Dependency Parsing and Its Application using Hierarchical Structure in Japanese Language

Kazuki Ono

Division of Infrastructures Management
KLab Inc.
6-10-1 Roppingi, Minato, Tokyo 106–6122, Japan
Email: ono-ka@klab.com

Kenji Hatano

Faculty of Culture and Information Science
Doshisha University
1-3 Tatara-Miyakodani, Kyotanabe, Kyoto 610–0394, Japan
Email: khatano@mail.doshisha.ac.jp

***Abstract*—Conventional Japanese dependency parsing methods are primarily based on the bi-nominal model between phrases, which has a limitation related to the order of phrases. Accordingly, the length of a phrase, which depends on the language dependency, is limited. In this paper, we propose a novel dependency parsing method for Japanese based on an extended hierarchical language model simulated by the hierarchical Pitman-Yor process in order to overcome the abovementioned limitation. We also evaluate the accuracy of the proposed dependency parsing method as well as its practicality to detect the topics of each document. Experimental results show that the proposed method can parse dependencies in long, complex sentences and can allocate topics to each document relatively well compared with the conventional method. Consequently, it can be said that the proposed method is feasible in the research fields of both Japanese dependency parsing and topic modeling.**

Keywords-Dependency Parsing; Syntax Analysis; Syntax Tree Modeling; Topic Modeling.

I. INTRODUCTION

Asian languages are based on case grammar, and their sentences have the following word order: subject, object, and verb (SOV), leading to these languages being termed SOV languages. Japanese also has the same characteristics; however, its phrase order is relatively unfettered.

In Japanese, postpositions represent the syntactic function of each phrase, while in English, the syntactic function of each phrase is represented by the phrase order. Moreover, phrases that contain one or more postpositions following a noun, which play a similar role to the declension of nouns, have been devised and used for syntactically analyzing Japanese sentences. Hence, it may be said that a syntactic analysis is equivalent to parsing dependencies between phrases. This is because the semantic rules of Asian languages are explained as relationships of phrases. As a result, almost all dependency parsing methods in such languages extract these semantic rules and utilize them for analyzing relationships of phrases in Asian languages.

For instance, we propose a dependency parsing method for Japanese based on tree-substitution grammar (TSG) [1], which has originally attracted attention as a language model in the syntactic analysis used for English [2]. This is because TSG constructs a hierarchical structure defined as a set of rewrite rules of context-free grammar (CFG) and helps to establish the highest accuracy of English parsing by learning it [3], [4]. Alternatively, other dependency parsing methods utilize a bi-nominal dependency model to decide whether a dependency relation exists between pairs of phrases or not [5]–[7]. The reason why they did not employ a rule-based approach like us is that the computational cost of our approach is very expensive.

It is true that parsing dependencies based on TSG is computationally expensive; however, syntax trees generated by TSG describe the order of phrases, whereas those generated using the bi-nominal dependency model only represent relationships between phrases. In short, TSG-based syntax trees retain not only the relationships between phrases but also the orders of phrases; therefore, they facilitate a precise extraction of the dependencies between phrases. As a result, dependency parsing of Asian languages based on the bi-nominal dependency model is less successful. We believe that the accuracy of the dependency parsing of Asian languages can be improved by applying TSG to them.

In this paper, consequently, we propose a TSG-based dependency parsing method for Japanese. The proposed method is characterized by the handling of an exchangeable sequence of phrases in case grammar to utilize the extracted hierarchical structure using TSG. That is to say, we can generate a dependency parsing model by calculating the probabilities of integrating phrase dependencies from supervised training data with the abovementioned hierarchical structure. Thus, it can be said that the proposed mothod is a novel dependency parsing method, with hierarchical structure relations, for the Japanese because it can handle the relationships of each of the phrase dependencies.

Moreover, we discuss a possible beneficial effect of the proposed dependency parsing method: to estimate the topic of a document on the Web. Usually, the topic of a document is estimated using an n-gram model. In this approach, the topic model is constructed using the word distribution ignoring the order of words, and hence, it may not be accurate. Therefore,

our topic model is constructed using a topic distribution based on the modification relations in documents.

The remainder of this paper is organized as follows: In Section II, we explain the basic issues related to our study; in particular, we provide further details about the statistical language model and its application, latent Dirichlet allocation. In Section III, we discuss a conventional dependency parsing method for the Japanese language and its application to topic estimation. In Section IV, we describe the proposed dependency parsing method and its evaluation. In Section V, we describe the topic model built using the proposed dependency parsing method and its evaluation. Finally, in Section VI, we conclude our paper and present directions for future work.

II. Basic Issues

In this section, we will discuss the basic issues related to our study.

A. Statistical Language Model

The fundamental research purpose of linguistics is to find rules and characteristics through observations of an existing language [8]. For example, word segmentation of Asian languages is a crucial research topic, because these languages do not have any explicit boundary between words. However, most people can recognize the boundaries by using heuristics and can detect each word in a sentence. These heuristics are generalized into the rules and characteristics of the language, and are used as grammar rules.

On the other hand, there are concerns that such rules and characteristics should be statistically defined from an enormous document set. This is because the rules and characteristics found by linguists might not be objective outputs but subjective ones. Thus, some computational linguists have tried to statistically define certain rules and characteristics by using an enormous document set. In the case of the word segmentation described above, the probability of word segmentation can be calculated using the Dirichlet and the Pitman-Yor processes [9], [10]. Compared with the Dirichlet process, the Pitman-Yor process is more efficient in and effective for modeling the existing languages (i.e., n-gram language model); therefore, we propose a dependency parsing method based on the Pitman-Yor process.

The Pitman-Yor process is a non-parametric Bayesian model [11]. It is a stochastic process that generates an infinite discrete probability distribution G, which is denoted by $\mathrm{PY}(d, \theta, G_0)$. $\mathrm{PY}(d, \theta, G_0)$ has three parameters: d denotes a discount parameter with $0 \leq d \leq 1$, θ represents a strength parameter that meets the condition $\theta \geq -d$, and G_0 indicates a base distribution over a probability space. In the research field of natural language processing, the probability space is usually generated from the probabilities of phrase occurrences.

When d is zero, $\mathrm{PY}(d, \theta, G_0)$ becomes the Dirichlet process denoted by $\mathrm{DP}(\theta, G_0)$. In short, because $\mathrm{DP}(\theta, G_0)$ can generate an infinite-dimensional Dirichlet distribution, $\mathrm{PY}(d, \theta, G_0)$ can also support it. As outlined above, d denotes a concentration parameter for G_0; therefore, we can define the following equation:

$$G \sim \mathrm{PY}(d, \theta, G_0) \tag{1}$$

That is, $\mathrm{PY}(d, \theta, G_0)$ is an extended version of $\mathrm{DP}(\theta, G_0)$. Let W be a fixed and finite vocabulary of V phrases. The Pitman-Yor process can generate a vector of phrase probability $G(w)$ for each phrase $w \in W$. $\mathrm{DP}(\theta, G_0)$ is approximated by the Dirichlet distribution $D(\theta G_0(w_1), \ldots, \theta G_0(w_i), \ldots, \theta G_0(w_r))$ that expresses any division of the disintegration space, where the size of the phenomenon space is r for observing any phrase w_i, as follows:

$$\mathrm{DP}(\theta, G_0) \sim D(\theta G_0(w_1), \ldots, \theta G_0(w_i), \ldots, \theta G_0(w_r)) \tag{2}$$

Here, $G_0(w_i)$ denotes an integral of the base distribution G_0; therefore, it is equal to the sum of the probabilities of phrase occurrences in the disintegration space. In short, we can say that the Pitman-Yor process is a recursive stochastic process whose input is a set of phrases w_i and its base distribution is G_0.

In general, these procedures for generating an n-gram distribution drawn from G are often referred to as the Chinese restaurant process [12]. In this process, we fancifully imagine a restaurant with an infinite number of tables whose capacities are infinite. The existing distribution of customers (phrases) who come to the restaurant and the tables (discount strength) with a one-by-one dish (phrase vocabulary) is based on the n-gram model, denoted by G, and the hypothesis of the tables elicits the base distribution G_0. In the Pitman-Yor process, the customers are compared to phrases such that a customer continues to sit at the same table if the customer coming to the restaurant is the same. However, if it is another customer that had not previously entered the restaurant, the customer sits down at a new table. Given this scenario, we can express the Pitman-Yor process as follows:

$$\mathrm{PY}(d, \theta, G_0) = \frac{c_k - d}{\theta + c_\cdot} + \frac{\theta + dt}{\theta + c_\cdot} p(w_k) \tag{3}$$

where t denotes the number of customers under the base distribution G_0, c_k indicates the number of species, $c_\cdot$ represents the total number of customers, and $p(w_k)$ refers to the probability of a customer visiting the restaurant. Here, the parameters d and θ are non-parametric. Therefore, we have to generate the distribution approximated by the base distribution G_0 with these parameters using a Gibbs sampling algorithm, such as the Markov Chain Monte Carlo method [13].

Note that $p(w_k)$ in Equation (3) denotes the probability of word w_k in the document set in the case of computational linguistics. However, phrases are given various types of contexts so that $p(w_k)$ is not a uniform state in the document set. In short, we cannot directly apply the Pitman-Yor process to an n-gram model.

In fact, based on an n-gram language distribution over which the current phrase is given, various contexts can be formulated using an hierarchical Pitman-Yor process [14]. The hierarchical Pitman-Yor process is also a stochastic process that is based on a hierarchical extension of the Pitman-Yor process. Consequently, the n-gram distribution G_u can be generated by the Pitman-Yor process using the base distribution $G_{\pi(u)}$, which is generated in the given context $\mathbf{u}$ as follows:

$$G_{\mathbf{u}} \sim \mathrm{HPY}(d_{\pi|\mathbf{u}|}, \theta_{\pi|\mathbf{u}|}, G_{\pi(\mathbf{u})}) \tag{4}$$

where the strength and discount parameters are calculated on the basis of the length of context $\mathbf{u}$, $\pi(\mathbf{u})$ denotes the suffix of $\mathbf{u}$ consisting of all but the earliest phrase, and $G_{\pi(\mathbf{u})}$ represents a vector of probabilities of its context. When we recursively use a stochastic process such as $G_{\pi(\pi(\mathbf{u}))}$ over $G_{\pi(\mathbf{u})}$ using Equation (4), we can define the stochastic process that generates the n-gram distribution. This process is repeated until we get G_ϕ as the base distribution, the vector of probabilities over the current phrase given the empty context ϕ.

As described above, the structure of the stochastic process generated by the hierarchical Pitman-Yor process is expressed as a suffix tree with depth n. Each node in the suffix tree corresponds to a context consisting of up to $(n-1)$ phrases, and each child corresponds to the addition of a different word to the beginning of the context. The hierarchical Pitman-Yor process can generate an n-gram language model precisely as demonstrated in the language model based on Kneser-Ney smoothing [15]. Given the above explanation, we can express the hierarchical Pitman-Yor process as follows:

$$\begin{aligned}&\mathrm{HPY}(d_{\pi|\mathbf{u}|}, \theta_{\pi|\mathbf{u}|}, G_{\pi(\mathbf{u})}) \\ &\quad = \frac{c(w_k|\mathbf{u}) - d_{\pi|\mathbf{u}|}}{\theta_{\pi|\mathbf{u}|} + c_{\pi|\mathbf{u}|}} + \frac{\theta_{\pi|\mathbf{u}|} + d_{\pi|\mathbf{u}|} t_{\pi|\mathbf{u}|}}{\theta_{\pi|\mathbf{u}|} + c_{\pi|\mathbf{u}|}} p(w_k|\mathbf{u'}) \quad (5)\end{aligned}$$

where $t_{\pi|\mathbf{u}|}$ denotes the number of customers under the base distribution $G_{\pi(\mathbf{u})}$; $c(w_k|\mathbf{u})$, the number word w_k when the state has context $\mathbf{u}$; $c_{\pi|\mathbf{u}|}$, the total number of words given context $\mathbf{u}$; and $p(w_k|\mathbf{u'})$, the probability of generating a word when the state has context $\mathbf{u'}$.

B. Latent Dirichlet Allocation

Latent Dirichlet process (LDA) is one of the language models used in a one-document case, while the n-gram model is used in a one-sentence case. In LDA, we assume that the document consists of several topics; this is similar to a probabilistic latent semantic analysis (pLSA) [16]. Therefore, LDA is considered a pLSA model with a topic distribution by introducing variational Bayesian methods. Since its development, this model has attracted attention in the research field of topic models.

Both LDA and pLSA models create a word distribution by ignoring the order of words in a sentence and utilizing the idea of bag-of-words. In each topic model, the probabilistic models are built using a document set D containing a set of words W. However, their representation schemes are different.

In the earliest years, pLSA was called aspect model [17]. The aspect model is a latent variable model for co-occurrence data that associates an unobserved class variable $z \in Z = \{z_1, \ldots, z_K\}$ that accompanies a bag-of-words $w \in W = \{w_1, \ldots, w_M\}$ in a document $d \in D = \{d_1, \ldots, d_N\}$, where M denotes the number of bag-of-words, z represents a topic class, and K refers to the number of topics, because the topic model is based on the bag-of-words in pLSA. In other words, the generation process of the model is as follows:

- The document is selected on the basis of $P(d)$.
- The unobserved class variable z is also selected on the basis of $P(z|d)$.
- The word w is generated if we can calculate $P(w|z)$.

Consequently, we obtain two pairs of data (d, w), and the corresponding joint probability model can be expressed as follows:

$$\begin{aligned}P(d, w) &= P(d)P(w|d) \\ P(w|d) &= \sum_{z \in Z} P(w|z)P(z|d) \quad (6)\end{aligned}$$

This process is shown in Figure 1 using a graphical model, which is used for a graph-based representation. This is commonly utilized in fields such as probability theory and statistics.

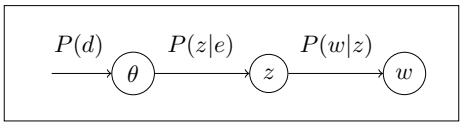


Figure 1. Graphical model representation of the pLSA model using asymmetric parameterization.

Here, we attempt to reverse the conditional probability $P(z|d)$ using Bayes' theorem and can define $P(z|d)$ as follows:

$$P(z|d) = \frac{P(z)P(d|z)}{P(d)} \quad (7)$$

Using Equations (6) and (7), we can obtain the model expressed by the following formula:

$$P(d, w) = \sum_{z \in Z} P(z)P(w|z)P(d|z) \quad (8)$$

Figure 2 shows a graphical model of Equation (8). This model shows that both d and w are independently conditioned and calculated using z. In Equation (8), $P(z)$, $P(w|z)$, and $P(z|d)$

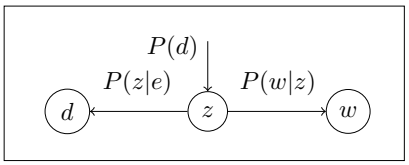


Figure 2. Graphical model representation of the pLSA model using symmetric parameterization.

can be calculated if the following log-likelihood function is maximized:

$$L = \sum_{d \in D} \sum_{w \in W} n(d, w) \log P(d, w) \quad (9)$$

where $n(d, w)$ denotes the number of bag-of-words w in the document d. In order to obtain an appropriate result of Equation (8), we have to maximize L in Equation (9). To do so, we use the expectation maximization (EM) algorithm [18]. Thus, we can build a topic model by using topic z corresponding to (d, w).

LDA also uses Z, which indicates a topic for each word in a document, as a latent variable. In LDA, z_{ij} denotes a latent topic of word w_{ij} in a document i. We can draw it as the following graphical model as Figure 3:

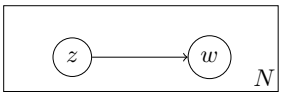


Figure 3. Initial LDA graphical model.

In addition, if the probability of generating word w_l in topic k is assumed to be ϕ_{kl}, the probability of generating w_{ij} in topic w_{ij} can be calculated; the corresponding graphical model is shown in Figure 4.

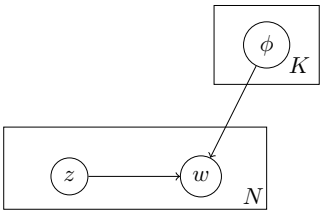


Figure 4. Adding parameter ϕ that denotes the generation probability of a word for a given topic.

Here, Z can be calculated using θ, because z_{ij} is approximated by θ_i, which is a topic distribution of document i. Therefore, we add this process to the graphical model described immediately before (see Figure 5).

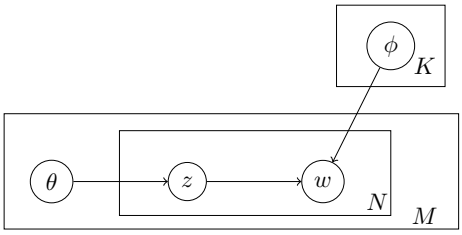


Figure 5. Adding parameter θ that denotes the topic distribution per documents.

Further, θ can also be approximated by using a Dirichlet distribution $Dir(\alpha)$ for a given parameter α. From the beginning, θ denotes the topic distribution of each document, expressing the word distribution composed of the bag-of-words; in short, θ satisfies $\{\theta_1, \theta_2, \ldots, \theta_K\} \in \theta$, where K denotes the number of topics. This means that α is composed of a set of α_k corresponding to θ_k, which is an element of θ, because θ_k denotes the topic k assigned to the documents. As is the case with θ, ϕ can be approximated by using the Dirichlet distribution $Dir(\beta)$, which is assigned a parameter β. That is, ϕ satisfies $\{\phi_1, \phi_2, \ldots, \phi_L\} \in \phi$, where L denotes the number of topics, and β consists of a set of β_l corresponding to ϕl, which is an element of ϕ because of the reason described above.

In light of the above explanation, LDA can be modeled using θ, which is a topic generation probability of each document, ϕ, which is a word generation probability of each topic, and Z, which is the topic of each word. The graphical model of LDA is shown in Figure 6.

In addition, α and β denote the parameters of the Dirichlet distributions θ and ϕ, and these can be estimated using the Markov Chain Monte Carlo method.

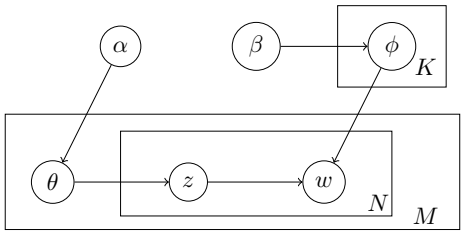


Figure 6. Graphical model of LDA.

III. Related Work

In this section, we first discuss a conventional dependency parsing method for Japanese and its limitations. The problem of the conventional methods can be solved by introducing the concept of TSG, which obtains highly precise English language parsing with a hierarchical structure of phrases. Therefore, we will now discuss a TSG-based dependency parsing method for the Japanese language.

We will also discuss an application of the proposed dependency parsing method and explain the importance of dependency parsing for natural language processing in Japanese.

A. Dependency Parsing for Japanese

As described in Section II-A, a Japanese sentence does not have an explicit boundary between words except punctuation marks. Therefore, in the first stage of computational linguistics for Japanese, single characters in a sentence and the relationships between two characters were focused upon. Currently, word segmentation in Japanese sentences is performed using an unsupervised approach [10].

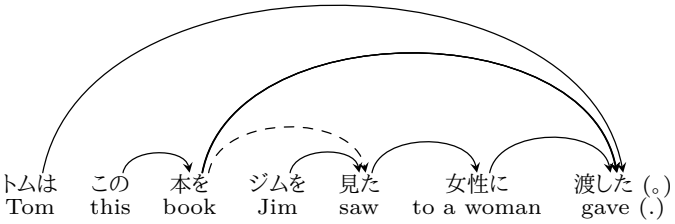
Dependency parsing in Japanese sentences also focuses upon relationships between two phrases in the beginning. This is called a bi-nominal approach, which generates a syntax tree for a sentence. This approach is a conventional method for parsing the dependencies between phrases in Japanese, and some methods using this approach have been developed thus far [5], [7]. These methods generate a syntax tree model to determine whether there is a dependency in Japanese between phrases in a sentence or not on the basis of features such as parts of speech and inflectional forms. They also conduct a syntactic analysis by using the generated syntax tree based on the following algorithm:

1) Check all the phrases in a sentence to ascertain whether they have a dependency with others located on the right side of the syntax tree.
2) Designate any phrase that has a dependency in Step 1 as the analysis result. At this time, the referrer of the dependency is excluded from the target of Step 1.
3) Repeat Steps 1 and 2 and keep the analysis results until all but the final phrase has a dependency.

The conventional methods described above consider only the relationship between two phrases in a sentence; however, do

not do otherwise. Therefore, they might misjudge some of the dependencies shown in Figure III-A.

The Japanese sentence "トムはこの本をジムを見た女性に渡した。" is shown in Figure 1. This sentence has a complex structure, and its English translation is "Tom gave this book to a woman who saw Jim." Originally, the phrase "本を (the book)" takes the dependency depicted by the solid line to another phrase "渡した (gave)," because "the book" is the object of "gave." However, sometimes the dependency from "本を (the book)" to "見た (saw)" as depicted by the dashed line in Figure III-A is extracted by the bi-nominal approaches. This is because the bi-nominal model cannot consider the hierarchical structure of the sentence. Thus, it is impossible to perform accurate parsing for a statement that has a complex structure.



(Tom gave this book to a woman who saw Jim.)

Figure 7. Example of parsing in Japanese.

Therefore, we propose a new method for parsing dependencies in Japanese on the basis of the hierarchical structure defined by the phrase relationships.

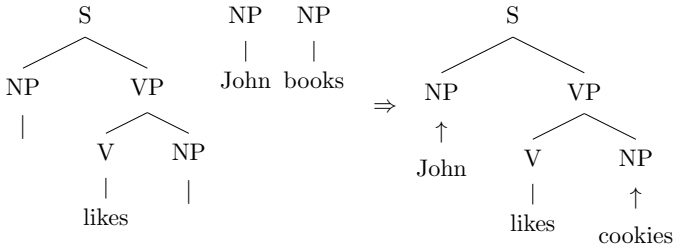
B. Tree-Substitution Grammar

Tree-substitution grammar (TSG) is an extension of context-free grammar (CFG) [3]. Both CFG and TSG have rewrite rules, called productions, which are used for constructing syntax trees by replacing nonterminal symbols with elementary trees, which are a part of the syntax tree.

The TSG production replaces a nonterminal symbol with an elementary tree whose depth is greater than one, while the CFG production does the same with an elementary tree whose depth is exactly one. As a result, the TSG production has a hierarchical structure based on a context. TSG is 4 tuple and is defined as $G = \{T, N, S, R\}$, where T denotes a set of terminal symbols; N, a set of nonterminal symbols; $S(\in N)$, a set of the distinguished root nonterminals; and R, a set of productions. In general, the syntax tree of an English sentence is described as a tree structure because of its phrase-structure rule.

Here, the root node is labeled with a nonterminal symbol, and its leaf nodes are labeled with either terminal symbols or nonterminal symbols. In short, the syntax tree of the input sentence is constructed by recursively replacing the nonterminal symbols with the elementary trees.

Figure 8 shows an example of parsing from the syntax tree, which is labeled on the left. For example, the S $\rightarrow$ (NP (VP (V like) NP)) production rewrites the nonterminal symbol S



Upper arrow "↑" indicate substitution sites in TSG.

Figure 8. A syntax tree consisting of three elementary trees.

with the fragment (S NP (VP (V likes) NP)). This syntax tree has two NPs as its nonterminal symbols, and the production rewrites these nonterminal symbols with the fragments (NP John) and (NP cookies).

In short, a derivation creates a tree by starting with the root symbol and rewriting (substituting) it with an elementary tree, then continuing to rewrite frontier nonterminals with elementary trees until there are no remaining nonterminals.

In TSG, the replacement of nonterminal symbols with elementary trees is performed in an arbitrary manner; however, we can calculate the probabilities of the rewrite rules in the syntax tree. Probabilistic tree-substitution grammar (PTSG) is an extension of TSG whose productions have their probabilities $P(e|c)$, where the elementary tree e replaces a nonterminal symbol c. $P(e|c)$ is statistically calculated using training data. The distribution of the PTSG production G_c can be generated by the hierarchical Pitman-Yor process described before using nonterminal symbol c insted of context $\mathbf{u}$ in Equation (4), so that G_c is defined as follows:

$$G_c \sim \mathrm{HPY}(d_c, \theta_c, G_{\pi(c)}) \tag{10}$$

where d_c and θ_c denote the hyperparameters of the hierarchical Pitman-Yor process when we use the nonterminal symbol c, and $G_{\pi(c)}$ represents a distribution over the infinite-dimensional distribution of the elementary tree with c.

To generate the elementary tree e, we now draw e_1 from G_ϕ, thereby obtaining an elementary tree with nonterminal symbol $c_1, \ldots, c_m$, and then, draw $e_2, \ldots, e_m$ in turn from the base measure $G_{\pi(c_1)}, \ldots, G_{\pi(c_m)}$. The above process is repeated until a full tree is generated.

However, a problem associated with the segmentation of elementary trees arises when PTSG is derived. Gibbs sampling, in which random variables are repeatedly sampled on the basis of the current values of all other random variables in the model, is usually used for solving this problem. Figure 9 shows an example of the segmentation of a syntax tree. In the procedures described above, we can perform dependency parsing based on the hierarchical structure. This method, state-of-the-art English dependency parsing [4], is known as symbol-refined TSG (SR-TSG). SR-TSG applies *symbol refinement* techniques [19], [20] to parse English sentences. These techniques can also be used in parsing methods that do not consider the hierarchical structure of a sentence, such as CFG. Thus, although not directly related to TSG, SR-TSG can be used for constructing symbol-refined syntax trees and is considered a state-of-the-art method.

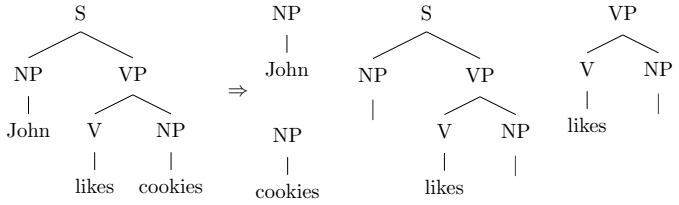


Figure 9. Elementary trees based on PTSG.

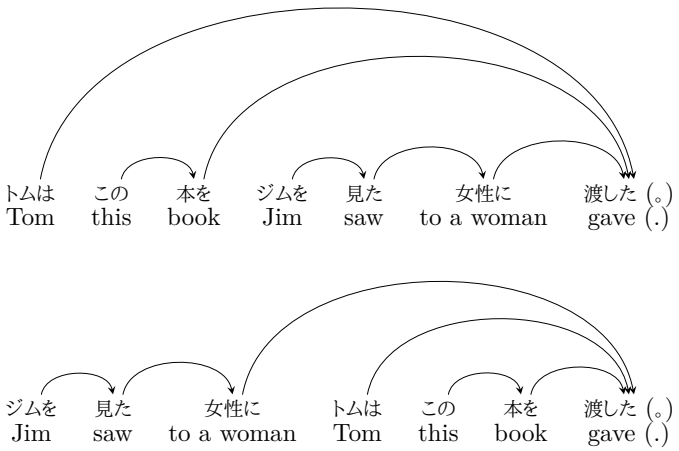


Figure 10. Two Japanese sentences that have the same structure.

However, TSG cannot support dependency parsing of Japanese sentences, because its syntax tree does not have a tree structure, which represents the syntactic function of each phrase, as required for the postpositions in the Japanese language. For example, two sentences are shown in Figure 10. The phrase order is different in these two sentences, while each phrase has the same dependencies. Therefore, it is impossible to conduct dependency parsing for Japanese sentences by using TSG, because TSG is only applicable to the language whose structure can be expressed as a syntax tree. As a result, we propose a novel dependency parsing method for the Japanese language that considers both the hierarchical structure and the order of phrases using an extension of TSG.

C. Topic Model

In general, a topic model called LDA or its derivations consider a bag-of-words to be an n-gram distribution in each document, and assign latent topics to each document if the generated topic distribution is the same as the observed n-gram distribution. However, the topics estimated by the generated topic model depend only on the word distribution. This is because such topics are characterized on the basis of the word distribution, which is constituted by the observed words, in each document. This is primarily caused by not considering grammatical factors such as the order of words and the dependency structures in a document. Therefore, a general topic model faces the problem of assigning incorrect topics to the documents.

For all of these reasons, modern topic models applicable to natural English language processing utilize the rules of grammar as their features, and some studies manage hierarchization of the syntactic structures and the syntax trees based on them [21], [22]. Similarly, the topic model applicable to natural Japanese language processing uses syntax rules as its features. For example, Kitajima et al. defined the features except an n-gram distribution of words as events that are mainly composed of phrase dependencies [23]. This topic model treats an event in units of not words but phrases to extract the events from each document. As a result, they estimate the topics of the documents on the basis of the distribution of the events.

Here, we would like to consider the *event* in [23]. We have already explained that this event is composed of a phrase dependency; in other words, it is an example of a phenomenon expressed by characters (e.g., What is happening? Who felt how?). From the standpoint of the research field of natural language processing, the event consists of pairs of two phrases with a dependency in each document. Consequently, they are expressed such as $(subject, predicate)$ or $(predicate_1, predicate_2)$. Because $(subject, predicate)$ denotes the movement or atmosphere of *subject* using *predicate*; moreover, $(predicate_1, predicate_2)$ describes the characteristic and atmosphere of *predicate*$_2$ using *predicate*$_1$. Therefore, we can detect events by using a dependency parsing technique if we take note of the pairs described above.

The part of speech of the subject is usually a noun or an unknown word, and that of the predicate is usually a verb, an adjective, or an adjectival noun, according to the idea presented in [23]. Therefore, a topic model based on the pairs of two phrases is constructed using the result of dependency parsing. Consequently, we can estimate the latent topics considering a grammatical factor by performing the topic estimation of every document.

IV. Hierarchical Dependency Parsing

As described in Section III, the dependency parsing of Japanese language should consider both the hierarchical structure and the order of phrases in a sentence. Existing works just consider the relationship between two phrases in a sentence, so that they might have the potential for extracting false dependencies. The hierarchical structure which is focused in this paper is called *weak context* in the research field of natural language processing. In this section, we explain our method for considering a weak context in a sentence by using an extension of TSG.

A. Syntax Tree Construction

In order to consider the dependencies between phrases, we generate a syntax tree based on the n-gram model made from the occurrence of dependencies. We also calculate the probabilities of the referrer phrases subject to the occurrence of the destruction phrase. In Figure 11, for example, the dependency between "トムは (Tom)" and "渡した (gave)" means that referrer phrase "トムは (Tom)" depends on the referenced phrase "渡した (gave)." If we can extract this dependency, we can calculate the occurrence probability of the dependency between phrase "トムは (Tom)" and "渡した (gave)" as the conditional probability P_D(トムは (Tom)| 渡した (gave)), which can be calculated using the prior probability P_D(渡した (gave)) in the bi-gram model.

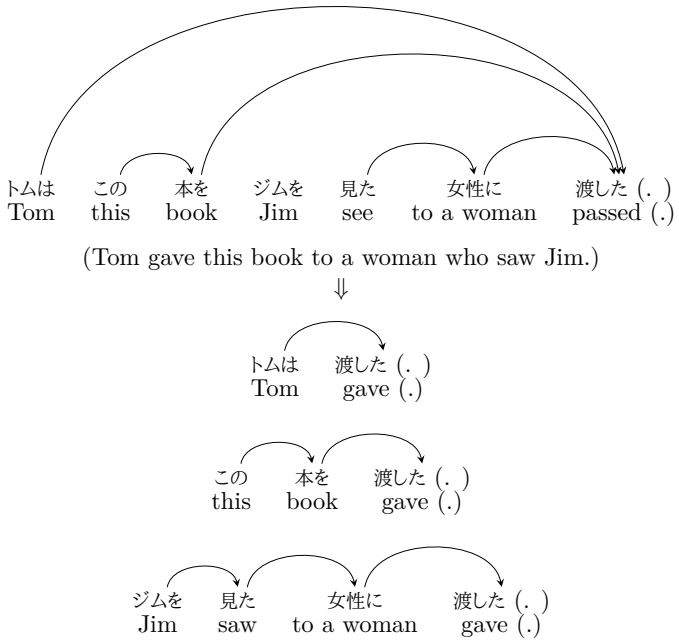


Figure 11. Hierarchical dependencies in the Japanese language.

At present, we cannot construct an n-gram model that reflects the occurrence of phrases if n is small. Conversely, the size of the n-gram model is large if n is large. In order to solve this problem, we use the variable-order Pitman-Yor language model [24], which is an extension of the hierarchical Pitman-Yor process [14]. This technique can help to treat n as a variable in the n-gram model. Using the variable-order Pitman-Yor language model, we can generate a syntax tree model considering the number of phrases. When we calculate the probability of the referrer phrase being the root node of the elementary tree by Equations (4) and (5) just like the hierarchical Pitman-Yor process.

For example, Figure 11 shows three elementary trees comprising the end of a sentence. In short, we obtain three elementary trees called the weak context and can calculate the probability $P_D(\cdot|$ 渡した (gave)) of the end of the sentence being the root node of the observed syntax tree.

Thus, we can precisely generate a syntax tree with a weak context dependency.

B. Hierarchical Dependency Parsing Algorithm

It is known that dependencies in the Japanese language have the following limitations:

- Japanese is a head-final language. Except for the rightmost one, thus, each segment modifies exactly one segment among the phrases appearing to its right.
- The dependencies do not cross one another.

Since the syntax trees are constructed using the method illustrated in Figure 12 (a), each tree whose root node is the phrase at the end of the sentence, is hierarchical; therefore, we parse the sentence in a bottom-up manner by using an algorithm called CYK [25], which is based on a depth-first search. The phrase at the end of the sentence gets the dependency from the phrase immediately before it; therefore, we regard the phrase at the end of the sentence and the phrase immediately before it as the weak context dependency and find another dependency in the sentence.

In Figure 12 (b), for example, the phrase "渡した (gave)" is the phrase at the end of the sentence; therefore, we obtain the dependency from "女性に (to a woman)." As a result, we can calculate the probability of the dependency P_D(女性に (to a woman)| 渡した (gave)). In the same manner, we can calculate different probabilities of dependencies P_D(見た (saw)| 女性に渡した (gave to a woman)) and P_D(見た (saw)| 渡した (gave)), if we assume that there is a dependency between "女性に (to a woman)" and "渡した (gave)". Presently, we compare P_D(見た (saw)| 女性に渡した (gave to a woman)) with P_D(見た (saw)| 渡した (gave)), and then, extract the dependency between the phrases. If P_D(見た (saw)| 女性に渡した (gave to a woman)) is larger than P_D(見た (saw)| 渡した (gave)), we decide that there is a dependency between "見た (saw)" and "女性に渡した (gave to a woman)." This process is repeated until we get to the phrase at the beginning of the sentence. In the end, all the phrases for which the relationships are adjudged to be dependencies form the syntax tree of the sentence.

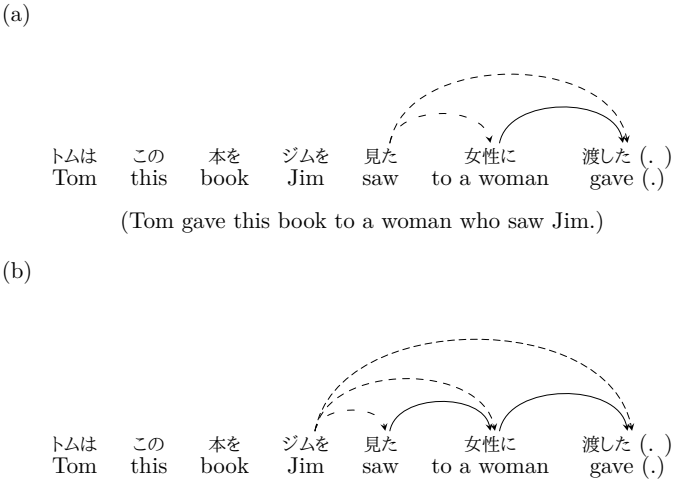


Figure 12. An example of dependency parsing processes using the hierarchical dependency model for Japanese.

C. Experimental Evaluation of Proposed Dependency Parsing

To evaluate the proposed method , which considers a weak context in the Japanese language, we compared it with a conventional method using the Kyoto University Text Corpus [26]. The Kyoto University Text Corpus is a Japanese text corpus for handling the dependencies of morphemes and segments in the Japanese language, and is commonly used for evaluating the effectiveness of the morphological analysis and dependency parsing of the Japanese language. It consists of newspaper articles published over a period of five days; however, we utilized only one day's articles (1,100 sentences) for the dependency parsing because it takes more time to generate syntax trees and we need to decrease the processing time.

We chose CaboCha [5] as the conventional method, because CaboCha is considered the most effective method for parsing dependencies in the Japanese language. The Kyoto

University Text Corpus was also used for the evaluation of CaboCha.

The following procedures are our experimental evaluation for obtaining dependencies in the Japanese language:

1) In order to create the syntax trees of sentences, we first extract the dependencies between phrases in each sentence.
2) We construct syntax trees focusing on the phrases at the end of the sentences assigned to the root node of each syntax tree.
3) We apply the hierarchical Pitman-Yor process and estimate parameter θ by using the Gibbs iteration.
4) We divide the sentences into segments after performing a morphological analysis of the Japanese language and check the segment sequences against our syntax trees.
5) We parse dependencies while continuing to extract the dependencies between phrases from the phrase at the end of an input sentence.

We also performed the same process for an experimental evaluation using CaboCha on the basis of a bi-nominal model [5]. To compare the proposed method with CaboCha, we utilized the following measurement denoted by R that is frequently used for evaluating the accuracy of the dependencies:

$$R = \frac{Y}{X} \tag{11}$$

where X denotes the number of dependencies in the test data, and Y represents the number of dependencies extracted by each parser (the proposed method and CaboCha). Of course, dependencies extracted by each parser are contained in the test data; therefore, R is usually termed recall in the field of information retrieval [27]. In order to calculate R by using each parser, we randomly selected 200 sentences from newspaper articles published on the other four days. In other words, we focused on the efficacy of the dependencies extracted by each method. Table I indicates that the method is, unfortunately, inferior to CaboCha from the viewpoint of effectiveness.

TABLE I. EXPERIMENTAL RESULTS 1

	CaboCha	Proposed method
R	0.871	0.769

This is because the proposed method utilizes syntax trees not only using a small quantity features for parsing dependencies in Japanese language. For which, CaboCha can parse dependencies in Japanese language with various features (word, lexical form of word, POS tag, and inflectional form), so that the accuracies of parsing dependencies in Japanese language are effective using CaboCha. Therefore, the precision of the proposed method can be improved further.

TABLE II. EXPERIMENTAL RESULTS 2

	Proposed method	CaboCha
Long, complex sentences	0.700	0.550

However, the proposed method has a competitive advantage for accurately extracting dependencies from long Japanese sentences whose syntax tree has a complex structure. In other words, we have to evaluate Japanese sentences from which CaboCha cannot extract dependencies. Therefore, we randomly selected 20 long sentences (10 percent of the test data used in the previous experiment) with relatively complex structures.

Table II shows that the proposed method could accurately extract dependencies from 14 of the 20 sentences, while CaboCha could do this only for 11 of the sentences. In short, the proposed method could parse dependencies of long, complex Japanese sentences better than CaboCha. This is attributed to the fact that it is possible for the parser to interpret the meaning of the postposition in some cases, even if the syntax tree model does not have a hierarchical structure. However, the syntax trees constructed using the proposed method contain not only the relationships between phrases but also the hierarchical structure of each phrase. Consequently, the proposed method can restrain a failed analysis of a complex Japanese sentence carried out using a conventional method.

V. APPLICATION OF PROPOSED PARSER

We evaluate not only the performance of proposed dependency parsing method but also its practicality to estimate topic modeling. Before we explain the contribution of our topic model generated using the proposed dependency parsing method, we describe the problem that the topic model generated using the conventional method has.

A. Problems of Event-based Topic Model

As described in Section III-C, the recent topic model is not built using word distribution. One such topic model proposed by Kitajima et al. is called an event-based topic model and is built using dependencies between phrases in each document [23]. In [23], the dependencies between phrases are extracted using a Japanese dependency parser called CaboCha [5], which is the most famous Japanese dependency parser. After extracting the dependencies between phrases, auxiliary verbs and postpositions are removed from the dependencies to extract the relationships between words. As a result, the event-based topic model is considered to be built using the dependencies between words.

However, in the case of the Japanese language, the auxiliary verbs and postpositions play a role in denoting case and tense to constitute grammar, whereas in other languages, they are not essential words. Therefore, we believe that both auxiliary verbs and postpositions should not be removed from the dependencies between phrases, and dependencies extracted by a Japanese dependency parser should be directly used for building an event-based topic model.

In the event-based topic model, we can extract the event related to the topic of the sentence, because we focus on its subject. However, we cannot accurately extract topics when we want to extract events from a sentence with a complex structure, because only $(subject, predicate)$ and $(predicate_1, predicate_2)$ are focused upon.

For example, if there is a Japanese noun phrase "海がきれいな景色の美しいホテル (the hotel with a magnificent view of the beautiful sea)," it is performed using a Japanese dependency parser, as shown in Figure 13. Therefore, from the noun phrase, we can extract four events, namely (海が, きれいな), (きれいな, ホテル), (景色の, 美しい), and (美しい, ホテル). These events are translated into English as follows:

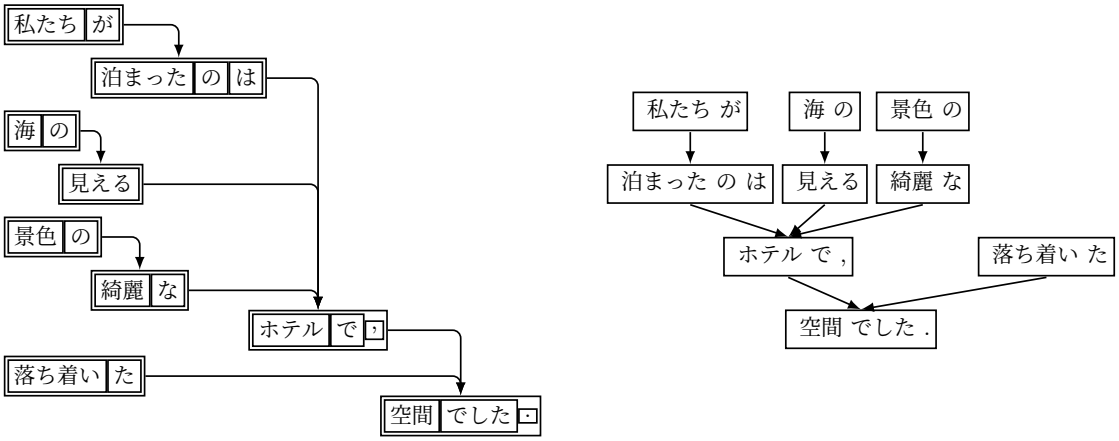


Figure 14. Example of the syntax tree of the parsing sentences.

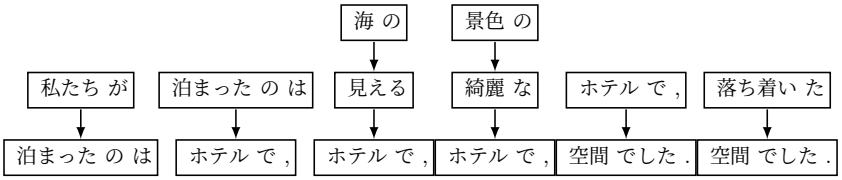


Figure 15. Extraction of phrases based on the hierarchical dependency syntax model.

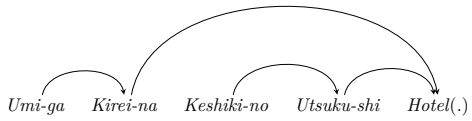


Figure 13. Example of a syntax tree of a phrase.

- (海が, きれいな): beautiful sea
- (きれいな, ホテル): beautiful hotel
- (景色の, 美しい): magnificent view
- (美しい, ホテル): magnificent hotel

From these events, we can find two meanings “きれいな (beautiful)” and “美しい (magnificent)” related to “ホテル (hotel).” However, only “美しい (magnificent)” qualifies “ホテル (hotel),” while “きれいな (beautiful)” qualifies “海が (sea).” As a result, there may be a vague relation between words if we generate an event-based topic model by using their dependencies. For these reasons, we have to detect the hierarchical structure of the events to extract them accurately.

This problem is caused by the application of a bi-nominal dependency parsing in a morphological analysis of Japanese sentences; therefore, we use the proposed dependency parsing, which can extract dependencies between phrases with complex structures, and utilize the dependencies as features to build our topic model. By using the proposed approach, we obtain variable orders of the dependencies; therefore, there is only a slight chance of finding a vague relation between phrases. Thus, the proposed approach has the potential to build a more appropriate topic model than the conventional method. This also helps us to evaluate the practical uses of the proposed dependency parsing.

B. Event Structure-based Topic Model

In order to construct a topic model using the proposed dependency parsing method for Japanese sentences, the following steps are executed:

1) Using the proposed dependency parsing method described in Section IV, we parse each sentence in a given document set and extract the dependencies between phrases in a variable order, and construct a syntax tree by using the extracted dependencies.
2) We divide the syntax tree into appropriate size of dependencies between phrases.
3) We can generate a topic model using the appropriate size of dependencies.

Using a Japanese sentence “私たちが泊まったのは，海の見える景色のきれいなホテルで，落ち着いた空間でした (The hotel with a magnificent view of beautiful sea which we stayed had a relief space),” we can construct a syntax tree shown on the left of Figure 14. The purpose of constructing the syntax tree is to obtain the appropriate size of dependencies between phrases, as shown in Figure 15, because the quality of the topic model depends on that of the extracted dependencies between phrases. Therefore, we extract them by reconstructing the tree structure whose root node denotes the last phrase “空間でした (had a relief space),” as shown on the right of Figure 14, by using the hierarchical dependency syntax model. Finally, we extract the set of phrases (element trees) shown in Figure 15.

Note that we do not use all the morphemes contained in the extracted phrases, as in the case of an event-based

topic model. That is, we remove only the phrases containing a postposition that expresses the case in each sentence, and use the remaining phrases (e.g., noun, verb, and adjective) as features of the proposed topic model. This procedure clarifies the topic boundaries.

As described above, the proposed topic model is built using the LDA shown in the graphical model that is replaced w with ph in Figure 6.

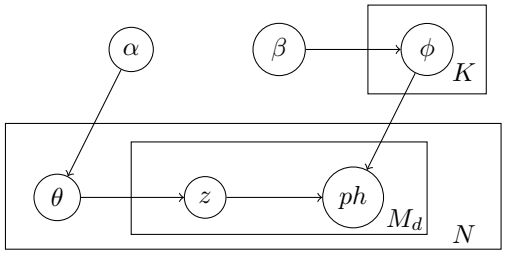


Figure 16. A graphical model of LDA with our extension.

In brief, a set of documents ph is expressed as the distribution of the appropriate size of phrases in each document. That is, θ is generated on the basis of the topic distribution corresponding to the phrases, and ϕ is generated on the basis of the phrase distribution corresponding to the topics. The parameters K, M_d, and N shown in Figure 16 denote the number of topics, the number of phrases with the most appropriate size of dependency in a document d, and the number of documents, respectively. As a result, this graphical model corresponds to a general LDA graphical model, replacing a word with a phrase having the most appropriate dependency size.

C. Experimental Evaluation of Proposed Topic Model

To evaluate the event structure-based topic model based on the proposed dependency parser, we compare our topic model with an event-based topic model. It is difficult to define the evaluation measure of this experiment; however, we evaluate each topic model in terms of whether it can explain the details of each detected topic or not.

First, we use the review data of *Rakuten Travel* [1], which is a travel website that provides information regarding hotel accommodation, to build the topic models. The use of these data allows us to score each review dataset on a five-point scale with respect to the quality of the hotel room and the hotel's geographical convenience. In short, there is a correlative relationship between the text of the review data and its score; therefore, we can allocate topics to each document on the basis of this correlative relationship. As a result, we can consider this correlation as the indication of each topic model because the generated topic model expresses this correlation through the elements in the documents.

We also consider that a document containing a number of topics is suitable for evaluating a topic model, because the text data are of varying lengths. As a result, we use review data whose length is approximately 100 words.

A large variety of parameters in each topic model are decided by reference to [23]. That is, the number of documents M is 2,000 which is randomly extracted from the entire review dataset, and the number of topics in the selected documents is 50. Hyperparameters α and β are also decided by a collapsed Gibbs sampler known as the Markov Chain Monte Carlo method. Table IV shows the results of this experiment obtained using conventional (left-hand side) and proposed topic models (right-hand side).

Here, we focus on two phrases: one is "チャペル–ある (be – a chapel)" at topic 3 on the left-hand side of Table III, and the other is "チャペル–ある–華やか (be – brilliant – a chapel)" at topic 4 on the right-hand side. Compared with these phrases, our topic model can extract a modifier "華やか (brilliant)" of a noun "チャペル (a chapel)." This should be valued because we can verify whether the topic has a positive aspect or not depending on whether the topic model can detect the modifier. From Table III, it seems that our topic model can detect a number of modifiers and can explain the contents of each topic easily by using them; therefore, our topic model can correctly classify documents compared with the event-based topic model, because our topic model can allocate appropriate phrases to each topic.

VI. Conclusion and Future Work

In this paper, we proposed a method for parsing dependencies in the Japanese language by using syntax trees based on a hierarchical structure, which are constructed by analyzing the modification relations in the considered documents. By considering these relations, the proposed method could solve the problems described in Section I and improve the accuracy of dependency parsing for long, complex Japanese sentences compared with the conventional method.

Moreover, topic estimation using the proposed dependency parsing method could be performed well compared with the existing method on the basis of the word distribution and by ignoring the order of words. That is, an appropriate dependency parsing method helps us not only to parse modification relations well but also to apply another research field of natural language processing; therefore, we could demonstrate the necessity for conducting basic research such as that on a dependency parsing method.

In the near future, we intend to utilize the order of words as well as the results of the syntactic and case structure analyses to improve the accuracy of the dependencies in the Japanese language. We also plan to conduct experimental evaluations using all the data in the Kyoto University Text Corpus.

Acknowledgment

We would like to express our sincere gratitude to all who commented on this work at the Eighth International Conference on Digital Society (ICDS 2014) for their assistance in identifying the weaknesses in our work and for giving suggestions on how to improve this study. We would also like to thank Rakuten, Inc. and the National Institute of Informatics for permission to use the resources of Rakuten Travel. This work was partly supported by a Japan Society for the Promotion of Science (JSPS) Grant-in-Aid for Scientific Research (A) (Grant Number 25240014) and Grant-in-Aid for Exploratory Research (Grant Number 25540150).

[1] http://travel.rakuten.co.jp/

TABLE III. EVENT BASED TOPIC MODEL

topic id	phrases with dependency relation
0	初めて–利用 泊まる–思う 西葛西–多い
1	部屋–狭い キャンペーン–着ける ツイン–宿泊 料金–宿泊
2	立地–良い ホテル–窓口 ツイン–部屋 とても–良い
3	駅–近い チャペル–ある 客室–忘れる 江坂–歩く
4	シングル–泊まる ホテル–泊る 以前–利用 カード–支払い
5	ホテル–部屋 ツイン–利用 シングル–快適 シングル–ある
6	ホテル–最寄り できる–ホテル ページ–見る 国道–ある
7	シングル–宿泊 ツイン–部屋 赤坂–向かう 京都–行く
8	以前–利用 皆さん–ご 繁華–困る 名古屋–出る
9	昨年–利用 交通–便利 まだ–おすすめ ９月–情報
10	フロント–対応 シングル–予約 ミナミ–中心 提供–地図
11	今回–宿泊 ９月–利用 部屋–広い 北大–いえる
12	部屋–狭い 私–利用 残念–ある 地元–住む
13	ホテル–窓口 シングル–予約 ツイン–宿泊 宿泊–教える
14	シングル–思う ・–点 部屋–きれい 部屋–狭い
15	静か–落ち着く ここ–数 除く–良い １月–連
16	先日–宿泊 ツイン–部屋 初めて–泊まる W–部屋
17	何–利用 シングル–泊まる 別館–ない 知人–ため
18	止まる–離れる 確か–広い 宿泊–考える 平成–前
19	シングル–予約 低層–なる ベッド–足元 窓口–宿泊
20	部屋–広い 部屋–狭い シングル–泊まる 日航–近く
21	今回–利用 私–泊まる 泊まる–する 昨年–利用
22	一般–ビジネス ．–広い 車–行う １０月–利用
23	駅–近い ツイン–部屋 ホテル–窓口 私–住む
24	ツイン–泊まる 駅–近い 泊まる–日 先日–利用
25	宿泊–ある ホテル–窓口 一泊–快適 シングル–泊まる
26	私–予約 ホテル–窓口 ここ–情報 ツイン–泊まる
27	ページ–予約 部屋–バス 部屋–広い 国際–便利
28	部屋–広い 駅–歩く 駅–近い 窓口–宿泊
29	ホテル–窓口 今回–宿泊 値段–割 会員–皆さん
30	駅–近い フロント–対応 ホテル–ある 宿泊–ホテル
31	シングル–泊まる ポイント–利用 先日–宿泊 今回–利用
32	結構–きれい 部屋–綺麗 ２月–利用 とにかく–安い
33	部屋–臭い 仕事–遅い 泊る–ツイン ３月–する
34	ずいぶん–訪れる 広い–情報 非常–する 念願–北海道
35	ホテル–窓口 宿泊–ある 窓口–利用 駅–便利
36	駅–便利 部屋–広い 利用–持てる 場所–わかる
37	ツイン–泊まる ツイン–部屋 ホテル–窓口 昨日–泊まる
38	ホテル–窓口 ここ–泊まる ダブル–利用 方–泊まる
39	ホテル–ある 大阪–慣れる 泊まる–感想 ホテル–近く
40	何–利用 部屋–宿泊 窓口–なる 髭–剃る
41	シングル–利用 年末–利用 皆さん–情報 初めて–泊まる
42	シングル–宿泊 ツイン–泊まる ９月–利用 フロント–対応
43	部屋–広い 今年–出来る 駅–近く 駅–近い
44	部屋–狭い 値段–割 シングル–宿泊 ツイン–シングル
45	近く–ある シングル–宿泊 ホテル–部屋 ネット–利用
46	初めて–利用 今回–利用 遊べる–ホテル 窓口–予約
47	部屋–きれい シングル–泊まる 値段–割 ここ–投稿
48	シングル–一泊 料金–安い 良い–良い 部屋–きれい
49	大変–感謝 ダブル–感じ 立派–ホテル 仕事–便利

TABLE IV. HIERARCHICAL EVENT BASED TOPIC MODEL

topic id	phrases with dependency relations
0	泊まる–思う–使える 西葛西–多い–抱く ベッド–快適–窮屈
1	部屋–狭い–きれい 安い–部屋 キャンペーン–着ける ロビー–作り–豪華
2	突然–止まる とても–良い–ホテル ホテル–窓口 ３月–よい–便利
3	以前–利用–とき 江坂–歩く 完成–きれい–助かる 中間–泊まる–部屋
4	カード–支払い–拒否 チャペル–ある–華やか ９月–中旬–利用
5	ホテル–泊る–トイレ 立地–良い–ある ツイン–ホテル–ある
6	ホテル–最寄り–駅 ＪＲ–線路–横 ページ–見る–予約
7	京都–行く 下–記述–もの 蒲田–駅–良い 観光–利用–持てる
8	価格–魅力–難 シングル–充分–思う ホテル–窓口–利用 ６月–下旬
9	まだ–おすすめ–思う シングル–出来る–思える おそらく–行う–点
10	ミナミ–中心 客室–改修–行う 立地–不便–便利 名古屋–ホテル–埋まる
11	ホテル–みる–行ける なかなか–きれい–ホテル すすきの–繁華–近く
12	シングル–思う–落ち着く 残念–ある–アクセス 地元–住む–する ５月–宿泊
13	ホテル–窓口–利用 以前–宿泊–時 部屋–入る–差し込む 町外れ–小高い–丘
14	ホテル–窓口–予約 ・–点–親切 別–予約–予約 部屋–狭い–いい
15	ここ–数–泊まる 私–泊まる–建物 駅–近い–利用 ちょっと–歩く–ある
16	W–部屋–つかう 泊まる–思う–泊まる 部屋–狭い–ありがたい
17	別館–ない–不便 一–なる–思う 部屋–バス–広め 新しい–ある–ホテル
18	ホテル–窓口–予約 止まる–離れる–眠れる 平成–前–言う 窓口–ある
19	ホテル–窓口–予約 相対–狭い–狭い 低層–なる–なる 昨年–９月–利用
20	ホテル–窓口–利用 みなさん–情報 宿泊–入る–心細い 議論–届く–届く
21	シングル–タイプ–宿泊 何–利用–利用 思う–ほど–奇麗
22	駅–便利–きれい ．–広い–朝食 メッセージ–ある–思う やはり–予定–観る
23	宿泊–ところ–思う コンビニ–ない–難点 クリスマス–土日–一泊
24	泊まる–日–雨 ロビー–大理石–ホテル 何–宿泊–最高 部屋–狭い–いい
25	午前–着く–ため 少し–前 私–予約–もらう 中–上–指摘
26	ホテル–窓口–予約 尾道–繁華–離れる 福岡ドーム–観戦–ため
27	部屋–バス–狭い ホテル–窓口–予約 私–泊まる–狭い 妻–東京–行く
28	名古屋–ある–立地 ホテル–ある–いい 昨年–夏–滞在 見かけ–古い–裏腹
29	ポイント–利用–いい 本–宿泊–感じる 去年–宿泊 立地–離れる–こたえる
30	ページ–予約 結構–きれい–コストパフォーマンス
31	ホテル–窓口–予約 温泉–予約–遠い 結構–きれい–ある 建物–古い–良い
32	宿泊–ホテル 妻–二–泊まる シングル–泊まる–狭い とにかく–安い
33	ずいぶん–訪れる–持てる 泊る–ツイン–気分 ３月–する–なる
34	場所–わかる–回る 広い–情報–みる 非常–する–ホテル 部屋–広い
35	ほんとに–ラブホテル–静か 朝食–バイキング 部屋–入る–差し込む
36	小松–右–清潔 １０月–ツイン–困る 部屋–綺麗–充実
37	大きな–執務 建物–言える–良い １月–週末–利用 地下–ある–いい
38	ツイン–部屋–宿泊 近く–ある–ない ダブル–利用–満足 連–欠点–厳しい
39	大阪–慣れる 宿泊–もてる–静か 泊まる–感想–する 非常–ホテル–滞在
40	部屋–用意–囚人 依然–宿泊–事 髭–剃る–入れる メッセージ–ある–思う
41	最寄り駅–行ける–ない 中洲–駅 先週–出張–際 観光–応対–ほう
42	年末–利用–思う 他–皆さん–情報 昨年–夏–滞在
43	今年–出来る 値段–割–高い 私–行く–ホテル シングル–する–ある
44	利用–なる ツイン–広い–気分 部屋–狭い 地下鉄–駅–約
45	ホテル–窓口–利用 ネット–利用–なる 広い–良い–ホテル 室内–する–部屋
46	新宿–駅–到着 近く–ある 窓口–予約–利用 いい–もと
47	ノート–もつ–出張 ここ–投稿–高い 利用–満足–感じ 中間–泊まる–部屋
48	立地–いく–ある 部屋–きれい–快適 良い–良い–広い 料金–安い–最高
49	大変–感謝–利用 ダブル–感じ–きれい 立派–ホテル–ゴージャスん

REFERENCES

[1] K. Ono and K. Hatano, “A dependency parsing method based on the hierarchical structure in Japanese language,” in The Eighth International Conference on Digital Society, 2014, pp. 203–208.

[2] M. Post and D. Gildea, “Weight pushing and binarization for fixed-grammar parsing,” in Proceedings of the 11th International Conference on Parsing Technologies, 2009, pp. 89–98.

[3] P. Blunsom and T. Cohn, “Unsupervised induction of tree substitution grammars for dependency parsing,” in Proceedings of Conference on Empirical Methods in Natural Language Processing, 2010, pp. 1204–1213.

[4] H. Shindo, Y. Miyao, A. Fujino, and M. Nagata, “Statistical Parsing with Probabilistic Symbol-Refined Tree Substitution Grammars,” in Proceedings of International Joint Conference on Artificial Intelligence, 2013, pp. 3082–2086.

[5] T. Kudo and Y. Matsumoto, “Japanese dependency analysis using cascaded chunking,” in Proceedings of Conference on Natural Language Learning, 2002, pp. 63–69.

[6] Y. Cheng, M. Asahara, and Y. Matsumoto, “Machine learning-based dependency analyzer for chinese,” Journal of Chinese Language and Computing, vol. 15, no. 1, 2005, pp. 13–24.

[7] M. Sassano, “Linear-time dependency analysis for japanese,” in Association for Computational Linguistics, 2004.

[8] N. Chomsky, The Logical Structure of Linguistic Theory. Springer, 1975.

[9] S. Goldwater, T. L. Griffiths, and M. Johnson, “Contextual Dependencies in Unsupervised Word Segmentation,” in ACL-44 Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics. ACL, 2006, pp. 673–680.

[10] D. Mochihashi, T. Yamada, and N. Ueda, “Bayesian unsupervised word segmentation with nested pitman-yor language modeling,” in ACĹ09 Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing. ACL, 2009, pp. 100–108.

[11] J. Pitman and M. Yor, “The Two-Parameter Poisson-Dirichlet Distribution Derived from a Stable Subordinator,” The Annals of Probability, vol. 25, no. 2, 1997, pp. 855–900.

[12] D. J. Aldous, Ecole d’Ete St Flour 1983, ser. Lecture Notes in Mathematics 1117. Springer, 1985, ch. Exchangeability and Related Topics, pp. 1–198.

[13] H. Ishwaran and L. F. James, “Generalized weighted Chinese restaurant processes for species sampling mixture models,” Statistica Sinica, vol. 13, 2003, pp. 1211–1235.

[14] Y. W. Teh, “A hierarchical Bayesian language model based on Pitman-Yor processes,” in Proceedings of Association for Computational Lin-

guistics, 2006, pp. 985–992.

[15] R. Kneser and H. Ney, "Improved backing-off for M-gram language modeling," in Proceedings of International Conference on Acoustics, Speech, and Signal Processing, 1995, pp. 181–184.

[16] T. Hofmann, "Probabilistic latent semantic analysis," in Proceedings of Uncertainty in Artificial Intelligence, 1999, pp. 289–296.

[17] ——, "Probabilistic latent semantic indexing," in Proceedings of the Annual International ACM SIGIR Conference, 1999, pp. 50–57.

[18] G. J. McLachlan and T. Krishnan, The EM Algorithm and Its Extensions., 2nd ed. Wiley-Interscience, 2007.

[19] M. Collins, "Head-Driven Statistical Models for Natural Language Parsing," Association for Computational Linguistics, vol. 29, no. 4, 2003.

[20] T. Matsuzaki, Y. Miyao, and J. Tsujii, "Probabilistic CFG with Latent Annotations," in Proceedings of Association for Computational Linguistics, 2005, pp. 75–82.

[21] J. Boyd-graber and D. Blei, "Syntactic Topic Models," in Neural Information Processing Systems, 2008, pp. 185–192.

[22] Y. Hu and J. Boyd-graber, "Efficient Tree-Based Topic Modeling," in Association for Computational Linguistics, 2012, pp. 275–279.

[23] R. Kitajima and I. Kobayashi, "Latent topic estimation based on events in a document," Advanced Computational Intelligence and Intelligent Informatics, 2012, pp. 603–610.

[24] D. Mochihashi and E. Sumita, "The infinite markov model," in Proceedings of Annual Conference on Neural Information Processing Systems, 2007, pp. 1017–1024.

[25] N. Chomsky, "Three models for the description of language," IRE Transactions on Information Theory IT-2, vol. 2, no. 3, 1956, pp. 113–124.

[26] S. Kurohashi and M. Nagao, "Building a Japanese Parsed Corpus while Improving the Parsing System," in Proceedings of Language Resources and Evaluation Conference, 1998, pp. 719–724.

[27] R. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval: The Concepts and Technology behind Search (ACM Press Books), 2nd ed. Addison-Wesley Professional, Feb. 2011.