Knowledge Discovery of Semantic Relationships between Words Using Nonparametric Bayesian Graph Model

Issei Sato
Graduate School of
Information Science
and Technology
The University of Tokyo, Japan
sato@r.dl.itc.u-tokyo.ac.jp

Minoru Yoshida
Information Technology Center
The University of Tokyo, Japan
mino@r.dl.itc.u-tokyo.ac.jp

Hiroshi Nakagawa
Information Technology Center
The University of Tokyo, Japan
nakagawa@dl.itc.u-tokyo.ac.jp

ABSTRACT
We developed a model based on nonparametric Bayesian modeling for automatic discovery of semantic relationships between words taken from a corpus. It is aimed at discovering semantic knowledge about words in particular domains, which has become increasingly important with the growing use of text mining, information retrieval, and speech recognition. The subject-predicate structure is taken as a syntactic structure with the noun as the subject and the verb as the predicate. This structure is regarded as a graph structure. The generation of this graph can be modeled using the hierarchical Dirichlet process and the Pitman-Yor process. The probabilistic generative model we developed for this graph structure consists of subject-predicate structures extracted from a corpus. Evaluation of this model by measuring the performance of graph clustering based on WordNet similarities demonstrated that it outperforms other baseline models.

Categories and Subject Descriptors
G.3 [Probability and Statistics]: Nonparametric statistics

General Terms
Algorithms

Keywords
Probabilistic Generative Model, Nonparametric Bayes, Graph Clustering, Text Mining

1. INTRODUCTION
Semantic knowledge about words for particular domains is increasingly important in text mining, information retrieval, speech recognition, and so on. We describe an unsupervised approach based on a probabilistic generative model to automatically discovering semantic relationships between words taken from a corpus. A probabilistic generative model is used for modeling the data generation process. By focusing on the generation process, we can extract specific properties of the data such as a latent topic or document category. Moreover, our approach is based on the distributional hypothesis, which is the basis of statistical semantics and states that words that occur in the same context tend to have a similar meaning. This hypothesis is attracting a great deal of attention in the field of cognitive science [12].

We focused on the syntactic structures found in a sentence and used them as contextual information. In particular, the subject-predicate structure was taken as a syntactic structure with the noun as the subject and the verb as the predicate. That is, the observed data consisted of pairs of a subject (noun) and a predicate (verb) extracted from a corpus. The process for generating this data was modeled using a probabilistic generative model. The extracted predicates were regarded as the context of the subject and vice versa.

These extracted pairs of a subject and a predicate are regarded as a graph structure in which each vertex corresponds to a subject or a predicate and each edge means that the corresponding pair of a subject and a predicate is actually observed in the corpus. This graph structure is a disassortative graph structure in which the vertices have most of their connections outside their group. Figure 1 shows an example of a disassortative graph structure.

We extended the interpretation of the distributional hypothesis to "words that have links to the same vertices tend to have a similar meaning". The purpose of our study was to develop a probabilistic generative model for this disassortative graph structure.

The probabilistic generative model proposed by Newman et al. [13] for a disassortative graph structure can be used to generate an assortative graph structure in which all the vertices are divided into groups such that the members of each group are mostly connected to other member of the same group, in a manner similar to a social network. We called this the "Newman model".

The Newman model assumes that the vertices in a graph are divided into $T$ classes or groups, and each class that corresponds to
is subdivided into clusters without any overlap. The MCL algorithm has been applied to various domains including corpus linguistics. Dorow et al. [4, 5] described an unsupervised algorithm that automatically discovers word senses from text by using a graph model representing words and the relationships between them. Sense clusters are iteratively updated by clustering a local graph of similar words around an ambiguous word. Gfeller et al. [8] used this algorithm to clean up a dictionary of synonyms that appeared to be ambiguous, incomplete, and inconsistent.

Relational learning, which has also received a great deal of attention, is used, for example, to find social roles in social network data and to discover an ontology in a particular domain. The stochastic block model is a well-known model for relational learning in sociology. Kemp et al. [10] described the infinite relational model (IRM), which is a stochastic block model for entity-relation modeling. Entities are partitioned into clusters, and the number of clusters is estimated using the Dirichlet process (DP). They used it to automatically extract a biomedical ontology from the Unified Medical Language System. A vertex can be regarded as an entity, and a link between vertices can be regarded as the relationship between the vertices. It is straightforward to use relational models for modeling the graph generation process.

3. NEWMAN MODEL

In this section, we describe the Newman model and the expansion of it by using the Dirichlet process (DP) [7].

3.1 Newman model

\( Y \) is the vertex set, \( v \) is a vertex; i.e., \( v \in Y \). \( V \) is the number of vertices. \( T \) is the number of classes. Suppose that the vertices fall into \( T \) classes with probability \( \pi \), where \( \pi_t \) is the probability that a vertex is assigned to class \( t \). Vertex \( j \) belongs to class \( t \), indicated by \( z_j = t \). Each class has a probability \( \phi_{tk} \) that a link from a particular vertex in class \( t \) connects to vertex \( i \). A link from vertex \( j \) to vertex \( i \) is indicated by \( l_{ij} = i \). Each vertex links to other vertices in accordance with \( \phi \). That is, vertex \( j \) links to vertex \( i \) in accordance with \( \phi_{z_j, i} \). The generation process for link \( l_j \) is represented by

\[
\begin{align*}
z_j & \sim \text{Multi}(\pi) \\
l_j & \sim \text{Multi}(\phi_{z_j}),
\end{align*}
\]

where \( \phi_{z_j} = (\phi_{z_{j,1}, \phi_{z_{j,2}}, \cdots, \phi_{z_{j,V}}}) \), and \( \text{Multi}(\cdot) \) is a multinomial distribution.

The parameters \( z, \phi, \) and \( \pi \) are estimated using the EM algorithm:

\[
\begin{align*}
p(z_j = t) &= \frac{\pi_t \prod_i \phi_{z_j}^{A_{ji}}}{\sum_s \pi_s \prod_i \phi_{z_s}^{A_{ji}}} \\
\pi_t &= \left\{ \begin{array}{ll}
p(z_j = t) & \text{if there is an edge from } j \text{ to } i \\
0 & \text{otherwise}
\end{array} \right.
\end{align*}
\]

where \( A \) is an adjacency matrix with elements \( A_{ji} = 1 \) if there is an edge from \( j \) to \( i \); otherwise \( A_{ji} = 0 \).

3.2 DP-Newman Model

As mentioned above, a problem with the Newman model is that one must decide the number of classes in advance. This problem can be solved by using DP [7][2]. We describe DP by adapting DP to Newman Model.

2. RELATED WORK

The use of graph structures for lexical acquisition of words has received much attention. The relationships between graph structures and human languages have been explained using power-law distributions like Zipf’s law [3]. Graph clustering algorithms are important for unsupervised lexical acquisition of words.

The Markov cluster (MCL) algorithm, a well known graph clustering method proposed by Van Dongen [17], consists of two steps, expansion and inflation, which are processed alternatively. A graph method proposed by V an Dongen [17], consists of two steps, important for unsupervised lexical acquisition of words.

distributions like Zipf’s law [3]. Graph clustering algorithms are used much attention. The relationships between graph structures and human languages have been explained using power-law distributions like Zipf’s law [3]. Graph clustering algorithms are important for unsupervised lexical acquisition of words.
Suppose that $\phi_i$ is distributed in accordance with the Dirichlet distribution $H(\tau)$; i.e., $\phi_i \sim H(\tau)$, where $\tau$ is a parameter of the Dirichlet distribution. $G$ is a random probability measure over $\phi$:

$$G \sim DP(\alpha_0, H(\tau)),$$

where $\alpha_0$ is the DP concentration parameter, and $H$ is the base measure, which is the Dirichlet distribution here. The generation process for link $l_j$ is represented by

$$\phi_{x_j} \sim G$$

$$l_j \sim Multi(\phi_{x_j}).$$

The probability of $z_j$ given $z_{-j} = \{z\} \setminus z_j$ and adjacency matrix $A$ is formulated on the basis of Bayesian theory:

$$p(z_j = t | A, z_{-j}) \propto p(A | z_j = t, A_{-j}, z_{-j})p(z_j = t | z_{-j}),$$

where $A = (A_1, A_2, \cdots, A_V)$, and $A_{-j} = A \setminus A_j$. We can adapt a Gibbs sampler algorithm for estimating $z_j$ by using Eq. (9).

$$p(A | z_j = t, A_{-j}, z_{-j}) = \frac{\Gamma(V + \sum_{v \neq j} \sum_{A_{vi} = t} z_i^v)}{\Gamma(V + \sum_{v} \sum_{A_{vi}} z_i^v)} \prod_i \frac{\Gamma(V + \sum_{v \neq j} A_{vi} z_i^v)}{\Gamma(V + \sum_{v} A_{vi} z_i^v)},$$

where $\Gamma(\cdot)$ is the gamma function, and $z_i^v$ indicates $1$ if $z_v = t$; otherwise, it indicates $0$.

$$p(z_j = t | z_{-j}) = \begin{cases} \frac{n_{t,j}^{-1}}{V + \alpha_0} & \text{if } t \text{ previously used} \\ \frac{V + \alpha_0}{V + \alpha_0} & \text{if } t \text{ is new} \end{cases},$$

where $n_{t,j}^{-1}$ is the number of vertices except vertex $j$ assigned to class $t$; i.e., $n_{t,j}^{-1} = \sum_{v \neq j} z_i^t$.

This calculation can be described using a Chinese restaurant metaphor [1]. Consider a Chinese restaurant with an infinite number of tables, each of which represents a different class. Each table has an infinite seating capacity. Customers enter the restaurant and seat themselves (each customer corresponds to a vertex). The first customer sits at the first available table, and each subsequent customer sits at an occupied table with a probability proportional to the number of customers already sitting there; i.e., $n_t$, or at an unoccupied table with probability proportional to $\alpha_0$, which is estimated by auxiliary variable sampling [6].

4. PROPOSED MODEL

Suppose that each vertex has multiple classes, and that links from a vertex are generated in accordance with its classes. This generation process can be modeled using the hierarchical Dirichlet process (HDP) [16]. First, we describe the generation process by using the restaurant representation in Figure 4. Then, we formulate the generation process using HDP and PYP.

4.1 Generation of Graph Using HDP

Each vertex is or has a restaurant. Each restaurant has tables, and each table is served a menu corresponding to a class. There are an infinite variety of menus. A menu lists specific dishes that represent links to certain vertices. Vertex $v$ generates a link to vertex $i$ as follows. A customer enters restaurant $v$, sits at a table with menu $t$ and orders dish $l_i$. How a customer selects and enters a restaurant is described in Section 4.3. Figure 4 shows the restaurant representation of the graph.

The model is formulated as follows.

$$l_j \sim \text{Multi}$$

$v$ is generated from class $t$. $H$ is the $V$-dimension Dirichlet distribution, which is a probabilistic measure with $\phi_t = (\phi_{1,t}, \cdots, \phi_{V,t})$; i.e., $\phi_t \sim H(\tau)$. $G_0$ is the random probabilistic measure over class. $G_v$ is the random probabilistic measure over class given vertex $v$, as shown in the following formulas.

$$G_0 \sim DP(\gamma, H(\tau))$$

$$G_v \sim DP(\alpha, G_0)$$

$$\phi_{x_j} \sim G_v$$

$$l_j \sim Multi(\phi_{x_j}),$$

where $\alpha$ and $\gamma$ are the DP concentration parameters.

The following formulas represent the stick-breaking representation of the generation process [15][16].

$$\phi_t \sim H(\tau), \ t = 1, 2, \cdots$$

$$\beta_t \sim Beta(1, \gamma), \ t = 1, 2, \cdots$$

$$\beta_t = \frac{\beta_t \prod_{k=1}^{t-1} (1 - \beta_k)}{\sum_{k=1}^{t-1} \beta_k}$$

$$z_j \sim Multi(\theta_v)$$

$$l_j \sim Multi(\phi_{x_j}),$$

where $Beta(\cdot)$ is the Beta distribution, $\theta_v$ is the probability that class $t$ is generated in vertex $v$, and $\beta$ is the base probability of $\theta_v$.

4.2 Inference

An observation is a link from vertex $j$ to vertex $i$; that is, a customer in restaurant $j$ orders dish $i$. We estimate the menu (i.e., class), $z_{ji}$, being used by this customer. The $z_{ji} = t$ indicates that the class of vertex $j$ is class $t$ when vertex $j$ generates a link to vertex $i$. We use Gibbs sampling to infer the posterior distribution over latent class $z_{ji}$, similar to HDP [16].

$v$ is an index of restaurants (vertices).

$t$ is an index of menus (classes).

$k$ is an index of tables.

Let $n_{v,tk}$ be the number of customers in restaurant $v$ sitting at table $k$ with menu $t$.

Let $m_{vtk}$ be the number of tables in restaurant $v$ serving menu $t$.

A dot denotes marginal counts; for example, $n_{v,tk}$ is the number of customers sitting at table $k$ in restaurant $v$. $n_{vtk}$ is the number of customers being served menu $t$ in restaurant $v$ (number of occurr-
ferences of class \( t \) in vertex \( v \), and \( m_v \) is the number of tables in restaurant \( v \). A customer eating dish \( i \) on menu \( t \) in restaurant \( j \) is represented by \( z_{ji} = t \). The number of customers eating dish \( i \) on menu \( t \) in all restaurants is represented by \( n_{ti} \), which is the number of links to vertex \( i \) for class \( t \).

The posterior probability over class is calculated using Bayes theorem.

\[
p(z_{ji} | z^{-ji}, A_{ji} = 1, v_j, A^{-ji}, v^{-j}) = p(A_{ji} = 1 | z_{ji}, A^{-ji}, z^{-ji}) p(z_{ji} | z^{-ji}, v_j, v^{-j})
\]

\[
p(A_{ji} = 1 | z_{ji} = t, A^{-ji}, z^{-ji}) = \frac{n_{ji} - d + \tau}{n_{ji} + V_t},
\]

where \( n_{ji} = \sum_{v \neq j} A_{vi} \delta(z_{vi} = t) \) means the number of customers eating dish \( i \) on menu \( t \) with a customer eating dish \( i \) in restaurant \( j \) removed from the calculation.

\[
p(z_{ji} = t | z^{-ji}, v_j, v^{-j}) = \begin{cases} 
\frac{n_{ji} - d + \tau}{n_{ji} + V_t} & \text{if } t \text{ previously used} \\
\frac{n_{ji} + \alpha}{n_{ji} + \alpha + \tau} & \text{if } t \text{ is new}
\end{cases}
\]

The \( \beta \) is sampled using

\[
(\beta_1, \cdots, \beta_T, \beta_{\text{new}}) \sim \text{Dir}(m_1, \cdots, m_T, \tau, \gamma),
\]

where \( m_{vt} \) is obtained using functions AddCustomer and RemoveCustomer (see Appendix A) and \( \alpha \) and \( \gamma \) are estimated by auxiliary variable sampling (see Appendix B).

We identify class \( z_{ji} \) of vertex \( j \) linking to vertex \( i \) as

\[
z_{ji} = \arg\max_i (P(z_{ji} | A_{ji}, v_j)).
\]

We cluster vertex \( j \) in a graph by using this estimated class. Different classes can be assigned to the same vertex depending on the vertices to which the vertex links. Therefore, a vertex are not always be determined to belong to only one class; i.e., there is soft clustering.

### 4.3 Modeling Degree of Graph Using PYP

We model the selection of vertices that link to another vertex. That is, we select vertex \( v \) from a probabilistic distribution over vertices and generate a link from \( v \) to another vertex by using the process described in Section 4. In the restaurant representation, we model how a customer selects and enters a restaurant (vertex). The selection is related to vertex degree because the number of customers selecting a vertex is equal to the degree of the vertex. The modeling of vertex degree is useful for understanding the graph structure.

The probabilistic model of graph degree is based on PYP [14] because it generates a power-law distribution. We assume the vertex space size is \( V \). Let \( G \) be a random probability measure over vertices; \( G(v) \) is the probability that vertex \( v \) is selected to link to another vertex. We select a vertex \( v \) from \( G \) and link from the selected vertex to another vertex by using the process described in Section 4.1. \( G \) is distributed in accordance with PYP:

\[
G \sim \text{PY}(d, \lambda, U),
\]

where \( d(0 \leq d < 1) \) is called the discount parameter \( \lambda \). \( \beta > -d \) is called the concentration parameter \( \beta \) and \( U \) denotes the uniform distribution over vertices, s.t. \( U(v) = \frac{1}{d} \).

The probability of the \( n \)-th selection of vertex \( x_n \), given that some selected vertices \( x_{n-1} \), is given by

\[
p(x_n = v | x_{n-1}) = \begin{cases} 
\frac{n_{v-d} - d}{n_{v-d} + \lambda} & \text{if } v \text{ previously selected} \\
\frac{n_{v-d} + \lambda}{n_{v-d} + \lambda + d} & \text{if } v \text{ is new selected}
\end{cases}
\]

5. EXPERIMENT

In an evaluation experiment, we investigated the syntactic structures found in sentences. In particular, we took a subject-predicate structure as a syntactic structure with a noun as the subject and a verb as the predicate. The observed data were pairs of a subject (noun) and a predicate (verb) extracted from a corpus. The extraction was done using a head-driven phrase structure grammar (HPDG) parser called Enju1. We evaluated the proposed model by measuring its performance for graph clustering based on WordNet similarities. We compared the results of our model with those of the DP-Newman model, the infinite relational model (IRM), and the Markov cluster algorithm (MCL).

1http://www-tsujii.is.s.u-tokyo.ac.jp/enju
5.1 Datasets

We used the Reuters corpus\(^1\) and the LISA corpus\(^2\). LISA stands for "Library and Information Science Abstracts".

We considered two graph structures from Reuters. "Reuters1" is a graph constructed from the pairs in the corpus with frequencies of over one. That is, if a pair was observed even once, we made an edge between the subject and predicate. "Reuters2" is a graph constructed from the pairs in the corpus with frequencies of over two. "LISA" is a graph constructed from the pairs in the corpus with frequencies over one.

The number of links (extracted pairs) of a subject and a predicate were 33,054 for Reuters1 and 19,202 for Reuters2. The number of subjects was 4,438, and that of predicates was 2,583 for Reuters1. The number of subjects was 1,991, and that of predicates was 1,207 for Reuters2. The number of links (extracted pairs) of a subject and a predicate was 3,401 for LISA. The number of subjects was 858, and that of predicates was 838 for LISA.

5.2 WordNet Similarity

We used the WordNet similarity of words as used by Hagiwara et al. \(^9\) to evaluate our model. WordNet similarity is based on the thesaurus tree structure in WordNet. The similarity between word sense \(w\) and \(v\) is calculated as follows.

Assume that word \(w\) has synset \(w_i\) \((i = 1, 2, \ldots, n)\) and word \(v\) has synset \(v_j\) \((j = 1, 2, \ldots, m)\). Let \(d_i\) be the depth of node \(w_i\) from the root node. Let \(d_j\) be the depth of node \(v_j\) from the root node. Let \(\delta_{\text{mca}}\) be the maximum depth of common ancestors.

\[
\text{WordNetSim}(w, v) = \max_{i,j} \frac{2\delta_{\text{mca}}}{d_i + d_j + \delta_{\text{mca}}} \quad (30)
\]

The idea behind this similarity was derived from Lin’s method \(^11\). Figure 6 shows a calculation example of WordNet Similarity.

5.3 Evaluation Scheme

We evaluated how well each algorithm performed by measuring the quality of output clusters in terms of WordNet similarities. We used a label-prediction-based accuracy measure because it can be easily applied to both soft- and hard-clustering algorithms. For target word \(w\), we predicted whether other words were similar or dissimilar to it. The set of target words, \(\mathcal{W}\), was constructed using pairs of nouns and verbs included in WordNet in a corpus. We selected a set of words for which the similarity to target word \(w(\in \mathcal{W})\) was above a threshold and labeled those words +. We also selected a set of words for which the similarity to the target word was below a threshold and labeled those words −. The number of words, \(N(w)\), in each set for each target word \(w\) was equal.

To predict these correct labels from the output of the algorithms, we created a similar word cluster, \(C(w)\), for \(w\) by merging all the clusters generated by the algorithm that contained \(w\) into a new cluster. If word \(x\) is in \(C(w)\), the label of \(x\) is predicted to be +; otherwise, it is predicted to be −. Therefore, the accuracy for word \(w\) is given by

\[
\text{accuracy}(w) = \frac{\sum_{x \in C(w)} \delta_+(x)}{2N(w)}, \quad (31)
\]

where \(\delta_+(x) = 1\) if \(x \in C(w)\) and the correct label of \(x\) is + or \(x \notin C(w)\) and the correct label of \(x\) is −; otherwise, it is 0. The total accuracy was calculated by taking the average accuracy for all words:

\[
\text{accuracy}_{\text{ave}} = \frac{1}{|\mathcal{W}|} \sum_{w \in \mathcal{W}} \text{accuracy}(w). \quad (32)
\]

We set the threshold for WordNet similarity to 0.8. We calculated and averaged the accuracy separately for three domains in accordance with the degree distribution in the graph (as illustrated in Figure 7). Words in the target word set were ranked in accordance with the degrees of their corresponding vertices in the graph. We roughly separated the target word set into three domains: domain 1 is the set of words with high degrees, domain 2 is the set of words with medium-level degrees, and domain 3 is the set of words with low degrees. This separation was done to clarify the type of words for which each algorithm is good from the viewpoint of vertex degrees (i.e., the number of adjacent words). As described in the next section, this separation enables clear discrimination among the algorithms which algorithms perform well for high-degree vertices and which ones perform well for low-degree ones. Table 1 lists the base information for each dataset with respect to the number of words and average degree in each domain.

---

\(^1\)http://www.daviddlewis.com/resources/testcollections/reuters21578/

\(^2\)http://www.library.pitt.edu/articles/database_info/lisa.html
Figure 7: Three Domains in Accordance with Degree Distribution

### Table 1: Base Information for Datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>domain1</th>
<th>domain2</th>
<th>domain3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reuters1 (nouns)</td>
<td>41</td>
<td>1212</td>
<td>2925</td>
</tr>
<tr>
<td>no. of words</td>
<td>124.1</td>
<td>17.4</td>
<td>1.8</td>
</tr>
<tr>
<td>average order</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reuters1 (verbs)</td>
<td>25</td>
<td>749</td>
<td>1808</td>
</tr>
<tr>
<td>no. of words</td>
<td>226.7</td>
<td>28.8</td>
<td>2.3</td>
</tr>
<tr>
<td>average order</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reuters2 (nouns)</td>
<td>17</td>
<td>520</td>
<td>1255</td>
</tr>
<tr>
<td>no. of words</td>
<td>75.2</td>
<td>10.4</td>
<td>1.6</td>
</tr>
<tr>
<td>average order</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reuters2 (verbs)</td>
<td>12</td>
<td>349</td>
<td>845</td>
</tr>
<tr>
<td>no. of words</td>
<td>113.4</td>
<td>16.9</td>
<td>1.9</td>
</tr>
<tr>
<td>average order</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LISA (nouns)</td>
<td>8</td>
<td>249</td>
<td>601</td>
</tr>
<tr>
<td>no. of words</td>
<td>48.9</td>
<td>8.7</td>
<td>1.4</td>
</tr>
<tr>
<td>average order</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LISA (verbs)</td>
<td>8</td>
<td>243</td>
<td>587</td>
</tr>
<tr>
<td>no. of words</td>
<td>79.6</td>
<td>8.3</td>
<td>1.3</td>
</tr>
<tr>
<td>average order</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 5.4 Results

We compared the results for our model with those for the DP-Newman model, the IRM, and the MCL algorithm. We also considered frequency-weighted versions of the DP-Newman model and the proposed model. Since the DP-Newman model and the proposed model are based on a mixture of multinomial distributions, they can be easily modified to take integers greater than 1 as the value of $A_{ij}$. Therefore, they can use the frequency of pairs of a subject and a predicate as a weight for links. If the value of $A_{ij}$ is $a$, it means that the link between vertex $j$ and vertex $i$ occurred $a$ times where the occurrences are treated individually and assigned different class variables. In restaurant representation, the frequency (weight) indicates the number of customers ordering the same dish. We assumed that such customers are individually seated to tables. "DP-Newman-f" and "Proposed-f" indicate frequency-weighted models.

Hereafter, we call MCL, IRM, and DP-Newman(-f) the "baselines".

A comparison with respect to accuracy is shown in Figures 8-13. The proposed algorithm performed especially well for domain-2 words and not particularly well for domain-1 words. In contrast, the baselines generally performed best for domain-1 words although not as well as the proposed algorithm. Moreover, their performances did not differ drastically across the three domains.

We think that the MCL’s poor performance was due to the fact that it was a model proposed for extracting community graphs, which are assortative graphs in general. Therefore, it might not work well for disassortative graph structures like predicate-argument graphs used in our study.

The IRM performed especially poorly for domain 3. This is because the vertices in domain 3 tended to be categorized into a large cluster by IRM. DP-Newman-f performed a bit better than DP-Newman in some cases thanks to its use of frequency information.

The proposed model outperformed the baselines for domains 2 and 3 for all corpora and performed somewhat worse for domain 1. The vertices in domain 1 tended to have multiple classes because domain 1 is the set of vertices with high degrees. In other words, many vertices in domain 1 tended to be categorized into the same class when the output of the proposed algorithm was used because it is a soft clustering algorithm. We solved this problem by introducing link weighting (Proposed-f). Although Proposed-f seats customers ordering the same dish individually to tables, the customers sit in the almost same table in our observation. It means that one menu on a particular table had large influence on the selection of the dish. It reduced the number of classes assigned to each link, resulting in improvement of accuracy in domain-1. One inexplicable finding is the lower performance of Proposed-f for verb clustering for the LISA corpus. In this clustering, the estimated number of classes with Proposed-f was smaller than with Proposed. This would explain the lower performance of Proposed.

Finally, we turn to discount parameter $d$, which was described in Section 4.3. Estimating $d$ helps us gain a better understanding of the obtained graph structures, especially the distribution of vertex degrees in a graph. We explained in Section 4.3 that if $d$ approaches one, the distribution of vertex degrees in a graph becomes a power-law distribution.

For example, we estimated $d = 0.0997$ for a graph of nouns and $d = 0.99946$ for a graph of verbs in Reuters1. These mean that the distribution of vertex degrees in the graph for Reuters1 had a power-law distribution.

Moreover, we can calculate the probability that a new word (verb) is selected using Eq. (29). For example, in Reuters1, the probability for the new selection of a noun was 0.1333; in contrast, the probability for the new selection of a verb was 0.0823. This means that it is easy for a new noun to be selected (generated) compared with a new verb. This is a reasonable result given that new nouns are generally generated more frequently than verbs because nouns tend to indicate concrete entities and verbs tend to indicate more abstract concepts, and new concrete entities appear more frequently than new abstract concepts.

### 5.5 Extracted Relationships between Words

Finally, we present several examples of relationships between words extracted from the Reuters corpus. Table 2 shows examples of extracted relationships and their linking vertices with respect to estimated classes.

For example, for the link to "land" relationship, the observed vertices linking to "land" are seven words. Each word is categorized into one of two classes, 25 or 32. Class 25 indicates "physically landing things", and Class 32 indicates "conceptually landing things". The words "flight", "ball", "airport", and "aircraft" are all related to landing. For the link to "dog" relationship, the observed vertices linking to "dog" are four words. The words
in class 31, “worry” and “fear”, have a similar meaning. Those in class 107, “stock” and “performance”, are seemingly unrelated. However, when they are used with □ dog □ (“stock dog” and “performance dog”), “stock” and “performance” have meanings similar to “working”.  

6. CONCLUSION
We investigated the syntactic structures found in sentences and used them as contextual information. In particular, we took the subject-predicate structure as a syntactic structure and constructed a disassortative graph structure. We also developed a probabilistic model for generating the graph structure. This model performed better than other graph clustering algorithms in terms of WordNet similarities. It can also be used in other domains where the data can be represented by a graph structure. Future work includes investigating such applications.

7. ACKNOWLEDGEMENTS
This research was funded in part by a MEXT Grant-in-Aid for Scientific Research on Priority Areas “i-explosion” in Japan.

8. REFERENCES


APPENDIX

A. ADD CUSTOMER AND REMOVE CUSTOMER

AddCustomer(v, t) adds a new customer using menu t into restaurant v as follows.
- Using proportion \( \frac{n_{vt} + \alpha}{n_{vt} + b + \alpha} \), sit a customer at table \( k \) in restaurant \( v \); that is, increment \( n_{vkt} \).
- Using proportion \( \frac{\alpha}{n_{vkt} + \alpha} \), sit a customer at a new table \( k^{\text{new}} \); that is, increment \( m_{kt} \) and set \( n_{vkt} = 1 \).

RemoveCustomer(v, t) removes a customer using menu t from restaurant v as follows.
- Using proportion \( \frac{n_{vt} + \alpha}{n_{vt} + b + \alpha} \), remove a customer from table \( k \) in restaurant \( v \); that is, decrement \( n_{vkt} \).
- If table \( k \) becomes unoccupied, remove table \( k \) from restaurant \( v \); that is, decrement, \( m_{kt} \).

B. ESTIMATE PARAMETERS OF HDP

The concentration parameters \( \alpha \) and \( \gamma \) of HDP can be estimated by auxiliary variable sampling [6, 16]. The \( \alpha \) is estimated by first sampling auxiliary variables \( w_v \) and \( y_v \) for each restaurant (vertex) \( v \) given \( \alpha \).

\[
\begin{align*}
w_v & \sim Beta(\alpha + 1, n_{w_v}) \quad (33) \\
y_v & \sim Bern\left(\frac{\alpha + 1}{n_{y_v}}\right) \quad (34)
\end{align*}
\]

where \( Bern(\cdot) \) is the Bernoulli distribution. Next, \( \alpha \) is sampled given \( w \) and \( y \).

\[
\alpha \sim Gamma(a_\alpha + \sum_v (m_{v\cdot} - y_v), b_\alpha - \sum_v \log w_v)(35)
\]

where \( Gamma(\cdot) \) is the gamma distribution, and \( a_\alpha \) and \( b_\alpha \) are hyper parameters.
\[ \gamma \text{ is estimated as follows by first sampling auxiliary variable } \kappa, \text{ and then sampling } \gamma. \]

\[ \kappa \sim \text{Beta}(\gamma, \sum_{v} m_{v}) \quad (36) \]

\[ \gamma \sim \text{Gamma}(a_{\gamma} + T - 1, b_{\gamma} - \log \kappa), \quad (37) \]

where \( T \) is the number of generated classes (kinds of menu), and \( a_{\gamma} \) and \( b_{\gamma} \) are hyper parameters. We set all hyper parameters to 1.

### C. ESTIMATE PARAMETERS OF PYP

Discount parameter \( d \) and concentration parameter \( \lambda \) of PYP can be estimated by auxiliary variable sampling similar to that used by Escobar and West [6].

First, auxiliary variables \( c_{vu}, \ w, \) and \( y_{k} \) are sampled for each restaurant (vertex) \( v \).

\[ c_{vu} \sim \text{Bern}(\frac{u - 1}{u - d})(u = 1, 2, \ldots, n_{v} - 1) \quad (38) \]

\[ w \sim \text{Beta}(\lambda + 1, \sum_{v} n_{v} - 1) \quad (39) \]

\[ y_{k} \sim \text{Bern}(\frac{\lambda}{\lambda + dk})(k = 1, 2, \ldots, s - 1). \quad (40) \]

Next, given the auxiliary variables, \( \lambda \) and \( d \) are sampled.

\[ d \sim \text{Beta}(a_{d} + \sum_{k=1}^{s-1} (1 - y_{k}), b_{d} + \sum_{v} \sum_{u=1}^{n_{v}-1} (1 - c_{vu})) \quad (41) \]

\[ \lambda \sim \text{Gamma}(a_{\lambda} + \sum_{k=1}^{s-1} y_{k}, b_{\lambda} - \log w), \quad (42) \]

where \( a_{d}, \ b_{d}, \ a_{\lambda}, \) and \( b_{\lambda} \) are hyper parameters. We set all hyper parameters to 1.

### D. PROPOSED MODEL WITH FREQUENCY

The posterior probability over class is calculated instead of Eq (22)-(25) as follows. We introduce \( A_{jci} \) that takes 1 if customer \( c \) eats dish \( i \) in restaurant \( j \), and otherwise 0; i.e., \( A_{ji} = \sum_{c} A_{jci} \). The \( z_{jci} = t \) indicates that customer \( c \) eating dish \( i \) in restaurant \( j \) uses menu \( t \).

\[
p(z_{jci} | z^{-jci}, A_{jci} = 1, v_{j}, A^{-jci}, v^{-j}) \]

\[
= p(A_{jci} = 1 | z_{jci}, A^{-jci}, z^{-jci})p(z_{jci} | z^{-jci}, v_{j}, v^{-j}) \quad (43) \]

\[
p(A_{jci} = 1 | z_{jci} = t, A^{-jci}, z^{-jci}) = \frac{n_{ji}^{-jci} + \tau}{n_{ji}^{-jci} + \nu_{j} + \nu}, \quad (44) \]

where \( n_{ji}^{-jci} = \sum_{v \neq j, c \neq c} A_{vei} \delta(z_{vei} = t) \) means the number of customers eating dish \( i \) on menu \( t \) with customer \( c \) eating \( i \) in restaurant \( j \) removed from the calculation.

\[
p(z_{jci} = t | z^{-jci}, v_{j}, v_{j}) = \begin{cases} 
\frac{n_{ji}^{-jci} + \alpha}{n_{ji}^{-jci} + \alpha} & \text{if } t \text{ previously used} \\
\frac{n_{ji}^{-jci} + \alpha}{n_{ji}^{-jci} + \alpha} & \text{if } t \text{ is new} 
\end{cases} \quad (45) \]
Figure 8: Accuracy of Noun Clustering for Reuters1

Figure 9: Accuracy of Verb Clustering for Reuters1

Figure 10: Accuracy of Noun Clustering for Reuters2

Figure 11: Accuracy of Verb Clustering for Reuters2

Figure 12: Accuracy of Noun Clustering for LISA

Figure 13: Accuracy of Verb Clustering for LISA