i \neq j}^{k} P\left(\boldsymbol{y}^i \mid w\right)}_{\text{weight}} \quad (37)$$

where $P(\boldsymbol{y}^j)$ in (37), representing the likelihood of the observation, can be calculated using the backward filtering procedure in the HDP-HSMM. Probability $P(\boldsymbol{y}^i | w)$ can also be calculated in the same way as (30) if w is given. The HDP-HSMM also provides a sampling procedure for $P(w|\boldsymbol{y}^j)$. Therefore, if we consider $P(w|\boldsymbol{y}^j)$ as the proposed distribution and $P(\boldsymbol{y}^j) \prod_{i \neq j}^{k} P(\boldsymbol{y}^i | w)$ as a weight, the SIR procedure can be employed [50]. Specifically, after a set of w are sampled from the proposed distribution $P(w \mid \boldsymbol{y}^j)$ $j = 1, 2, \ldots, k$, a final sample is drawn from the set with a probability proportional to each sample's weight. Using this procedure, the proposed model can approximately sample a latent letter sequence w_i for the ith latent word.

C. Sampling Model Parameters

After sampling latent words $\{z_s\}$ for each observation data and sampling letter sequences for the latent words, other parameters can be updated. Parameters of the LM, i.e., $\{\pi_i^{\text{LM}}\}$ and β^{LM}, can be updated on the basis of latent word sequences. Parameters of the WM, i.e., $\{\pi_j^{\text{WM}}\}$ and β^{WM}, can be updated on the basis of sampled letter sequences for latent words. Parameters for the AM, i.e., $\{\omega_j\}$ and $\{\theta_j\}$, can be updated if each hidden state x_t is determined for each y_t. During the SIR process for sampling a letter sequence, $\{\bar{w}_s^m\}$ in Algorithm 1 are subsidiarily obtained. To accelerate the mixing rate, the subsidiary sampling results $\{\bar{w}_s^m\}$ obtained in the SIR are used for updating the AM parameters. These parameters can be

sampled in the same way as the HDP-HSMM. For more details, we refer to the original paper in which the HDP-HSMM were introduced [8]. Finally, the overall sampling procedure is obtained, as described in Algorithm 1.

D. NPB-DAA

Based on the generative model, HDP-HLM, and its inference algorithm shown in Algorithm 1, the proposed NPB-DAA is obtained, finally. By assuming HDP-HLM as a generative model of observed time series data, and by inferring latent variables of the model, we can analyze latent double articulation structure, i.e., hierarchically organized latent words and phonemes, of the data in an unsupervised manner. We call the novel unsupervised DAA NPB-DAA.

V. Experiment 1: Synthetic Data

We conducted an experiment using a synthetic dataset that explicitly has a double articulation structure to validate our proposed method.

A. Conditions

To validate the ability of our proposed method to infer a latent double articulation structure in time series data, we applied the proposed NPB-DAA based on the HDP-HLM to synthetic time series data. The conventional DAA was employed as a comparative method. The time series data are generated using five letters $\{j\}_{j\in J} = \{1, 2, 3, 4, 5\}$ and four words $\{w\}_{w\in W} = \{[1, 3, 5], [3, 2], [4, 1, 5, 2], [1, 5]\}$ where J is a set of letters and W is a set of words. The four words were generated randomly. The sequence $w_i = [w_{i1}, w_{i2}, \ldots, w_{iL_i}]$ represents a word that is generated by combining $\{w_{i1}, w_{i2}, \ldots, w_{iL_i}\}$ sequentially where w_{ik} denotes the kth letter of w_i. The durations of the letters were assumed to follow Poisson distributions and their parameters were drawn from a Gamma distribution whose parameters were $\alpha = 50$ and $\beta = 10$. The emission distribution was assumed to be a Gaussian distribution whose parameters were $\mu = 5i, \sigma^2 \in \{0.1, 0.5, 1.0\}$, where i represents the index of latent letters. The variance of the emission distribution was changed in stages, and the inference results were compared. Forty time series data items were generated from 20 types of latent word sequences. Sixteen of them were pairs of words in W, e.g., ([1, 3, 5], [1, 5]) and ([3, 2], [3, 2]). Four of them were three-word sentences, e.g., ([3, 2], [1, 3, 5], [1, 5]). A sequence of latent words is represented by $(w_1, w_2, \ldots, w_n)$. Two observations were generated from each word sequence.

We set the parameters of the NPB-DAA as follows: the hyperparameters for the latent LM were $\gamma^{\mathrm{LM}} = 10.0, \alpha^{\mathrm{LM}} = 10.0$, and the maximum number of words was six for weak-limit approximation. The hyperparameters for the latent WM were $\gamma^{\mathrm{WM}} = 10.0, \alpha^{\mathrm{WM}} = 10.0$, and the maximum number of letters was seven for weak-limit approximation. The hyperparameters of the duration distributions were set to $\alpha = 50$ and $\beta = 10$, and those of the emission distributions were set to $\mu_0 = 0, \sigma_0^2 = 1.0, \kappa_0 = 0.01, \nu_0 = 1$. The Gibbs sampling procedure was iterated 100 times.

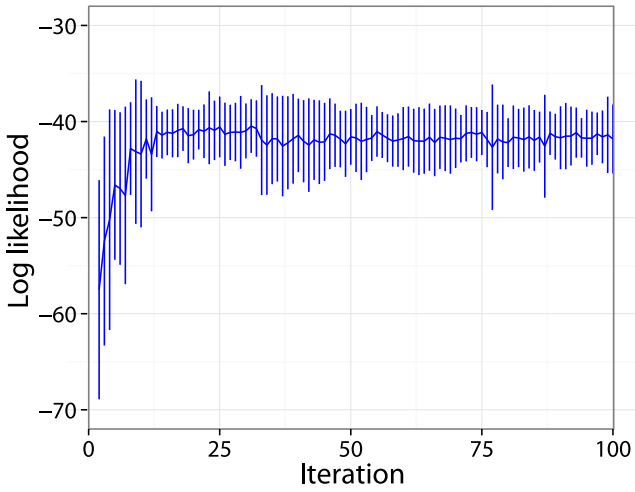


Fig. 4. Log-likelihood profile through Gibbs sampling ($\sigma^2 = 1.0$).

For the conventional DAA, we set the hyperparameters of the sticky HDP-HMM to be as similar to those of the NPB-DAA as possible. In this condition, the latticelm software[1] developed by Neubig *et al.* [25] was used for NPYLM. The hyperparameters of the NPYLM used in the conventional DAA were set to $\alpha = 0.1$ and $d = 0.1$.

The hyperparameters in the NPB-DAA were heuristically given in a top-down manner by referring to the size of the state space and the approximate duration of a phoneme. Those of the Pitman–Yor LM were set to the default values of the software.

B. Results

The average log-likelihood is shown in Fig. 4, where error bars represent the standard deviation of 30 trials. These results show that the proposed inference procedure worked appropriately, gradually sampling more probable latent variables as the iterations increased.

In contrast with ordinal speech recognition tasks, the target task (language acquisition and double articulation analysis) is an unsupervised learning task. Specifically, it is a clustering task. Therefore, it is difficult to evaluate the methods' performance from the viewpoint of precision and recall because the estimated index of a cluster and the label corresponding to the ground truth data are usually different. We evaluated the obtained result using the adjusted rand index (ARI), which quantifies the performance of a clustering task [51]. If all data items are clustered randomly or only to one cluster, the ARI becomes 0. By contrast, if the results of clustering are the same as those of the ground truth data, the ARI becomes 1.

Table I shows the ARI for the estimated latent letters. The ARI for estimated latent letters shows how accurately each method estimated latent letters, which correspond to phonemes in speech signals. Table II shows the ARI for estimated latent

[1] latticelm: http://www.phontron.com/latticelm/index.html.

TABLE I
ARI FOR ESTIMATED LATENT LETTERS

σ^2	0.1	0.5	1.0
Conventional DAA (sticky HDP-HMM)	0.845	0.832	0.649
NPB-DAA	**0.984**	**0.895**	**0.938**

TABLE II
ARI FOR ESTIMATED LATENT WORDS

σ^2	0.1	0.5	1.0
Conventional DAA (sticky HDP-HMM + NPYLM)	0.122	0.107	0.125
NPB-DAA	**0.594**	**0.509**	**0.618**

words. The ARI for estimated latent words shows how accurately each method estimated latent letters, which correspond to words in speech signals. In both tables, each column shows ARIs for different σ^2. A higher ARI implies more accurate estimation of the latent variables.

Although the ARI for the latent letters obtained by conventional DAA decreases when the variance σ^2 increases, that of NPB-DAA did not decrease as much. As the ARIs for latent words show, the performance of word segmentation by conventional DAA was poor, even when the ARI for latent letters was larger than 0.8. In contrast, the ARI for latent words estimated by NPB-DAA was over 0.5 in all conditions. This shows that the NPB-DAA can mitigate the ill effects of phoneme recognition errors in the word segmentation task, and obtained knowledge about words can improve phoneme recognition performance by using contextual information. Fig. 5 shows the change in ARI through iterations in the case of $\sigma^2 = 1.0$. This shows that the ARI also increased gradually while log likelihood increases, as in Fig. 4. These results suggest that the NPB-DAA is an appropriate generative model because better word segmentation performance corresponded to higher likelihood of the model.

To check the effects of the limit on weak-limit approximation, we ran an experiment where the maximum number of letters was 20 for weak-limit approximation. The ARI for the estimated latent words were $\{0.682, 0.650, 0.604\}$, those for estimated latent letters were $\{0.967, 0.899, 0.878\}$, and the estimated number of latent letters were $\{5.6, 6.3, 6.6\}$ on average for $\sigma^2 = \{0.1, 0.5, 1.0\}$. This result shows that our model can work appropriately to estimate the number of latent states owing to the nature of Bayesian nonparametrics when the limit is sufficiently large.

An example of estimated latent variables is shown in Fig. 6, which shows the results for time series data generated from the latent word sequence $([3, 2], [1, 3, 5], [1, 5])$. The input time series data is shown at the very top of the figure. The top of each panel shows the true latent letters or latent words, whereas the panel beneath shows the inferred results. The vertical axes represent the iteration of the Gibbs sampling. In Fig. 6, the figure in the middle shows a latent word sequence estimated using the proposed method, and the figure at the bottom shows the estimated boundaries of the latent words. These results show that the inference procedure works consistently and can estimate an adequate boundary for the latent words given the data.

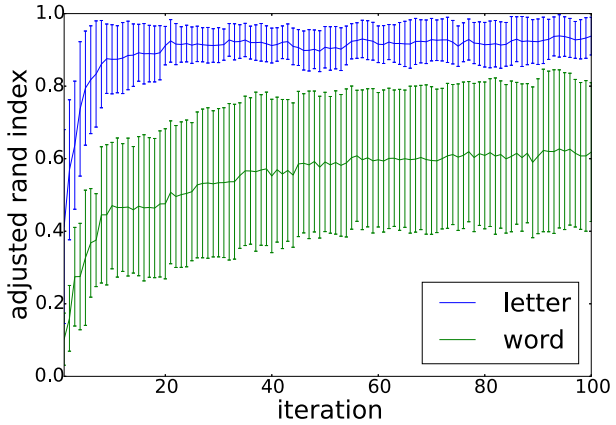


Fig. 5. ARI profile through Gibbs sampling ($\sigma^2 = 1.0$).

These results show that the proposed method is a more effective machine learning method for estimating a latent double articulation structure embedded in time series data.

VI. EXPERIMENT 2: CONTINUOUS JAPANESE VOWEL SPEECH SIGNAL

In the second experiment, we evaluated our proposed method using Japanese vowel speech signals to test the applicability of the proposed method to actual human continuous speech signal.

A. Conditions

We prepared four datasets. Each dataset corresponds to a speaker, and consisted of 60 audio data items. We asked two male and two female Japanese speakers to read 30 artificial sentences aloud two times at a natural speed, and recorded his/her voice. The 30 sentences were prepared using five words {aioi, aue, ao, ie, uo}, which consisted of five Japanese vowels {a, i, u, e, o} representing {ä, i, ɯ$^{\beta}$, e̞, o̞} in phonetic symbols, respectively. By reordering the five words, we prepared 25 two-word sentences, e.g., "ao aioi," "uo aue," and "aioi aioi," and five three-word sentences, i.e., "uo aue ie," "ie ie uo," "aue ao ie," "ao ie ao," and "aioi uo ie." The set of two-word sentences consisted of all types of word pairs ($5 \times 5 = 25$). The set of three-word sentences were generated randomly. A separate model was trained on each dataset corresponding to each speaker, and evaluated.

The recorded data were encoded into 13-D mel-frequency cepstrum coefficient (MFCC) time series data using the HMM Toolkit.[2] The frame size and shift were set to 25 and 10 ms, respectively. Twelve-dimensional MFCC data was obtained as input data by eliminating power information from the original 13-D MFCC data. As a result, 12-D time series data at a frame rate of 100 Hz were obtained.

The hyperparameters for the latent LM were set to $\gamma^{\mathrm{LM}} = 10.0$ and $\alpha^{\mathrm{LM}} = 10.0$, and the maximum number of words was set to seven for weak-limit approximation. The hyperparameters for the latent WM were $\gamma^{\mathrm{WM}} = 10.0$ and $\alpha^{\mathrm{WM}} = 10.0$, and the maximum number of letters was seven for weak-limit

[2]HMM Toolkit: http://htk.eng.cam.ac.uk/.

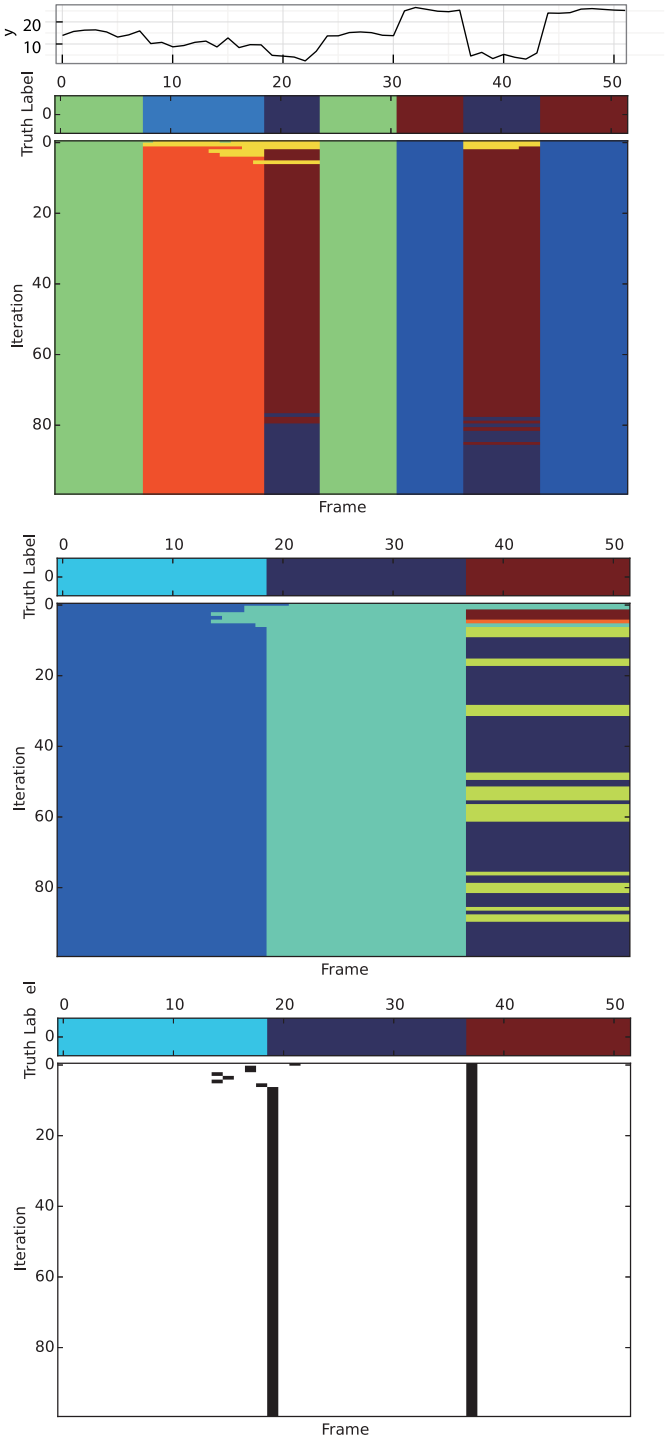


Fig. 6. Example of inference results for sample data ([3, 2], [1, 3, 5], [1, 5]) and $\sigma^2 = 1.0$: (top) observation data, (upper middle) latent letters, (lower middle) latent words, and (bottom) the boundaries of latent words. Different colors denote different states.

approximation. The hyperparameters of the duration distributions were set to $\alpha = 200$ and $\beta = 10$, and those of the emission distributions were set to $\mu_0 = 0, \sigma_0^2 = 1.0, \kappa_0 = 0.01$, and $\nu_0 = 17 = (\text{dimension}+5)$.

For the conventional DAA, we set the hyperparameters of the sticky HDP-HMM to be as similar to those of the NPB-DAA as possible. The hyperparameters for the NPYLM used in the conventional DAA were set to $\alpha = 0.1$ and $d = 0.1$. The Gibbs sampling procedure was iterated 100 times. With different random number seeds, 20 trials were performed.

The parameters in the NPB-DAA were given in a top-down manner heuristically by referring to the size of the state space and the approximate duration of a phoneme. Those of the Pitman–Yor LM were set to the default values of the software.

As a baseline method, we employed an open-source continuous speech recognition engine, Julius,[3] which is widely used in Japanese speech recognition tasks. Julius's AM is trained by using a large number of speech data in a supervised manner. We prepared four conditions for Julius. The first one was called "Julius (phoneme + NPYLM)." In this condition, we used Julius as a phoneme recognition system by preparing a phoneme dictionary containing five Japanese vowels {a, i, u, e, o}. Moreover, Julius's dictionary also contains silB and silE to represent silence due to system requirements. After encoding continuous speech signals into phoneme sequences using Julius as a phoneme recognizer, unsupervised morphological analysis based on the NPYLM was conducted to discover words and an LM. The second condition was called "Julius (phoneme + latticelm)." In this condition, we also used latticelm, which is an unsupervised morphological analyzer for lattice output from an ASR system. The method was proposed by Neubig *et al.* [25] as an extension of Mochihashi's [13] NPYLM. In this condition, the latticelm software was used too.

In the third and fourth conditions, called "Julius (monophone + word dictionary)" and "Julius (triphone + word dictionary)," respectively, we prepared a complete word dictionary that contained all of the words that appeared in the target speech signal, i.e., {aioi, aue, ao, ie, uo}, for Julius. This condition provides almost an upper bound for the performance of our task. Except for in Julius (triphone + word dictionary), Julius uses a monophone-based AM contained in the dictation kit. The AM is trained in a supervised manner using a large number of labeled speech data. Julius (triphone + word dictionary) used a triphone-based acoustic model for comparison.

B. Results

We provided word and letter ground truth labels to all frames of the speech signal data and evaluated the relationship between the truth labels and estimated latent letter and word indices.

The results are shown in Table III. Check marks in the AM and LM columns indicate that the method used a pretrained AM and the given true LM, respectively. Letter ARI shows the ARI of phoneme clustering. A high letter ARI means more accurate phoneme acquisition and recognition. Word ARI shows the ARI of word clustering. A higher word ARI means more accurate word discovery and recognition.

[3]Open-Source Large Vocabulary CSR Engine Julius: http://julius.sourceforge.jp/. The Linux binary dictation-kit-v4.3.1-linux.tgz was used in this experiment. The software encodes the recorded data into 36-D MFCC data including dynamic features and uses them for speech recognition.

TABLE III
ARI FOR ESTIMATED LATENT LETTERS AND WORDS

Method	Letter ARI	Word ARI	AM	LM
NPB-DAA (MAP)	**0.596**	**0.529**		
NPB-DAA	0.561	0.401		
Conventional DAA	0.590	0.090		
Julius (phoneme dictionary + NPYLM)	0.486	0.297	✓	
Julius (phoneme dictionary + latticelm)	0.554	0.337	✓	
Julius (monophone + word dictionary)	0.586	0.487	✓	✓
Julius (triphone + word dictionary)	0.548	0.616	✓	✓

Each row corresponds to each method explained in the conditions. A separate model was trained on the data from each speaker, i.e., each model was trained to be a speaker-dependent model. We compute the averages of the results across the four speakers, and show them in Table III. The results of "NPB-DAA" and conventional DAA show the ARI averaged over 20 trials. In contrast, "NPB-DAA [maximum *a posteriori* probability (MAP)]" obtained the MAP of the 20 trials. An advantage of the NPB-DAA is that the method can calculate the posterior probability of a given dataset after the learning phase because the NPB-DAA is derived from a generative model, i.e., HDP-HLM, which integrates the LM and AM. In contrast with the conventional DAA and similar methods that do not have appropriate generative models, the NPB-DAA can obtain an appropriate learning result by referring to the probability. The rows with MAP in Table III show that this probability is an adequate criterion for selecting a learning result.

The results show that the NPB-DAA (MAP) outperformed not only the conventional DAA but also Julius-based word discovery systems whose AMs were trained in supervised manner. One reason is that the AMs of the DAAs were trained only from one participant's speech signals, in contrast, Julius's AM was trained by the speech signals of many speakers. In other words, NPB-DAA acquired speaker-dependent AM in contrast with that Julius used speaker-independent AM. This adaptation of AM to the speaker must have increased the NPB-DAA's performance.

The results show that a naive application of the NPYLM to recognized phoneme sequences results in poor word acquisition performance, especially in conventional DAA. Because the theory of the NPYLM does not presume that letter sequences have recognition errors, the existence of phoneme recognition error deteriorates word segmentation performance. The methods that simply apply an NPYLM to obtained phoneme sequences, i.e., the conventional DAA and Julius (phoneme dictionary + NPYLM), output bad results in the word ARI compared with those of the letter ARI. However, latticelm, which presumes phoneme recognition errors to some extent, could not dramatically improve the performance of word acquisition in our experimental setting.

In contrast, Julius (triphone + word dictionary) improved its word ARI performance with respect to letter ARI performance. Julius (monophone + word dictionary) also kept its performance high with respect to the word recognition task compared with the phoneme recognition task. We note that the word error rate was 32.8% and the phoneme error rate was 28.1% in Julius (monophone + word dictionary).

In the research field of ASR, it is widely known that a good LM improves word and phoneme recognition performance. The NPB-DAA could not improve the performance of word ARI with respect to letter ARI performance. However, it obtained an adequate LM and prevented the score of the word ARI from becoming far worse than that of the letter ARI. To achieve such an error-proof word acquisition, the direct inference of latent words are important in NPB-DAA. In the inference procedure described in Section III, latent words are sampled directly without sampling latent letters while marginalizing all possible latent letter sequences. This achieves an effect similar to that of a given LM in the inference process

Typical examples of the estimation results are shown in Table IV for NPB-DAA and conventional DAA. Each number in parentheses represents an estimated phoneme label, each space represents a phoneme boundary, each number in bold style represents a sampled index of a word, and "/" represents a boundary between successive words. For example, "ao ie" was divided into two words, i.e., "5 0 1" and "6 3 4 6," in the NPB-DAA results, and their word indices were 3 and 4. In Table IV, the sampled letters corresponding to the word "ie" are underlined. Although conventional DAA could not estimate ie as a single word, the NPB-DAA could estimate ie to be a single word: "4." In the conventional DAA results, several phoneme recognition errors can be found. The errors completely deteriorated the following chunking process, i.e., unsupervised morphological analysis using an NPYLM, as past research has frequently pointed out. As shown in Table IV, NPB-DAA had some phoneme recognition errors. However, in the NPB-DAA, latent words are sampled on the basis of the marginalized phoneme distribution before sampling concrete phoneme sequences. This property of the sampling procedure seemed to improve the performance of NPB-DAA.

An example of the estimated latent variables is shown in Fig. 7, which shows the results for time series data corresponding to a vowel sequence, ao ie ao. The input time series data, i.e., 12-D MFCC time series data, are shown at the top of the figures. The middle and the bottom figures show the inference process. The top of each figure shows the true latent letters or latent words, whereas the bottom shows the inferred result. The vertical axes represent the number of Gibbs sampling iterations. This shows that the inference procedure worked for human vowel sequence data, and could estimate an adequate unit for each word.

Let us further examine the characteristics of the segmentation results of the NPB-DAA. Table IV shows that some of the estimated latent words have a latent letter "6" at their head or tail. The latent letter 6 represents silence observed during the transition from one vowel to another. Silence in speech signals and the transitional sounds observed between two phonemes were treated in the same manner as other uttered sounds in our model. The question of whether such signals should be treated in the same way as other sounds in a generative model calls

TABLE IV
EXAMPLE WORD DISCOVERY RESULTS

Vowel sequence	Estimated NPB-DAA results	Estimated conventional DAA results
ao ie	**3** (5 0 1) / **4** (6 3 4 6)	**226** (2 0 3 4 1 5 4 1)
ao ie ao	**3** (5 0 1) / **4** (6 3 4 6) / **3** (5 0 1) / **0** (6 4 6)	**494** (3) / **675** (2 3 0) / **374** (1 5 4 1 2 0 1)
aue ie	**6** (6 5 1 2 6 4) / **4** (6 3 4 6)	**329** (2 3 8 4 5 4 1)
ie ie	**4** (6 3 4 6) / **4** (6 3 4 6)	**389** (5 4 1 4 1 5 4 1)
ie uo	**4** (6 3 4 6) / **5** (5 1 2) / **3** (5 0 1)	**401** (5 4 1 8 0 1)
ie aioi	**4** (6 3 4 6) / **1** (5 6 4 6 3 6 1) / **4** (6 3 4 6)	**813** (5 4 1 2 4 5) / **832** (4 3 0 3 4 5 1)

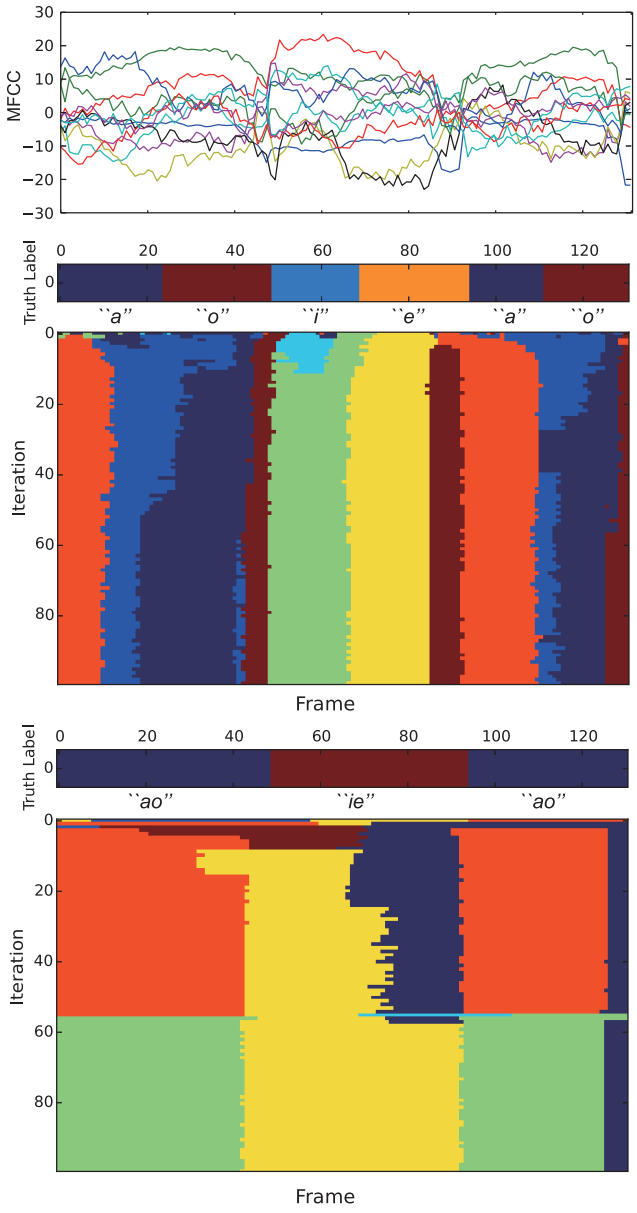


Fig. 7. Example of inference results for ao ie ao. MFCC feature vectors are plotted in the top panel. The middle and bottom panels show the inference results of latent letters and latent words, respectively. Different colors denote different states.

for further investigation. In our model, a phoneme is simply represented by a single Gaussian distribution, although many past speech recognition systems assign a richer structure to a phoneme, e.g., a three-state left-to-right HMM with Gaussian mixture model emission distributions. There is room for investigating whether a phoneme model, i.e., a latent letter, should itself have a more complex structure, or if a double articulation hierarchy is sufficient from the viewpoint of unsupervised word discovery tasks.

An interesting result that represents a characteristic of the NPB-DAA is the latent word "**4** (6 3 4 6)" estimated at the end of "ie aioi." The speech signals corresponding to this "**4**" were a kind of transitional sound observed following "aioi." The NPB-DAA directly inferred the latent word by marginalizing latent letters. In this case, it seems that "**4**" was more likely than other latent words, and the NPB-DAA hence generated this result. This can be regarded as a side effect of our approach, i.e., the marginalization of latent letter sequences in a latent word. We are confident that the marginalization of latent letters and the direct inference of word sequences are important to improving the performance of the unsupervised word segmentation of continuous speech signals, but there is room to consider this side effect.

Note that the NPB-DAA performed unsupervised word discovery under the condition that the training data consisted of speech signals uttered by one speaker, in contrast with Julius, whose AM was trained using many speakers' speech signals. Speaker-independent, unsupervised word discovery from continuous speech signals remains a challenging problem because the acoustic features of phonemes heavily depend on the speaker. When we gave four speakers' speech signals to the NPB-DAA at the same time, the letter ARI and the word ARI decreased to 0.297 and 0.104, respectively. By contrast, those produced by Julius with a triphone AM and a true word dictionary were 0.552 and 0.599, respectively. In the experiment, 120 audio data items that were recorded by asking two male and two female Japanese speakers to read 30 artificial sentences were used, i.e., a half of the data items used in the main experiment due to computational cost. It was observed that speaker "dependent" phoneme models were obtained by the NPB-DAA, i.e., speech signals representing the same phoneme uttered by deferent persons tended to be clustered to different latent letters. To develop a machine learning method that enables a robot to obtain LM and AM independent of speakers, or automatically adapting to different speakers is one of our future challenges.

VII. CONCLUSION

In this paper, we proposed NPB-DAA for direct and simultaneous acquisition of LM and AM from continuous speech signals in an unsupervised manner. For this purpose, we proposed an integrative generative model called the HDP-HLM by extending HDP-HSMM. Based on the generative model, we derived an inference procedure by extending the blocked Gibbs

sampler originally proposed for HDP-HSMM. The method is expected to enable a developmental robot to simultaneously obtain LM and AM directly from continuous speech signals. To evaluate the performance of the proposed method, two experiments were performed. In the first experiment, the proposed method was applied to synthetic data, and it was shown that the method can successfully infer latent words embedded in time series data in an unsupervised manner. In the second experiment, we applied the proposed method to actual human Japanese vowel sequences. The result showed that the proposed method outperformed a conventional two-stage sequential method, conventional DAA, and a baseline ASR method.

One of the most important challenges in our future work is to achieve complete human language acquisition from speech signals. We did not achieve complete language acquisition from speech signals that includes consonants as well as vowels in this paper. Language acquisition from more natural speech signals like child-directed speech by human parents are also part of our future work. To achieve these aims, we still have two main problems: feature extraction and computational cost.

To address these problems, more sophisticated feature extraction methods are needed. Deep learning has gained attention recently because of its impressive feature extraction performance. Integrating a deep learning method into the NPB-DAA should improve its performance.

Computational cost is another problem. Even though the size of the dataset used in experiment 2 was very small, it took approximately 240 min for 100 iterations using an Intel Xeon CPU E5-2650 v2 2.60 GHz, 8 cores $\times$ 16 CPU. In particular, the computational cost of the blocked Gibbs sampler was $O(TL_{\max}d_{\max}^3N_{\max}^2)$, where $L_{\max}$ is the maximum number of latent letters for a word, $d_{\max}$ is the maximum duration of a word, and $N_{\max}$ is the maximum number of words. To apply the proposed method to a larger dataset, improving its computational cost will be necessary.

Currently, the accuracy of the language acquisition is still limited, as shown in Table III. In this paper, we focused on a language acquisition method based on distributional cues and proposed a mathematical model for language acquisition. Obviously, distributional cues are not enough for more accurate language acquisition. As suggested by several computational and robotic studies, making use of co-occurrence cues improves the accuracy of language acquisition [23], [24]. The proposed HDP-HLM is a fully probabilistic generative model. Therefore, introducing other factors into consideration is relatively easier than for other heuristic models. This is also advantage of our approach. Combining prosodic and co-occurrence cues into the NPB-DAA, and obtaining a more accurate and more plausible constructive developmental language acquisition model is also a direction for future research.

REFERENCES

[1] R. N. Aslin, J. Z. Woodward, N. P. LaMendola, and T. G. Bever, "Models of word segmentation in fluent maternal speech to infants," in *Signal to Syntax: Bootstrapping From Speech to Grammar in Early Acquisition*, J. L. Morgan and K. Demuth, Eds. New York, NY, USA: Psychology Press, 1995, pp. 117–134.

[2] T. Kawaharay *et al.*, "Free software toolkit for Japanese large vocabulary continuous speech recognition," in *Proc. Int. Conf. Spoken Lang. Process. (ICSLP)*, Beijing, China, 2000, pp. 3073–3076.

[3] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Audio, Speech, Language Process.*, vol. 20, no. 1, pp. 30–42, Jan. 2012.

[4] J. R. Saffran, E. L. Newport, and R. N. Aslin, "Word segmentation: The role of distributional cues," *J. Memory Lang.*, vol. 35, no. 4, pp. 606–621, 1996.

[5] J. R. Saffran, R. N. Aslin, and E. L. Newport, "Statistical learning by 8-month-old infants," *Science*, vol. 274, no. 5294, pp. 1926–1928, 1996.

[6] E. D. Thiessen and J. R. Saffran, "When cues collide: Use of stress and statistical cues to word boundaries by 7- to 9-month-old infants," *Develop. Psychol.*, vol. 39, no. 4, pp. 706–716, 2003.

[7] P. K. Kuhl, "Cracking the speech code: How infants learn language," *Acoust. Sci. Technol.*, vol. 28, no. 2, pp. 71–83, 2007.

[8] M. J. Johnson and A. S. Willsky, "Bayesian nonparametric hidden semi-Markov models," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 673–701, Jan. 2013.

[9] M. R. Brent, "An efficient, probabilistically sound algorithm for segmentation and word discovery," *Mach. Learn.*, vol. 34, nos. 1–3, pp. 71–105, 1999.

[10] A. Venkataraman, "A statistical model for word discovery in transcribed speech," *Comput. Linguist.*, vol. 27, no. 3, pp. 351–372, 2001.

[11] S. Goldwater, T. L. Griffiths, and M. Johnson, "Contextual dependencies in unsupervised word segmentation," in *Proc. 21st Int. Conf. Comput. Linguist. 44th Annu. Meeting Assoc. Comput. Linguist.*, 2006, pp. 673–680.

[12] S. Goldwater, T. L. Griffiths, and M. Johnson, "A Bayesian framework for word segmentation: Exploring the effects of context," *Cognition*, vol. 112, no. 1, pp. 21–54, Jul. 2009.

[13] D. Mochihashi, T. Yamada, and N. Ueda, "Bayesian unsupervised word segmentation with nested Pitman-Yor language modeling," in *Proc. Joint Conf. 47th Annu. Meeting ACL 4th Int. Joint Conf. Nat. Lang. Process. AFNLP (ACL-IJCNLP)*, Singapore, 2009, pp. 100–108.

[14] M. Johnson and S. Goldwater, "Improving nonparameteric Bayesian inference: Experiments on unsupervised word segmentation with adaptor grammars," in *Proc. Human Lang. Technol. Annu. Conf. North American Chapter Assoc. Comput. Linguist.*, Boulder, CO, USA, 2009, pp. 317–325.

[15] M. Chen, B. Chang, and W. Pei, "A joint model for unsupervised Chinese word segmentation," in *Proc. Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*, Doha, Qatar, 2014, pp. 854–863.

[16] P. Magistry, "Unsupervized word segmentation: The case for Mandarin Chinese," in *Proc. 50th Annu. Meeting Assoc. Comput. Linguist. Short Papers*, vol. 2. 2012, pp. 383–387.

[17] S. Sakti, A. Finch, R. Isotani, H. Kawai, and S. Nakamura, "Unsupervised determination of efficient Korean LVCSR units using a Bayesian Dirichlet process model," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Prague, Czech Republic, 2011, pp. 4664–4667.

[18] D. K. Roy and A. P. Pentland, "Learning words from sights and sounds: A computational model," *Cogn. Sci.*, vol. 26, no. 1, pp. 113–146, 2002.

[19] N. Iwahashi, "Interactive learning of spoken words and their meanings through an audio-visual interface," *IEICE Trans. Inf. Syst.*, vol. E91-D, no. 2, pp. 312–321, 2008.

[20] N. Iwahashi, "Language acquisition through a human–robot interface by combining speech, visual, and behavioral information," *Inf. Sci.*, vol. 156, nos. 1–2, pp. 109–121, 2003.

[21] N. Iwahashi, K. Sugiura, R. Taguchi, T. Nagai, and T. Taniguchi, "Robots that learn to communicate: A developmental approach to personally and physically situated human-robot conversations," in *Proc. Dialog Robots Papers AAAI Fall Symp.*, Arlington County, VA, USA, 2010, pp. 38–43.

[22] T. Araki *et al.*, "Online learning of concepts and words using multimodal LDA and hierarchical Pitman-Yor language model," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Vilamoura, Portugal, Oct. 2012, pp. 1623–1630.

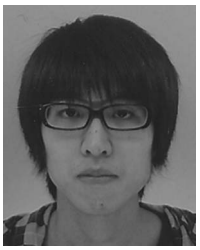
[23] T. Nakamura *et al.*, "Mutual learning of an object concept and language model based on MLDA and NPYLM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Chicago, IL, USA, 2014, pp. 600–607.

[24] R. Taguchi, Y. Yamada, K. Hattori, T. Umezaki, and M. Hoguro, "Learning place-names from spoken utterances and localization results by mobile robot," in *Proc. Interspeech*, Florence, Italy, 2011, pp. 1325–1328.

[25] G. Neubig, M. Mimura, S. Mori, and T. Kawahara, "Bayesian learning of a language model from continuous speech," *IEICE Trans. Inf. Syst.*, vol. E95-D, no. 2, pp. 614–625, 2012.
[26] J. Heymann, O. Walter, R. Haeb-Umbach, and B. Raj, "Unsupervised word segmentation from noisy input," in *IEEE Workshop Autom. Speech Recognit. Understand. (ASRU)*, Olomouc, Czech Republic, 2013, pp. 458–463.
[27] J. Heymann, O. Walter, R. Haeb-Umbach, and B. Raj, "Iterative Bayesian word segmentation for unsupervised vocabulary discovery from phoneme lattices," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Florence, Italy, 2014, pp. 4057–4061.
[28] M. Elsner, S. Goldwater, N. Feldman, and F. Wood, "A joint learning model of word segmentation, lexical acquisition, and phonetic variability," in *Proc. Conf. Empir. Methods Nat. Lang. Process.*, Seattle, WA, USA, 2013, pp. 42–54.
[29] B. M. Lake, G. K. Vallabha, and J. L. McClelland, "Modeling unsupervised perceptual category learning," *IEEE Trans. Auton. Mental Develop.*, vol. 1, no. 1, pp. 35–43, Jun. 2009.
[30] G. K. Vallabha, J. L. McClelland, F. Pons, J. F. Werker, and S. Amano, "Unsupervised learning of vowel categories from infant-directed speech," *Proc. Nat. Acad. Sci. USA*, vol. 104, no. 33, pp. 13273–13278, 2007.
[31] N. H. Feldman, T. L. Griffiths, S. Goldwater, and J. L. Morgan, "A role for the developing lexicon in phonetic category acquisition," *Psychol. Rev.*, vol. 120, no. 4, pp. 751–778, 2013.
[32] C.-Y. Lee and J. Glass, "A nonparametric Bayesian approach to acoustic model discovery," in *Proc. 50th Annu. Meeting Assoc. Comput. Linguist. Long Papers*, 2012, pp. 40–49.
[33] C.-Y. Lee, Y. Zhang, and J. R. Glass, "Joint learning of phonetic units and word pronunciations for ASR," in *Proc. Conf. Empirical Methods Nat. Lang. Process.*, 2013, pp. 182–192.
[34] H. Brandl, B. Wrede, F. Joublin, and C. Goerick, "A self-referential childlike model to acquire phones, syllables and words from acoustic speech," in *Proc. IEEE Int. Conf. Develop. Learn.*, Monterey, CA, USA, Aug. 2008, pp. 31–36.
[35] O. Walter, T. Korthals, R. Haeb-Umbach, and B. Raj, "A hierarchical system for word discovery exploiting DTW-based initialization," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand. (ASRU)*, Olomouc, Czech Republic, 2013, pp. 386–391.
[36] H. Kamper, A. Jansen, and S. Goldwater, "Fully unsupervised small-vocabulary speech recognition using a segmental Bayesian model," in *Proc. INTERSPEECH*, Dresden, Germany, 2015, pp. 678–682.
[37] C.-Y. Lee, T. J. O. Donnell, and J. Glass, "Unsupervised lexicon discovery from acoustic input," *Trans. Assoc. Comput. Linguist.*, vol. 3, pp. 389–403, Feb. 2015.
[38] T. Taniguchi and S. Nagasaka, "Double articulation analyzer for unsegmented human motion using Pitman-Yor language model and infinite hidden Markov model," in *Proc. IEEE/SICE Int. Symp. Syst. Integr. (SII)*, Kyoto, Japan, 2011, pp. 250–255.
[39] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky, "A sticky HDP-HMM with application to speaker diarization," *Ann. Appl. Stat.*, vol. 5, no. 2A, pp. 1020–1056, 2009.
[40] K. Takenaka, T. Bando, S. Nagasaka, T. Taniguchi, and K. Hitomi, "Contextual scene segmentation of driving behavior based on double articulation analyzer," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Vilamoura, Portugal, 2012, pp. 4847–4852.
[41] T. Taniguchi, S. Nagasaka, K. Hitomi, N. P. Chandrasiri, and T. Bando, "Semiotic prediction of driving behavior using unsupervised double articulation analyzer," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Alcalá de Henares, Spain, 2012, pp. 849–854.
[42] T. Taniguchia, S. Nagasaka, K. Hitomi, K. Takenaka, and T. Bando, "Unsupervised hierarchical modeling of driving behavior and prediction of contextual changing points," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 1746–1760, Aug. 2014.
[43] S. Nagasaka, T. Taniguchi, G. Yamashita, K. Hitomi, and T. Bando, "Finding meaningful robust chunks from driving behavior based on double articulation analyzer," in *Proc. IEEE/SICE Int. Symp. Syst. Integr. (SII)*, Fukuoka, Japan, 2012, pp. 535–540.
[44] T. Bando, K. Takenaka, S. Nagasaka, and T. Taniguchi, "Unsupervised drive topic finding from driving behavioral data," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Gold Coast, QLD, Australia, 2013, pp. 177–182.
[45] T. Bando, K. Takenaka, S. Nagasaka, and T. Taniguchi, "Automatic drive annotation via multimodal latent topic model," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Tokyo, Japan, 2013, pp. 2744–2749.
[46] K. Takenaka, T. Bando, S. Nagasaka, and T. Taniguchi, "Drive video summarization based on double articulation structure of driving behavior," in *ACM Multimedia*, Nara, Japan, 2012, pp. 1169–1172.
[47] K. P. Murphy. (Nov. 2002). *Hidden Semi-Markov Models (HSMMs)*. [Online]. Available: http://www.cs.ubc.ca/~murphyk/Papers/segment.pdf
[48] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical Dirichlet processes," *J. Amer. Stat. Assoc.*, vol. 101, no. 476, 2006, pp. 1566–1581
[49] J. Sethuraman, "A constructive definition of Dirichlet priors," *Stat. Sinica*, vol. 4, no. 2, pp. 639–650, 1994.
[50] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics* (Intelligent Robotics and Autonomous Agents Series). Cambridge, MA, USA: MIT Press, 2005.
[51] L. Hubert and P. Arabie, "Comparing partitions," *J. Classif.*, vol. 2, no. 1, pp. 193–218, 1985.

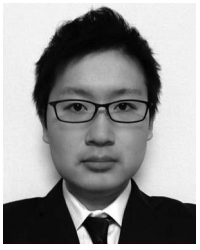
Tadahiro Taniguchi received the M.E. and Ph.D. degrees from Kyoto University, Kyoto, Japan, in 2003 and 2006, respectively.

From 2005 to 2006, he was a Japan Society for the Promotion of Science (JSPS) Research Fellow (DC2) with the Department of Mechanical Engineering and Science, Graduate School of Engineering, Kyoto University, where he was a JSPS Research Fellow (PD), from 2006 to 2007. From 2007 to 2008, he was a JSPS Research Fellow with the Department of Systems Science, Graduate School of Informatics, Kyoto University. From 2008 to 2010, he was an Assistant Professor with the Department of Human and Computer Intelligence, Ritsumeikan University, Kyoto, where he has been an Associate Professor, since 2010. His current research interests include machine learning, emergent systems, and semiotics.



Shogo Nagasaka received the B.E. and M.E. degrees in information science and engineering from Ritsumeikan University, Kyoto, Japan, in 2012 and 2014, respectively.

His current research interests include time series analysis, machine learning, and emergent systems.



Ryo Nakashima received the B.E. degree in information science and engineering from Ritsumeikan University, Kyoto, Japan, in 2014.

His current research interests include machine learning and language acquisition.