Segmentation-Based Formulation of Slot Filling Task for Better Generative Modeling

Kei Wakabayashi, Johane Takeuchi, and Mikio Nakano

Abstract Slot filling is a fundamental task in spoken language understanding that is usually formulated as a sequence labeling problem and solved using discriminative models such as conditional random fields and recurrent neural networks. One of the weak points of this discriminative approach is robustness against incomplete annotations. For obtaining a more robust method, this paper leverages an overlooked property of slot filling tasks: Non-slot parts of utterance follow a specific pattern depending on the user's intent. To this end, we propose a generative model that estimates the underlying pattern of utterances based on a segmentation-based formulation of slot-filling tasks. The proposed method adopts nonparametric Bayesian models that enjoy the flexibility of the phrase distribution modeling brought by the new formulation. The experimental result demonstrates that the proposed method performs better in a situation that the training data with incomplete annotations in comparison to the BiLSTM-CRF and HMM.

1 Introduction

Slot filling is a task that estimates the speaker's intent in the form of slot representation. For example, the utterance "Remind me to call John at 10 to 9 am tomorrow" contains two pieces of information that the system is required to extract for setting a reminder; {**time**: "10 to 9 am tomorrow"} and {**subject**: "call John"}. We use the

K. Wakabayashi (✉)
University of Tsukuba, Tsukuba, Japan
e-mail: kwakaba@slis.tsukuba.ac.jp

J. Takeuchi
Honda Research Institute Japan Co., Ltd., Saitama, Japan
e-mail: johane.takeuchi@jp.honda-ri.com

M. Nakano
Honda Research Institute Japan Co., Ltd. (Currently with C4A Research Institute, Inc.), Saitama, Japan
e-mail: mikio.nakano@c4a.jp

S. Stoyanchev et al. (eds.), *Conversational AI for Natural Human-Centric Interaction*,
Lecture Notes in Electrical Engineering 943,
https://doi.org/10.1007/978-981-19-5538-9_2

O	O	O	B-subject	I-subject	O	B-time	I-time	I-time	I-time	I-time
remind	me	to	call	john	at	10	to	9	am	tomorrow

Fig. 1 Sequence labeling formulation of slot filling

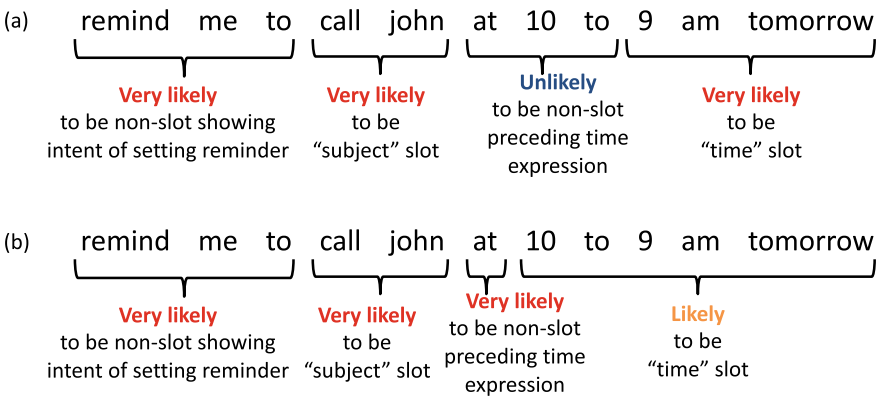


Fig. 2 The segmentation-based approach. Detecting slot parts are formulated as a task that finds the best partition in terms of the joint probability of both the slot and non-slot phrases. The proposed method prefers the segmentation in case (**b**) than in (**a**) because it gives a higher joint likelihood

term *slot* to refer to variables such as **time** and **subject** that are filled by substrings in an utterance.

Slot filling is usually formulated as a sequence labeling task with IOB tagging scheme, which is generally used in phrase extraction tasks such as named entity recognition [10]. Figure 1 shows an example of the sequence labels. On the basis of this formulation, the existing studies have applied discriminative models including conditional random fields (CRFs) [18] and neural networks [13, 17, 29].

One of the weak points in the supervised learning approach is the robustness against incomplete annotations [7]. In practice, obtaining high-quality annotation for phrase extraction is expensive and not scalable [24]. When there is a missing annotation on a substring, the model will be trained to assign O tags for the substrings since we have no way to know if it is missing or truly a non-slot part.

In this paper, we explore an approach that leverages an overlooked characterization of slot filling that is not shared with other phrase extraction tasks: Non-slot substrings also follow a specific pattern. For example, when a user has an intent of setting a reminder, her utterance likely starts with "remind me" to show her intent. On this idea, the slot filling can be formulated as a task that splits an utterance into segments and estimates the role of each segment as Fig. 2 shows.

To this end, we propose a generative model that allows us to induce the roles of non-slot substrings in a similar way to unsupervised grammar induction methods [8, 16]. The proposed model adopts Pitman-Yor Chinese restaurant processes (PYCRPs), which reflects the power-law property inhered in natural language phrases [4],

for defining phrase probabilities. Instead of the word-by-word generative process assumed by conventional models such as hidden Markov models (HMMs), the proposed generative model fully enjoys the flexibility of the phrase distribution modeling brought by the new formulation. In the experiment, we will show that the proposed model is capable of capturing the latent structure of utterances, and therefore, more robust against missing annotations in training data.

The contribution of this paper is summarized as follows:

- We propose a new representation of slot filling task, particularly for a better generative modeling. We also propose a Bayesian model that leverages the flexibility of modeling provided by this formulation by adopting a nonparametric Bayesian model for phrase distribution.
- We empirically show that the generative methods on this formulation are more robust than neural networks when the annotation is highly incomplete. We also show the proposed model has good interpretability thanks to this formulation based on segment-wise pattern recognition.

2 Related Work

Although the sequence labeling formulation is the dominant approach in recent years, there have been different ways to formalize slot filling tasks, including formulation as machine translation task [12], decoding problem with finite-state transducers [3, 9], parsing task with context-free grammar [22]. When we can fix the set of possible values that can be put in each slot, we can formulate a slot filling task as a value-based classification task [6]. Wakabayashi et al. [27] extend the classification-based slot filling method by considering the likelihood of non-slot phrases. To define the likelihood of phrases, they proposed a probabilistic model based on nonparametric Bayesian models that are similar to ours. However, their formulation is classification-based; therefore, it requires candidates of slot filling output fed by the N-best results of another discriminative model such as CRF.

While the discriminative modeling based on neural networks is extensively studied [18, 30], the generative approach still has an advantage when we have incomplete and noisy annotated sentences as training data. When we use crowdsourcing to obtain labeled sentence, we need to handle the training data that includes erroneous annotation [15, 19]. In this situation, HMM-based generative models achieve a better accuracy compared to methods that are based on discriminative models [14]. Simpson et al. [23] further improve the accuracy by integrating prior distribution into the generative models. These methods treat the true labels as latent variables and estimate them in a Bayesian estimation manner in the representation of word-by-word sequence labeling formulation. Applying the proposed segmentation-based formulation to these models for crowdsourced annotations will be a subject of future work.

The proposed method can be viewed as a kind of grammar induction [8, 16] since the method attempts to induce the roles of non-slot parts in an unsupervised manner. From this viewpoint, the proposed method is characterized as follows: (i) Flat (non-hierarchical) latent structure is assumed. (ii) Partial supervision on slot part is available instead of inducing fully model-driven grammatical units. Ponvert et al. [16] proposed an unsupervised shallow parsing (i.e., chunking) method that induces labeled segmentation, which is compatible with the characteristics (i) of ours, In comparison to Ponvert's method, our proposed method uses nonparametric Bayesian language modeling to handle longer phrases than grammatical units with a partially supervised training algorithm. Some language models developed for unsupervised morphological analysis [4, 26] adopt the Pitman-Yor process, which inspires our model. However, these models are designed to find the morphological units by embedding n-gram probability over segments. Our proposed method is designed to capture patterns of non-slot phrases supposing that partial supervision on slot part is available, which is novel even in the context of grammar induction.

3 Segmentation-Based Formulation of Slot Filling Task

The proposed formulation regards a slot filling task as a labeled segmentation of a given sentence. For example, the case (b) in Fig. 2 divides the sentence into four segments, "remind me to", "call john", "at" and "10 to 9 am tomorrow", and attaches labels "non-slot 2", "subject slot", "non-slot 4" and "time slot", respectively. Let $x_{1:T} = x_1, \ldots, x_T$ be a sequence of tokens[1] and $b_{1:K} = b_1, \ldots, b_K$ be indices of the last token of each segment where $b_k < b_{k+1}$. The segmentation in Fig. 2b is represented as $b_1 = 3$, $b_2 = 5$, $b_3 = 6$, and $b_4 = 11$. We denote the sequence of the segment labels by $y_{1:K} = y_1, \ldots y_K$. The subsequence of tokens that represents the k-th segment is denoted by $s_k = x_{b_{k-1}+1:b_k}$. b_K equals T because the last segment ends with the last token. We also define $b_0 = 0$ for convenience.

For slot filling tasks, a set of slots $\mathscr{Z}$ (e.g., {**time**, **subject**}) and a set of training data are given. The instance of the training data is a pair of a sentence and a slot annotation. For example, the annotation for the sentence in Fig. 2 is { **subject**: "call john", **time**: "10 to 9 am tomorrow" }. In the proposed method, we assume that the non-slot parts also have latent segments. For these segments, we assign a non-slot label that reflects a pattern of non-slot parts. We denote a set of the non-slot labels by $\mathscr{U}$ and assume that each non-slot label in $\mathscr{U}$ is associated with its particular phrase distribution. Consequently, the set of labels is defined as $\mathscr{Y} = \mathscr{Z} \cup \mathscr{U}$. We emphasize that the training data only have slot annotations so that the non-slot labels are latent variables. In the proposed method, the non-slot labels are estimated by Gibbs sampling as we present later.

[1] In the experiment, we used word as a token for English and character as a token for Japanese.

3.1 Definition of Generative Models

We consider a generative model in the following form.

$$p(x_{1:T}, y_{1:K}, b_{1:K}) = p(x_{1:T}, b_{1:K}|y_{1:K})p(y_{1:K}) \tag{1}$$

We assume that $p(y_{1:K})$ follows a Markov model with a parameter $\Theta = \theta_1, \ldots, \theta_{|\mathscr{Y}|}$.

$$p(y_{1:K}) = \prod_{k=1}^{K} p_{cat}(y_k|\theta_{y_{k-1}}) \tag{2}$$

$p_{cat}(y_k|\theta_{y_{k-1}})$ is a transition probability that follows a categorical distribution with parameter $\theta_{y_{k-1}}$. θ_y follows a Dirichlet distribution of a hyperparameter γ. The joint distribution $p(x_{1:T}, b_{1:K}|y_{1:K})$ is assumed to be decomposable into segments.

$$p(x_{1:T}, b_{1:K}|y_{1:K}) = \prod_{k=1}^{K} \mathscr{P}_{y_k}(x_{b_{k-1}+1:b_k}) \tag{3}$$

Given the label y_k, a sequence of characters in the k-th segment is generated by a *slot model* $\mathscr{P}_y$, which we present in the following subsections. In the slot model, we represent a phrase as a sequence of characters $s = c_1, \ldots, c_L$ where c_l is a character,[2] instead of the sentence-dependent representation $x_{b_{k-1}+1:b_k}$ [31]. $\mathscr{P}_y$ is a probabilistic model over the infinite set of token sequences $\mathscr{V}$ represented below.

$$\mathscr{V} = \{c_1, \ldots, c_L | c_l \in \mathscr{C}, L \geq 0\}$$

where $\mathscr{C}$ is the set of characters that potentially appear in the input sentences, including the whitespace character. We call an element of $\mathscr{V}$ as a phrase.

3.1.1 N-Gram Slot Model

One of the simplest ways to define a distribution on $\mathscr{V}$ is to adopt an N-gram model. We also explicitly formulate the probability of the phrase length to define a distribution such that the sum of the probability is 1 over $\mathscr{V}$ [31]. The probability of phrase $s = c_1, \ldots, c_L$ is defined as the product of the n-gram probability of the character sequence and the probability that the phrase length is L.

[2] We can formulate the language models for phrases based on token sequence representation, but we prefer the character sequence modeling because the model can get more flexibility. This choice does not affect the overall framework of the proposed method.

$$p_{ngsm}(s = c_1, \ldots, c_L|\psi, \xi) = p_{cat}(L|\psi) \prod_{l=1}^{L} p_{cat}(c_l|\xi_{c_{l-n+1:l-1}})$$

$p_{cat}(L|\psi)$ is defined as a L_{max}-dimensional categorical distribution. $p_{cat}(c_l|\xi_{c_{l-n+1:l-1}})$ is a categorical distribution of a character depending on the n-gram context $c_{l-n+1:l-1}$. Dirichlet distributions with parameter η_1, η_2 are assumed as the priors of ψ and ξ, respectively. We call this model n-gram slot model (NGSM).

3.1.2 Pitman-Yor Process Slot Model

In slot filling tasks, users tend to use specific common phrases. For example, the **time** slot only takes the expressions of time and date, and as a result, a small number of expressions are expected to be used repeatedly. To reflect this observation, we present Pitman-Yor process slot model (PYPSM) for modeling the phrase distribution. PYPSM adopts a Pitman-Yor Chinese restaurant process (PYCRP) that entails power-law distributions over $\mathscr{V}$ [11, 20].

PYPSM is a model that generates phrases $s_1, \ldots, s_N$ where $s_i = c_{i1}, \ldots, c_{iL_i}$ based on the generative process shown in Fig. 3 **(Left)**. The PYPSM has two latent variables; $\boldsymbol{\phi} = \{\phi_1, \ldots, \phi_M\}$ ($\phi_m \in \mathscr{V}$) that is a series of phrases that have been seen before[3] and $a_{1:N} = a_1, \ldots, a_N$ ($1 \leq a_i \leq M$) that associates each observation s_i to one of the elements of $\boldsymbol{\phi}$. Initially, $\boldsymbol{\phi}$ is empty and $M = 0$. For each step to generate a phrase s_i, the process draws a_i depending on $a_{1:i-1}$ from the following distribution.

$$p(a_i = m|a_{1:i-1}) = \begin{cases} \frac{n_m - \beta}{i-1+\alpha} & 1 \leq m \leq M \\ \frac{M\beta+\alpha}{i-1+\alpha} & m = M+1 \end{cases} \tag{4}$$

n_m is the frequency of m in $a_{1:N}$, i.e., $n_m = \sum_{i=1}^{N} \delta_{a_i=m}$ where δ_p is an indicator function that returns 1 if the proposition p is true and 0 otherwise. α and β are hyper-parameters of PYCRP that controls the strength of the power-law property. If $a_i = M + 1$ is drawn from the distribution above, the process generates a new phrase for s_i from the NGSM p_{ngsm}, which is known as base distribution [31]. If $a_i \leq M$, the process generates s_i as the same phrase generated before, ϕ_{a_i}.

The PYPSM assigns a large probability to "memorized" phrases in ϕ but does not fix a set of possible phrases predefined in advance, which matches the tendency of slot filling tasks. When all the latent variables $a_{1:N}$ and $\phi_{1:M}$ generated up to the N-th observation are given, the predictive distribution of the next phrase s_{N+1} can be described as follows by marginalizing out a_{N+1} and ϕ_{M+1}.

$$p_{pypsm}(s_{N+1}|a_{1:N}, \phi_{1:M}, \psi, \xi) = \sum_{m=1}^{M} \frac{n_m - \beta}{N+\alpha} \delta_{\phi_m = s_{N+1}} + \frac{M\beta+\alpha}{N+\alpha} p_{ngsm}(s_{N+1}|\psi, \xi) \tag{5}$$

[3] In contrast to the major usage of CRP that constitutes an infinite mixture model [25], ϕ_{a_i} is not a parameter for another distribution but an observable phrase ($s_i = \phi_{a_i}$).

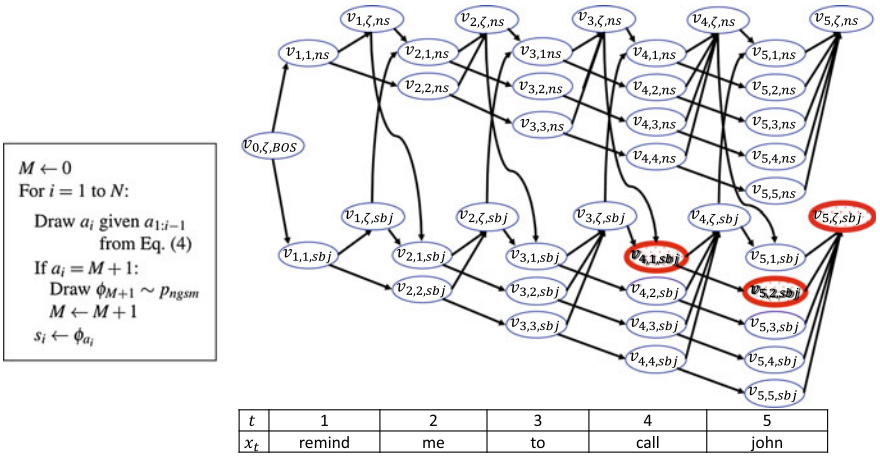


Fig. 3 **(Left)** Generative process of PYPSM. **(Right)** Lattice for forward-backward sampling and Viterbi algorithm on the segmentation-based formulation. The emphasized nodes correspond to a slot annotation in training data (**subject**: "call john") that is available in the training phase

This distribution predicts the next phrase as either (i) a phrase that has been observed at least once in $s_{1:N}$ for probability $\frac{N-M\beta}{N+\alpha}$, or (ii) a phrase that is newly generated by the NGSM for probability $\frac{M\beta+\alpha}{N+\alpha}$. We use p_{pypsm} as the language model $\mathscr{P}_y$.

3.2 Training of PYPSMs by Collapsed Gibbs Sampling

The annotation provided in training data consists of multiple pairs of (slot, value) to be extracted from a given utterance. For example, the annotation for the sentence in Fig. 2 is { **subject**: "call john", **time**: "10 to 9 am tomorrow" }. This supervision partially determines y and b in the proposed model, but the boundaries and the labels for non-slot parts are still hidden. In this paper, we present a collapsed Gibbs sampling method to make an inference on these latent variables.

Let $X = x^{(1)}_{1:T_i}, \ldots, x^{(N)}_{1:T_N}$ be a set of training sentences and $Z = z^{(1)}, \ldots, z^{(N)}$ be the corresponding annotation. The set of latent variables that the collapsed Gibbs sampler draws is $\{y, b, a\}$. When y, b, and $a_{\setminus i}$ are given,[4] the sample of a_i can be obtained from Eq. (4) easily. However, y and b involve the sequence structure, so that we need an efficient sampler. The conditional distribution of y and b that is required to compose the sampler is below.

[4] The index $\setminus i$ indicates a set of the variables except for the ith variable.

$$
\begin{aligned}
&p(b^{(i)}, y^{(i)} | X, Z, b^{(\backslash i)}, y^{(\backslash i)}, a, \phi) \\
&\propto p(x^{(i)}, b^{(i)} | y^{(i)}, x^{(\backslash i)}, y^{(\backslash i)}, b^{(\backslash i)}, a, \phi) p(y^{(i)} | y^{(\backslash i)}) \delta_{S(z^{(i)}, y^{(i)}, b^{(i)})} \\
&\approx \prod_{k=1}^{K} \mathscr{P}_{y_k^{(i)}}(x^{(i)}_{b_{k-1}^{(i)}+1:b_k^{(i)}} | z^{(i)}, x^{(\backslash i)}, y^{(\backslash i)}, b^{(\backslash i)}, a, \phi) \prod_{k=1}^{K} p(y_k^{(i)} | y_{k-1}^{(i)}, y^{(\backslash i)}) \delta_{S(z^{(i)}, y^{(i)}, b^{(i)})} \quad (6)
\end{aligned}
$$

$S(z^{(i)}, y^{(i)}, b^{(i)})$ is a proposition that checks if the labeled segment $(y^{(i)}, b^{(i)})$ is consistent with the supervision $z^{(i)}$. The approximation we applied above ignores the non-Markov dependency between the local random variables in the ith sentence.[5] Each factor in Eq. (6) can be calculated as follows.

$$
\begin{aligned}
\mathscr{P}_{y_k^{(i)}}(x^{(i)}_{b_{k-1}^{(i)}+1:b_k^{(i)}} | x^{(\backslash i)}, y^{(\backslash i)}, b^{(\backslash i)}, a, \phi) &= p_{pypsm}(s_{N+1} = x^{(i)}_{b_{k-1}^{(i)}+1:b_k^{(i)}} | a, \phi, \hat{\psi}_{y_k^{(i)}}, \hat{\xi}_{y_k^{(i)}}) \\
p(y_k^{(i)} | y_{k-1}^{(i)}, y^{(\backslash i)}) &= p_{cat}(y_k^{(i)} | \hat{\theta}_{y_{k-1}^{(i)}})
\end{aligned}
$$

$\hat{\theta}$, $\hat{\psi}_{y_k^{(i)}}$ and $\hat{\xi}_{y_k^{(i)}}$ are respectively the expected value of the variable with the posterior distribution given $x^{(\backslash i)}, y^{(\backslash i)}, b^{(\backslash i)}$.[6] The sample from the distribution of Eq. (6) can be obtained by using a sequence-structured sampling method called forward-backward sampling [21] based on a lattice illustrated in Fig. 3 **(Right)**. Unlike the dynamic programming for HMMs, the proposed model requires restoring the range that corresponds to the phrase for calculating the phrase probability. For this reason, the state in the lattice retains the number of tokens contained in the current segment.

The nodes $v_{t,\tau,y}$ in Fig. 3 **(Right)** such as $v_{1,1,ns}$ and $v_{5,2,sbj}$ indicate a combination of position and label (t, τ, y) that means x_t is the τ-th token of a segment having label y. For example, $v_{1,1,ns}$ indicates x_1 ("remind") is interpreted as the first token for ns (**non-slot**) segment, and $v_{5,2,sbj}$ indicates x_5 ("john") is considered the second token for sbj (**subject slot**) segment. The node $v_{t,\zeta,y}$ $(\tau = \zeta)$ indicates that a segment with the label y is terminated at x_t. Any possible labeled segmentation $(b_{1:K}, y_{1:K})$ has a one-to-one relationship with a path from the node $v_{0,\zeta,BOS}$ to a node $v_{T,\zeta,y}$. Slot annotations in training data can be represented as a set $\mathscr{L}$ of nodes that a path needs to visit. When we denote a slot annotation by a tuple of label and range $(y, i : j)$, $\mathscr{L}$ contains $\{v_{t,\tau,y}\}_{i \le t \le j, 1 \le \tau \le j-i}$ and $\{v_{j,\zeta,y}\}$. For example, the elements in $\mathscr{L}$ corresponding to a slot annotation (**subject**, 4 : 5) are the red nodes in Fig. 3 **(Right)**.

For the forward-backward sampling, we first compute forward probabilities $\alpha(v_{t,\zeta,y}) \equiv \sum_k p(x_{1:t}, b_k = t, y_k = y)$ and $\alpha(v_{t,\tau,y}) \equiv \sum_k p(x_{1:t-\tau}, b_{k-1} = t - \tau, y_k = y)$ by using the following recursive formulas with the base $\alpha(v_{1,\zeta,BOS}) = 1$.

---

[5] As described in [28], the effect of this approximation that ignores the local count is sufficiently small when there are many short sentences. This case applies to the slot filling task.

[6] We can substitute the variables with the expected values because the predictive distribution of a Dirichlet-categorical distribution with $p_{dir}(\theta|\alpha)$ and $p_{cat}(x|\theta)$ equals $p(x_N = k | x_{1:N-1}) = \int p(x_N = k|\theta) p(\theta | x_{1:N-1}) d\theta = \frac{\alpha_k + \sum_{i=1}^{N-1} \delta(x_i = k)}{\sum_k \alpha_k + N - 1} = p_{cat}(x|\theta = E_{p(\theta|x_{1:N-1})}[\theta])$.

$$\alpha(v_{t,\tau,y}) = \left(\delta_{\tau=1} \sum_{y' \in \mathscr{Y}} \alpha(v_{t-1,\zeta,y'}) p(y|\theta_{y'}) + \delta_{\tau>1} \alpha(v_{t-1,\tau-1,y}) \right) \delta_{\neg excluded(t,\tau,y)} \tag{7}$$

$$\alpha(v_{t,\zeta,y}) = \sum_{\tau=1}^{t} \alpha(v_{t,\tau,y}) \mathscr{P}_y(x_{t-\tau+1:t}) \tag{8}$$

To exclude a path that does not follow the training annotations, we define a predicate $excluded(t, \tau, y) \Leftrightarrow \exists \tau', y'[v_{t,\tau',y'} \in \mathscr{L} \wedge (\tau', y') \neq (\tau, y)]$ and use it in (7). The backward sampling starts with drawing a sample of the label of the last segment denoted by $\tilde{y}_{\mathscr{K}}$. Then, the segment lengths $\tilde{\tau}_\kappa$ and the segment labels $\tilde{y}_{\kappa-1}$ are sampled recursively in a backward order for $\kappa = \mathscr{K}, \mathscr{K}-1, \mathscr{K}-2, \ldots$. Let $\tilde{b}_{\mathscr{K}} = T$ and $\tilde{b}_{\kappa-1} = \tilde{b}_\kappa - \tilde{\tau}_\kappa$. The sampling repeats until $\tilde{b}_{\kappa-1} = 0$ is obtained. The conditional distributions that the sampler draws from are represented by using the forward probabilities as a straightforward extension of the backward sampling for HMMs [2].

After the sampling iterations, we obtain a sample of the latent variables for all sentences. While Monte Carlo estimation generally takes an average of multiple samples, we simply use a single sample of segmentation to estimate the posterior of the model parameters [1]. The computational complexity of the algorithm that processes one sentence is $\mathscr{O}(TL_{max}^2)$ because the dominant factor Eq. (8) requires computations for $t = 1$ to T and $\tau = 1$ to $min(L_{max}, t)$, and each computation of $\mathscr{P}_y(x_{t-\tau+1:t})$ involves the calculation of N-gram probability that requires τ iteration.

3.3 Finding the Most Likely Labeled Segmentation

To complete the slot filling task, we need to find the most likely labeled segmentation regarding the trained PYPSMs. Such segmentation can be obtained by an algorithm to find the shortest path on the lattice. We define a cost function $f : E \to \mathbb{R}$ to make the sum of the costs in a path to be equivalent to the negative log likelihood of the corresponding labeled segmentation. For an initialization edge in E_I, the cost is the negative log probability of the corresponding label transition, $f(v_{t,\zeta,y} \to v_{t+1,1,y'}) = -\log p(y'|\theta_y)$. No cost is imposed to cross a continuation edge in E_C, i.e., $f(v_{t,\tau,y} \to v_{t+1,\tau+1,y}) = 0$. For a termination edge in E_T, the cost is the negative log probability of the phrase in the current segment, which is calculated by using a PYPSM, $f(v_{t,\tau,y} \to v_{t,\zeta,y}) = -\log \mathscr{P}_y(x_{(t-\tau+1):t})$. Under this definition of the cost function, the shortest path that minimizes the sum of the costs is guaranteed to correspond to the labeled segmentation that maximizes Eq. (1).

4 Experiment

4.1 Datasets

We use two datasets called DSTC corpus and weather corpus to evaluate the proposed methods. The DSTC corpus is a collection of English utterances for restaurant search provided at the dialog state tracking challenge 3 [5]. We extracted the first utterance in each dialog session, which typically describes the preference of restaurant. The sentences that contain no slot information are excluded. The corpus consists of 1,441 sentences with five possible slot types, **area**, **food**, **price range**, **type**, and **children allowed**. We manually identified the substring that expresses the slot value if the slot value does not match with any substring in the sentence.

The weather corpus is an in-house dataset of Japanese utterances that ask about the weather (for example, "Tell me the amount of precipitation in Tokyo tomorrow." in Japanese). The utterances are collected from users who accessed to a prototype dialog system that can reply about weather information. We manually annotated slot information to all the utterance for three slot types, **when**, **where**, and **what**. The weather corpus consists of 1,442 sentences.

For each dataset, we splitted the set of sentences into evaluation (90%) and validation (10%) subsets for hyperparameter search. From the evaluation set, we organized train/test subsets in a 10-fold cross validation manner.

4.2 Settings

The proposed method has two variants: **PYPSM** presented in Section 3, and **NGSM** that uses n-gram slot models instead of PYPSMs. By using the validation set, hyperparameter search is conducted for deciding the PYCRP parameters α, β, the number of non-slot labels $|\mathscr{U}|$, and the context length in N-gram N, for each dataset. The best configuration was $\alpha = 1.0$, $\beta = 0.1$, $|\mathscr{U}| = 3$, and $N = 4$ for the DSTC dataset and $\alpha = 1.0$, $\beta = 0.1$, $|\mathscr{U}| = 5$, and $N = 3$ for weather dataset. The Gibbs sampling and Viterbi decoding are applied to both method for training and inference. We compared the accuracy of the proposed method with the following existing methods:

- **BiLSTM-CRF** Neural network proposed in [10] with word and character embeddings and bidirectional LSTM. We set the hidden dimension of LSTM to 128 and the number of LSTM layers to 2 based on the hyperparameter search.
- **HMM** Hidden Markov model, which is a generative model based on sequence labeling formulation with IOB2 tagging scheme. The HMM is trained in a fully supervised manner by associating the sequence labels to the hidden states.

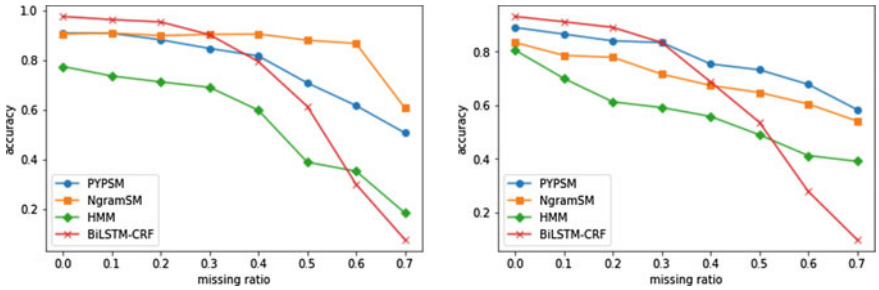


Fig. 4 Accuracy of slot estimation in DSTC corpus **(Left)** and weather corpus **(Right)**

We implemented the proposed method in Java and the BiLSTM-CRF in Python with Anago library.[7] We set $L_{max} = 32$. We treat a word as a unit of token for the DSTC corpus and a character as a token for the weather corpus. This is because Japanese sentences do not contain whitespace letters that indicate word boundaries. The n-gram models in the proposed method (NGSM and PYPSM) are based on characters including the whitespace letters for both datasets.

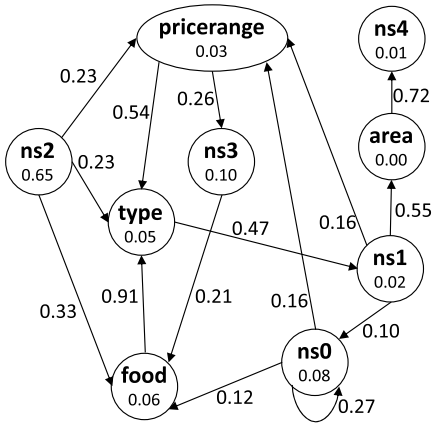
We calculated the slot estimation accuracy with 10-fold cross validation. The accuracy is defined as the ratio of the number of utterances that have the exactly correct slot estimation against the number of all test utterances. For the experiment on incomplete annotation, we simulate the missing annotation by dropping the annotated slot information randomly in various missing ratio from 0.0 (complete annotation) to 0.7 (highly incomplete annotation).

4.3 Results

Figure 4 shows the estimation accuracy. The horizontal axis indicates the missing ratio of annotation and the vertical axis indicates the averaged accuracy of the 10-fold cross validation. The accuracy of BiLSTM-CRF is the highest among all the methods when the missing rate is low. However, the performance of BiLSTM-CRF apparently degrades as the missing rate is higher. The generative models including the proposed method and HMM seem to be more tolerant against the missing annotation. Compared with the HMM, the proposed method significantly improves the estimation performance. This implies that the proposed segmentation-based formulation is more suitable than sequence labeling formulation for slot filling tasks.

Another advantage of the proposed method is the interpretability of model parameters. Figure 5 is a diagram representing the PYPSM model parameters we obtained on DSTC dataset with missing ratio 0.0 (annotated completely). The left side of the figure represents the transition parameters among labels. The values in the nodes

[7] https://github.com/Hironsan/anago.



ns0	a (28), an(12), I would like (10), to find a (10), the (9), uh (8), a tv (7), and (6), trying (5)
ns1	in the (101), in (59), with (11), I (11), with a (10), and (8), that (5), can I (5), should (4), hello (4)
ns2	I'm looking for a (161), I'm looking for an (29), I want to find a (22), a (21), looking for a (12)
ns3	price range (57), I am looking for a (20), yes I'm looking for a (13), unintelligible (8), about the (7)
ns4	area (112), and it should be in the (26), and it should have (10), be (5), tv(5), please (4), it (4), a television (4)

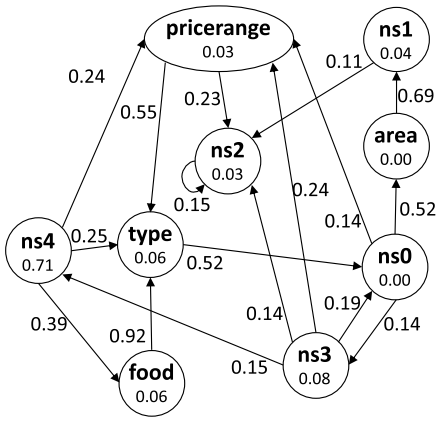
Fig. 5 Parameters of PYPSM obtained by training on DSTC dataset with missing ratio 0.0. (**Left**) Transition parameters among slot labels and non-slot labels (ns0, ..., ns4). The numbers by the edges are the transition probability $p(y_k|y_{k-1})$. The numbers in the circles are the initial label probability $p(y_1)$. (**Right**) Substrings recognized as a non-slot part in the test dataset. The numbers in parentheses are the frequency

indicate the initial probability for the corresponding label. The right side of the figure shows the list of top phrases assigned to the non-slot labels in the test data (the parenthesized numbers are the frequency). We can reconstruct typical utterances by examining this diagram. For example, one of the likely paths is **ns2** (e.g., "I'm looking for a"), **food** (e.g., "Italian"), **type** (e.g., "restaurant"), **ns1** (e.g., "in the"), **area** (e.g., "new chesterton"), **ns4** (e.g., area).

The robustness of the proposed method against a high missing ratio can be observed also in the invariance of the extracted pattern. A diagram for the model trained on DSTC with missing ratio 0.7 is shown in Fig. 6. The structure of the transition pattern resembles the diagram in Fig. 5, and the path we observed in Fig. 5 can be found in this diagram too. This indicates that the proposed model is capable of capturing the structural pattern of utterances from the partial annotations.

Table 1 shows examples of the prediction by models trained on the DSTC dataset with missing ratio 0.7. For the first example, the BiLSTM-CRF failed to detect the area slot even though it is a typical way to mention area information. We believe the area slot could be detected if the BiLSTM-CRF is trained on the perfectly annotated dataset. Contrarily, the PYPSM could detect the slot information probably because of the robust modeling of the non-slot segments.

The second example is the case that demonstrates the effectiveness of explicit probabilistic modeling on the phrases. The PYPSM is less likely to misrecognize phrases that are observed during the training because of the "memorizing" property [4]. On the other hand, on imperfect training data, discriminative models tend to be uncertain about the label for the phrase "chinese" should be **food** or not.



ns0	in the (127), in (60), and (35), with (13), and it should have (10), I (8), please (4), should have (4)
ns1	area (110), about the (10), ah (9), can I (6), you (3), me (3), allow children (3), I would like (3)
ns2	price range (60), a tv (9), unintelligible (8), internet connection and (7), the (7), be (5)
ns3	it should be (30), a (19), an(12), hi (10), with a (10), uh (8), and I (8), to (6), and (6), I need to find a (4)
ns4	I'm looking for a (163), I'm looking for an (29), a (27), I want to find a (22), I am looking for a (20)

Fig. 6 Parameters of PYPSM obtained by training on DSTC dataset with missing ratio 0.7

Table 1 Examples of the prediction by models trained on the DSTC dataset with missing ratio 0.7. Asterisk (*) indicates misrecognition

Utterance	BiLSTM-CRF	PYPSM
Expensive restaurant in the trumington area	**pricerange**: expensive,	**pricerange**: expensive,
	type: restaurant,	**type**: restaurant,
	area: (None) (*)	**area**: trumington
I'm looking for a chinese and it should be in the cherry hinton area	**type**: chinese (*),	**food**: chinese,
	area: cherry hinton	**area**: cherry hinton
I want to find a chinese take away	**food**: chinese take away	**food**: chinese (*),
		food: take away (*)

The third example shows the downside of the memorizing property of the proposed method. While "chinese take away" is another genre of food than "chinese", the PYPSM discretely assigns high probability to "chinese" and recognizes it as an independent slot value. For this example, "take away" is also recognized as another **food** slot value. This kind of generalization error might be mitigated by introducing a constraint that prevents such a split recognition of the same slot values.

5 Conclusion

In this study, we proposed a new formulation of slot filling tasks that is based on an inference of the most likely labeled segmentation. The proposed method considers the probabilities of both slot segments and non-slot segments by a Bayesian model that produces the power-law distribution of phrases. The experimental results show that the proposed method is more accurate than neural network methods when the missing ratio of annotation is high. We empirically showed the proposed model has good interpretability thanks to the formulation based on segment-wise pattern recognition. Future work includes the exploration of more accurate models that are based on the segmentation-based formulation.

Acknowledgements This work was partially supported by JSPS KAKENHI Grant Number 19K20333.

References

1. Bishop C (2006) Pattern recognition and machine learning. Springer
2. Chib S (1996) Calculating posterior distributions and modal estimates in markov mixture models. J Econom 75:79–97
3. Fukubayashi Y, Komatani K, Nakano M, Funakoshi K, Tsujino H, Ogata T, Okuno HG (2008) Rapid prototyping of robust language understanding modules for spoken dialogue systems. In: Proceedings of IJCNLP, pp 210–216
4. Goldwater S, Griffiths TL, Johnson M (2011) Producing power-law distributions and damping word frequencies with two-stage language models. J Mach Learn Res 12:2335–2382
5. Henderson M (2015) Machine learning for dialog state tracking: a review. In: Proceedings of international workshop on machine learning in spoken language processing
6. Henderson MS (2015) Discriminative methods for statistical spoken dialogue systems. PhD thesis, University of Cambridge
7. Jie Z, Xie P, Lu W, Ding R, Li L (2019) Better modeling of incomplete annotations for named entity recognition. In: Proceedings of NAACL: HLT, pp 729–734
8. Jin L, Schwartz L, Doshi-Velez F, Miller T, Schuler W (2021) Depth-bounded statistical PCFG induction as a model of human grammar acquisition. Comput Linguist Assoc Comput Linguist 47(1):181–216
9. Komatani K, Katsumaru M, Nakano M, Funakoshi K, Ogata T, Okuno HG (2010) Automatic allocation of training data for rapid prototyping. In: Proceedings of COLING
10. Lample G, Ballesteros M, Subramanian S, Kawakami K, Dyer C (2016) Neural architectures for named entity recognition. arXiv:1603.01360 [cs.CL]
11. Lim KW, Buntine W, Chen C, Du L (2016) Nonparametric Bayesian topic modelling with the hierarchical Pitman-Yor processes. Int J Approx Reason 78(C):172–191
12. Macherey K, Och FJ, Ney H (2001) Natural language understanding using statistical machine translation. In: Proceedings of EUROSPEECH, pp 2205–2208
13. Mesnil G, Dauphin Y, Yao K, Bengio Y, Deng L, Hakkani-Tur D, He X, Heck L, Tur G, Yu D, Zweig G (2015) Using recurrent neural networks for slot filling in spoken language understanding. IEEE/ACM Trans Audio, Speech, Lang Process 23(3):530–539
14. Nguyen AT, Wallace BC, Li JJ, Nenkova A, Lease M (2017) Aggregating and predicting sequence labels from crowd annotations. In: Proceedings ACL, pp 299–309
15. Niu J, Penn G (2019) Rationally reappraising ATIS-based dialogue systems. In: Proceedings ACL, pp 5503–5507

16. Ponvert E, Baldridge J, Erk K (2011) Simple unsupervised grammar induction from raw text with cascaded finite state models. In: Proceedings ACL, pp 1077–1086
17. Qin L, Liu T, Che W, Kang B, Zhao S, Liu T (2021) A co-interactive transformer for joint slot filling and intent detection. In: Proceedings ICASSP, pp 8193–8197
18. Raymond C, Riccardi G (2007) Generative and discriminative algorithms for spoken language understanding. In: Proceedings of Interspeech
19. Rodrigues F, Pereira F, Ribeiro B (2014) Sequence labeling with multiple annotators. Mach Learn 95(2):165–181
20. Sato I, Nakagawa H (2010) Topic models with power-law using Pitman-Yor process. In: Proceedings KDD
21. Scott SL (2002) Bayesian methods for hidden markov models: recursive computing in the 21st century. J Am Stat Assoc 97:337–351
22. Seneff S (1992) TINA: a natural language system for spoken language applications. Comput Linguist 18(1):61–86
23. Simpson ED, Gurevych I (2019) A bayesian approach for sequence tagging with crowds. In: Proceedings EMNLP, pp 1093–1104
24. Snow R, O'Connor B, Jurafsky D, Ng AY (2008) Cheap and fast—but is it good? evaluating non-expert annotations for natural language tasks. In: Proceedings EMNLP, pp 254–263
25. Teh YW, Jordan MI, Beal MJ, Blei DM (2005) Hierarchical dirichlet processes. J Am Stat Assoc 101:1566–1581
26. Uchiumi K, Tsukahara H, Mochihashi D (2015) Inducing word and part-of-speech with pitman-yor hidden semi-markov models. In: Proceedings ACL-IJCNLP
27. Wakabayashi K, Takeuchi J, Funakoshi K, Nakano M (2016) Nonparametric Bayesian models for spoken language understanding. In: Proceedings EMNLP
28. Wang P, Blunsom P (2013) Collapsed variational Bayesian inference for hidden Markov models. In: Proceedings AISTATS, pp 599–607
29. Xu P, Sarikaya R (2013) Convolutional neural network based triangular CRF for joint intent detection and slot filling. In: Proceedings of IEEE workshop on automatic speech recognition and understanding
30. Yadav V, Bethard S (2018) A survey on recent advances in named entity recognition from deep learning models. In: Proceedings COLING
31. Zhai K, Boyd-graber J (2013) Online latent dirichlet allocation with infinite vocabulary. In: Proceedings of ICML