Bayesian Network Models of Subcategorization and Their MDL-Based Learning from Corpus

Takashi Miyata Takehito Utsuro Yuji Matsumoto
Graduate School of Information Science, Nara Institute of Science and Technology
Takayama-cho 8916-5, Ikoma-shi, Nara 630-01, Japan
Email: {takashi,utsuro,matsu}@is.aist-nara.ac.jp
Phone: +81-743-72-5245, Fax: +81-743-72-5249

1997.10.15

Abstract

We formalize two probabilistic models of verbal subcategorization based on the Bayesian network which treat dependencies among case slots as well as the class generalization of adjunct/argument nouns. We implement algorithms for obtaining locally optimal models based on the MDL principle and evaluate the obtained models in terms of syntactic disambiguation task. In the task, we compare the two models: (i) a model combining the two models each of which separately considers dependencies among case slots and those among class generalizations, and (ii) a model which considers the two types of dependencies together. The former model can detect dependencies among class generalizations of the adjunct/argument nouns in contrast to (Li and Abe, 1996), which contribute to increasing the performance of the task.

1 Introduction

Availability of large corpora in recent years has made it possible to automatically acquire various kinds of linguistic knowledge. Lexical/semantic collocational knowledge of verbs is one of these useful knowledge. It can be applied to ranking parses in syntactic analysis, clustering nouns, and so forth. In general, when learning lexical/semantic collocational knowledge of verbs from corpora, it is necessary to consider the two issues: case markers that the verb has and class generalizations of the nouns marked by the case marker. Case markers and class generalizations of the nouns depend not only on the verb but also on each other. So there are three kinds of dependencies:

- Dependency between case markers:

  For example, it is generally true that two cases marked by the same case marker do not appear together.
  "*He arrived at Tokyo at Kyoto."

- Dependency between the case marker and the class generalization:

  For example, it is generally true that the subject of motion verb is apt to be in the class animate.
  "John walks to the station."

- Dependency between the class generalizations:

  For example, it is generally true that when the noun marked by the preposition 'to' of the verb 'write' is in the class human, the subject of the verb is also in the class human.
  "John wrote to Kim every week."

There are various researches on extracting and/or making use of lexical/semantic collocational knowledge of verbs. (Resnik, 1993) and (Li and Abe, 1995) studied the methods to find an optimal abstraction level of an argument noun in a tree-structured thesaurus and evaluated the obtained abstraction levels by their usability in syntactic disambiguation. Their methods, however, are limited to only one argument. (Li and Abe, 1996) studied a method for learning dependencies between case slots and evaluated the discovered dependencies in the syntactic disambiguation task. They first obtained optimal abstraction levels of the argument nouns by the method proposed in (Li and Abe,

1995), and then tried to discover dependencies between the class-based case slots. They reported that dependencies were discovered only at the slot-level and not at the class-level. (Haruno, 1995) applied an information-theoretic data compression technique to corpus-based case frame learning, and proposed a method of finding case frames of verbs as compressed representation of verb-noun collocational data in a corpus. The work concentrated on the extraction of declarative representation of case frames and did not consider their performance in sentence parsing. (Utsuro and Matsumoto, 1997) proposed a method of learning probabilistic subcategorization preference considering dependencies among case markers and class generalization of the case-marked nouns based on some statistical heuristics. Although their formalization was not a probabilistic one, they evaluated the obtained dependencies by the syntactic disambiguation task and achieved 72% – 97% accuracy. (Utsuro et al., 1997) reformalized this model based on Maximum Entropy model learning method (Berger et al., 1996).

In contrast to those previous works, this paper proposes a method for learning a probabilistic model of subcategorization preference of verbs. We adopt a Bayesian network (Pearl, 1988), which encodes probabilistic relationships among a set of variables. Compared with the Maximum Entropy model learning approach, Bayesian network approach has an advantage that the result of model learning is easier for a human to understand. This is because, in the Maximum Entropy model learning method, a probabilistic model is represented by a number of pairs of a feature and a parameter, and a feature corresponds to a combination of several factors in a Bayesian network.

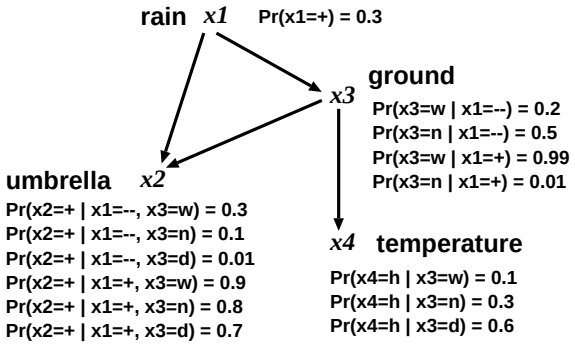
When constructing a Bayesian network from raw data, it is important to make a trade-off between accuracy and usefulness of the constructed network. More complex networks allow for more accurate models, but at the same time such models are computationally and conceptually more difficult to use. In addition, the raw data itself is only an approximated picture of the true underlying distribution. All we can do is to approximate the underlying distribution from the raw data, so it seems only reasonable to prefer approximations that are more useful. To make this trade-off, it has been proposed (Bouckaert, 1993; Suzuki, 1993; Lam and Bacchus, 1994; Suzuki, 1996) to use a well-studied formalism: Rissanen's minimum description length (MDL) principle (Rissanen, 1984). The MDL principle makes it possible to learn a less complex network if that network is sufficiently accurate, and at the same time, it is capable of learning a complex network if no simpler network is sufficiently accurate. In this paper, we use the MDL principle as a trade-off between accuracy and usefulness of the constructed Bayesian network.

Most previous works on constructing a Bayesian network based on the MDL principle (Bouckaert, 1993; Suzuki, 1993; Lam and Bacchus, 1994; Suzuki, 1996) dealt only with finding appropriate dependencies between variables. In the task of subcategorization preference learning, however, we have to search for a noun class generalization level as well as dependencies among cases. We realize the process of searching for locally optimal noun class generalization levels by (1) dividing one of the element of the domain of a vertex in the Bayesian network into the set of its subclasses or (2) splitting of a vertex itself in the Bayesian network, where the resulting vertices are the ones that are subordinate to the original vertex. In the proposed method, a Bayesian network is constructed according to a *hill-climbing algorithm*. In each step of the Bayesian network construction, the change of the description length of the network is calculated for every possible *linking* of two vertices, *dividing* an element of the domain of a vertex, or *splitting* a vertex. Then, one of these operations with the greatest reduction in the description length is chosen. The construction algorithm terminates if none of the operations on any vertices can reduce the description length.

The rest of the paper is organized as follows: Section 2 and 4 give a general explanation on Bayesian network and MDL principle which is used as a guideline of learning. Section 3 shows a model of verbal subcategorization frames and extends it to probabilistic one. Section 5 explains our model construction algorithm. Section 6 describes the syntactic disambiguation task for evaluation and the results of the evaluation. Finally Section 7 concludes this paper.

2 Bayesian Network

Notation and Example First we begin with a general explanation of the Bayesian network. Bayesian network is a directed acyclic graph (DAG) where each vertex represents the event (i.e. the random variable or *attribute*) and each edge represents the dependency between the two events. When there is an edge from vertex u to vertex v, u is called a *parent* of v and v is called a *child* of u. Each vertex is assigned a set of *conditional probabilities*. Figure 1 shows an example of the Bayesian network that has the four vertices: x_1 for raining, x_2 for people putting up umbrellas, x_3 for the condition of the ground, and x_4 for the temperature. The domain of each attribute is displayed in the figure. For example, $\Pr(x_3 = w \mid x_1 = +)$, which is one of the conditional probabilities at the vertex x_3, means the conditional probability of the ground being wet ($x_3 = w$) under the condition that it is raining ($x_1 = +$). The reason why the conditional probability $\Pr(x_3 = d \mid x_1 = +)$ is omitted is that the constraint $\sum_{a \in \{w,n,d\}} \Pr(x_3 = a \mid x_1 = +) = 1$ must be satisfied. Each vertex x_i has the (# combinations of the variations of the parents of x_i) $\times$ ((# variations of



attribute	domain
x_1 (whether it is raining)	$\{+, -\}$
x_2 (whether people put up umbrellas)	$\{+, -\}$
x_3 (the condition of the ground)	$\{w, n, d\}$
x_4 (the temperature)	$\{h, c\}$

Figure 1: Example of Bayesian Network

$x_i) - 1)$ entries. Appendix A formally defines Bayesian network and some notations used in the description of our model construction algorithm. Note that the more edges a network has, the more fit to a given data the network has, and the more entries of conditional probabilities the network must contain. Finding an optimal point in this trade-off between fit to the data and the number of variations of conditional probabilities is formally described in the MDL principle, which is explained in Section 4.

Calculation of Joint Probability According to the Bayes' law, the probability of each attribute x_i being a certain value a_i, which is called the *joint probability*, can be calculated as follows:

$$\begin{aligned}\Pr(x_1 = a_1,\, x_2 = a_2,\, \ldots,\, x_n = a_n) \;=\; & \Pr(x_1 = a_1) \\ & \times \Pr(x_2 = a_2 \mid x_1 = a_1) \\ & \times \Pr(x_3 = a_3 \mid x_1 = a_1,\, x_2 = a_2) \\ & \quad \ldots \\ & \times \Pr(x_n = a_n \mid x_1 = a_1,\, x_2 = a_2, \ldots,\, x_{n-1} = a_{n-1}) \qquad (1)\end{aligned}$$

For example, the joint probability that "it is not raining, people put up umbrellas, the condition of the ground is normal, and the temperature is high" is calculated by the Bayesian network in Figure 1 as follows:

$$\begin{aligned}&\Pr(x_1 = -,\, x_2 = +,\, x_3 = n,\, x_4 = h) && (2)\\ &= \Pr(x_1 = -)\Pr(x_3 = n \mid x_1 = -)\Pr(x_2 = + \mid x_1 = -,\, x_3 = n)\Pr(x_4 = h \mid x_1 = -,\, x_3 = n,\, x_2 = +) && (3)\\ &= \Pr(x_1 = -)\Pr(x_3 = n \mid x_1 = -)\Pr(x_2 = + \mid x_1 = -,\, x_3 = n)\Pr(x_4 = h \mid x_3 = n) && (4)\\ &= (1 - 0.3) \times 0.5 \times 0.1 \times 0.3 = 0.0105 && (5)\end{aligned}$$

Note that the equation $\Pr(x_4 = h \mid x_1 = -,\, x_3 = n,\, x_2 = +) = \Pr(x_4 = h \mid x_3 = n)$ holds in this Bayesian network, since there are no edges neither from x_1 nor from x_3 to x_4. Appendix A describes the calculation of the joint probability in general cases.

3 Subcategorization Preference

We adopt the two kinds of attributes that characterize the subcategorization frame of a given verb: *sorts* of the case markers and *class generalizations* of the nouns marked by case markers. We put an assumption that each class generalization of the noun in a case is disjoint in order to avoid trivial dependencies between them.[1] In this paper, we describe probabilistic model learnings of subcategorization of Japanese verbs, but the method is applicable to any other languages in which case markers are syntactically distinguished.

[1] When noun class C_1 is subordinate to noun class C_2, "subordinate to C_1" entails "subordinate to C_2" due to the transitivity of subordinate-relationship. This means the attribute "subordinate to C_1" always depends on attribute "subordinate to C_2."

3.1 Notations and Definitions

First we introduce the following two notations and a definition for describing our algorithm, which concern a kind of feature-structure. Next we describe how this kind of feature-structure can be used to constrain the possible patterns of case-marked argument/adjunct nouns of a verb.

Verb-Noun Collocation A *verb-noun collocation* is a feature structure for the collocation of a verb and its argument/adjunct nouns. A verb-noun collocation $e^{(v)}$ of verb v is represented as a feature structure consisting of all the pairs of co-occurring case markers p_i and classes c_i of the case-marked nouns in the thesaurus[2]:

$$e^{(v)} = \begin{bmatrix} p_1 : & c_1 \\ p_2 : & c_2 \\ & \vdots \\ p_k : & c_k \end{bmatrix} \tag{6}$$

For example, the following feature structure represents collocation "Kodomo-ga kouen-de hana-wo tumda. (A child picked flowers in the park.":

$$e^{(\text{tumu/pick})} = \begin{bmatrix} \textit{ga}\ (\text{NOM}) : & \text{kodomo (child)} \\ \textit{de}\ (\text{`in'}) : & \text{kouen (park)} \\ \textit{wo}\ (\text{ACC}) : & \text{hana (flowers)} \end{bmatrix} \tag{7}$$

We assume that a *thesaurus* is a tree structured type hierarchy in which each node represents a semantic class. We introduce binary relation $\preceq_c$ as the superordinate-subordinate relation between two classes in the thesaurus; $c_1 \preceq_c c_2$ means that c_1 is subordinate to c_2.

Subcategorization Frame A *subcategorization frame* of verb v is a feature structure that restricts the possible collocations between v and its case-marked argument/adjunct nouns. A subcategorization frame $s^{(v)}$ is represented as follows:

$$s^{(v)} = \begin{bmatrix} p_1 : & \begin{bmatrix} \text{pol} : & \text{pol}_1 \\ \text{cnt} : & c_1 \end{bmatrix} \\ p_2 : & \begin{bmatrix} \text{pol} : & \text{pol}_2 \\ \text{cnt} : & c_2 \end{bmatrix} \\ & \vdots \\ p_l : & \begin{bmatrix} \text{pol} : & \text{pol}_l \\ \text{cnt} : & c_l \end{bmatrix} \end{bmatrix} \tag{8}$$

Each pol_i is a *polarity* for the case marker p_i, whose value is + (positive) or − (negative). They mean whether an argument/adjunct noun marked by p_i can co-occur with the verb v. Sense restriction c_i of each case-marked argument/adjunct noun is a class at an arbitrary level of the thesaurus. The case-marked class restriction $\left[p_i : \begin{bmatrix} \text{pol} : & - \\ \text{cnt} : & c_i \end{bmatrix} \right]$ with negative polarity in subcategorization frame $s^{(v)}$ means that those argument/adjunct nouns which are subordinate to c_i with case marker p_i cannot appear in the verb-noun collocation of v. For example, the following feature structure represents a subcategorization frame of verb *tumu* (pick) that restricts the nouns in the cases marked by *ga* (NOM), *he* (DAT), and *ni* (DAT):

$$s^{(\text{tumu})} = \begin{bmatrix} \textit{ga} : & \begin{bmatrix} \text{pol} : + \\ \text{cnt} : \mathsf{human} \end{bmatrix} \\ \textit{he} : & [\ \text{pol} : - \ ] \\ \textit{ni} : & [\ \text{cnt} : \mathsf{human} \ ] \end{bmatrix} \tag{9}$$

We also denote the result of the unification of two subcategorization frames s_1 and s_2 as $s_1 \wedge s_2$.

[2] Here we ignore sense ambiguities of case-marked nouns. We briefly mention how we deal with sense ambiguities of case-marked nouns in the current implementation in Section 6.

Subsumption Relation We introduce *subsumption relation* $\sqsubseteq$ between verb-noun collocation $e^{(v)}$ and subcategorization frame $s^{(v)}$ of verb v as follows:

$e^{(v)} \sqsubseteq s^{(v)}$ iff

1. for each case marker p_i in $s^{(v)}$ that does not have negative polarity:
   (a) when p_i has positive polarity in $s^{(v)}$, there exists the same case marker p_i in $e^{(v)}$ and
   (b) when the noun class of p_i is c_i^s and $e^{(v)}$ has the same case marker p_i in $s^{(v)}$, then the noun class c_i^e of p_i in $e^{(v)}$ satisfies $c_i^e \preceq_c c_i^s$.
2. for each case marker p_i in $s^{(v)}$ that has negative polarity:
   (a) $e^{(v)}$ does not have the same case marker p_i or
   (b) $s^{(v)}$ has the noun class c_i^s in p_i but the noun class c_i^e in p_i in $e^{(v)}$ does not satisfy $c_i^e \preceq_c c_i^s$.

Note that this definition of the subsumption allows the case where verb-noun collocation $e^{(v)}$ does not have the same case marker as in subcategorization frame $s^{(v)}$. For example, subsumption relation $e^{(\text{tumu})} \sqsubseteq s^{(\text{tumu})}$ on $e^{(\text{tumu})}$ in Equation 7 and $s^{(\text{tumu})}$ in Equation 9 holds.[3]

In the following explanations we focus on a certain verb and omit v of verb-noun collocations $e^{(v)}$ and subcategorization frames $s^{(v)}$ for the simplicity.

3.2 Probabilistic Models of Subcategorization Preference

We measure the *subcategorization preference* of verb-noun collocation e by the probability $\Pr(e \sqsubseteq s)$ that e is observed satisfying given dependencies restricted with the subcategorization frame s. Since the subcategorization frame s can be regarded as the unification of the set of the 'elementary' frames $[p_i : v_i]$, this probability is calculated as follows:

$$\Pr(e \sqsubseteq s) = \frac{\Pr(e \sqsubseteq [p_1 : v_1],\ e \sqsubseteq [p_2 : v_2],\ \ldots,\ e \sqsubseteq [p_n : v_n])}{L(c_1)\,L(c_2)\,\ldots\,L(c_n)} = \prod_{i=1}^{n} \frac{\Pr(e \sqsubseteq [p_i : v_i] \mid e \sqsubseteq S_i)}{L(c_i)} \tag{10}$$

where $s = \bigwedge_{i=1}^{n} [p_i : v_i]$, c_i is the class generalization in v_i, and $L(c_i)$ is the number of the leaf nodes subordinate to c_i in the thesaurus. S_i is the unification of arbitrary set of $[p_j : v_j]$, which is determined based on the structure of a given Bayesian network. The numerator of Equation 10 is equivalent to the joint probability of e being subsumed by 'elementary' subcategorization frames $[p_i : v_i]$. The factor $(L(c_i))^{-1}$ in Equation 10 represents the probability of each word in v_i being generated among the set of words in class c_i.[4]

According to the types of the values v_i of case markers p_i in the subcategorization frame s, we can construct the following three types of Bayesian network models for a set of subcategorization frames that have fixed number of case-markers:

Polarity Model: When each v_i contains a polarity only, each attribute in the Bayesian network is a case marker and the domain of the values is $\{+, -\}$. Figure 2 illustrates the Bayesian network where the attributes x_1, x_2, and x_3 correspond to the case markers *ga*, *de*, and *wo*, respectively. The subcategorization preference of $e^{(\text{tumu})}$ in Equation 7 against one of the subcategorization frames in the network is as follows:

$$\begin{aligned}
\Pr(e^{(\text{tumu})} \sqsubseteq \begin{bmatrix} ga: & [\text{pol}:+] \\ de: & [\text{pol}:+] \\ wo: & [\text{pol}:+] \end{bmatrix}) \;&=\; \Pr(e \sqsubseteq [ga : [\text{pol} : +]]) \\
&\quad \times \Pr(e \sqsubseteq [de : [\text{pol} : +]] \mid e \sqsubseteq [ga : [\text{pol} : +]]) \\
&\quad \times \Pr(e \sqsubseteq [wo : [\text{pol} : +]] \mid e \sqsubseteq \begin{bmatrix} ga : [\text{pol} : +] \\ de : [\text{pol} : +] \end{bmatrix}) \\
&=\; \Pr(e \sqsubseteq [ga : [\text{pol} : +]])
\end{aligned} \tag{11}$$

[3] When a subcategorization frame specifies the class c_i of its argument/adjunct noun only, this subcategorization frame *does* subsume the verb-noun collocation that does not have the case-marked noun in the class c_i. Contrast to the case marker *ni* (DAT), if $e^{(\text{tumu})}$ did not have the case $[ga : \text{kodomo}]$, $e^{(\text{tumu})} \sqsubseteq s^{(\text{tumu})}$ did not hold.

[4] We assume each probability of the word in the same class being generated is the same.

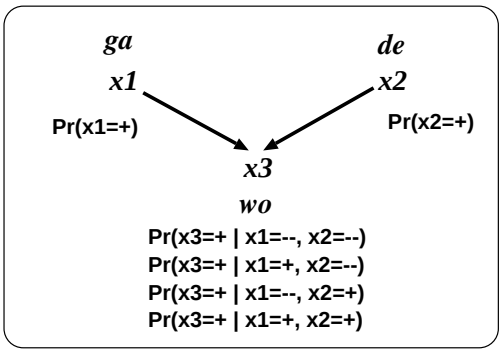


Figure 2: Polarity Model

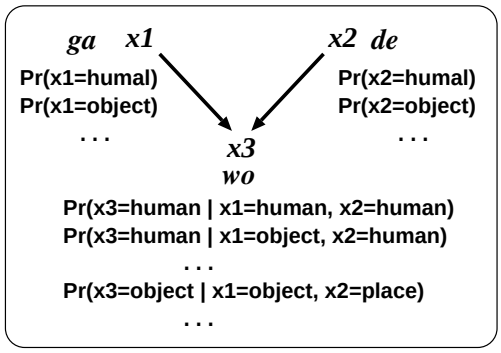


Figure 3: Class Model

$$\times \Pr(e \sqsubseteq [de : [\text{pol} : +]])$$
$$\times \Pr(e \sqsubseteq [wo : [\text{pol} : +]] \mid e \sqsubseteq \begin{bmatrix} ga : [\text{pol} : +] \\ de : [\text{pol} : +] \end{bmatrix}) \quad (12)$$
$$= \Pr(x_1 = +) \Pr(x_2 = +) \Pr(x_3 = + \mid x_1 = +, x_2 = +) \quad (13)$$

The conditional probability corresponding to wo is $\Pr(e \sqsubseteq [wo : [\text{pol} : +]] \mid e \sqsubseteq \begin{bmatrix} ga : [\text{pol} : +] \\ de : [\text{pol} : +] \end{bmatrix}) = \Pr(x_3 = + \mid x_1 = +, x_2 = +)$ since there are the two dependencies from ga and from de to wo in the network. Contrast to this, the conditional probability $\Pr(e \sqsubseteq [de : [\text{pol} : +]] \mid e \sqsubseteq [ga : [\text{pol} : +]])$ corresponding to de becomes $\Pr(e \sqsubseteq [de : [\text{pol} : +]]) = \Pr(x_2 = +)$ since there is no parent of de in the network.

Class Model: When each v_i contains a class generalization only, each attribute in the Bayesian network is a case marker and the domain of the values is a set of disjoint class generalizations in the thesaurus (i.e. *cut* of the thesaurus). Figure 3 illustrates the Bayesian network where the attributes x_1, x_2, and x_3 correspond to the case markers ga, de, and wo, respectively. The subcategorization preference of $e^{(\text{tumu})}$ in Equation 7 against one of the subcategorization frames in the network is as follows:

$$\Pr(e^{(\text{tumu})} \sqsubseteq \begin{bmatrix} ga : & [\text{cnt} : \mathsf{human}] \\ de : & [\text{cnt} : \mathsf{place}] \\ wo : & [\text{cnt} : \mathsf{object}] \end{bmatrix}) = \frac{1}{L(\mathsf{human})\, L(\mathsf{place})\, L(\mathsf{object})}$$
$$\times \Pr(e \sqsubseteq [ga : [\text{cnt} : \mathsf{human}]])$$
$$\times \Pr(e \sqsubseteq [de : [\text{cnt} : \mathsf{place}]] \mid e \sqsubseteq [ga : [\text{cnt} : \mathsf{human}]])$$
$$\times \Pr(e \sqsubseteq [wo : [\text{cnt} : \mathsf{object}]] \mid e \sqsubseteq \begin{bmatrix} ga : [\text{cnt} : \mathsf{human}] \\ de : [\text{cnt} : \mathsf{place}] \end{bmatrix}) \quad (14)$$

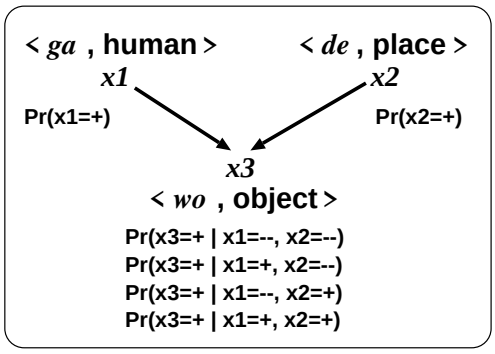


Figure 4: Mixed Model

$$
\begin{aligned}
&= \frac{1}{L(\mathsf{human})\,L(\mathsf{place})\,L(\mathsf{object})} \\
&\quad \times \Pr(e \sqsubseteq [ga : [\mathrm{cnt} : \mathsf{human}]]) \\
&\quad \times \Pr(e \sqsubseteq [de : [\mathrm{cnt} : \mathsf{place}]]) \\
&\quad \times \Pr(e \sqsubseteq [wo : [\mathrm{cnt} : \mathsf{object}]] \mid e \sqsubseteq \begin{bmatrix} ga : [\mathrm{cnt} : \mathsf{human}] \\ de : [\mathrm{cnt} : \mathsf{place}] \end{bmatrix}) \qquad (15) \\
&= \frac{1}{L(\mathsf{human})\,L(\mathsf{place})\,L(\mathsf{object})} \\
&\quad \times \Pr(x_1 = \mathsf{human})\Pr(x_2 = \mathsf{place}) \\
&\quad \times \Pr(x_3 = \mathsf{object} \mid x_1 = \mathsf{human},\, x_2 = \mathsf{place}) \qquad (16)
\end{aligned}
$$

Mixed Model: When each v_i contains both a polarity and a class generalization, each attribute in the Bayesian network is a pair of case marker and class of the noun and the domain of the values is $\{+, -\}$. Figure 4 illustrates the Bayesian network where the attributes x_1, x_2, and x_3 correspond to the pairs of $\langle ga, \mathsf{human}\rangle$, $\langle de, \mathsf{place}\rangle$, and $\langle wo, \mathsf{object}\rangle$, respectively. The subcategorization preference of $e^{(\mathrm{tumu})}$ in Equation 7 against one of the subcategorization frames in the network is as follows:

$$
\begin{aligned}
\Pr(e^{(\mathrm{tumu})} \sqsubseteq \begin{bmatrix} ga : \begin{bmatrix} \mathrm{pol} : + \\ \mathrm{cnt} : \mathsf{human} \end{bmatrix} \\ de : \begin{bmatrix} \mathrm{pol} : + \\ \mathrm{cnt} : \mathsf{place} \end{bmatrix} \\ wo : \begin{bmatrix} \mathrm{pol} : + \\ \mathrm{cnt} : \mathsf{object} \end{bmatrix} \end{bmatrix}) &= \frac{1}{L(\mathsf{human})\,L(\mathsf{place})\,L(\mathsf{object})} \\
&\quad \times \Pr(e \sqsubseteq \begin{bmatrix} ga : \begin{bmatrix} \mathrm{pol} : + \\ \mathrm{cnt} : \mathsf{human} \end{bmatrix} \end{bmatrix}) \\
&\quad \times \Pr(e \sqsubseteq \begin{bmatrix} de : \begin{bmatrix} \mathrm{pol} : + \\ \mathrm{cnt} : \mathsf{place} \end{bmatrix} \end{bmatrix} \mid e \sqsubseteq \begin{bmatrix} ga : \begin{bmatrix} \mathrm{pol} : + \\ \mathrm{cnt} : \mathsf{human} \end{bmatrix} \end{bmatrix}) \\
&\quad \times \Pr(e \sqsubseteq \begin{bmatrix} wo : \begin{bmatrix} \mathrm{pol} : + \\ \mathrm{cnt} : \mathsf{object} \end{bmatrix} \end{bmatrix} \mid e \sqsubseteq \begin{bmatrix} ga : \begin{bmatrix} \mathrm{pol} : + \\ \mathrm{cnt} : \mathsf{human} \end{bmatrix} \\ de : \begin{bmatrix} \mathrm{pol} : + \\ \mathrm{cnt} : \mathsf{place} \end{bmatrix} \end{bmatrix}) \qquad (17) \\
&= \frac{1}{L(\mathsf{human})\,L(\mathsf{place})\,L(\mathsf{object})}
\end{aligned}
$$

$$\times \Pr(e \sqsubseteq \begin{bmatrix} ga : \begin{bmatrix} \text{pol} : + \\ \text{cnt} : \mathsf{human} \end{bmatrix} \end{bmatrix})$$

$$\times \Pr(e \sqsubseteq \begin{bmatrix} de : \begin{bmatrix} \text{pol} : + \\ \text{cnt} : \mathsf{place} \end{bmatrix} \end{bmatrix})$$

$$\times \Pr(e \sqsubseteq \begin{bmatrix} wo : \begin{bmatrix} \text{pol} : + \\ \text{cnt} : \mathsf{object} \end{bmatrix} \end{bmatrix} \mid e \sqsubseteq \begin{bmatrix} ga : \begin{bmatrix} \text{pol} : + \\ \text{cnt} : \mathsf{human} \end{bmatrix} \\ de : \begin{bmatrix} \text{pol} : + \\ \text{cnt} : \mathsf{place} \end{bmatrix} \end{bmatrix}) \quad (18)$$

$$= \frac{\Pr(x_1 = +)\Pr(x_2 = +)\Pr(x_3 = + \mid x_1 = +, x_2 = +)}{L(\mathsf{human})\,L(\mathsf{place})\,L(\mathsf{object})} \quad (19)$$

where x_1, x_2, and x_3 are abbreviations for $\langle ga,\mathsf{human}\rangle$, $\langle de,\mathsf{place}\rangle$, and $\langle wo,\mathsf{object}\rangle$, respectively.

4 MDL Principle

The MDL (Minimum Description Length) principle is one of the criteria in model selection. It is designed so as to "select the model that has as much fit to a given data as possible and that is as simple as possible." The MDL principle selects the model that minimizes the amount called *description length*. The description length $l(M, D)$ of the probabilistic model M for the data D is defined as follows:

$$l(M, D) = -\log L_M(D) + \frac{1}{2} N_M \log |D| \quad (20)$$

where $L_M(D)$ is the likelihood of the model M to the data D, N_M is the number of the parameters in the model M, and $|D|$ is the size of the data D. According to Equation 20, the model should increase $L_M(D)$ and decrease N_M in order to minimize the description length $l(M, D)$. Increasing $L_M(D)$ corresponds to seeking for a model that has more fit to the data D and decreasing N_M corresponds to seeking for a simpler model.

When a data is given as a set of combinations of attributes, each probability of the combination can be calculated as the joint probability of the attributes. The number of parameters is the sum of the number of entries at each vertex as explained in the previous section. So we can define the description length of a given Bayesian network B for a given set C of verb-noun collocations. The formal definition is as follows:

$$l(B, C) = -\sum_{e_i \in C} \log \Pr^{(B)}(e_i \sqsubseteq s_i) + \frac{1}{2} \left(\sum_{x_j \in V} (|\mathrm{dom}(x_j)| - 1) \prod_{x_k \in \pi(x_j)} |\mathrm{dom}(x_k)| \right) \log |C| \quad (21)$$

where s_i is one of the subcategorization frames represented by the Bayesian network B, V is the set of vertices in the network B, and $\pi(x_i)$ is the set of the parents of vertex x_i. Each subcategorization preference $\Pr^{(B)}(e_i \sqsubseteq s_i)$ in the first term can be calculated as described in Section 3.2.

5 Model Construction Algorithm

We adopt a hill-climbing algorithm or a greedy search in order to find better models for Polarity, Class, and Mixed Models. The following algorithm constructs a better model by transforming a given model into one that has shorter description length preserving it as a DAG. Note that this algorithm is not guaranteed to obtain the optimal model.[5]

1. Let $B_s^{(0)}$ be the initial graph that has no edges. Set the initial parameter of each vertex in $B_s^{(0)}$ according to the corpus $C = \{e_1, e_2, \ldots, e_M\}$.

2. Let $B_{s1}^{(n+1)}$, $B_{s2}^{(n+1)}$, ..., $B_{sk}^{(n+1)}$ be the transformed graphs of $B_s^{(n)}$ by applying the possible transformational operations.

[5] A naive exhaustive search instead of a greedy search is too computationally expensive to execute. Branch and Bound Method used in (Suzuki, 1996) is one of the promising methods to improve the search efficiency.

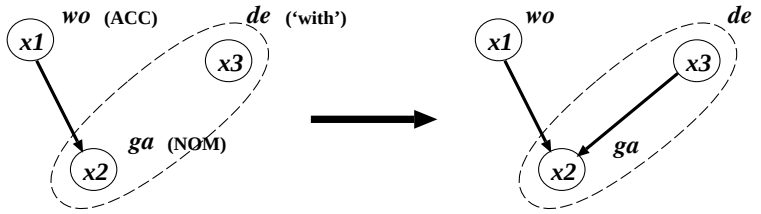


Figure 5: Vertex Linking Operation

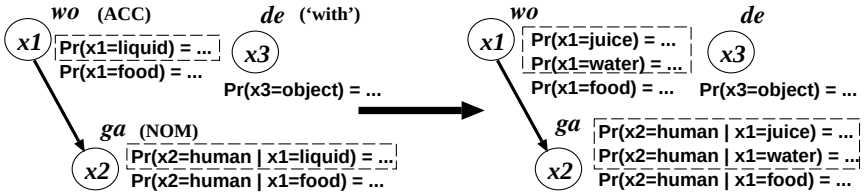


Figure 6: Class-Domain Dividing Operation

3. For each $B_{si}^{(n+1)}$, set the parameter of each vertex in the graph according to corpus C and calculate the description length $l(B_{si}^{(n+1)}, C)$ of each graph.

4. Choose the graph whose description length $l(B_{si}^{(n+1)}, C)$ is the shortest of them and let it be $B_s^{(n+1)}$.

5. Repeat the above until $l(B_s^{(n+1)}, C)$ cannot be decreased from $l(B_s^{(n)}, C)$.

For each model, the set of vertices (i.e. attributes) in the initial graph $B_s^{(0)}$ and the transformational operations from $B_s^{(n)}$ into $B_{si}^{(n+1)}$ are defined as follows:

Polarity Model: the set of vertices in the initial graph is the case markers that appear in the given corpus C. The domain of each vertex is $\{+, -\}$. The transformational operations are only to connect two vertices with a directed edge. We call this operation *vertex linking*. Figure 5 illustrates the example that adds an edge from the vertex x_3 corresponding to the case marker *de* ('with') to the vertex x_2 corresponding to the case marker *ga* (NOM).

Class Model: the set of vertices in the initial graph is the case markers that appear in the given corpus C. The domain of each vertex is the children of the root node of the thesaurus. The transformational operations are: (a) to connect two vertices with a directed edge (i.e. vertex linking operation) and (b) to 'refine' the domain of a vertex, that is, to replace a class generalization c_j in the domain $\mathrm{dom}(x_i)$ of vertex x_i with subclasses of c_j.[6] We call this operation *class-domain dividing*. Figure 6 illustrates the example that refine the domain of the vertex x_1 corresponding to the case marker *wo* (ACC) by dividing the class generalization liquid in the domain into its subclasses juice and water. Note that the entries of the vertex x_2 which is a child of the vertex x_1 are also divided according to this operation.

Mixed Model: the set of vertices in the initial graph is any pairs $\langle p_i, c_j \rangle$ of case markers p_i that appear in the given corpus C and the children c_j of the root node of the thesaurus. The domain of each vertex is $\{+, -\}$. The transformational operations are: (a) to connect two vertices with a directed edge (i.e. vertex linking) and (b) to split a vertex $\langle p_i, c_j \rangle$ into $\langle p_i, c_{j1} \rangle$, $\langle p_i, c_{j2} \rangle$, ..., $\langle p_i, c_{jk} \rangle$ according to the subclasses c_{j1}, c_{j2}, ..., c_{jk} of the class generalization c_j.[7] We call this operation *class-vertex splitting*. Figure 7 illustrates the example that splits

[6] This operation does not change the structure of the graph.

[7] We restrict the application of this operation within the case where the vertex $\langle p_i, c_j \rangle$ does not have edges in order to avoid managing a huge number of possible split operations. On the other hand, we could adopt *merging* operation instead of splitting, which begins with vertices whose class restrictions are the leaf level of the thesaurus and then gradually merge them, keeping the graph as a DAG. The disadvantages of this algorithm, however, are (1) that the computational complexities at the preparation and the earlier steps are greater than those of the proposed one and (2) that the more general models we want to obtain with respect to the class restrictions, the more steps of iteration are required.

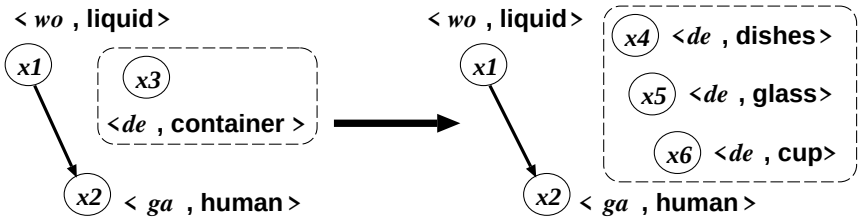


Figure 7: Class-Vertex Splitting Operation

the vertex x_3 corresponding to the attribute $\langle de, \mathsf{container} \rangle$ into the vertices x_4, x_5, and x_6 according to the subclasses dishes, glass, and cup of the class container.

Note that each operation is applicable as long as the result of the transformation is also a DAG. The parameter assignment of each vertex in the transformed graph is calculated according to the frequency of each case combination, which is the maximum likelihood estimation. Their calculation are described in detail in Appendix B.

6 Experiment and Evaluation

As a training and test corpus, we used the EDR Japanese bracketed corpus (EDR, 1995), which contains about 210,000 sentences collected from newspaper and magazine articles. From the EDR corpus, we extracted verbs which appear more than 300 times in the corpus and picked up seven verbs in order to evaluate the constructed models in the syntactic disambiguation task for various pairs of two verbs. These verbs are *agaru* (increase or elevate), *tigau* (differ), *kau* (buy), *motoduku* (base), *ouziru* (respond), *sumu* (live), and *tunagaru* (connect).

We used 'Bunrui Goi Hyou' (BGH) (NLRI, 1993) as the Japanese thesaurus. BGH has six-layered abstraction hierarchy and more than 60,000 words are assigned at the leaves and its nominal part contains about 45,000 words. Five classes are allocated at the second level with the root node being the first level. Note that the BGH concept ID of each case-marked noun cannot always be determined uniquely due to the word sense ambiguity. In such cases, we enumerate all the possible patterns of the concept IDs in the collocation and assume their frequencies to be equal. For example, when two nouns X and Y in a collocation have ambiguities $\{C_1, C_2\}$ and $\{D_1, D_2, D_3\}$, respectively, we divide the collocation into six ones, (C_1, D_1), (C_1, D_2), ..., (C_2, D_3) and assign the frequency 1/6 to each of them.

6.1 Example of Constructed Bayesian Network

Figure 8 illustrates some Bayesian networks for verb *kau* (buy) obtained by the algorithm. The tuples of numbers $[1.xxx\ldots]$ in the Class and Mixed Model is the BGH concept ID of the case-marked noun. The Polarity and the Class Models have the 11 case markers. In the Polarity model, the dependencies among the five case markers of them are obtained while in the Class Model, the dependencies among the eight case markers of them are obtained. The Mixed Model has the 31 vertices and the eight of them are dependent. The most frequent case is *wo* (ACC), which appears in about 75% of the collocations. This fact also has some effects on the constructed models, that is, almost all the cases depend on the cases marked by *wo*.

6.2 Evaluation of Subcategorization Preference

6.2.1 Syntactic Disambiguation Task

We evaluate the performance of the constructed Bayesian networks in the following syntactic disambiguation task by subcategorization preference. Consider the following verb-final Japanese sentence with a subordinate clause:

$$N_x - p_x - N_{11} - p_{11} - \ldots - N_{1m} - p_{1m} - V_1 - N_{21} - p_{21} - \ldots - N_{2n} - p_{2n} - V_2. \tag{22}$$

where N_x and N_{ij} are nouns, p_{ij} are case-marking post-positional particles, and V_1 and V_2 are verbs. The first verb V_1 is the head verb of the subordinate clause. We consider the subcategorization ambiguity of the post-positional phrase $N_x - p_x$, that is, whether $N_x - p_x$ is subcategorized for by V_1 or V_2. The following illustrates an example where $N_x - p_x$ is *zitensya de*:

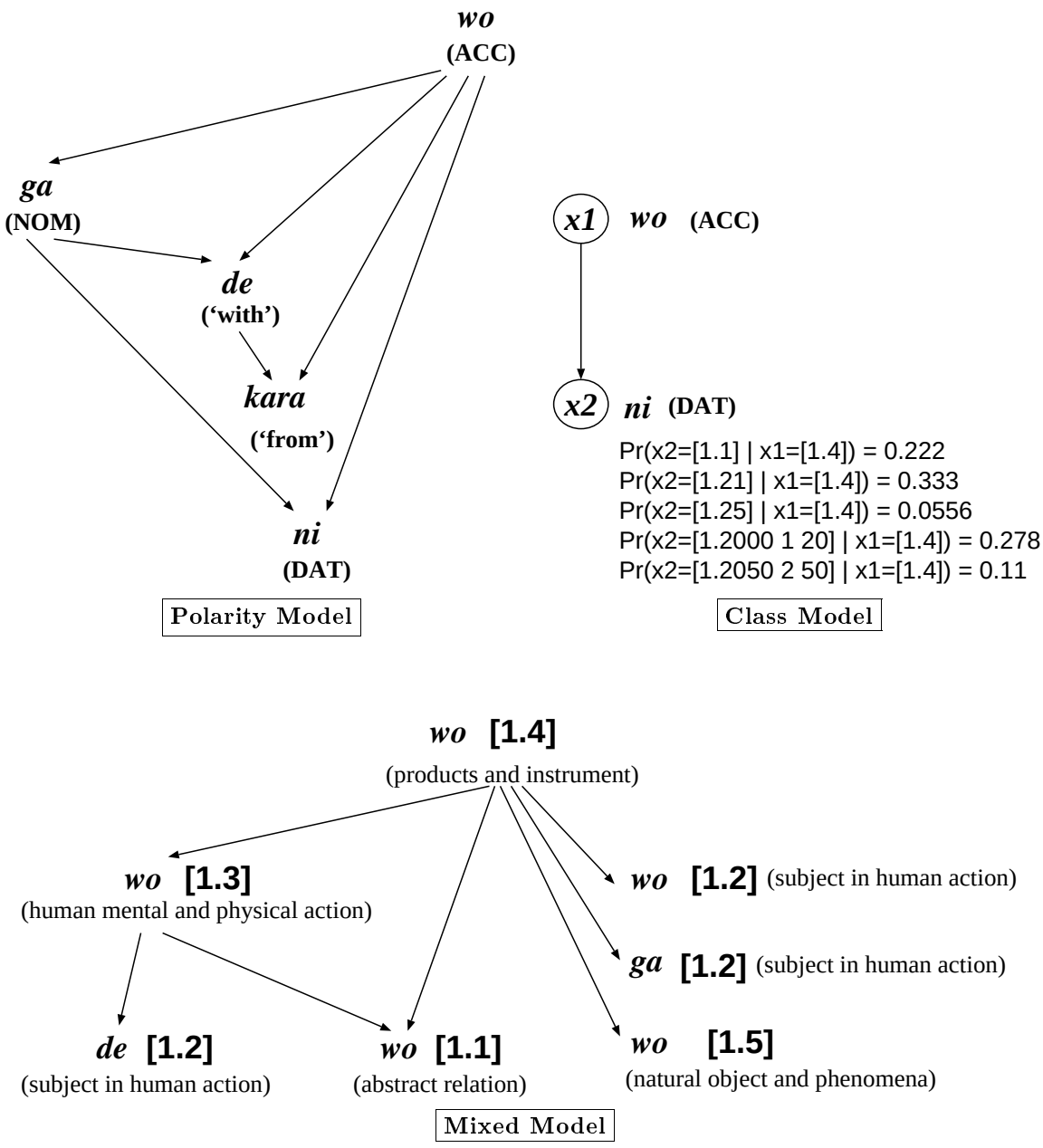


Figure 8: Structures of Constructed Bayesian Networks (excerpt)

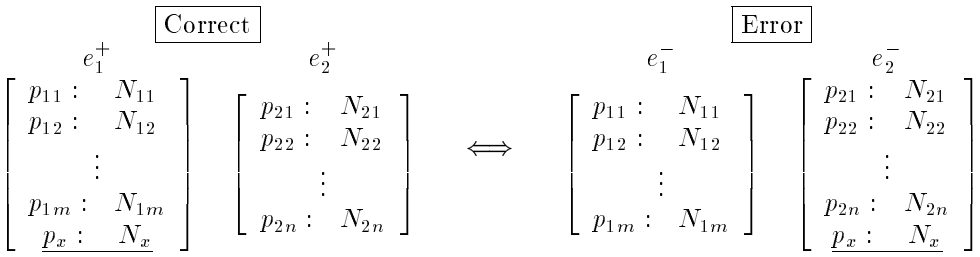


Figure 9: Generation of Test Collocations

(*Taro ga*)	*zitensya de*	*nisi*	*he*	*hasiru*	*Hanako*	*wo*	*eki*	*made*	*oikaketa.*
(Taro NOM)	bicycle by	the west	to	run	Hanako	OBJ	station	to	chase.

(If the phrase "*zitensya de* (by bicycle)" modifies the verb "*hasiru* (run)", the sentence means that "Taro chased Hanako, who ran to the west by bicycle, to the station." On the other hand, If the phrase "*zitensya de* (by bicycle)" modifies the verb "*oikaketa* (chase)", the sentence means that "Taro chased Hanako, who ran to the west, to the station by bicycle.")

We use held-out verb-noun collocations of the verbs V_1 and V_2 which are not used in the training. They are like those verb-noun collocations in the left side in Figure 9. Next, we generate erroneous verb-noun collocations of V_1 and V_2 as those in the right side Figure 9, by choosing a case element $[p_x : N_x]$ at random and moving it from V_1 to V_2. Then, we compare the preference of each collocation between the correct and erroneous pairs. Let e_1^+ (resp. e_1^-) be a collocation of verb V_1 which contains (does not contain) the case element $[p_x : N_x]$ in question and let e_2^- (e_2^+) be defined in the same way. Let the positive threshold for confidence be T. The case element $[p_x : N_x]$ in question is judged to be attached to verbs V_1 and V_2 according to the following criteria:

$$\begin{array}{lcl} \Pr(e_1^+)\Pr(e_2^+) - \Pr(e_1^-)\Pr(e_2^-) > T & \Longrightarrow & [p_x : N_x] \text{ is attached to } V_1. \\ \Pr(e_1^+)\Pr(e_2^+) - \Pr(e_1^-)\Pr(e_2^-) < -T & \Longrightarrow & [p_x : N_x] \text{ is attached to } V_2. \\ \text{Otherwise} & \Longrightarrow & \text{do not decide.} \end{array}$$

We have tested the two probabilistic models: *Polarity+Class Model* and *Mixed Model*. The first model constructs and uses both of the Polarity Model and the Class Model on the same verb. Each subcategorization frame s can be written as the unification of s_p and s_c, where s_p is the subcategorization frame that only specifies polarities and s_c is the subcategorization frame that only specifies class generalizations. If we assume that the polarities are independent of the class generalizations, we can approximate the conditional probability $\Pr(e \sqsubseteq s_p \wedge s_c \mid e \sqsubseteq s'_p \wedge s'_c)$ as the product of $\Pr(e \sqsubseteq s_p \mid e \sqsubseteq s'_p)$ and $\Pr(e \sqsubseteq s_c \mid e \sqsubseteq s'_c, e \sqsubseteq s_p \wedge s'_p)$. The Polarity+Class Model is the model that calculates each conditional probability as such product.

6.2.2 Results

Applicability and Precision According to the above criteria, the greater threshold T is, the less applicable the model is. Figure 10 shows the trade-offs between applicability and precision for various values of T. For those pairs of verbs in Figure 10, the Polarity+Class Model performs (almost) the best.

With these results, it is quite interesting to compare the precisions of the four models when the applicability is close to 1. First, in the case of (a) "*agaru* (increase or elevate)" vs. "*tigau* (differ)", the precisions of the three models other than the Polarity+Class Model are similar and much less than that of the Polarity+Class Model. In this case, combination of the Polarity and the Class Models achieves a great increase in the precision of the syntactic disambiguation task. Next, in the case of (b) "*agaru* (increase or elevate)" vs. "*motoduku* (base)", the precisions of the Polarity+Class Model and the Class Model are comparable and are much greater than other two models. In this case, the Class Model seems to contribute to raising the precision of the syntactic disambiguation task. On the other hand, in the case of (c) "*agaru* (increase or elevate)" vs. "*kau* (buy)", the precisions of the Polarity+Class Model and the Polarity Model are comparable and it seems that the Polarity Model contributes to raising the precision. Finally, in the case of (d) "*motoduku* (base)" vs. "*tunagaru* (connect)", the result is similar to the case (a), although the precisions become less in the order of the Polarity+Class, Mixed, Polarity, and Class Models.

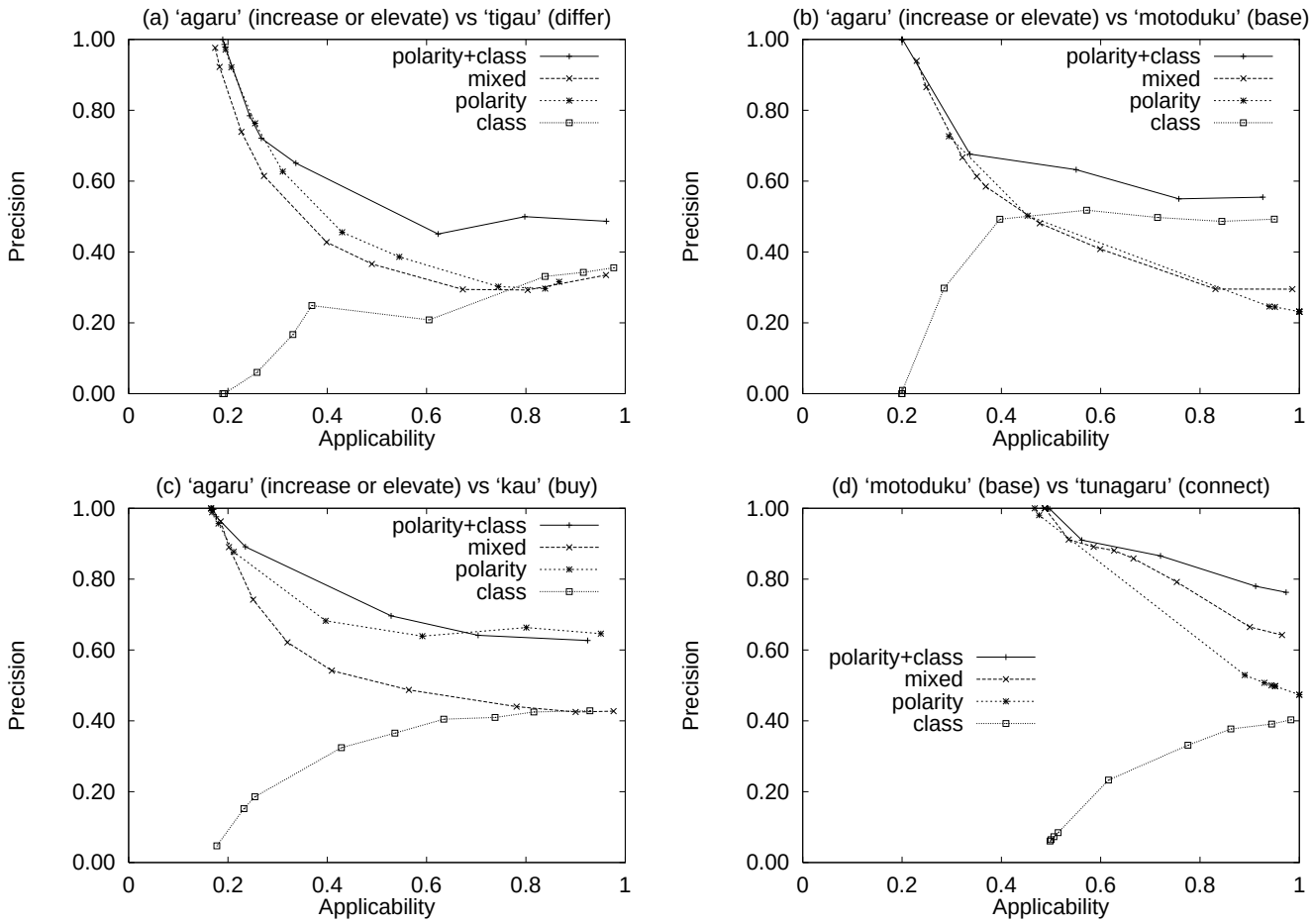


Figure 10: Trade-off between Applicability and Precision

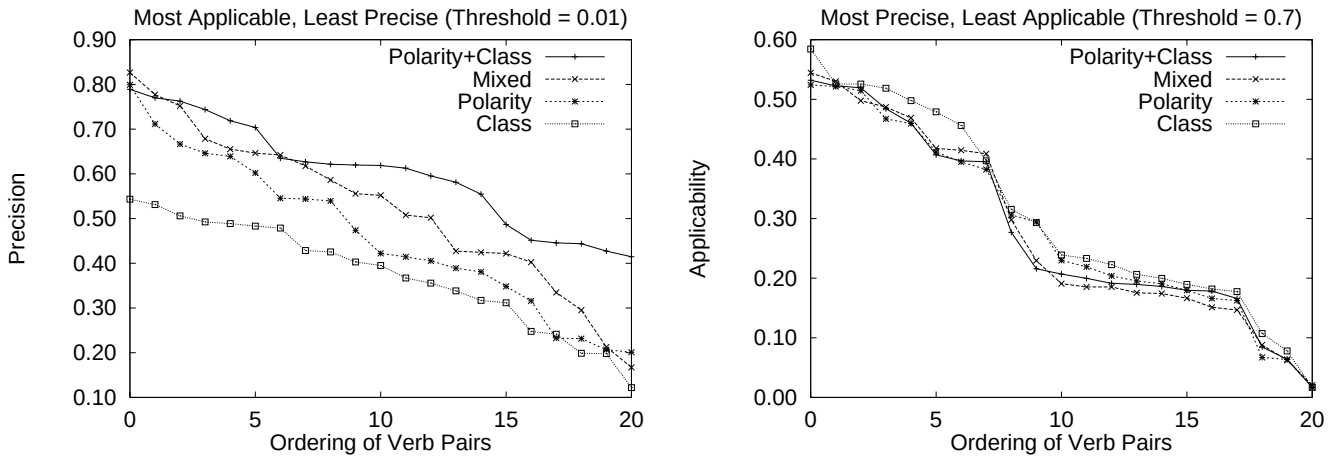


Figure 11: Orderings of Precisions and Applicabilities

Ordering of Precisions and Applicabilities Since there are seven verbs, the number of possible test is ${}_7C_2 = 21$. Figure 11 shows the orderings of the precisions and the applicabilities of these test. The left-hand graph plots the precisions with threshold $T = 0.01$, which roughly corresponds to the applicability 0.8–0.9. The right-hand graph plots the applicabilities with threshold $T = 0.7$, which roughly corresponds to the precision 0.9–1.0. In contrast to the precisions, the applicabilities of the four models are nearly the same. These graphs indicate that the Polarity+Class Model performs better than the Mixed Model does. The Mixed Model performs the best (82.7 %) in the case of the pair of "*ouziru* (respond)" and "*sumu* (live)" and the worst (16.7 %) in the case of the pair of "*tigau* (differ)" and "*motoduku* (base)". The Polarity+Class Model performs the best (78.9 %) in the case of the pair of "*sumu* (live)" and "*tunagaru* (connect)" and the worst (41.5 %) in the case of the pair of "*kau* (buy)" and "*ouziru* (respond)". The Polarity+Class Model achieves precisions higher than 70% for about 29% (6/21) pairs of verbs, and those higher than 60% for about 62% (13/21) of pairs.

6.2.3 Discussion

The result of the experiment strongly indicates that the Polarity+Class Model is superior to the Mixed Model in the syntactic disambiguation task, though the former is an approximation of the latter. This difference of the two models is mainly caused by the effect of the Class Model. Especially, this section discusses the following two reasons: (i) the effect of *positive/negative* dependencies, (ii) the probabilities of the *negative* co-occurrence in the Mixed Model.

First, the Polarity and the Mixed Models can represent both positive and negative dependencies while the Class Model can represent only the positive dependencies. By positive dependencies, we mean that the indicated co-occurrence is positive, i.e., indicating that case-markers p_i and p_j have such positive dependencies that they co-occur within a verb-noun collocation of the verb. By negative dependencies, we mean that the indicated co-occurrence is negative, i.e., indicating that case-markers p_i and p_j have such negative dependencies that they do not co-occur within a verb-noun collocation of the verb.

In the Mixed Model, the positive/negative dependencies are represented within a single model. In the Polarity and Mixed Models, almost all the dependencies included in the constructed Bayesian network are *negative* dependencies. Therefore, subtle differences of positive dependencies are ignored in the Mixed Model, because almost all the dependencies are negative and both positive/negative dependencies are represented within a single model. On the other hand, in the Polarity+Class Model, the Class Model represents only the positive dependencies, which are combined with the negative dependencies represented in the Polarity Model. In this case, subtle differences of positive dependencies are can be detected within the Class Model and these differences can also be detected in the combined model.

Second, in the Mixed Model, the polarities of all the classes of case-marked nouns are included in the model. Thus, when calculating the probability of the given verb-noun collocation, the probabilities of all the classes of case-marked nouns have to be included in the probability of the given verb-noun collocation, even though those classes do

not actually appear within the verb-noun collocation. On the other hand, in the Polarity+Class Model, the Polarity Model includes the polarities of case-markers only, and not the polarities of the classes of case-marked nouns. And, the Class Model do not include the probabilities of the classes of case-marked nouns which do not actually appear within the give verb-noun collocation. Therefore, in the Polarity+Class Model, the polarities of the classes of case-marked nouns have much less effect than in the Mixed Model.

7 Conclusion

We categorized dependencies among cases of a verb into the three types. According to this categorization, we formalized verbal subcategorization preference based on the Bayesian network and proposed the two probabilistic models: the Polarity+Class Model and the Mixed Model. We evaluated these two models by the syntactic disambiguation task and obtained the result that the Polarity+Class Model was superior to the Mixed Model. As a related work, (Li and Abe, 1996) studied a method for learning dependencies between case slots and evaluated the discovered dependencies in the syntactic disambiguation task. In Contrast to (Li and Abe, 1996), we could extract dependencies among class generalizations of nouns in the cases, which contribute to increasing the performance of the syntactic disambiguation task. As a further work, we are planning to investigate the relationship between the description length of the probabilistic model and the performance in the syntactic disambiguation task. This might make it possible to experimentally evaluate the appropriateness of the MDL Principle in terms of the syntactic disambiguation task.

References

Adam L. Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

Remco R. Bouckaert. 1993. Probabilistic network construction using the minimum description length principle. In Michael Clarke, Rudolf Kruse, and Serafín Moral, editors, *Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, volume 747 of *LNCS*, pages 41–48. Springer-Verlag. (Proceedings of European Conference ECSQARU'93, Granada, Spain, November 8–10, 1993).

(Japan Electronic Dictionary Research Institute, Ltd) EDR. 1995. EDR electronic dictionary technical guide.

M. Haruno. 1995. Verbal case frame acquisition as data compression. In *Proceedings of the Fifth International Workshop on Natural Language Understanding and Logic Programming (NLULP 5)*, pages 45–50.

W. Lam and F. Bacchus. 1994. Learning Bayesian belief networks: An approach based on the MDL principle. *Computational Intelligence*, 10(3):269–293.

Hang Li and Naoki Abe. 1995. Generalizing case frames using a thesaurus and the MDL principle. In *Proceedings of International Conference on Recent Advances in Natural Language Processing*, pages 239–248, Bulgaria, September.

Hang Li and Naoki Abe. 1996. Learning dependencies between case frame slots. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING '96)*, volume 1, pages 10–15, Copenhagen, Denmark, August.

(National Language Research Institute) NLRI. 1993. Word list by semantic principles. (in Japanese).

Judea Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, Los Altos, CA.

P. Resnik. 1993. Semantic classes and syntactic ambiguity. In *Proceedings of the Human Language Technology Workshop*, pages 278–283.

Jorma Rissanen. 1984. Universal coding, information, prediction, and estimation. *IEEE Transactions on Information Theory*, IT-30(4):629–636, July.

Joe Suzuki. 1993. A construction of Bayesian networks from databases based on an MDL principle. In *Proceedings of Uncertainty in AI*, pages 266–273.

Joe Suzuki. 1996. Learning Bayesian belief networks based on the minimum description length principle: An efficient algorithm using the B&B technique. In *Proceedings of the International Conference on Machine Learning*, pages 462–470.

Takehito Utsuro and Yuji Matsumoto. 1997. Learning probabilistic subcategorization preference by identifying case dependencies and optimal noun class generalization level. In *Proceedings of Applied Natural Language Processing*, pages 364–371, Washington, DC, USA, April.

Takehito Utsuro, Takashi Miyata, and Yuji Matsumoto. 1997. Maximum entropy model learning of subcategorization preference. In *Proceedings of the 5th Workshop on Very Large Corpora*, pages 246–260, Beijing, China, August.

A Formal Definition of Bayesian Network

Let $X = (x_1, x_2, \ldots, x_N)$ be an N dimensional random variable. Each attribute x_i is also a random variable. Let $I = \{1, 2, \ldots, N\}$ as an index set. Define σX as a k dimensional attribute vector $\sigma X = (x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(k)})$, where $\sigma = \{\sigma(1), \sigma(2), \ldots, \sigma(k)\}$ is an element of the power set 2^I. For instance, when $\sigma = \{2, 4\}$ and $X = (x_1, x_2, x_3, x_4, x_5)$, σX becomes (x_2, x_4).

Bayesian Network is represented as a DAG whose vertices are elements x_i of X. We denote the graph as B_s. When there is an edge from vertex x_i to vertex x_j in B_s, x_i is called the *parent* of x_j and x_j is called the *child* of x_i. We denote the set of parents of vertex x_i as $\pi(x_i)$. We also denote the function of a set of vertices v to a set of their indices as $\iota(v)$. For instance, when $v = \{x_1, x_3, x_8\}$, $\iota(v)$ becomes $\{1, 3, 8\}$.

When vertex x_i is a child of x_j, suppose that the size of the domain of the values for vertices x_i and x_j be k_i and k_j, respectively. Then vertex x_i is assigned $(k_i - 1) \prod_{x_j \in \pi(x_i)} k_j$ parameters. Each parameter means a *conditional probability* of vertex x_i being a certain value under the condition that the combination of the values for the parents of x_i is a certain pattern. We can denote these parameters by π and ι as follows:

$$\bigcup_A \{\Pr(x_i = a_i \mid \iota(\pi(x_i))X = \iota(\pi(x_i))A)\} \tag{23}$$

where A ranges over the all combinations $(a_1, a_2, \ldots, a_N)$ of possible values for $(x_1, x_2, \ldots, x_N)$. We denote an assignment of parameters as B_p and the assignment of parameters for vertex x_i as $B_p(x_i)$. We also denote the Bayesian Network whose structure is B_s and assignment of parameters is B_p as $B = (B_s, B_p)$. The joint probability $\Pr(X = A)$ can be calculated as follows:

$$\Pr(X = A) = \prod_{x_i} \Pr(x_i = a_i \mid \iota(\pi(x_i))X = \iota(\pi(x_i))A) \tag{24}$$

where x_i ranges over all the vertices in the given Bayesian network.

B Update of Parameter Assignment for Each Model

We abbreviate $\iota(\pi(x_i))X = \iota(\pi(x_i))A$ as $X_i = A_i$ for the simplicity. Let the number of occurrences of the verb-noun collocation that satisfies the condition $x_i = a_i$, $x_j = a_j$, ... as $C(x_i = a_i, x_j = a_j, \ldots)$, which is abbreviated as $C(X_k = A_k)$ for arbitrary k. Table 1 shows the update of the parameter assignment for each model.

$$B_p^{(n)}(x_i) = \left\{ \Pr(x_i = + \mid X_i = A_i) = \frac{C(x_i = +, X_i = A_i)}{C(X_i = A_i)} \;\middle|\; A_i \in \{+,-\}^{|\pi(x_i)|} \right\} \tag{25}$$

$$B_p^{(n)}(x_i) = \left\{ \Pr(x_i = c_k \mid X_i = A_i) = \frac{C(x_i = c_k, X_i = A_i)}{C(X_i = A_i)} \;\middle|\; c_k \in \mathrm{dom}(x_i), A_i \in \prod_{x_j \in \pi(x_i)} \mathrm{dom}(x_j) \right\} \tag{26}$$

$$B_p^{(n)}(x_i) = \left\{ \Pr(\langle p_i, c_i \rangle = + \mid X_i = A_i) = \frac{C(\langle p_i, c_i \rangle = +, X_i = A_i)}{C(X_i = A_i)} \;\middle|\; A_i \in \{+,-\}^{|\pi(x_i)|} \right\} \tag{27}$$

Table 1: Parameter Assignment for Each Model