INFORMATION
SCIENCE
TECHNICAL
REPORT

NAIST-IS-TR2009003
ISSN 0919-9527

Bypassed Alignment Graph for Learning Coordination in Japanese Sentences: Supplementary Material

Hideharu Okuma, Kazuo Hara, Masashi Shimbo,
and Yuji Matsumoto

July 2009

NAIST

〒 630-0192
奈良県生駒市高山町 8916-5
奈良先端科学技術大学院大学
情報科学研究科

Graduate School of Information Science
Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara 630-0192, Japan

Bypassed Alignment Graph for Learning Coordination in Japanese Sentences: Supplementary Material

Hideharu Okuma Kazuo Hara Masashi Shimbo Yuji Matsumoto
Computational Linguistics Laboratory
Graduate School of Information Science
Nara Institute of Science and Technology
8916-5 Takayama, Ikoma Nara 630-0192, Japan
{okuma.hideharu01,kazuo-h,shimbo,matsu}@is.naist.jp

July 27, 2009

Abstract

We presented an improved coordination analysis method for Japanese in our ACL-IJCNLP 2009 short paper entitled *Bypassed alignment graph for learning coordination in Japanese sentences*. This method can determine (i) whether coordination exists in a sentence, in addition to (ii) disambiguating its scope if coordination exists. Ability to cope with these two tasks simultaneously is essential for Japanese coordinate structure analysis. This technical report contains details omitted from the ACL-IJCNLP paper: the labeling scheme for processing Japanese, features used for our method, and how we extracted coordination scopes from the EDR Corpus for the experiments reported in the paper.

1 Introduction

1.1 Outline

This technical report contains material omitted, due to lack of space, from our short paper [8].

In that paper, we extended the coordinate structure analysis model of Shimbo and Hara [10] to (i) determine whether a coordination exists in a sentence, in addition to (ii) disambiguate the scope of the coordination (if it exists). A method that can simultaneously cope with task (i) in addition to (ii) is crucial for processing Japanese sentences, as some frequent coordination markers in Japanese are ambiguous and their presence does not reliably tell the presence of coordinate structure. In English, by contrast, detection of coordination (task (i)) is not an issue, since a small number of coordinate conjunctions (e.g., "and, " "or") and conjunctive phrases ("as well as," "not only ... but also") almost always indicate the presence of coordination and pattern matching suffice for task (i). As a result, nearly all previous studies on coordination in English are concerned with the disambiguation of coordination from a sentence or phrase in which coordination is a priori known to be present. Shimbo and Hara's model is mainly designed for disambiguating coordination scopes as well. As a result, naive adaptation of their model to Japanese does not perform well.

This technical report consists of three parts. Section 2 describes the modification to labeling scheme required to adopt Shimbo and Hara's model to Japanese. Section 3 describes the features used to obtain the experimental results of [8]. Section 4 presents how we extracted gold coordination scopes from the EDR corpus [4] for the experiments.

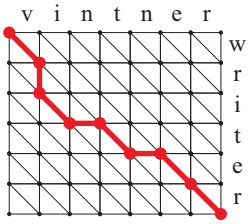


Figure 1: Alignment graph

Before presenting these main parts, we briefly review sequence alignment, and the definition of the alignment graph for coordinate structure analysis, as used by Shimbo and Hara[1].

1.2 Alignment graph for coordinate structure analysis

The objective of sequence alignment is to find how (dis)similar two sequences are. Dissimilarity is evaluated by the cost of transforming one sequence to the other, through the repeated application of basic edit operations; substitution of one character to another, inserting a character to the first sequence, and deleting a character from the first sequence. Each edit operation incurs a cost depending on the type of the edit and the character(s) involved in the edit, and the cost of an edit-operation series is the sum of the costs of its constituent edit operations. The *edit distance* is a measure of dissimilarity between two sequences, and is defined as the minimum cost of transforming one sequence to the other. Similarly, if *rewards* are associated to edit operations in place of costs, a similarity measure can be defined as the highest rewards among all possible series of edit operations.

An *alignment graph* (also called *edit graph*) [3] is a data structure for computing sequence alignment. An alignment graph for two strings "vintner" and "writer" is shown in Figure 1. Given two sequences whose similarity we like to evaluate, the two sequences are respectively associated to the rows and columns of a graph. By mapping each substitution to a diagonal arc, deletion to a vertical arc, and insertion to a horizontal arc, a series of edit operations transforming one sequence to another corresponds one-to-one to a path starting from the top-left corner node and arriving at the bottom-right corner of the graph. For details, see textbooks on biological sequence alignment, for instance [2, 3]. This correspondence allows us to reduce the problem of finding a minimum cost edit operations to a shortest path problem in an alignment graph, in which arc costs are defined as the corresponding edit operation costs.

In coordinate structure analysis, on the other hand, the objective is to find similar subsequences (or phrases) in a single sentence. If such subsequences exist, they are likely to be coordinating conjuncts; it has been observed that coordination often consists of conjuncts having similar syntactic constructs [9].

To evaluate the similarity between conjuncts, we use a sequence alignment technique. However, since we have only one sequence (sentence) and not two, the same sentence is associated to both rows and columns of the alignment graph. Because the same sequence is associated to both rows and columns, we only need to consider the half of the alignment graph, separated by the diagonal connecting the upper left corner (initial node) and the lower right corner (terminal node) of the alignment graph. We use the upper-right half, as shown in Figure 2. Further, we are interested in the similarity of segments, but not a whole sequence. To indicate segments, Shimbo and Hara associated a label, either *Inside* or *Outside*, to each of the arcs in the path. With this labeling, the horizontal and vertical spans of an *Inside* path segment determines the scope of two conjuncts.

[1] Kurohashi and Nagao [6] first proposed a similar data structure.

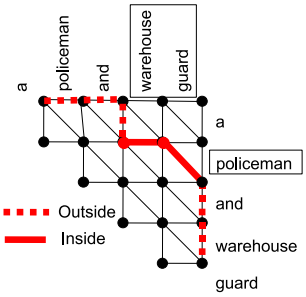
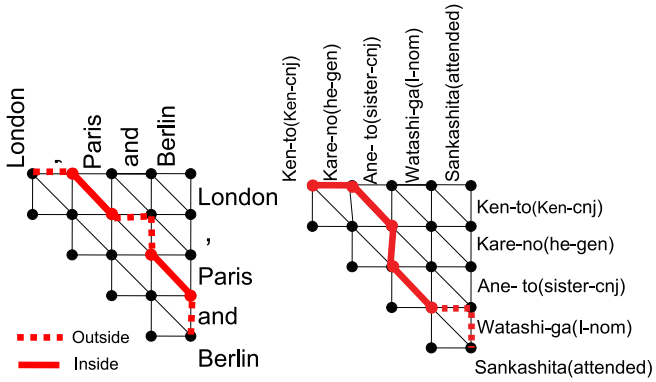


Figure 2: Triangular alignment graph for coordinate structure analysis



(a) English
"London, Paris and Berlin"

(b) Japanese
"Ken, his sister, and I attended."

Figure 3: A coordination with three conjuncts represented as two chainable paths (a) in English, and (b) in Japanese. In (a), two paths are disjoint, but in (b) they are not.

2 Arc labeling scheme for Japanese bunsetsu-level coordinate structure analysis

Shimbo and Hara's model deals with coordination comprising three or more conjuncts, by first breaking it down to consecutive pairs of conjuncts. For example, coordination "(A), (B), and (C)" with three conjuncts is broken down to two pairwise coordinations (A, B) and (B, C), while ignoring (A,C). When represented as paths in an alignment graph, these pairs yield a set of *chainable* paths [3]. Figure 3(a) depicts an English coordination consisting of three conjuncts. As seen from the figure, conjuncts are separated by a comma and a coordinate conjunction "and". Consequently, two paths are disjoint in the graph.

The same disjointness does not hold in bunsetsu-based Japanese coordination processing. In Japanese coordination analysis, the basic unit of processing is a *bunsetsu* which consists of one or more content words followed by zero or more function words. In particular, conjunction markers like particle "*to*-**cnj**"[2] and punctuations are also treated as function words, and thus absorbed in bunsetsus; "*Kare* (he) *to* (conjunctive particle)" is an example of a single bunsetsu. Consequently, two paths can occur adjacent to each other without an intervening bunsetsu, and the boundary of the two paths may not be uniquely

[2]In this technical report, we use the following abbreviations: **cnj** (conjunction), **gen** (genitive), **nom** (nominative), **acc** (accusative), and **top** (topic maker).

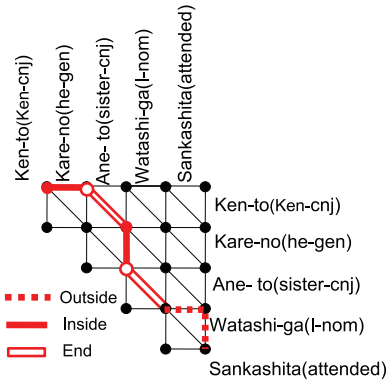


Figure 4: Chainable coordination paths in *End*/*Inside*/*Outside* model. The end of paths are identified with the "End" (white) arc.

determined in the alignment graph. Figure 3(b) shows a Japanese coordination with three conjuncts, for illustration.

To avoid such uncertainty, we introduce an additional label, *End*, to distinguish the first arc in a path from the rest. This makes a total of three labels, *End* in addition to *Inside* and *Outside* originally used by Shimbo and Hara. We call this set of labels "EIO labeling" scheme. For illustration, the white arcs in Figure 4 represent arcs with the *End* label, i.e., the last arc in individual paths.

Note that we could equally consider a "BIO" label set consisting of *Beginning*, *Inside*, and *Outside*; we add a label *Beginning* indicating the start of a path instead of *End*. We use the EIO labeling scheme as it gave a slightly better results than the BIO labeling in a preliminary experiment.

3 Features

Features can be divided into sentence-internal features and external features, depending on whether or not their values can be determined from the properties of the sentence in which they occur. We will discuss these two types of features in Sections 3.1 and 3.2, respectively. A summary of these features is given in Section 3.3.

In our model, all features are binary. If the condition associated with a feature is satisfied, it takes a value of 1; otherwise, it is 0.

3.1 Sentence-internal features

The sentence-internal features are the features determined solely on words/bunsetsus in a given sentence, without referring to any external data sources.

To define features, we first represent each bunsetsu in a sentence as a vector of *attributes* listed in Table 1. These include the part-of-speech (POS) label of the head word (last content word in a bunsetsu), its POS subcategory label, conjugation forms of verbs and adjectives, etc.

In contrast to attributes which are the property of a single bunsetsu, *features* are essentially defined for two (or more) bunsetsus; recall that features are assigned to an arc or two consecutive arcs in the alignment graph, and they determine the likelihood score of two words/bunsetsus making a coordination. In our model, all features are defined as binary indicator functions, most of which ask whether one or more attributes take specific values at the neighbor of an arc. One example of a feature assigned to a

Table 1: Attributes

Name	Description
BaseForm	Base form of the head word
Pos	Part-of-speech (POS) of the head word
PosSubcat	POS subcategory of the head word
ConjForm	Conjugate form of the head word
Type	Whether head is a noun, predicate, or others
BunruiCode	Five-digit code assigned to the head word by Bunrui-Goi-Hyo thesaurus
AuxBaseForm	Base form of the last verb/auxiliary verb in a bunsetsu
AuxPos	POS of the last verb/auxiliary verb
AuxPosSubcat	POS subcategory of the last verb/auxiliary verb
AuxConjForm	Conjugate form of the last verb/auxiliary verb
Particle	Surface and POS of particles in a bunsetsu
Punctuation	Punctuation marks in a bunsetsu
Digit	Presence of digits in a bunsetsu
ProperNoun	Presence of proper nouns in a bunsetsu
Location	Presence of location names in a bunsetsu
Person	Presence of person names in a bunsetsu
Organization	Presence of organization names in a bunsetsu
Surface	Surface form

diagonal arc at row i and column j of the alignment graph is

$$f = \begin{cases} 1 & \text{if } \textit{POS}[i] = \text{Noun}, \textit{POS}[j] = \text{Adjective, the EIO label of the arc is } \textit{Inside} \text{ and the} \\ & \text{arc direction is diagonal,} \\ 0 & \text{otherwise.} \end{cases}$$

where $\textit{POS}[i]$ denote the *POS* attribute (see Table 1) of the ith bunsetsu in a sentence. Below we describe some features which are different from Shimbo and Hara's original model for English.

Surface attributes In a preliminary experiment, we found attributes such as the surface forms of content words and their stems are too sparse to be effective in this task. Our feature set hence does not include indicator functions for the *Surface* attribute in general, except where EIO labels change. And even in the latter case, we only take the surface form of particles into account.

More complex combination of attributes In Shimbo and Hara's method, features are defined as the indicator function of a single attribute in a word, or the indicator of the same attribute in a pair of words. However, such simple features cannot represent a variety of cue expressions for coordinations in Japanese. Some of cue expressions are combinations of particles, punctuations and inflected forms. Thus we use indicator functions based on combinations of different attributes.

3.2 External features

We introduce two types of external features: (i) thesaurus features and (ii) collocation features.

Thesaurus features The thesaurus-based features are computed from *Bunrui Goi Hyo* [7], a Japanese thesaurus containing 96,051 Japanese words. Each word in the thesaurus is assigned a five-digit code called *bunrui goi code*, denoting the position of the word in the thesaurus tree. Each places in the code

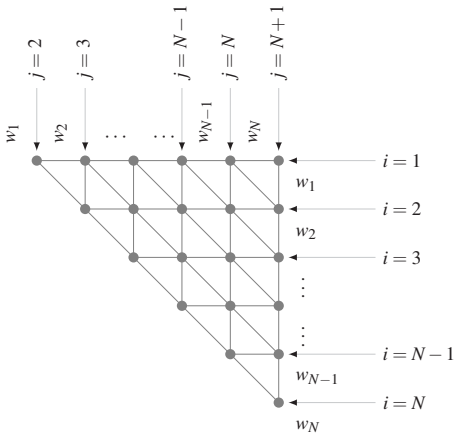


Figure 5: Node coordinates (i, j) in the triangular edit graph for a sentence $w_1, w_2, \ldots, w_N$ of length N. The first coordinate i is the row index and the second coordinate j is the column index.

represent the levels of the tree, with the most significant place representing the level closest to the root (top-level) of the tree. Thus, the degree of semantic similarity of two words can be evaluated by how many digits are shared by their *bunrui goi* codes, counting from the most significant places until a mismatched digit is found. *CommonBunruiDigits*, the first type of our thesaurus based features, exactly measures this degree of similarity. The use of this type of feature follows a rule used by Kurohashi and Nagao for their coordinate structure analysis method [6].

The second type of our thesaurus features is the indicator of the *bunrui goi codes* for the head words in two bunsetsus[3]. Because the number of code combinations can be huge, this type of feature was not used by Kurohashi and Nagao's rule-based method.

Collocation features The other type of external features we introduce is based on collocation statistics. The intuition behind these features is that given a phrase "A B and C", if both "A B" and "A C" are frequently observed in a corpus, it is more likely that the coordinate structure takes the form (A ((B) and (C))) rather than ((A B) and (C)). If "A C" is infrequent, the latter is more likely.

For the collocation features, we first collect the pairs of word-particle sequences from the ten years' volume of Mainichi Newspaper articles, and then apply the likelihood-ratio test [1] to extract collocations from these pairs. Note that the Mainichi Newspaper corpus has no overlap with the Encyclopedia section of the EDR corpus used in our experiment.

The collocation features are implemented as the indicator of whether or not the head words of two bunsetsus are found in the collected collocation database. They are assigned to the consecutive pairs of arcs in which the EIO label changes from either *Outside* to *Inside*, *Outside* to *End*, or *End* to *Outside*.

3.3 List of all features

Before summarizing all the features used in our experiments, let us first define the coordinates of nodes in a triangular alignment graph. Given a sentence consisting of N bunsetsus, the first coordinate $1 \leq i \leq N$ of the coordinate pair (i, j) denotes the row in the graph, and the second coordinate $2 \leq j \leq N+1$ denotes

[3]The second thesaurus feature (indicator of the *bunrui goi* codes for two head words) are represented as instances of $\mathit{UnorderedPair}_{t,\mathsf{a},*,*}$ with attribute $\mathsf{a} = \mathsf{BunruiCode}$ in Table 2.

Table 2: List of features assigned to an arc of type *t* emanating from the node at coordinates (i, j).

Arc type(s) *t*	Attribute *a*	Feature function templates
$\langle$D, I$\rangle$, $\langle$D, E$\rangle$	*Pos*, *PosSubcat*, *Type*, *BunruiCode*, *Particle*, *Punctuation*, *AuxConjForm*, *Digit*, *ProperNoun*, *Location*, *Person*, *Organization*	$\mathit{RowSingle}_{t,a,0,*}(i)$, $\mathit{ColSingle}_{t,a,0,*}(j)$, $\mathit{RowSeqPair}_{t,a,-1,*,*}(i)$, $\mathit{RowSeqPair}_{t,a,1,*,*}(i)$, $\mathit{ColSeqPair}_{t,a,-1,*,*}(j)$, $\mathit{ColSeqPair}_{t,a,1,*,*}(j)$, $\mathit{Match}_{t,a}(i,j)$
	Pos, *PosSubcat*, *Type*, *BunruiCode*	$\mathit{UnorderedPair}_{t,a,*,*}(i,j)$
	AuxConjForm, *Particle*, *Punctuation*, *Digit*, *ProperNoun*, *Location*, *Person*, *Organization*	$\mathit{OrthPair}_{t,a,a,*,*}(i,j)$
	BaseForm	$\mathit{Match}_{t,a}(i,j)$
		$\mathit{CommonBunruiDigits}_{t,k}(i,j)$, $\mathit{CommonChars}_{t,k}(i,j)$, $\mathit{CommonFuncWords}_{t,k}(i,j)$
$\langle$H, $*\rangle$	*Pos*, *PosSubcat*, *Part*, *AuxConjForm*, *Punctuation*	$\mathit{RowSingle}_{t,a,-1,*}(i)$, $\mathit{RowSingle}_{t,a,0,*}(i)$, $\mathit{ColSingle}_{t,a,0,*}(j)$, $\mathit{RowSeqPair}_{t,a,-1,*,*}(i)$, $\mathit{ColSeqPair}_{t,a,-1,*,*}(j)$, $\mathit{ColSeqPair}_{t,a,1,*,*}(j)$
$\langle$V, $*\rangle$	*Pos*, *PosSubcat*, *Particle*, *AuxConjForm*, *Punctuation*	$\mathit{RowSingle}_{t,a,0,*}(i)$, $\mathit{ColSingle}_{t,a,-1,*}(j)$, $\mathit{ColSingle}_{t,a,0,*}(j)$, $\mathit{RowSeqPair}_{t,a,-1,*,*}(i)$, $\mathit{RowSeqPair}_{t,a,1,*,*}(i)$, $\mathit{ColSeqPair}_{t,a,-1,*,*}(j)$

the column. The initial node has the coordinates $(1, 2)$ and the terminal is at coordinates $(N, N + 1)$. Arcs connecting a node at row (or column) i and one at row (or column) $i + 1$ is associated with the ith bunsetsu in the given sentence. See Figure 5 for illustration.

Table 2 lists the features assigned to an arc emanating from node at the (i, j) coordinates. Table 3 lists those assigned to a pair of consecutive arcs whose joint (the tail of the first arc and the head of the second arc) is at (i, j). Features are specified in these two tables using the indicator functions listed in Table 4. Notice that these are the features before distance-based feature decomposition we described in Section 3.2 of our ACL-IJCNLP short paper.

In these tables, the wild card $*$ matches arbitrary strings. When a match occurs, the matched value is substituted for $*$ in the feature function template to obtain a realization of the template. An arc type is a combination $\langle X, Y\rangle$ of a direction $X \in \{\text{D(iagonal)}, \text{H(orizontal)}, \text{V(ertical)}\}$ and an EIO label $Y \in \{\text{E(nd)}, \text{I(nside)}, \text{O(utside)}\}$. For arc pairs, $t = \langle X, Y\rangle{\rightarrow}\langle X', Y'\rangle$ denotes that the first arc is of type $\langle X, Y\rangle$ and the second is $\langle X', Y'\rangle$.

The combination of the function name and its subscripts determines the identity of a feature; i.e., two features are treated as different if the name or any of the subscripts (after variable substitution) is different. Arguments in the parentheses are not part of feature identity, so feature specifications that only differ in parenthesized arguments are *aliased* (i.e., treated as the same one feature; a single weight is associated to this feature by the perceptron algorithm). For example, $f_{1,2,3}(4, 5)$ and $f_{1,2,3}(6, 7)$ are the same feature since they have the same name and subscripts—albeit their values may be different depending on the arcs to which these features are assigned. And $f_{1,2,3}(4, 5)$ and $f_{6,7,8}(4, 5)$ are different features because they have different subscripts ($\langle 1, 2, 3\rangle$ vs. $\langle 6, 7, 8\rangle$). On the other hand, even if associated conditions are the same for two features, they are treated as different if the function names or subscripts are different.

Table 3: List of features assigned to a pair of consecutive arcs whose joint node is at coordinates (i, j).

Arc pair(s) *t*	Attributes *a*, *b*, *c*	Feature function templates
$\langle *, \mathrm{I}\rangle \rightarrow \langle *, \mathrm{O}\rangle$ $\langle *, \mathrm{E}\rangle \rightarrow \langle *, \mathrm{O}\rangle$ $\langle *, \mathrm{O}\rangle \rightarrow \langle *, \mathrm{I}\rangle$ $\langle *, \mathrm{O}\rangle \rightarrow \langle *, \mathrm{E}\rangle$ $\langle *, \mathrm{E}\rangle \rightarrow \langle *, \mathrm{I}\rangle$ $\langle *, \mathrm{E}\rangle \rightarrow \langle *, \mathrm{E}\rangle$	*AuxPos*, *AuxPosSubcat*, *AuxConjForm*, *Particle*, *Punctuation*	$\mathit{RowSingle}_{t,a,-1,*}(i)$, $\mathit{RowSingle}_{t,a,0,*}(i)$, $\mathit{ColSingle}_{t,a,-1,*}(j)$, $\mathit{ColSingle}_{t,a,0,*}(j)$, $\mathit{RowSeqPair}_{t,a,-1,*,*}(i)$, $\mathit{ColSeqPair}_{t,a,-1,*,*}(j)$, $\mathit{OrthPair}_{t,a,a,-1,-1,*,*}(i,j)$, $\mathit{OrthPair}_{t,a,a,-1,0,*,*}(i,j)$, $\mathit{OrthPair}_{t,a,a,0,-1,*,*}(i,j)$, $\mathit{OrthPair}_{t,a,a,0,0,*,*}(i,j)$
	Particle, *Punctuation*, *AuxConjForm*	$\mathit{RowPair}_{t,a,b,-1,*,*}(i)$, $\mathit{ColPair}_{t,a,b,-1,*,*}(j)$
	Particle, *Punctuation*, *AuxConjForm*, *AuxPosSubcat*	$\mathit{OrthPair}_{t,a,b,-1,-1,*,*}(i,j)$
	Particle, *Punctuation*, *AuxConjForm*	$\mathit{Triple}_{t,a,b,c,*,*,*}(i-1, i-1, j-1)$
		$\mathit{Distance}_{t,k}(i,j)$
$\langle *, \mathrm{I}\rangle \rightarrow \langle *, \mathrm{O}\rangle$ $\langle *, \mathrm{E}\rangle \rightarrow \langle *, \mathrm{O}\rangle$		$\mathit{Colloc}_t(i-1, j)$, $\mathit{Colloc}_t(j-1, j)$, $\mathit{PostCollocPair}_t(i,j)$
$\langle *, \mathrm{O}\rangle \rightarrow \langle *, \mathrm{I}\rangle$ $\langle *, \mathrm{O}\rangle \rightarrow \langle *, \mathrm{E}\rangle$		$\mathit{Colloc}_t(i-1, i)$, $\mathit{Colloc}_t(i-1, j)$, $\mathit{PreCollocPair}_t(i,j)$

For instance, in Table 4, *RowSingle* and *ColSingle* have identical conditions, but are treated as distinct features. Here, the two features are used to distinguish which of the row-wise or column-wise bunsetsus satisfies the condition. Hence in Tables 2 and 3, the argument passed to *RowSingle* is always the row index i and not column index j, while *ColSingle* are always given j (column index) and not i.

4 Extracting coordination scopes from the EDR corpus

In [8], we used the Encyclopedia section of the EDR corpus [4] for evaluation. The Encyclopedia section is largest in the EDR corpus in terms of the ratio of sentences containing coordinations to all sentences. In the EDR corpus, all sentences are segmented into words, and annotated with part-of-speech[4] Each sentence is accompanied by a syntactic tree and a semantic frame. Syntactic tree information is based on dependency structure and is given as S-expressions. Semantic information is provided apart from the syntactic tree in a semantic frame.

In the EDR corpus, coordination is represented in the semantic frame as a type of relation (named "and") connecting two words. We can obtain two coordinated series of words by combining this relation with the information in the syntactic tree. Figure 6 is an example of a syntactic tree in the EDR corpus. "W" denotes a leaf (a word) and the attached number is a word ID. In this example, the words with IDs of 2 and 5 are labeled as a coordination (i.e., relationship "and" is defined over them) in the semantic frame (not shown in the figure) for this sentence. Using this information, we can find two coordinated phrases "*kogakuteki hoho* (optical method)" and "*denkiteki hoho* (electrical method)" by tracking the nodes of the syntactic tree upwards starting from the two leaves (2 and 5) until a common ancestor node, viz. "S$*$" in the above example, is met. The word indices appearing below this node is 1 (*kogakuteki*) through 5 (second *hoho*), which exactly specify the scope of the desired coordination.

[4] In our experiments, however, we did not use the word segmentation and part-of-speech information in the EDR corpus. Instead, word segmentation and parts-of-speech were respectively generated by morphological analyzer JUMAN, which is the only input format accepted by one of the compared method, KNP [5].

Table 4: List of indicator functions for defining features. $F \equiv X$ signifies function F takes value 1 if condition X is met, or 0 if not. $\mathsf{a}[i]$, $\mathsf{b}[i]$ and $\mathsf{c}[i]$ respectively denote attribute a, b, c of the ith bunsetsu. Variables x, y, and z denote arbitrary attribute values, and i, j and k word positions in a sentence. δ, ϵ and n denote an integer.

$\mathit{Match}_{t,\mathsf{a}}(i,j)$	$\equiv$	$\mathsf{a}[i] = \mathsf{a}[j]$
$\mathit{RowSingle}_{t,\mathsf{a},\delta,x}(i)$	$\equiv$	$\mathsf{a}[i+\delta] = x$
$\mathit{ColSingle}_{t,\mathsf{a},\delta,x}(j)$	$\equiv$	$\mathsf{a}[j+\delta] = x$
$\mathit{RowSeqPair}_{t,\mathsf{a},\delta,x,y}(i)$	$\equiv$	$\mathsf{a}[i+\delta] = x$ and $\mathsf{a}[i+\delta+1] = y$
$\mathit{ColSeqPair}_{t,\mathsf{a},\delta,x,y}(j)$	$\equiv$	$\mathsf{a}[j+\delta] = x$ and $\mathsf{a}[j+\delta+1] = y$
$\mathit{RowPair}_{t,\mathsf{a},\mathsf{b},\delta,x,y}(i)$	$\equiv$	$\mathsf{a}[i+\delta] = x$ and $\mathsf{b}[i+\delta] = y$
$\mathit{ColPair}_{t,\mathsf{a},\mathsf{b},\delta,x,y}(j)$	$\equiv$	$\mathsf{a}[j+\delta] = x$ and $\mathsf{b}[j+\delta] = y$
$\mathit{OrthPair}_{t,\mathsf{a},\mathsf{b},\delta,\epsilon,x,y}(i,j)$	$\equiv$	$\mathsf{a}[i+\delta] = x$ and $\mathsf{b}[j+\epsilon] = y$
$\mathit{UnorderedPair}_{t,\mathsf{a},x,y}(i,j)$	$\equiv$	($\mathsf{a}[i] = x$ and $\mathsf{a}[j] = y$) or ($\mathsf{a}[i] = y$ and $\mathsf{a}[j] = x$)
$\mathit{Triple}_{t,\mathsf{a},\mathsf{b},\mathsf{c},x,y,z}(i,j,k)$	$\equiv$	$\mathsf{a}[i] = x$ and $\mathsf{b}[j] = y$ and $\mathsf{c}[k] = z$
$\mathit{CommonChars}_{t,n}(i,j)$	$\equiv$	Content words in $\mathsf{Surface}[i]$ and $\mathsf{Surface}[j]$ have exactly n common characters
$\mathit{CommonFuncWords}_{t,n}(i,j)$	$\equiv$	$n =$ number of common function words in the ith and jth bunsetsus
$\mathit{CommonBunruiDigits}_{t,n}(i,j)$	$\equiv$	$n =$ number of the most significant digits common in $\mathsf{BunruiCode}[i]$ and $\mathsf{BunruiCode}[j]$
$\mathit{Distance}_{t,n}(i,j)$	$\equiv$	$j - i = n \quad (n = 1, \ldots, 10)$
$\mathit{Colloc}_{t}(i,j)$	$\equiv$	sequence $\mathsf{BaseForm}[i]\mathsf{BaseForm}[j]$ is a collocation
$\mathit{PreCollocPair}_{t}(i,j)$	$\equiv$	both $\mathit{Colloc}_{t}(i-1,j)$ and $\mathit{Colloc}_{t}(i-1,i)$ hold
$\mathit{PostCollocPair}_{t}(i,j)$	$\equiv$	both $\mathit{Colloc}_{t}(i-1,j)$ and $\mathit{Colloc}_{t}(j-1,j)$ hold

References

[1] Ted Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19:61–74, 1993.

[2] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

[3] Dan Gusfield. *Algorithms on Strings, Trees, and Sequences*. Cambridge University Press, 1997.

[4] Japanese Electronic Dictionary Research Institute. The EDR dictionary. `http://www2.nict.go.jp/r/r312/EDR/index.html`, 1995.

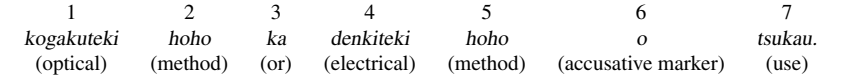
[5] Sadao Kurohashi. Kōpasu-ga saki-ka, pāsā-ga saki-ka (Which should we build first, a corpus, or a parser?). *Jōhō Shori (IPSJ Magazine)*, 41(7), 2000. In Japanese.

[6] Sadao Kurohashi and Makoto Nagao. A syntactic analysis method of long Japanese sentences based on the detection of conjunctive structures. *Computational Linguistics*, 20:507–534, 1994.

[7] The National Institute for Japanese Language. *Bunrui Goi Hyo*. Dainippon Tosho, 2004. In Japanese.

[8] Hideharu Okuma, Kazuo Hara, Masashi Shimbo, and Yuji Matsumoto. Bypassed alignment graph for learning coordination in Japanese sentences. In *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-IJCNLP 2009): Short Papers*, Singapore, 2009. To appear.

[9] Philip Resnik. Semantic similarity in a taxonomy. *Journal of Artificial Intelligence Research*, 11:95–130, 1999.



(a) Sentence

```
(M (S (t (S* (M (W 1)              ; 1: "kogakuteki" (optical)
                (t (W 2)))         ; 2: "hoho" (method)
             (W 3)                 ; 3: "ka" (or)
             (t (M (W 4)           ; 4: "denkiteki" (electrical)
                   (t (W 5))))))   ; 5: "hoho" (method)
      (W 6))                       ; 6: "o" (accusative case marker)
   (t (W 7)))                      ; 7: "tsukau" (use)
```

(b) Syntax tree

Figure 6: (a) An example sentence and (b) its associated syntax tree (an S-expression). The numbers in the S-expression represent the word index shown above the words in (a); Comments shown in each line following a ';' indicates the corresponding word and its translation.

[10] Masashi Shimbo and Kazuo Hara. A discriminative learning model for coordinate conjunctions. In *Proceedings of Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 610–619, 2007.