Neural Network-Based Model for Japanese Predicate Argument Structure Analysis

Tomohide Shibata and **Daisuke Kawahara** and **Sadao Kurohashi**
Graduate School of Informatics, Kyoto University
Yoshida-honmachi, Sakyo-ku, Kyoto, 606-8501, Japan
{shibata, dk, kuro}@i.kyoto-u.ac.jp

Abstract

This paper presents a novel model for Japanese predicate argument structure (PAS) analysis based on a neural network framework. Japanese PAS analysis is challenging due to the tangled characteristics of the Japanese language, such as case disappearance and argument omission. To unravel this problem, we learn selectional preferences from a large raw corpus, and incorporate them into a SOTA PAS analysis model, which considers the consistency of all PASs in a given sentence. We demonstrate that the proposed PAS analysis model significantly outperforms the base SOTA system.

1 Introduction

Research on predicate argument structure (PAS) analysis has been conducted actively these days. The improvement of PAS analysis would benefit many natural language processing (NLP) applications, such as information extraction, summarization, and machine translation.

The target of this work is Japanese PAS analysis. The Japanese language has the following characteristics:

- head final,
- free word order (among arguments), and
- postpositions function as (surface) case markers.

Japanese major surface cases are が (*ga*), を (*wo*), and に (*ni*), which correspond to Japanese postpositions (case markers). We call them nominative case, accusative case, and dative case, respectively. In this paper, we limit our target cases to these three cases. Note that though they are surface cases, they roughly correspond to Arg1, Arg2, and Arg3 of English semantic role labeling based on PropBank.

Japanese PAS analysis has been considered as one of the most difficult basic NLP tasks, due to the following two phenomena.

Case disappearance When a topic marker は (*wa*) is used or a noun is modified by a relative clause, their case markings disappear as in the following examples.[1]

(1) a. ジョンは パンを 食べた。 → ジョンが
John-TOP bread-ACC ate John-NOM
(John ate bread.)

b. パンは ジョンが 食べた。 → パンを
bread-TOP John-NOM ate bread-ACC
(John ate bread.)

(2) a. パンを 食べた ジョン を ... → ジョンが (食べた)
bread-ACC ate John-ACC John-NOM (ate)
(John, who ate bread, ...)

b. ジョンが 食べた パン が ...→ パンを (食べた)
John-NOM ate bread-NOM (ate) bread-ACC
(Bread, which John ate, ...)

In the example sentences (1a) and (1b), since a topic marker は is used, the NOM and ACC case markers disappear. In the example sentences (2a) and (2b), since a noun is modified by a relative clause, the NOM case of "ジョン" (John) for "食べた" (eat) and ACC case of "パン" (bread) for "食べた" disappear.

Argument omission Arguments are very often omitted in Japanese sentences. This phenomenon is totally different from English sentences, where the word order is fixed and pronouns are used con-

[1] In this paper, we use the following abbreviations: NOM (nominative), ACC (accusative), DAT (dative) and TOP (topic marker).

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 1235–1244,
Berlin, Germany, August 7-12, 2016.

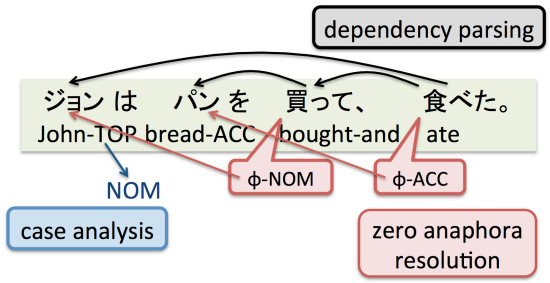

Figure 1: An example of PAS analysis. Input sentence: “ジョンはパンを買って、食べた。” (John bought bread, and ate it.)

sistently. For example, let us compare the following parallel Japanese and English sentences:

(3) a. ジョンは パンを 買って、 食べた。
John-TOP bread-ACC bought ate
b. John bought bread, and ate it.

The dependency parse of (3a) is shown in Figure 1. In general, the first phrase with a topic marker は is treated as modifying the final predicate according to the guidelines of Japanese dependency annotation. As a result, “買って” (bought) has no NOM argument (omitted), and “食べた” (ate) has no ACC argument. Note that “食べた” has an argument “ジョン” (John), but its case does not appear.

In the case of the parallel sentences (4) below, again we can witness the difficulty of Japanese PAS analysis.

(4) a. パンを 買った ジョンは 急いで 食べた。
bread-ACC bought John-TOP hurry ate
b. John who bought bread ate it in a hurry.

Although all the case arguments of the predicates “bought” and “ate” are explicit in (4b), the case of “ジョン” (John) for “買った” (bought) and that for “食べた” (ate) are hidden, and the ACC argument of “食べた” (ate) is omitted in (4a).

Many researchers have been tackling Japanese PAS analysis (Taira et al., 2008; Imamura et al., 2009; Hayashibe et al., 2011; Sasano and Kurohashi, 2011; Hangyo et al., 2013; Ouchi et al., 2015). However, because of the two aforementioned characteristics in Japanese sentences, the accuracy of Japanese PAS analysis for omitted (zero) arguments remains around 40%.

This paper proposes a novel Japanese PAS analysis model based on a neural network (NN) framework, which has been proved to be effective for several NLP tasks recently. To unravel the tangled situation in Japanese, we learn selectional preferences from a large raw corpus, and incorporate them into a SOTA PAS analysis model proposed by Ouchi et al. (2015), which considers the consistency of all PASs in a given sentence. This model is achieved by an NN-based two-stage model that acquires selectional preferences in an unsupervised manner in the first stage and predicts PASs in a supervised manner in the second stage as follows.

1. The most important clue for PAS analysis is selectional preferences, that is, argument prediction from a predicate phrase. For example, how likely the phrase “パンを買った” (bought bread) takes “ジョン” (John) as its NOM argument.

   Such information cannot be learned from a medium-sized PAS annotated corpus with size of the order of ten-thousand sentences; it is necessary to use a huge raw corpus by an unsupervised method. Ouchi et al. (2015) did not utilize such knowledge extracted from a raw corpus. Some work has utilized PMI between a predicate and an argument, or case frames obtained from a raw corpus. However, this is discrete word-based knowledge, not generalized semantic knowledge.

   As the first stage of the method, we learn a prediction score from a predicate phrase to an argument by an NN-based method. The resultant vector representations of predicates and arguments are used as initial vectors for the second stage of the method.

2. In the second stage, we calculate a score that a predicate in a given sentence takes an element in the sentence as an argument using NN framework. We use the prediction score in the first stage as one feature for the second stage NN. The system by Ouchi et al. (2015) used a manually designed feature template to take the interactions of the atomic features into consideration. In the case of an NN framework, no feature template is required, and a hidden layer in an NN can capture the interactions of the atomic features automatically and flexibly.

We demonstrate that the proposed PAS analysis model outperforms the SOTA system by Ouchi et al. (2015).

2 Related Work

Several methods for Japanese PAS analysis have been proposed. The methods can be divided into three types: (i) identifying one case argument independently per predicate (Taira et al., 2008; Imamura et al., 2009; Hayashibe et al., 2011), (ii) identifying all the three case arguments (NOM, ACC, and DAT) simultaneously per predicate (Sasano and Kurohashi, 2011; Hangyo et al., 2013), and (iii) identifying all case arguments of all predicates in a sentence (Ouchi et al., 2015). The third method can capture interactions between predicates and their arguments, and thus performs the best among the three types. This method is adopted as our base model (see Section 3 for details).

Most methods for PAS analysis handle both intra-sentential and inter-sentential zero anaphora. For identifying inter-sentential zero anaphora, an antecedent has to be searched in a broad search space, and the salience of discourse entities has to be captured. Therefore, the task of identifying inter-sentential zero anaphora is more difficult than that of intra-sentential zero anaphora. Thus, Ouchi et al. (2015) and Iida et al. (2015) focused on only intra-sentential zero anaphora. Following this trend, this paper focuses on intra-sentential zero anaphora.

Recently, NN-based approaches have achieved improvement for several NLP tasks. For example, in transition-based parsing, Chen and Manning (2014) proposed an NN-based approach, where the words, POS tags, and dependency labels are first represented by embeddings individually. Then, an NN-based classifier is built to make parsing decisions, where an input layer is a concatenation of embeddings of words, POS tags, and dependency labels. This model has been extended by several studies (Weiss et al., 2015; Dyer et al., 2015; Ballesteros et al., 2015). In semantic role labeling, Zhou and Xu (2015) propose an end-to-end approach using recurrent NN, where an original text is the input, and semantic role labeling is performed without any intermediate syntactic knowledge. Following these approaches, this paper proposes an NN-based PAS method.

3 Base Model

The model proposed by Ouchi et al. (2015) is adopted as our base model (Figure 2). We briefly introduce this base model before describing our proposed model.

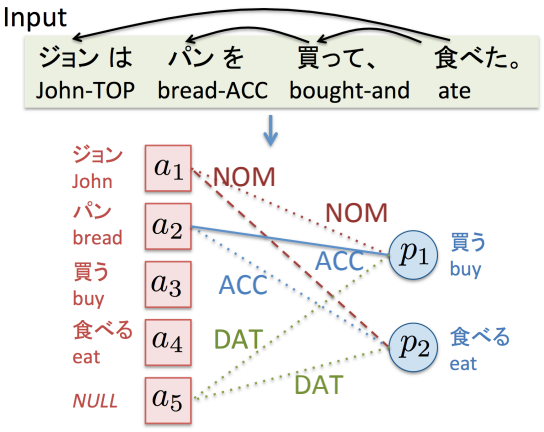

Figure 2: Our base model (Ouchi et al., 2015).

3.1 Predicate-Argument Graph

In this model, for an input sentence, a bipartite graph is constructed, consisting of the set of predicate and argument nodes. This is called Predicate-Argument Graph (PA Graph). A PA graph represents a possible interpretation of the input sentence, including case analysis result and zero anaphora resolution result.

A PA graph is a bipartite graph $\langle A, P, E \rangle$, where A is the node set consisting of candidate arguments, P is the node set consisting of predicates, and E is the set of edges. A PA graph is defined as follows:

$$\begin{aligned} A &= \{a_1, \ldots, a_n, a_{n+1} = \texttt{NULL}\} \\ P &= \{p_1, \ldots, p_m\} \\ E &= \{\langle a, p, c \rangle | deg(p, c) = 1, \\ &\quad \forall a \in A, \forall p \in P, \forall c \in C\} \end{aligned}$$

where n and m represent the number of predicates and arguments, and C denotes the case role set (NOM, ACC, and DAT). An edge $e \in E$ is represented by a tuple $\langle a, p, c \rangle$, indicating the edge with a case role c connecting a candidate argument node a and a predicate node p. $deg(p, c)$ is the number of the edges with a case role c outgoing from a predicate node p. An admissible PA graph satisfies the constraint $deg(p, c) = 1$, which means each predicate node p has only one edge with a case role c. A dummy node a_{n+1} is added, which is defined for the cases where a predicate requires no case argument (e.g. when the predicate node "ある" (exist) connects a NULL node with a case ACC, this means this predicate takes

no ACC argument) or the required argument does not appear in the sentence.

In the bipartite graph shown in Figure 2, the three kinds of edge lines have the meaning as follows:

solid line: the argument node and the predicate node has a dependency relation, and the argument node is followed by a case marking postposition. In this case, these nodes have a relation through its corresponding case marking postposition. Therefore, this edge is fixed.

dashed line: the argument node and the predicate node has a dependency relation, and the argument node is not followed by a case marking postposition. These nodes are likely to have a relation[2], but the case role is unknown. Identifying this case role corresponds to case analysis.

dotted line: the argument node and the predicate node do not have a dependency relation. Identifying this edge and its case role corresponds to zero anaphora resolution.

For an input sentence x, a scoring function $Score(x, y)$ is defined for a candidate graph y, and the PA graph that has the maximum score is searched.

$$\tilde{y} = \underset{y \in G(x)}{\operatorname{argmax}} Score(x, y) \tag{1}$$

where $G(x)$ is a set of admissible PA graphs for the input sentence x. $Score(x, y)$ is defined as follows[3]:

$$\sum_{e \in E(y)} score_l(x, e) + \sum_{e_i, e_j \in E_{pair}(y)} score_g(x, e_i, e_j). \tag{2}$$

$$\begin{aligned} score_l(x, e) &= \boldsymbol{\theta}_l \cdot \boldsymbol{\phi}_l(x, e) \\ score_g(x, e_i, e_j) &= \boldsymbol{\theta}_g \cdot \boldsymbol{\phi}_g(x, e_i, e_j) \end{aligned} \tag{3}$$

where $E(y)$ is the edge set on the candidate graph y, $E_{pair}(y)$ is a set of edge pairs in the edge set $E(y)$, $score_l(x, e)$ and $score_g(x, e_i, e_j)$ represent a local score for the edge e and a global score for the edge pair e_i and e_j, $\boldsymbol{\phi}_l(x, e)$ and $\boldsymbol{\phi}_g(x, e_i, e_j)$ represent local features and global features. While $\boldsymbol{\phi}_l(x, e)$ is defined for each edge e, $\boldsymbol{\phi}_g(x, e_i, e_j)$ is defined for each edge pair e_i, e_j $(i \neq j)$. $\boldsymbol{\theta}_l$ and $\boldsymbol{\theta}_g$ represent model parameters for local and global features. By using global scores, the interaction between multiple case assignments of multiple predicates can be considered.

[2]For example, in the sentence "今日は 暑い" (today-TOP hot), the predicate "暑い" does not take "今日", which represents time, as an argument. Therefore, these nodes do not always have a relation.

[3]Ouchi et al. (2015) introduce two models: Per-Case Joint Model and All-Cases Joint Model. Since All-Cases Joint Model performed better than Per-Case Joint Model, All-Cases Joint Model is adopted as our base model.

3.2 Inference and Training

Since global features make the inference of finding the maximum scoring PA graph more difficult, the randomized hill-climbing algorithm proposed in (Zhang et al., 2014) is adopted.

Figure 3 describes the pseudo code for hill-climbing algorithm. First, an initial PA graph $y^{(0)}$ is sampled from the set of admissible PA graph $G(x)$. Then, the union Y is constructed from the set of neighboring graphs $NeighborG(y^{(t)})$, which is a set of admissible graphs obtained by changing one edge in $y^{(t)}$, and the current graph $y^{(t)}$. The current graph $y^{(t)}$ is updated to a higher scoring graph $y^{(t+1)}$. This process continues until no more improvement is possible, and finally an optimal graph $\tilde{y}$ can be obtained.

Input: sentence x, parameter $\boldsymbol{\theta}$
Output: a locally optimal PA graph $\tilde{y}$
Sample a PA graph $y^{(0)}$ from $G(x)$
$t \leftarrow 0$
repeat
  $Y \leftarrow NeighborG(y^{(t)}) \bigcup y^{(t)}$
  $y^{(t+1)} \leftarrow \underset{y \in Y}{\operatorname{argmax}} Score(x, y; \boldsymbol{\theta})$
  $t \leftarrow t + 1$
until $y^{(t)} = y^{(t+1)}$
return $\tilde{y} \leftarrow y^{(t)}$

Figure 3: Hill climbing algorithm for obtaining optimal PA graph.

Given N training examples $D = \{(x, \hat{y})\}_k^N$, the model parameter $\boldsymbol{\theta}$ are estimated. $\boldsymbol{\theta}$ is the set of $\boldsymbol{\theta}_l$ and $\boldsymbol{\theta}_g$, and is estimated by averaged perceptron (Collins, 2002) with a max-margin framework (Taskar et al., 2005).

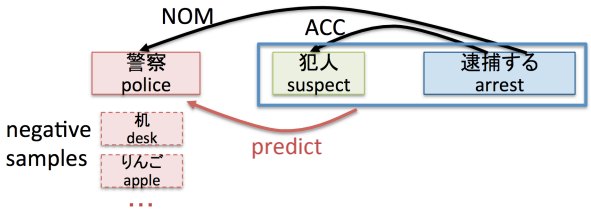

Figure 4: Argument prediction model. In the PAS "警察" (police) NOM "犯人" (suspect) ACC "逮捕" (arrest), "警察" with the NOM case is predicted given the predicate "逮捕" (arrest) and its ACC "犯人" (suspect).

4 Proposed Model

4.1 Argument Prediction Model

No external knowledge is utilized in the base model. One of the most important types of knowledge in PAS analysis is selectional preferences. Sasano and Kurohashi (2011) and Hangyo et al. (2013) extract knowledge of the selectional preferences in the form of case frames from a raw corpus, and the selectional preference score is used as a feature. In this work, *argument prediction model* is trained using a neural network from a raw corpus, in a similar way to Titov and Khoddam (2015) and Hashimoto et al. (2014).

PASs are first extracted from an automatically-parsed raw corpus, and in each PAS, the argument a_i is generated with the following probability $p(a_i|PAS_{-a_i})$:

$$p(a_i|PAS_{-a_i}) = \frac{\exp(\boldsymbol{v}_{a_i}^{\mathrm{T}} W_{a_i}^{\mathrm{T}}(W_{pred}\boldsymbol{v}_{pred} + \sum_{j\neq i} W_{a_j}\boldsymbol{v}_{a_j}))}{Z} \quad (4)$$

where PAS_{-a_i} represents a PAS excluding the target argument a_i, $\boldsymbol{v}_{pred}$, $\boldsymbol{v}_{a_i}$ and $\boldsymbol{v}_{a_j}$ represent embeddings of the predicate, argument a_i and argument a_j, and W_{pred}, W_{a_i}, and W_{a_j} represent transformation matrices for a predicate and an argument a_i and a_j. Z is the partition function.

Figure 4 illustrates the argument prediction model. The PAS "警察" (police) NOM "犯人" (suspect) ACC "逮捕" (arrest)" is extracted from a raw corpus, and the probability of NOM argument "警察" given the predicate "逮捕" and its ACC argument "犯人" is calculated.

All the parameters including predicate/argument embeddings and transformation matrices are trained, so that the likelihood given by Equation (4) is high. Since the denominator of Equation (4) is impractical to be calculated since the number of vocabulary is enormous, negative sampling (Mikolov et al., 2013) is adopted. In the example shown in Figure 4, as for a NOM argument, negative examples, such as "机" (desk) and "りんご" (apple), are drawn from the noise distribution, which is a unigram distribution raised to the 3/4th power.

In each PAS, all the arguments are predicted in turn. All the parameters are updated using stochastic gradient descent.

This model is first trained using the automatic parsing result on a raw corpus, and in performing PAS analysis described in Section 4.2, the score derived from this model is used as a feature.

4.2 Neural Network-Based Score Calculation

In the base model, the score for an edge (local score) or an edge pair (global score) is calculated using the dot product of a sparse high-dimensional feature vector with a model parameter, as shown in Equation (3). In our proposed model, these scores are calculated in a standard neural network with one hidden layer, as shown in Figure 5.

We first describe the calculation of the local score $score_l(x, e)$. A predicate p and an argument a are represented by embeddings (a d dimensional vector) $\boldsymbol{v}_p$ and $\boldsymbol{v}_a \in \mathbb{R}^d$, and $\boldsymbol{v}_{f_l} \in \mathbb{R}^{d_f}$ (d_f represents a dimensional of $\boldsymbol{v}_{f_l}$) represents a feature vector obtained by concatenating the case role between a and p, the argument prediction score obtained from the model described in Section 4.1, and the other atomic features. An input layer is a concatenation of these vectors, and then, a hidden layer $\boldsymbol{h}_l \in \mathbb{R}^{d_h}$ (d_h represents a dimension of the hidden layer) is calculated as follows:

$$\boldsymbol{h}_l = f(W_l^1[\boldsymbol{v}_p; \boldsymbol{v}_a; \boldsymbol{v}_{f_l}]) \quad (5)$$

where f is an element-wise activation function ($tanh$ is used in our experiments), and $W_l^1 \in \mathbb{R}^{d_h(2d+d_h)}$ is a weight matrix (for the local score) from the input layer to the hidden layer. The scalar score in an output layer is then calculated as follows:

$$score_l(x, e) = f(W_l^2 \boldsymbol{h}_l) \quad (6)$$

where $W_l^2 \in \mathbb{R}^{(2d+d_h)\cdot 1}$ is a weight matrix (for the local score) from the hidden layer to the output layer. By calculating the score in this way, all the

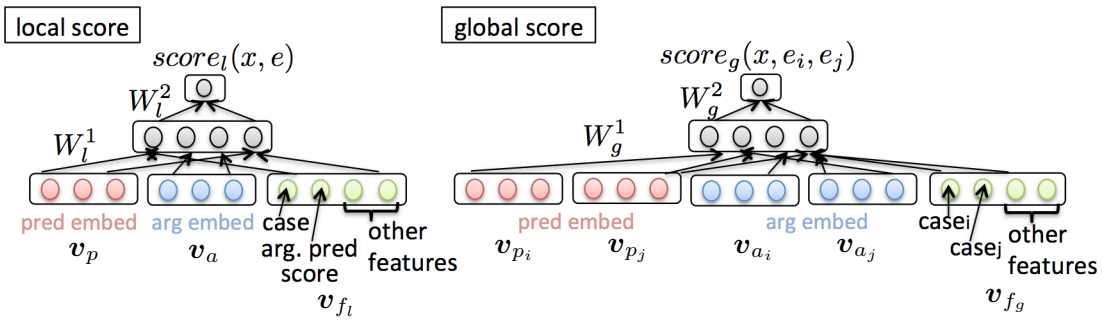

Figure 5: A score calculation in our proposed neural-network based model. The left part and right part represent a local and global score calculation.

combinations of features in the input layer can be considered.

Next we describe the calculation of the global score $score_g(x, e_i, e_j)$. In the base model, the two types of global features are utilized: one is for the two predicates having different arguments, and the other is for the two predicates sharing the same argument. The input layer is a concatenation of involving vectors of predicates/arguments and the other features $\boldsymbol{v}_{f_g}$. For example, when calculating the global score for the two predicates having different arguments, the input layer is a concatenation of the vectors of two predicates and two arguments and $\boldsymbol{v}_{f_g}$.

A hidden layer $\boldsymbol{h}_g$ is calculated as follows:

$$\boldsymbol{h}_g = f(W_g^1[\boldsymbol{v}_{p_i}; \boldsymbol{v}_{p_j}; \boldsymbol{v}_{a_i}; \boldsymbol{v}_{a_j}; \boldsymbol{v}_{f_g}]) \quad (7)$$

where W_g^1 is a weight matrix (for the global score) from the input layer to the hidden layer, $\boldsymbol{v}_{p_i}$ and $\boldsymbol{v}_{a_i}$ are the embeddings of the predicate/argument connected by e_i, and $\boldsymbol{v}_{p_j}$ and $\boldsymbol{v}_{a_j}$ are defined in the same way.

The scalar score in an output layer is then calculated as follows:

$$score_g(x, e_i, e_j) = f(W_g^2 \boldsymbol{h}_g) \quad (8)$$

where W_g^2 is a weight matrix (for the global score) from the hidden layer to the output layer.

4.3 Inference and Training

While inference is the same as the base model, training is slightly different.

In our proposed model, the model parameter $\boldsymbol{\theta}$ consists of the embeddings of predicates/arguments and weight matrices for the local/global score in the neural networks. Our objective is to minimize the following loss function:

case	# of dep arguments	# of zero arguments	total
NOM	1,402	1,431	2,833
ACC	278	113	391
DAT	92	287	379
ALL	1,772	1,831	3,603

Table 1: Test set statistics of the number of arguments.

$$J(\boldsymbol{\theta}) = \sum_k^N l_k(\boldsymbol{\theta}), \quad (9)$$

where

$$l_k(\boldsymbol{\theta}) = \max_{y_k \in G(x)} (Score(x_k, y_k; \boldsymbol{\theta}) - Score(x_k, \hat{y}_k; \boldsymbol{\theta}) + ||y_k - \hat{y}_k||_1), \quad (10)$$

and $||y_k - \hat{y}_k||_1$ denotes the Hamming distance between the gold PA graph $\hat{y}_k$ and a candidate PA graph y_k.

Stochastic gradient descent is used for parameter inference. Derivatives with respect to parameters are taken using backpropagation. Adam (Kingma and Ba, 2014) is adopted as the optimizer.

For initialization of the embeddings of a predicate/argument, the embeddings of the predicate/argument trained by the method described in Section 4.1 are utilized. The weight matrices are randomly initialized.

5 Experiment

5.1 Experimental Setting

The KWDLC (Kyoto University Web Document Leads Corpus) evaluation set (Hangyo et al., 2012) was used for our experiments, because it contains

a wide variety of Web documents, such as news articles and blogs. This evaluation set consists of the first three sentences of 5,000 Web documents. Morphology, named entities, dependencies, PASs, and coreferences were manually annotated.

This evaluation set was divided into 3,694 documents (11,558 sents.) for training, 512 documents (1,585 sents.) for development, and 700 documents (2,195 sents.) for testing. Table 1 shows the statistics of the number of arguments in the test set. While "dep argument" means that the argument and a predicate have a dependency relation, but a specified case marking postposition is hidden (corresponds to "dashed line" in Section 3.1), "zero argument" means that the argument and a predicate do not have a dependency relation (corresponds to "dotted line" in Section 3.1).

Since we want to focus on the accuracy of case analysis and zero anaphora resolution, gold morphological analysis, dependency analysis, and named entities were used.

The sentences having a predicate that takes multiple arguments in the same case role were excluded from training and test examples, since the base model cannot handle this phenomena (it assumes that each predicate has only one argument with one case role). For example, the following sentence,

(5) そんな 面白ネタ 満載な
such funny-material full
日々を 絵と共に
daily life-ACC picture-with
お届けします。,
report
(I report my daily life full of such funny materials along with pictures.)

where the predicate "お届けします" (report) takes both "日々" (daily life) and "絵" (picture) as ACC case arguments, was excluded from training and testing. About 200 sentences (corresponding to about 1.5% of the whole evaluation set) were excluded.

In this evaluation set, zero exophora, which is a phenomenon that a referent does not appear in a document, is annotated. Among five types of zero exophora, the two major types, "author" and "reader," are adopted, and the others are discarded. To consider "author" and "reader" as a referent, the two special nodes, AUTHOR and READER, are added as well as a NULL node in a PA graph of the base model. When the argument predication score is calculated for "author" or "reader," because its lemma does not appear in a document, for each noun in the following noun list of "author"/"reader" (Hangyo et al., 2013), the argument prediction score is calculated, and the maximum score is used as a feature.

- author: "私" (I), "我々" (we), "僕" (I), "弊社" (our company), · · ·
- reader: "あなた" (you), "客" (customer), "君" (you), "皆様"(you all), · · ·

In the argument prediction model training described in Section 4.1, a Japanese Web corpus consisting of 10M sentences was used. We preformed syntactic parsing with a publicly available Japanese parser, KNP[4]. The number of negative samples was 5, and the number of epochs was 10.

In the model training described in Section 4.3, the dimensions of both embeddings for predicates/arguments and hidden layer were set to 100. The number of epochs was set to 20, following the base model.

5.2 Result

We compared the following three methods:

- Baseline (Ouchi et al., 2015)
- Proposed model w/o arg. prediction score: in the PAS analysis model, the feature derived from the argument prediction model was not utilized. The embeddings of a predicate/argument were randomly initialized. This method corresponds to adopting the NN-based score calculation in the base model.
- Proposed model w/ arg. prediction score: the feature derived from the argument prediction model was utilized, and the embeddings of a predicate/argument were initialized with those obtained in the argument prediction model learning.

The performances of case analysis and zero anaphora resolution were evaluated by micro-averaged precision, recall, and F-measure. The precision, recall, and F-measure were averaged

[4]http://nlp.ist.i.kyoto-u.ac.jp/index.php?KNP

case	method	case analysis			zero anaphora		
		P	R	F	P	R	F
NOM	Baseline	0.880	0.868	0.874	**0.693**	0.377	0.488
	Proposed model w/o arg. prediction score	0.927	0.917	0.922	0.559	0.532	0.545
	Proposed model w arg. prediction score	**0.946**	**0.936**	**0.941**	0.568	**0.586**	**0.577**
ACC	Baseline	0.433	0.374	0.402	0.000	0.000	0.000
	Proposed model w/o arg. prediction score	0.805	0.553	0.656	0.151	0.060	0.085
	Proposed model w/ arg. prediction score	**0.890**	**0.658**	**0.756**	**0.297**	**0.124**	**0.173**
DAT	Baseline	0.224	**0.359**	0.276	0.531	0.059	0.107
	Proposed model w/o arg. prediction score	0.512	0.104	0.173	0.535	0.242	0.332
	Proposed model w/ arg. prediction score	**0.834**	0.185	**0.300**	**0.622**	**0.273**	**0.378**
ALL	Baseline	0.765	0.764	0.765	**0.686**	0.304	0.421
	Proposed model w/o arg. prediction score	0.908	0.818	0.860	0.544	0.458	0.497
	Proposed model w/ arg. prediction score	**0.937**	**0.853**	**0.893**	0.563	**0.509**	**0.534**

Table 2: Experimental results on the KWDLC corpus.

over 5 runs. Table 2 shows our experimental results. Our proposed method outperformed the baseline method by about 11 absolute points in F-measure. The comparison of "Proposed model w/o arg. prediction score" with the baseline showed that the neural network-based approach was effective, and the comparison of "Proposed model w/ arg. prediction score" with "Proposed model w/o arg. prediction score" showed that our arg. prediction model was also effective.

The following is improved by adding an argument prediction score.

(6) 久しぶりに　　パートですけど、
after a long time part-time job
働き始めて　新しい 一歩を
begin to work new　　step-ACC
踏み出しました。
step forward
(It's my first part-time job in a long time. I begin to work, and make a new step.)

While in the base model, the NOM arguments of the predicate "働き始める" (begin to work) and "踏み出す" (step forward) were wrongly classified as NULL, by adding an argument prediction score, they were correctly identified as "author."

The phenomenon "case disappearance" occurs in other languages such as Korean, and the phenomenon "argument omission" occurs in other languages such as Korean, Hindi, Chinese, and Spanish. We believe that our neural network approach to the argument prediction and the calculation of the local and global scores is also effective for such languages.

5.3 Error Analysis

Errors in our proposed model are listed below:

- Recall for ACC and DAT in both case analysis and zero anaphora resolution is low.

  One reason is that since the number of the ACC and DAT arguments is smaller than that of the NOM argument, the system tends to assign the ACC and DAT arguments with NULL. Another reason is that since this paper focuses on intra-sentential zero anaphora, the NULL arguments include arguments that appear in previous sentences as well as the case where a predicate takes no argument, which makes the training for NULL arguments difficult. We are planing to tackle with inter-sentential zero anaphora resolution.

- The distinction of "author" from NULL fails.

  (7) 肉を　　焼く だけが
  meat-ACC roast-only-NOM
  BBQ じゃない！
  BBQ-(COPULA)
  (Roasting meat isn't all in BBQ!)

  Although the NOM argument of the predicate "焼く" (roast) is "author," our proposed model wrongly classified it as NULL. Hangyo et al. (2013) identify mentions referring to an author or reader in a document, and utilize this result in the zero anaphora resolu-

tion. We plan to incorporate the author/reader identification into our model.

6 Conclusion

In this paper we presented a novel model for Japanese PAS analysis based on neural network framework. We learned selectional preferences from a large raw corpus, and incorporated them into a PAS analysis model, which considers the consistency of all PASs in a given sentence. In our experiments, we demonstrated that the proposed PAS analysis model significantly outperformed the base SOTA model.

In the future, we plan to extend our model to incorporate coreference resolution and inter-sentential zero anaphora resolution.

Acknowledgments

This work is supported by CREST, Japan Science and Technology Agency.

References

Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2015. Improved transition-based parsing by modeling characters instead of words with lstms. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 349–359, Lisbon, Portugal, September. Association for Computational Linguistics.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar, October. Association for Computational Linguistics.

Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 334–343, Beijing, China, July. Association for Computational Linguistics.

Masatsugu Hangyo, Daisuke Kawahara, and Sadao Kurohashi. 2012. Building a diverse document leads corpus annotated with semantic relations. In *Proceedings of the 26th Pacific Asia Conference on Language, Information, and Computation*, pages 535–544, Bali,Indonesia, November. Faculty of Computer Science, Universitas Indonesia.

Masatsugu Hangyo, Daisuke Kawahara, and Sadao Kurohashi. 2013. Japanese zero reference resolution considering exophora and author/reader mentions. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 924–934, Seattle, Washington, USA, October. Association for Computational Linguistics.

Kazuma Hashimoto, Pontus Stenetorp, Makoto Miwa, and Yoshimasa Tsuruoka. 2014. Jointly learning word representations and composition functions using predicate-argument structures. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1544–1555, Doha, Qatar, October. 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.

Ryu Iida, Kentaro Torisawa, Chikara Hashimoto, Jong-Hoon Oh, and Julien Kloetzer. 2015. Intra-sentential zero anaphora resolution using subject sharing recognition. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2179–2189, Lisbon, Portugal, September. 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 ACL-IJCNLP 2009 Conference Short Papers*, pages 85–88, Suntec, Singapore, August. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Hiroki Ouchi, Hiroyuki Shindo, Kevin Duh, and Yuji Matsumoto. 2015. Joint case argument identification for Japanese predicate argument structure analysis. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*,

pages 961–970, Beijing, China, July. 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*, pages 523–532, Honolulu, Hawaii, October. Association for Computational Linguistics.

Ben Taskar, Vassil Chatalbashev, Daphne Koller, and Carlos Guestrin. 2005. Learning structured prediction models: A large margin approach. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 896–903, New York, NY, USA. ACM.

Ivan Titov and Ehsan Khoddam. 2015. Unsupervised induction of semantic roles within a reconstruction-error minimization framework. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1–10, Denver, Colorado, May–June. Association for Computational Linguistics.

David Weiss, Chris Alberti, Michael Collins, and Slav Petrov. 2015. Structured training for neural network transition-based parsing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 323–333, Beijing, China, July. 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.

Jie Zhou and Wei Xu. 2015. End-to-end learning of semantic role labeling using recurrent neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1127–1137, Beijing, China, July. Association for Computational Linguistics.