Joint Case Argument Identification for Japanese Predicate Argument Structure Analysis

Hiroki Ouchi **Hiroyuki Shindo** **Kevin Duh** **Yuji Matsumoto**
Graduate School of Information and Science
Nara Institute of Science and Technology
8916-5, Takayama, Ikoma, Nara, 630-0192, Japan
{ ouchi.hiroki.nt6, shindo, kevinduh, matsu }@is.naist.jp

Abstract

Existing methods for Japanese predicate argument structure (PAS) analysis identify case arguments of each predicate without considering interactions between the target PAS and others in a sentence. However, the argument structures of the predicates in a sentence are semantically related to each other. This paper proposes new methods for Japanese PAS analysis to jointly identify case arguments of all predicates in a sentence by (1) modeling multiple PAS interactions with a bipartite graph and (2) approximately searching optimal PAS combinations. Performing experiments on the NAIST Text Corpus, we demonstrate that our joint analysis methods substantially outperform a strong baseline and are comparable to previous work.

1 Introduction

Predicate argument structure (PAS) analysis is a shallow semantic parsing task that identifies basic semantic units of a sentence, such as *who does what to whom*, which is similar to semantic role labeling (SRL)[1].

In Japanese PAS analysis, one of the most problematic issues is that arguments are often omitted in the surface form, resulting in so-called *zero-pronouns*. Consider the sentence of Figure 1.

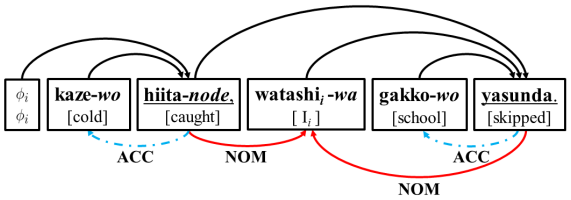

Figure 1: An example of Japanese PAS. The English translation is "Because ϕ_i caught a cold, I_i skipped school.". The upper edges are dependency relations, and the under edges are case arguments. "NOM" and "ACC" represents the nominative and accusative arguments, respectively. "ϕ_i" is a *zero-pronoun*, referring to the *antecedent* "watashi$_i$".

The case role label "NOM" and "ACC" respectively represents the nominative and accusative roles, and ϕ_i represents a zero-pronoun. There are two predicates "hiita (caught)" and "yasunda (skipped)". For the predicate "yasunda (skipped)", "watashi$_i$-*wa* (I_i)" is the "skipper", and "gakko-*wo* (school)" is the "entity skipped". It is easy to identify these arguments, since syntactic dependency between an argument and its predicate is a strong clue. On the other hand, the nominative argument of the predicate "hiita (caught)" is "watashi$_i$-*wa* (I_i)", and this identification is more difficult because of the lack of the direct syntactic dependency with "hiita (caught)". The original nominative argument appears as a zero-pronoun, so that we have to explore the *antecedent*, an element referred to by a zero-pronoun, as the argument. As the example sentence shows, we cannot use effective syntactic information for identifying such arguments. This type of arguments is known as **implicit arguments**, a very problematic language

[1]We use "PAS analysis" in this paper following previous work on Japanese PAS analysis.

Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pages 961–970, Beijing, China, July 26-31, 2015.

phenomenon for PAS analysis (Gerber and Chai, 2010; Laparra and Rigau, 2013).

Previous work on Japanese PAS analysis attempted to solve this problem by identifying arguments per predicate without considering ***interactions*** between multiple predicates and arguments (Taira et al., 2008; Imamura et al., 2009). However, implicit arguments are likely to be shared by semantically-related predicates. In the above example (Figure 1), the implicit argument of the predicate "hiita (caught)" is shared by the other predicate "yasunda (skipped)" as its nominative argument "watashi$_i$ (I$_i$)".

Based on this intuition, we propose methods to jointly identify optimal case arguments of all predicates in a sentence taking their interactions into account. We represent the interactions as a bipartite graph that covers all predicates and candidate arguments in a sentence, and factorize the whole relation into the second-order relations. This interaction modeling results in a hard combinatorial problem because it is required to select the optimal PAS combination from all possible PAS combinations in a sentence. To solve this issue, we extend the randomized hill-climbing algorithm (Zhang et al., 2014) to search all possible PAS in the space of bipartite graphs.

We perform experiments on the NAIST Text Corpus (Iida et al., 2007), a standard benchmark for Japanese PAS analysis. Experimental results show that compared with a strong baseline, our methods achieve an improvement of 1.0-1.2 points in F-measure for total case argument identification, and especially improve performance for implicit argument identification by 2.0-2.5 points. In addition, although we exploit no external resources, we get comparable results to previous work exploiting large-scale external resources (Taira et al., 2008; Imamura et al., 2009; Sasano and Kurohashi, 2011). These results suggest that there is potential for more improvement by adding external resources.

The main contributions of this work are: (1) We present new methods to jointly identify case arguments of all predicates in a sentence. (2) We propose global feature templates that capture interactions over multiple PAS. (3) Performing experiments on the NAIST Text Corpus, we demonstrate our methods are superior to a strong baseline and comparable to the methods of representative previous work.

2 Japanese Predicate Argument Structure Analysis

2.1 Task Overview

In Japanese PAS analysis, we identify arguments taking part in the three major case roles, **nominative** (NOM), **accusative** (ACC) and **dative** (DAT) cases, for each predicate. Case arguments can be divided into three categories according to the positions relative to their predicates (Hayashibe et al., 2011):

***Dep*:** The arguments that have direct syntactic dependency with the predicate.

***Zero*:** The implicit arguments whose antecedents appear in the same sentence and have no direct syntactic dependency with the predicate.

***Inter-Zero*:** The implicit arguments whose antecedents do not appear in the same sentence.

For example, in Figure 1, the accusative argument "gakko-*wo* (school)" of the predicate "yasunda (skipped)" is regarded as *Dep*, and the nominative argument "watashi$_i$-*wa* (I)" (the antecedent of zero-pronoun "ϕ_i") of the predicate "hiita (caught)" is *Zero*.

In this paper, we focus on the analysis for intra-sentential arguments (*Dep* and *Zero*). In order to identify inter-sentential arguments (*Inter-Zero*), it is required to search a much broader space, such as the whole document, resulting in a much harder analysis than intra-sentential arguments.[2] Therefore, we believe that quite different approaches are necessary to realize an inter-sentential PAS analysis with high accuracy, and leave it for future work.

2.2 Related Work

For Japanese PAS analysis research, the NAIST Text Corpus has been used as a standard benchmark (Iida et al., 2007). One of the representative researches using the NAIST Text Corpus is Imamura et al. (2009). They built three distinct models corresponding to the three case roles by extracting features defined on each pair of a predicate and a candidate argument. Using each model, they select the best candidate argument for each case per predicate. Their models are based on maximum entropy model and can easily incorporate various features, resulting in high accuracy.

[2] Around 10-20% in F measure has been achieved in previous work (Taira et al., 2008; Imamura et al., 2009; Sasano and Kurohashi, 2011).

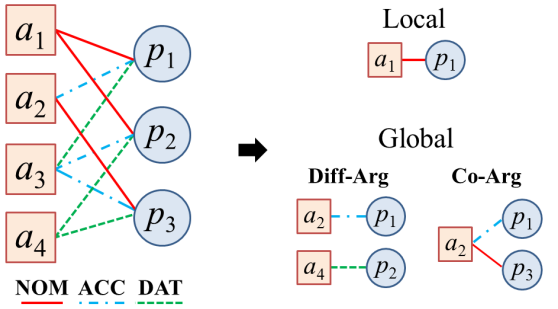

Figure 2: Intuitive image of a *predicate-argument graph*. This graph is factorized into the local and global features. The different line color/style indicate different cases.

While in Imamura et al. (2009) one case argument is identified at a time per predicate, the method proposed by Sasano and Kurohashi (2011) simultaneously determines all the three case arguments per predicate by exploiting large-scale case frames obtained from large raw texts. They focus on identification of implicit arguments (*Zero* and *Inter-Zero*), and achieves comparable results to Imamura et al. (2009).

In these approaches, case arguments were identified per predicate without considering interactions between multiple predicates and candidate arguments in a sentence. In the semantic role labeling (SRL) task, Yang and Zong (2014) pointed out that information of different predicates and their candidate arguments could help each other for identifying arguments taking part in semantic roles. They exploited a reranking method to capture the interactions between multiple predicates and candidate arguments, and jointly determine argument structures of all predicates in a sentence (Yang and Zong, 2014). In this paper, we propose new joint analysis methods for identifying case arguments of all predicates in a sentence capturing interactions between multiple predicates and candidate arguments.

3 Graph-Based Joint Models

3.1 A Predicate-Argument Graph

We define predicate argument relations by exploiting a bipartite graph, illustrated in Figure 2. The nodes of the graph consist of two disjoint sets: the left one is a set of *candidate arguments* and the right one is a set of *predicates*. In this paper, we call it a **predicate-argument (PA) graph**.

Each predicate node has three distinct edges corresponding to nominative (NOM), accusative (ACC), and dative (DAT) cases. Each edge with a case role label joins a candidate argument node with a predicate node, which represents a case argument of a predicate. For instance, in Figure 2 a_1 is the nominative argument of p_1, and a_3 is the accusative argument of p_2.

Formally, a PA graph is a bipartite graph $\langle A, P, E \rangle$, where A is the node set consisting of candidate arguments, P the node set consisting of predicates, and E the set of edges subject to that there is exactly one edge e with a case role label c outgoing from each of the predicate nodes p to a candidate argument node a. A PA graph is defined as follows:

$$A = \{a_1, ..., a_n, a_{n+1} = \texttt{NULL}\}$$
$$P = \{p_1, ..., p_m\}$$
$$E = \{\langle a, p, c \rangle \mid deg(p, c) = 1, \quad \forall a \in A,\ \forall p \in P,\ \forall c \in C\ \}$$

where $deg(p, c)$ is the number of edges with a case role c outgoing from p, and C is the case role label set. We add a dummy node a_{n+1}, which is defined for the cases where the predicate requires no case argument or the required case argument does not appear in the sentence. An edge $e \in E$ is represented by a tuple $\langle a, p, c \rangle$, indicating the edge with a case role c joining a candidate argument node a and a predicate node p. An admissible PA graph satisfies the constraint $deg(p, c) = 1$, representing that each predicate node p has only one edge with a case role c.

To identify the whole PAS for a sentence x, we predict the PA graph with an edge set corresponding to the correct PAS from the admissible PA graph set $G(x)$ based on a **score** associated with a PA graph y as follows:

$$\tilde{y} = \underset{y \in G(x)}{\text{argmax}}\ Score(x, y)$$

A scoring function $Score(x, y)$ receives a sentence x and a candidate graph y as its input, and returns a scalar value.

In this paper, we propose two scoring functions as analysis models based on different assumptions: (1) **Per-Case Joint Model** assumes the interaction between multiple predicates (***predicate interaction***) and the independence between case roles, and (2) **All-Cases Joint Model** assumes the interaction between case roles (***case interaction***) as well as the *predicate interaction*.

3.2 Per-Case Joint Model

Per-Case Joint Model assumes that different case roles are independent from each other. However, for each case, interactions between multiple predicates are considered jointly.

We define the score of a PA graph y to be the sum of the scores for each case role c of the set of the case roles C:

$$Score_{per}(x, y) = \sum_{c \in C} Score_c(x, y) \quad (1)$$

The scores for each case role are defined as the dot products between a weight vector $\boldsymbol{\theta_c}$ and a feature vector $\boldsymbol{\phi_c}(x, E(y, c))$:

$$Score_c(x, y) = \boldsymbol{\theta_c} \cdot \boldsymbol{\phi_c}(x, E(y, c)) \quad (2)$$

where $E(y, c)$ is the edge set associated with a case role c in the candidate graph y, and the feature vector is defined on the edge set.

The edge set $E(y, c)$ in the equation (2) is utilized for the two types of features, the **local features** and **global features**, inspired by (Huang, 2008), defined as follows:

$$\boldsymbol{\theta_c} \cdot \boldsymbol{\phi_c}(x, E(y, c)) =$$

$$\sum_{e \in E(y,c)} \boldsymbol{\theta_c}\, \boldsymbol{\phi_l}(x, e) + \boldsymbol{\theta_c}\, \boldsymbol{\phi_g}(x, E(y, c)) \quad (3)$$

where $\boldsymbol{\phi_l}(x, e)$ denotes the local feature vector, and $\boldsymbol{\phi_g}(x, E(y, c))$ the global feature vector. The local feature vector $\boldsymbol{\phi_l}(x, e)$ is defined on each edge e in the edge set $E(y, c)$ and a sentence x, which captures a predicate-argument pair. Consider the example of Figure 2. For Per-Case Joint Model, we use edges, $e_{a_1p_1}$, $e_{a_1p_2}$, and $e_{a_2p_3}$, as local features to compute the score of the edge set with the nominative case.

In addition, the global feature vector $\boldsymbol{\phi_g}(x, E(y, c))$ is defined on the edge set $E(y, c)$, and enables the model to utilize linguistically richer information over multiple predicate-argument pairs. In this paper, we exploit *second-order* relations, similar to the second-order edge factorization of dependency trees (McDonald and Pereira, 2006). We make a set of edge pairs E_{pair} by combining two edges e_i, e_j in the edge set $E(y, c)$, as follows:

$$E_{pair} = \{\, \{e_i, e_j\} \mid \forall e_i, e_j \in E(y, c),\ e_i \neq e_j \,\}$$

For instance, in the PA graph in Figure 2, to compute the score of the nominative arguments, we make three edge pairs:

$$\{\{e_{a_1p_1},\ e_{a_1p_2}\}, \{e_{a_1p_1},\ e_{a_2p_3}\}, \{e_{a_1p_2},\ e_{a_2p_3}\}\}$$

Then, features are extracted from these edge pairs and utilized for the score computation. For the accusative and dative cases, their scores are computed in the same manner. Then, we obtain the resulting score of the PA graph by summing up the scores of the local and global features. If we do not consider the global features, the model reduces to a per-case local model similar to previous work (Imamura et al., 2009).

3.3 All-Cases Joint Model

While Per-Case Joint Model assumes the *predicate interaction* with the independence between case roles, All-Cases Joint Model assumes the *case interaction* together with the *predicate interaction*. Our graph-based formulation is very flexible and easily enables the extension of Per-Case Joint Model to All-Cases Joint Model. Therefore, we extend Per-Case Joint Model to All-Cases Joint Model to capture the interactions between predicates and all case arguments in a sentence.

We define the score of a PA graph y based on the local and global features as follows:

$$Score_{all}(x, y) =$$

$$\sum_{e \in E(y)} \boldsymbol{\theta} \cdot \boldsymbol{\phi_l}(x, e) + \boldsymbol{\theta} \cdot \boldsymbol{\phi_g}(x, E(y)) \quad (4)$$

where $E(y)$ is the edge set associated with all the case roles on the candidate graph y, $\boldsymbol{\phi_l}(x, e)$ is the local feature vector defined on each edge e in the edge set $E(y)$, and $\boldsymbol{\phi_g}(x, E(y))$ is the global feature vector defined on the edge set $E(y)$.

Consider the PA graph in Figure 2. The local features are extracted from each edge:

Nominative : $e_{a_1p_1}, e_{a_1p_2}, e_{a_2p_3}$
Accusative : $e_{a_2p_1}, e_{a_3p_2}, e_{a_3p_3}$
Dative : $e_{a_3p_1}, e_{a_4p_2}, e_{a_4p_3}$

For the global features, we make a set of edge pairs E_{pair} by combining two edges e_i, e_j in the edge set $E(y)$, like Per-Case Joint Model. However, in the All-Cases Joint Model, the global features may involve different cases (i.e. mixing edges with different case roles). For the PA graph in Figure 2, we make the edge pairs $\{e_{a_1p_1}, e_{a_2p_1}\}$, $\{e_{a_3p_1},\ e_{a_1p_2}\}$, $\{e_{a_3p_2},\ e_{a_4p_3}\}$, and so on. From these edge pairs, we extract information as global features to compute a graph score.

Structure	Name	Description
Diff-Arg	**PAIR**	$\langle\ p_i$.rf $\circ\ p_j$.rf $\circ\ p_i$.vo $\circ\ p_j$.vo $\rangle$, $\langle\ a_i$.ax $\circ\ a_i$.rp $\circ\ p_i$.ax $\circ\ p_i$.vo $\rangle$, $\langle\ a_j$.ax $\circ\ a_j$.rp $\circ\ p_j$.ax $\circ\ p_j$.vo $\rangle$
	TRIANGLE	$\langle\ a_i$.ax $\circ\ a_i$.ax $\circ\ a_i$.rp $\circ\ a_j$.rp $\circ\ p_i$.ax $\circ\ p_i$.vo $\rangle$, $\langle\ a_i$.ax $\circ\ a_j$.ax $\circ\ a_i$.rp $\circ\ a_j$.rp $\circ\ p_j$.ax $\circ\ p_j$.vo $\rangle$,
	QUAD	$\langle\ a_i$.ax $\circ\ a_j$.ax $\circ\ a_i$.rp $\circ\ a_j$.rp $\circ\ p_i$.vo $\circ\ p_j$.vo $\rangle$ $\langle\ a_i$.ax $\circ\ a_j$.ax $\circ\ p_i$.ax $\circ\ p_j$.ax $\circ\ a_i$.rp $\circ\ a_j$.rp $\circ\ p_i$.vo $\circ\ p_j$.vo $\rangle$ $\langle\ a_i$.ax $\circ\ a_j$.ax $\circ\ p_i$.rf $\circ\ p_j$.rf $\circ\ a_i$.rp $\circ\ a_i$.rp $\circ\ p_i$.vo $\circ\ p_i$.vo $\rangle$
Co-Arg	**BI-PREDS**	$\langle\ a_i$.rp $\circ\ p_i$.rf $\circ\ p_j$.rf $\rangle$, $\langle\ a_i$.ax $\circ\ a_i$.rp $\circ\ p_i$.rf $\circ\ p_j$.rf $\rangle$
	DEP-REL	$\langle\ a_i$.ax $\circ\ a_i$.rp $\circ\ p_i$.ax $\circ\ p_j$.ax $\circ\ p_i$.vo $\circ\ p_j$.vo $\circ\ (x, y)$.dep $\rangle$ if x depends on y for x,y in (p_i,p_j), (a_i,p_i), (a_i,p_j), (p_i,a_i), (p_j,a_i)

Table 1: Global feature templates. p_i, p_j is a predicate, a_i is the argument connected with p_i, and a_j is the argument connected with p_j. Feature conjunction is indicated by $\circ$; ax=auxiliary, rp=relative position, vo=voice, rf=regular form, dep=dependency. All the features are conjoined with the relative position and the case role labels of the two predicates.

4 Global Features

Features are extracted based on **feature templates**, which are functions that draw information from the given entity. For instance, one feature template $\phi_{100} = a.ax \circ p.vo$ is a conjunction of two atomic features $a.ax$ and $p.vo$, representing an auxiliary word attached to a candidate argument ($a.ax$) and the voice of a predicate ($p.vo$). We design several feature templates for characterizing each specific PA graph. Consider the PA graph constructed from the sentence in Figure 1, and a candidate argument "kaze-*wo* (a cold)" and a predicate "hiita (caught)" are connected with an edge. To characterize the graph, we draw some linguistic information associated with the edge. Since the auxiliary word attached to the candidate argument is "*wo*" and the voice of the predicate is "*active*", the above feature template ϕ_{100} will generate a feature instance as follows.

$$(a.ax = wo) \circ (p.vo = active)$$

Such features are utilized for the local and global features in the joint models.

We propose the *global feature* templates that capture multiple PAS interactions based on the **Diff-Arg** and **Co-Arg** structures, depicted in the right part of Figure 1. The Diff-Arg structure represents that the two predicates have different candidate arguments, and the Co-Arg structure represents that the two predicates share the same candidate argument. Based on these structures, we define the global feature templates that receive a pair of edges in a PA graph as input and return a feature vector, shown in Table 1.

4.1 Diff-Arg Features

The feature templates based on the Diff-Arg structure are three types: **PAIR** (a pair of predicate-argument relation), **TRIANGLE** (a predicate and its two arguments relation), and **QUAD** (two predicate-argument relations).

PAIR These feature templates denote where the target argument is located relative to another argument and the two predicates in the Diff-Arg structure. We combine the relative position information (rp) with the auxiliary words (ax) and the voice of the two predicates (vo).

TRIANGLE This type of feature templates captures the interactions between three elements: two candidate arguments and a predicate. Like the PAIR feature templates, we encode the relative position information of two candidate arguments and a predicate with the auxiliary words and voice.

QUAD When we judge if a candidate argument takes part in a case role of a predicate, it would be beneficial to grasp information of another predicate-argument pair. The QUAD feature templates capture the mutual relation between four elements: two candidate arguments and predicates. We encode the relative position information, the auxiliary words, and the voice.

4.2 Co-Arg Features

To identify predicates that take implicit (*Zero*) arguments, we set two feature types, **BI-PREDS** and **DEP-REL**, based on the Co-Arg structure.

BI-PREDS For identifying an implicit argu-

Input: the set of cases to be analyzed C, parameter $\boldsymbol{\theta_c}$, sentence x
Output: a locally optimal PA graph $\tilde{y}$

1: Sample a PA graph $y^{(0)}$ from $G(x)$
2: $t \leftarrow 0$
3: **for** each case $c \in C$ **do**
4: **repeat**
5: $Y_c \leftarrow NeighborG(y^{(t)}, c) \cup y^{(t)}$
6: $y^{(t+1)} \leftarrow \underset{y \in Y_c}{\operatorname{argmax}} \boldsymbol{\theta_c} \cdot \boldsymbol{\phi_c}(x, E(y, c))$
7: $t \leftarrow t + 1$
8: **until** $y^{(t)} = y^{(t+1)}$
9: **end for**
10: **return** $\tilde{y} \leftarrow y^{(t)}$

Figure 3: Hill-Climbing for Per-Case Joint Model

Input: the set of cases to be analyzed C, parameter $\boldsymbol{\theta}$, sentence x
Output: a locally optimal PA graph $\tilde{y}$

1: Sample a PA graph $y^{(0)}$ from $G(x)$
2: $t \leftarrow 0$
3: **repeat**
4: $Y \leftarrow NeighborG(y^{(t)}) \cup y^{(t)}$
5: $y^{(t+1)} \leftarrow \underset{y \in Y}{\operatorname{argmax}} \boldsymbol{\theta} \cdot \boldsymbol{\phi}(x, E(y))$
6: $t \leftarrow t + 1$
7: **until** $y^{(t)} = y^{(t+1)}$
8: **return** $\tilde{y} \leftarrow y^{(t)}$

Figure 4: Hill-Climbing for All-Cases Joint Model

ment of a predicate, information of another semantically-related predicate in the sentence could be effective. We utilize bi-grams of the regular forms (rf) of the two predicates in the Co-Arg structure to capture the predicates that are likely to share the same argument in the sentence.

DEP-REL We set five distinct feature templates to capture dependency relations (dep) between the shared argument and the two predicates. If two elements have a direct dependency relation, we encode its dependency relation with the auxiliary words and the voice.

5 Inference and Training

5.1 Inference for the Joint Models

Global features make the inference of finding the maximum scoring PA graph more difficult. For searching the graph with the highest score, we propose two greedy search algorithms by extending the *randomized hill-climbing* algorithm proposed in (Zhang et al., 2014), which has been shown to achieve the state-of-the-art performance in dependency parsing.

Figure 3 describes the pseudo code of our proposed algorithm for Per-Case Joint Model. Firstly, we set an initial PA graph $y^{(0)}$ sampled uniformly from the set of admissible PA graphs $G(x)$ (line 1 in Figure 3). Then, the union Y_c is constructed from the set of neighboring graphs with a case $NeighborG(y^{(t)}, c)$, which is a set of admissible graphs obtained by changing one edge with the case c in $y^{(t)}$, and the current graph $y^{(t)}$ (line 5). The current graph $y^{(t)}$ is updated to a higher scoring graph $y^{(t+1)}$ selected from the union Y_c (line 6). The algorithm continues until no more score improvement is possible by changing an edge with the case c in $y^{(t)}$ (line 8). This repetition is executed for other case roles in the same manner. As a result, we can get a locally optimal graph $\tilde{y}$.

Figure 4 describes the pseudo code of the algorithm for All-Cases Joint Model. The large part of the algorithm is the same as that for Per-Case Joint Model. The difference is that the union Y consists of the current graph $y^{(t)}$ and the neighboring graph set obtained by changing one edge in $y^{(t)}$ regardless of case roles (line 4 in Figure 4), and that the iteration process for each case role (line 3 in Figure 3) is removed. The algorithm also continues until no more score improvement is possible by changing an edge in $y^{(t)}$, resulting in a locally optimal graph $\tilde{y}$.

Following Zhang et al. (2014), for a given sentence x, we repeatedly run these algorithms with K consecutive restarts. Each run starts with initial graphs randomly sampled from the set of admissible PA graphs $G(x)$, so that we obtain K local optimal graphs by K restarts. Then the highest scoring one of K graphs is selected for the sentence x as the result. Each run of the algorithms is independent from each other, so that multiple runs are easily executable in parallel.

5.2 Training

Given a training data set $D = \{(\hat{x}, \hat{y})\}_i^N$, the weight vectors $\boldsymbol{\theta}$ ($\boldsymbol{\theta_c}$) in the scoring functions of the joint models are estimated by using machine learning techniques. We adopt averaged perceptron (Collins, 2002) with a max-margin technique:

$$\forall i \in \{1, ..., N\},\ y \in G(x_i),$$
$$Score(\hat{x}_i, \hat{y}_i) \geq Score(\hat{x}_i, y) + \|\hat{y}_i - y\|_1 - \xi_i$$

where $\xi_i \geq 0$ is the slack variable and $\|\hat{y}_i - y\|_1$ is the Hamming distance between the gold PA graph $\hat{y}_i$ and a candidate PA graph y of the admissible PA graphs $G(x_i)$. Following Zhang et al. (2014), we select the highest scoring graph $\tilde{y}$ as follows:

$$\text{TRAIN} : \tilde{y} = \underset{y \in G(\hat{x}_i)}{\text{argmax}} \{Score(\hat{x}_i, y) + \|\hat{y}_i - y\|_1\}$$
$$\text{TEST} : \tilde{y} = \underset{y \in G(x)}{\text{argmax}} \{Score(x, y)\}$$

Using the weight vector tuned by the training, we perform analysis on a sentence x in the test set.

6 Experiment

6.1 Experimental Settings

Data Set We evaluate our proposed methods on the NAIST Text Corpus 1.5, which consists of 40,000 sentences of Japanese newspaper text (Iida et al., 2007). While previous work has adopted the version 1.4 beta, we adopt the latest version. The major difference between version 1.4 beta and 1.5 is revision of dative case (corresponding to Japanese case particle "ni"). In 1.4 beta, most of adjunct usages of "ni" are mixed up with the argument usages of "ni", making the identification of dative cases seemingly easy. Therefore, our results are not directly comparable with previous work.

We adopt standard train/dev/test split (Taira et al., 2008) as follows:

Train	Articles: Jan 1-11,	Editorials: Jan-Aug
Dev	Articles: Jan 12-13,	Editorials: Sept
Test	Articles: Jan 14-17,	Editorials: Oct-Dec

We exclude inter-sentential arguments (*Inter-Zero*) in our experiments. Our features make use of the annotated POS tags, phrase boundaries, and dependency relations annotated in the NAIST Text Corpus. We do not use any external resources.

Baseline We adopt the pointwise method (using only local features) proposed by Imamura et al. (2009) as the baseline. They built three distinct models corresponding to the three case roles. By using each model, they estimate the likelihood that each candidate argument plays a case role of the target predicate as a score, and independently select the highest scoring one per predicate.

	feature	*Dep*	*Zero*	*Total*
PC Joint	*local*	84.59	42.55	77.89
	+ *global*	85.51	44.54	78.85
AC Joint	*local*	84.17	41.33	77.43
	+ *global*	85.92	44.45	79.17

Table 2: Global vs Local features on the development sets in F-measures. "PC Joint" denotes the Per-Case Joint Model, and "AC Joint" denotes the All-Cases Joint Model.

Features The baseline utilizes the *Baseline Features* used in Imamura et al. (2009) and *Grammatical* features used in Hayashibe et al. (2009), as the "Local Features". In addition, the joint models utilize the "Global Features" in Table 1.

Implementation Details For our joint models with hill-climbing, we report the average performance across ten independent runs with 10 restarts, which almost reaches convergence [3]. We train the baseline and our joint models for 20 iterations with averaged perceptron.

6.2 Results

Local Features vs Global Features

Table 2 shows the effectiveness of the global features on the development sets. We incrementally add the global features to the both models that utilize only the local features. The results show that the global features improve the performance by about 1.0 point in F-measures in total. For and are particularly beneficial to the implicit (*Zero*) argument identification (an improvement of 1.99 points in Per-Case Joint Model and 3.12 points in All-Cases Joint Model).

Pointwise Methods vs Joint Methods

Table 3 presents the F-measures of the baseline and our joint methods on the test set of the NAIST Text Corpus. We used the bootstrap resampling method as the significance test. In most of the metrics, our proposed joint methods outperform the baseline pointwise method. Note that since Per-Case Joint Model yields better results compared with the baseline, capturing the *predicate interaction* is beneficial to Japanese PAS analysis. In addition, the joint methods achieve a considerable improvement of 2.0-2.5 points in F-measure for

[3] Performance did not change when increasing the number of restarts

Case	Type	# of Args.	Baseline	PC Joint	AC Joint
NOM	*Dep*	14055	86.50	87.54 †	**88.13** † ‡
	Zero	4935	45.56	47.62	**48.11**
	Total	18990	77.31	78.39 †	**79.03** † ‡
ACC	*Dep*	9473	92.84 ⋆	**93.09** † ⋆	92.74
	Zero	833	21.38	22.73	**24.43**
	Total	10306	88.86 ⋆	**89.00** † ⋆	88.47
DAT	*Dep*	2518	30.97	34.29 †	**38.39** † ‡
	Zero	239	0.83	0.83	**4.80**
	Total	2757	29.02	32.20 †	**36.35** † ‡
ALL	*Dep*	26046	85.06	85.79 †	**86.07** † ‡
	Zero	6007	41.65	43.60	**44.09**
	Total	32053	78.15	78.91 †	**79.23** † ‡

Table 3: F-measures of the three methods in the test sets. The bold values denote the highest F-measures among all the three methods. Statistical significance with $p < 0.05$ is marked with † compared with Baseline, ‡ compared with PC Joint, and ⋆ compared with AC Joint.

	Dep			*Zero*		
	NOM	ACC	DAT	NOM	ACC	DAT
TA08	75.53	88.20	89.51	30.15	11.41	3.66
IM09	87.0	93.9	80.8	50.0	30.8	0.0
S&K11	-	-	-	39.5	17.5	8.9
PC Joint	87.54	93.09	34.19	47.62	22.73	0.83
AC Joint	88.13	92.74	38.39	48.11	24.44	4.80

Table 4: Comparison with previous work using the NAIST Text Corpus in F-measure. TA08 is Taira et al. (2008), IM09 is Imamura et al. (2009), and S&K11 is Sasano & Kurohashi (2011). Their results are not directly comparable to ours since they use external resources and the NAIST Text Corpus 1.4 beta.

the implicit arguments (*Zero*), one of the problematic issues in Japanese PAS analysis.

Comparing the joint methods, each of our two joint methods is effective for a different case role. Per-Case Joint Model is better at the ACC case, and All-Cases Joint Model is better at the NOM and DAT cases. One of the possible explanations is that the distribution of ACC cases is different from NOM cases. While the ratio of *Dep* and *Zero* arguments for ACC cases is 90:10, the ratio for NOM cases is 75:25. This might have some negative effects on the ACC case identification with All-Cases Joint Model. However, in total, All-Cases Joint Model achieves significantly better results. This suggests that capturing *case interactions* improves performance of Japanese PAS analysis.

Existing Methods vs Joint Methods

To compare our proposed methods with previous work, we pick the three pieces of representative previous work exploiting the NAIST Text Corpus: Taira et al. (2008) (TA08), Imamura et al. (2009) (IM09), and Sasano and Kurohashi (2011) (S&K11). Sasano and Kurohashi (2011) focus on the analysis for the *Zero* and *Inter-Zero* arguments, and do not report the results on the *Dep* arguments. With respect to the *Dep* arguments, the All-Cases Joint Model achieves the best result for the NOM cases, Imamura et al. (2009) the best for the ACC cases, and Taira et al. (2008) the best for the DAT cases. In terms of the *Zero* arguments, Imamura et al. (2009) is the best for the NOM and ACC cases, and Sasano and Kurohashi (2011) the best for the DAT cases. Our joint methods achieve high performance comparable to Imamura et al. (2009).

However, because they used additional external resources and a different version of the NAIST Text Corpus, the results of previous work are not directly comparable to ours. Our research direction and contributions are orthogonal to theirs, and adding their external resources could potentially leads to much better results.

7 Conclusion

We have presented joint methods for Japanese PAS analysis, which model interactions between multiple predicates and arguments using a bipartite graph and greedily search the optimal PAS combination in a sentence. Experimental results shows that capturing the *predicate interaction* and *case interaction* is effective for Japanese PAS analysis. In particular, implicit (*Zero*) argument identification, one of the problematic issues in Japanese PAS analysis, is improved by taking such interactions into account. Since this framework is applicable to the *argument classification* in SRL, applying our methods to that task is an interesting line of the future research. In addition, the final results of our joint methods are comparable to representative existing methods despite using no external resources. For future work, we plan to incorporate external resources for our joint methods.

Acknowledgments

We are grateful to the anonymous reviewers. This work is partially supported by a JSPS KAKENHI Grant Number 26730121 and 15K16053.

References

Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1–8, Philadelphia, July. Association for Computational Linguistics.

Matthew Gerber and Joyce Chai. 2010. Beyond nombank: A study of implicit arguments for nominal predicates. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1583–1592, Uppsala, Sweden, July. Association for Computational Linguistics.

Yuta Hayashibe, Mamoru Komachi, and Yuji Matsumoto. 2011. Japanese predicate argument structure analysis exploiting argument position and type. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 201–209, Chiang Mai, Thailand, November. Asian Federation of Natural Language Processing.

Liang Huang. 2008. Forest reranking: Discriminative parsing with non-local features. In *Proceedings of 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 586–594, Columbus, Ohio, June. Association for Computational Linguistics.

Ryu Iida, Mamoru Komachi, Kentaro Inui, and Yuji Matsumoto. 2007. Annotating a japanese text corpus with predicate-argument and coreference relations. In *Proceedings of the Linguistic Annotation Workshop*, pages 132–139, Prague, Czech Republic, June. Association for Computational Linguistics.

Kenji Imamura, Kuniko Saito, and Tomoko Izumi. 2009. Discriminative approach to predicate-argument structure analysis with zero-anaphora resolution. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and 4th International Joint Conference on Natural Language Processing*, pages 85–88, Suntec, Singapore, August. Association for Computational Linguistics.

Egoitz Laparra and German Rigau. 2013. Impar: A deterministic algorithm for implicit semantic role labelling. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1180–1189, Sofia, Bulgaria, August. Association for Computational Linguistics.

Ryan McDonald and Fernando Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *Proceedings of the 11th conference on European Chapter of the Association for Computational Linguistics (EACL)*, pages 81–88, Trento, Italy, April. Association for Computational Linguistics.

Ryohei Sasano and Sadao Kurohashi. 2011. A discriminative approach to japanese zero anaphora resolution with large-scale lexicalized case frames. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 758–766, Chiang Mai, Thailand, November. Asian Federation of Natural Language Processing.

Hirotoshi Taira, Sanae Fujita, and Masaaki Nagata. 2008. A japanese predicate argument structure analysis using decision lists. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 523–532, Honolulu, Hawaii, October. Association for Computational Linguistics.

Haitong Yang and Chengqing Zong. 2014. Multi-predicate semantic role labeling. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 363–373, Doha, Qatar, October. Association for Computational Linguistics.

Yuan Zhang, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2014. Greed is good if randomized: New inference for dependency parsing. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1013–1024, Doha, Qatar, October. Association for Computational Linguistics.