Distance-Free Modeling of Multi-Predicate Interactions in End-to-End Japanese Predicate-Argument Structure Analysis

Yuichiroh Matsubayashi[♠◇] and **Kentaro Inui**[♠◇]
[♠]Graduate School of Information Sciences, Tohoku University
[◇]RIKEN Center for Advanced Intelligence Project
{y-matsu, inui}@ecei.tohoku.ac.jp

Abstract

Capturing interactions among multiple predicate-argument structures (PASs) is a crucial issue in the task of analyzing PAS in Japanese. In this paper, we propose new Japanese PAS analysis models that integrate the label prediction information of arguments in multiple PASs by extending the input and last layers of a standard deep bidirectional recurrent neural network (bi-RNN) model. In these models, using the mechanisms of pooling and attention, we aim to directly capture the potential interactions among multiple PASs, without being disturbed by the word order and distance. Our experiments show that the proposed models improve the prediction accuracy specifically for cases where the predicate and argument are in an indirect dependency relation and achieve a new state of the art in the overall F_1 on a standard benchmark corpus.

1 Introduction

A predicate-argument structure (PAS) is a structure that represents the relationships between a predicate and its arguments. Identifying PASs in Japanese text is a long-standing challenge chiefly due to the abundance of omitted (elliptical) arguments. In the example in Figure 1, the dative relation between *answer* and *reporters* is not explicitly indicated by the syntactic structure of the sentence. We regard such arguments as elliptical and call those argument slots *Zero* cases. 25% of the obligatory arguments in Japanese newspaper articles are reported to be elliptical.[1] The accuracy of identifying the fillers of such Zero cases remains only around 50% in terms of F_1 even if the task is restricted to the identification of intra-sentential predicate-argument relations (Matsubayashi and Inui, 2017).

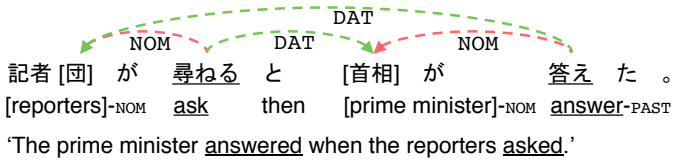
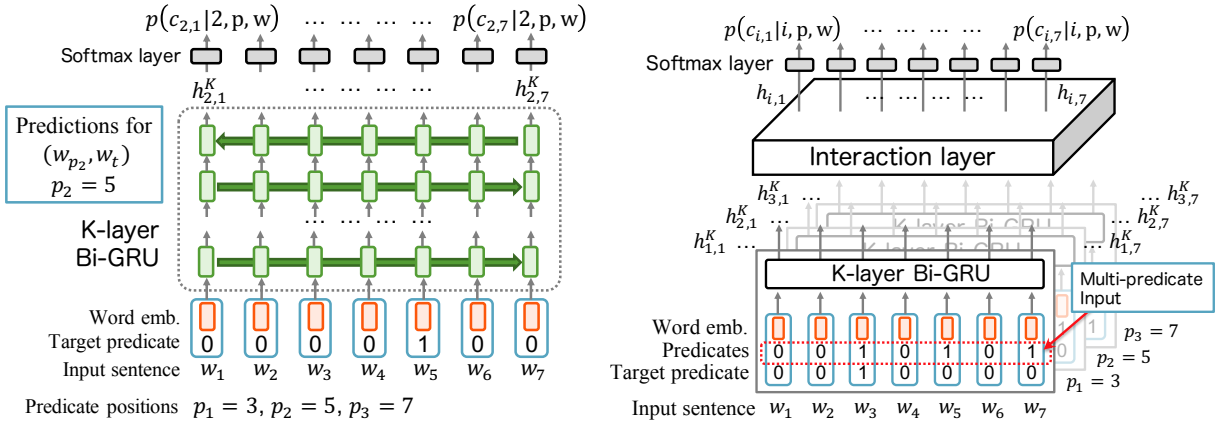


Figure 1: Example of PAS analysis. The dashed lines represent the predicate-argument relations. "[reporters]-NOM ask then" constitutes a subordinate clause and "[prime minister]-NOM answer-PAST" constitutes a matrix clause.

One promising approach for addressing this problem is to model argument sharing across multiple predicates (Iida et al., 2015; Ouchi et al., 2015; Ouchi et al., 2017). In Figure 1, for example, one can find very limited syntactic clues for predicting the long-distance dative relation between *answer* and *reporters*. However, the relation must be easy to identify for human readers who know that *the person who asks a question is likely to be answered*; namely, the nominative argument of *ask* is likely to be shared with the dative argument of *answer*. Capturing such inter-predicative dependencies has, therefore, been considered crucial of Japanese PAS analysis.

[1]Statistics from the NAIST Text Corpus 1.5. (Iida et al., 2017)

Proceedings of the 27th International Conference on Computational Linguistics, pages 94–106
Santa Fe, New Mexico, USA, August 20-26, 2018.



(a) Base model (b) Proposed models

Figure 2: Network structures of the base and proposed models.

With this goal in mind, Iida et al. (2015) constructed a *subject-shared predicate network* with an accurate recognizer of subject-sharing relations and deterministically propagated the predicted subjects to the other predicates in the graph. However, this method is applied only to subject sharing, so it cannot take into account the relationships among multiple argument labels.

More recently, as an end-to-end model considering multi-predicate dependencies, Ouchi et al. (2017) used Grid RNN to incorporate intermediate representations of the prediction for one predicate generated by an RNN layer into the inputs of the RNN layer for another predicate. However, in this model, since the information of multiple predicates also propagates through the RNNs, the integration of the prediction information is influenced by word order and distance, which is not necessarily important for aspects of syntactic and semantic relations. Consequently, there might be information loss caused by the surface distances of words, as previous work had pointed out for RNN language models (Linzen et al., 2016).

In this study, we propose new Japanese PAS analysis models that integrate the prediction information of arguments in multiple predicates. We extend a standard end-to-end style deep bi-RNN model (Figure 2a) and introduce components that consider the multiple predicate interactions into both the input and last layers (Figures 2b and 3). In contrast to Grid RNN, our extended models stack the extra layers using pooling and attention mechanisms on top of a deep bi-RNN so that they can directly associate the label prediction information for a target (predicate, word) pair with the predictions for words strongly related to the target pair. Through experiments, we show that the proposed models improve argument prediction accuracy, especially for the *Zero* cases, and achieve a new state-of-the-art performance in the overall F_1 on a standard benchmark corpus.

2 Task

In this paper, we employ a task definition based on the NAIST Text Corpus (NTC) (Iida et al., 2010; Iida et al., 2017), a commonly used benchmark corpus annotated with nominative (NOM), accusative (ACC), and dative (DAT) arguments for predicates. Given a tokenized sentence $\mathrm{w} = w_1, ..., w_n$ and its predicate positions $\mathrm{p} = p_1, ..., p_q$, our task is to identify at most one head of the filler tokens for each argument slot of each predicate. In this study, we follow the setting of Iida et al. (2015), Ouchi et al. (2017), and Matsubayashi and Inui (2017), and focus only on analyzing arguments in a target sentence. In addition, we exclude argument instances that are in the same *bunsetsu*, a base phrase unit in Japanese, as the target predicate, following Ouchi et al. (2017), which we will compare with the results in experiments.

The semantic labels used in NTC may seem to be rather syntactic as they are named nominative, accusative, etc. However, this annotation task markedly differs from shallow syntactic parsing and is, in fact, more like a semantic role labeling (SRL) task including implicit argument prediction. First, the semantic labels in NTC generalize case alteration caused by voice alteration and thus represent semantic roles analogous to ARG0, ARG1, etc. in the PropBank-style annotation (Palmer et al., 2005). Second,

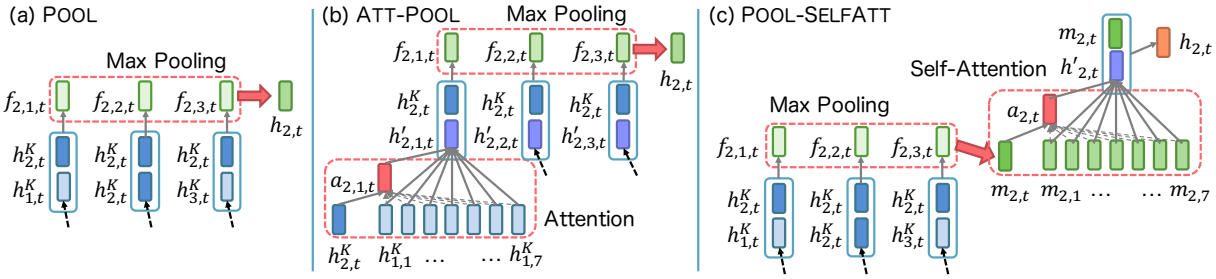


Figure 3: Three variants of interaction layers.

in the corpus, when an argument is omitted (i.e., zero-anaphora), the antecedent is identified with an appropriate semantic role, which is a prominent problem in Japanese semantic analysis and is the primary target of this study.

3 Base Model

Our proposed models extend end-to-end style SRL systems using deep bi-RNN (Zhou and Xu, 2015; He et al., 2017; Ouchi et al., 2017) to combine mechanisms that consider multiple predicate interactions. Figure 2a shows the network of our base model. Formally, given a word sequence $\mathrm{w} = w_1, ..., w_n$ and a target predicate position p_i in p, the model outputs a label probability for each word position: $p(c_{i,1}|i, \mathrm{p}, \mathrm{w}), ..., p(c_{i,n}|i, \mathrm{p}, \mathrm{w})$. Here, $c_{i,t} \in \{\texttt{NOM}, \texttt{ACC}, \texttt{DAT}, \texttt{NONE}\}$ represents the argument label of the word w_t for the target predicate w_{p_i}.

The input layer creates a vector $h^0_{i,t} \in \mathbb{R}^{d_w+1}$ for each pair of a predicate w_{p_i} and a word w_t by concatenating a word embedding $e(w_t) \in \mathbb{R}^{d_w}$ and a binary value representing the target predicate position in a method similar to that of He et al. (2017). The obtained vectors are then input into the deep bi-RNN, where the directions of the layers alternate (Zhou and Xu, 2015):

$$h^1_{i,t} = r^1(h^0_{i,t}, h^1_{i,t-1}), \quad h^k_{i,t} = \begin{cases} h^{k-1}_{i,t} + r^k(h^{k-1}_{i,t}, h^k_{i,t-1}) & (k \text{ is odd}) \\ h^{k-1}_{i,t} + r^k(h^{k-1}_{i,t}, h^k_{i,t+1}) & (k \text{ is even}) \end{cases} (k \geq 2). \tag{1}$$

Here, $h^k_{i,t} \in \mathbb{R}^{d_r}$ is the output of the k-th RNN layer for a pair (w_{p_i}, w_t), and r^k is a function representing the k-th RNN layer. We employ gated recurrent units (GRUs) (Cho et al., 2014) for the RNNs. In addition, we use the residual connections (He et al., 2016) following Ouchi et al. (2017). Then, a four-dimensional vector representing a probability $p(c_{i,t}|i, \mathrm{p}, \mathrm{w})$ is obtained by applying a softmax layer to each output of the last RNN layer $h^K_{i,t}$. For each argument label c of each predicate, we eventually select a word with the maximum probability that exceeds an output threshold θ_c.

4 Proposed Models

Our base model independently predicts the arguments of each predicate. In order to capture dependencies between the arguments of multiple predicates, we apply two extensions to our base model: a *multi-predicate input layer* and three variants of *interaction layers* on top of the deep bi-RNNs. Figures 2b and 3 show the network structures of the extended models.

In contrast to the Grid RNN model of Ouchi et al. (2017), where the information of multiple predicates propagates through the RNNs, our interaction layers use pooling and attention mechanisms to directly associate the label prediction information for a target (predicate, word) pair with that for words strongly related to the target pair, without being disturbed by word order and distance.

4.1 Interaction Layers

Pooling (POOL) Argument sharing across multiple predicates can be captured with both syntactic and semantic clues. At the syntactic level, we want to capture tendencies that, for example, the subject of the predicate of a matrix clause is likely to fill argument slots of other predicates in the same sentence.

At the semantic level, we want to model semantic dependencies between neighboring events such as *the person who asks a question is likely to be answered*, as in Figure 1. Our proposal is to capture both types of clues by incorporating a max pooling layer on top of the base model.

Specifically, as illustrated in Figure 3a, for each word w_t, we integrate the intermediate representation of label prediction for each predicate $h^K_{i,t}$ by applying max pooling to the vectors that represent pairs of prediction information for two predicates $h^K_{i,t}$ and $h^K_{j,t}$ (including the case $i = j$):

$$h_{i,t} = \mathrm{maxpool}_j(f_{i,j,t}), \quad \text{where} \quad f_{i,j,t} = \mathrm{ReLU}(W_f[h^K_{i,t}, h^K_{j,t}] + b_f). \tag{2}$$

In this equation, $\mathrm{maxpool}_j(f_{i,j,t})$ is an operation to extract the maximum value of each dimension in $\{f_{i,1,t}, ..., f_{i,q,t}\}$. The newly obtained vector $h_{i,t}$ for w_{p_i} and w_t is input into the softmax layer in the same manner as in the base model.

Attention-then-Pooling (ATT-POOL) Besides the argument sharing across multiple predicates, we would also like to capture dependencies between different arguments of a single predicate (and potentially, arguments of multiple predicates). For example, syntactically, two distinct argument slots of a single predicate are unlikely to share the same filler. Semantically, the subject of a predicate *take* is likely to be a person when its object is *a bread*, but is likely to be a company if the object is *a new employee*.

To capture such dependencies, we integrate the intermediate label prediction $h^K_{j,t'}$ of $w_{t'}$ for an arbitrary predicate w_{p_j} (including the case $i = j$) into the prediction of w_t for a target predicate w_{p_i}. In the integration, we aim to weigh the prediction information for $(w_{p_j}, w_{t'})$ based on its relatedness to the target pair (w_{p_i}, w_t) using the attention mechanism (Bahdanau et al., 2015). As in Figure 3b, we calculate a weight $a_{i,j,t}(t') \in \mathbb{R}$ for each of $h^K_{j,1}, ..., h^K_{j,n}$ on the basis of the prediction $h^K_{i,t}$ for the target pair and we obtain a weighted sum of $h^K_{j,t'}$ as a summary of the argument information of w_{p_j}, which is expected to be useful for the label prediction of (w_{p_i}, w_t):

$$h'_{i,j,t} = \textstyle\sum_{t'} a_{i,j,t}(t') \cdot h^K_{j,t'}, \quad \text{where} \quad a_{i,j,t}(t') = \frac{\exp(W_a g_{i,j,t,t'} + b_a)}{\sum_{t''} \exp(W_a g_{i,j,t,t''} + b_a)}, \tag{3}$$

$$g_{i,j,t,t'} = \tanh(W_g[h^K_{i,t}, h^K_{j,t'}] + b_g). \tag{4}$$

The obtained $h'_{i,j,t}$ are concatenated with the prediction for the target pair $h^K_{i,t}$ and linearly transformed with the ReLU activation. Max pooling is then applied to these vectors to combine the predictions for multiple predicates.

$$h_{i,t} = \mathrm{maxpool}_j(f_{i,j,t}), \quad \text{where} \quad f_{i,j,t} = \mathrm{ReLU}(W_f[h^K_{i,t}, h'_{i,j,t}] + b_f) \tag{5}$$

Pooling-then-Self-Attention (POOL-SELFATT) The ATT-POOL model involves a high computational cost because it must compute nq^2 different attentions regarding the number of words n and the number of predicates q in a sentence. Therefore, as illustrated in Figure 3c, in this model, we first apply the max pooling that we applied in the POOL model to reduce the sequences for which attentions must be computed by integrating the label predictions of w_t for all the other predicates in advance.

$$m_{i,t} = \mathrm{maxpool}_j(f_{i,j,t}), \quad \text{where} \quad f_{i,j,t} = \mathrm{ReLU}(W_f[h^K_{i,t}, h^K_{j,t}] + b_f) \tag{6}$$

Then, we combine the information in the obtained sequence $m_{i,1}, ..., m_{i,n}$ in a similar manner as in the ATT-POOL model using the attention mechanism, but this time, with self-attention, that is, computing the weights of the elements in the sequence based on the relatedness to the element inside the sequence.

$$h_{i,t} = \mathrm{ReLU}(W_h[m_{i,t}, h'_{i,t}] + b_h) \tag{7}$$

$$h'_{i,t} = \textstyle\sum_{t'} a_{i,t}(t') \cdot m_{i,t'}, \quad \text{where} \quad a_{i,t}(t') = \frac{\exp(W_a g_{i,t,t'} + b_a)}{\sum_{t''} \exp(W_a g_{i,t,t''} + b_a)} \tag{8}$$

$$g_{i,t,t'} = \tanh(W_g[m_{i,t}, m_{i,t'}] + b_g) \tag{9}$$

Consequently, the number of attentions that must be computed is reduced to nq.

Self-Attention (SELFATT) To conduct ablation tests to assess the impact of the proposed extensions, we also implemented a model only with self-attention. This model explicitly considers the relationships between arguments of a single predicate, but not arguments across multiple predicates.

$$h_{i,t} = \mathrm{ReLU}(W_h[h^K_{i,t}, h'_{i,t}] + b_h) \tag{10}$$

$$h'_{i,t} = \textstyle\sum_{t'} a_{i,t}(t') \cdot h^K_{i,t'}, \quad \text{where} \quad a_{i,t}(t') = \frac{\exp(W_a g_{i,t,t'} + b_a)}{\sum_{t''} \exp(W_a g_{i,t,t''} + b_a)} \tag{11}$$

$$g_{i,t,t'} = \tanh(W_g[h^K_{i,t}, h^K_{i,t'}] + b_g) \tag{12}$$

4.2 Multi-Predicate Input Layer (MP)

In addition, we add a simple but effective extension to the input layer. As He et al. (2016) reported, the information of the target predicate w_{p_i} propagates to the intermediate prediction $h^K_{i,t}$ of the candidate argument w_t through the deep bi-RNN by just adding a binary value representing the predicate position. Inspired by this finding, as shown in Figure 2b, in the input layer, we add another binary value that represents all the predicate positions to $h^0_{i,t}$, aiming to propagate multiple predicate information.

5 Experiments

We evaluated the impacts of our extensions and compared their performances to those of previous studies. Our main hypothesis is that the pooling and attention mechanisms are both useful for capturing different types of argument interactions as we explained in Section 4 and do work complementarily of each other to improve the prediction accuracy, especially for arguments in a long-distance dependency.

5.1 Settings

5.2 Dataset and Implementation Details

The experiments were performed on NTC 1.5. We divided the corpus into the commonly used divisions of training, development, and test sets (Taira et al., 2008), each of which includes 24,283, 4,833, and 9,284 sentences, respectively. NTC represents each argument of a predicate by indicating a coreference cluster in a text. For each given predicate-argument slot, we count a system's output as correct if the output token is included in the coreference cluster corresponding to the slot fillers. The evaluation is performed on the basis of the precision, recall, and F_1 score.

The hyperparameters were selected to obtain a maximum F_1 on the development set. The details of the hyperparameter selection and preprocessing are described in the supplemental material. In the following experiments, we train each model 10 times with the same training data and hyperparameters and then show the average scores.

5.3 Grid RNN Baseline (GRID)

In order to strictly compare the impact of our extensions to the method used for integrating multiple pieces of predicate information in the state-of-the-art end-to-end model, in addition to our base model, we replicated the method of Ouchi et al. (2017) by modifying Equations (1) of our base model as follows:

$$h^1_{i,t} = r^1([h^0_{i,t}, h^1_{i-1,t}], h^1_{i,t-1}), \quad h^k_{i,t} = h^{k-1}_{i,t} + \begin{cases} r^k([h^{k-1}_{i,t}, h^k_{i-1,t}], h^k_{i,t-1}) & (k \text{ is odd}) \\ r^k([h^{k-1}_{i,t}, h^k_{i+1,t}], h^k_{i,t+1}) & (k \text{ is even}) \end{cases} (k \geq 2), \tag{13}$$

if $1 \leq i \leq q$; otherwise, $h^k_{i,t} = \mathbf{0}$. The performance of this replicated model may not be strictly the same as that reported in Ouchi et al. (2017) due to discrepancies in the embeddings of inputs, hyperparameters (a training batch size, a hidden unit size, etc.), and training strategy (an optimizing algorithm, a regularization method, an early stopping method, etc.). The predicate positions $\mathrm{p} = p_1, ..., p_q$ are arranged in ascending order.

		All				F_1 at different dependency distances					
	Model	F_1 (%)	SD	Prec.	Rec.	*Dep*	*Zero*	2	3	4	≥ 5
Baseline Models	BASE ($d_r = 32, K = 8$)	81.22	±0.19	84.30	78.37	88.39	49.12	55.73	47.1	39	29
	GRID ($d_r = 32, K = 8$)	81.06	±0.31	84.33	78.04	88.17	48.73	55.26	47.5	39	28
	BASE	83.39	±0.13	85.85	81.07	89.90	54.37	61.09	53.8	44	31
	GRID	82.94	±0.17	85.38	80.63	89.51	53.57	60.28	52.4	44	32
Proposed Models	SELFATT	83.56	±0.22	85.91	81.34	90.06	54.84	61.36	54.3	**45**	32
	POOL	83.56	±0.16	86.05	81.21	90.00	54.81	61.54	54.3	**45**	31
	ATT-POOL	83.48	±0.24	85.97	81.12	89.98	54.57	61.19	54.0	44	32
	POOL-SELFATT	83.76	±0.17	86.11	81.54	90.17	55.19	62.10	54.0	**45**	32
	MP	83.67	±0.22	86.08	81.39	90.10	54.80	61.67	53.8	44	32
	MP-SELFATT	83.79	±0.22	86.11	**81.60**	90.22	55.26	61.88	54.3	**45**	**33**
	MP-POOL-SELFATT	**83.94**	±0.12	**86.58**	81.46	**90.26**	**55.55**	**62.44**	**54.7**	**45**	32
Previous SOTAs	Ouchi et al. (2017)	81.42				88.17	47.12				
	M&I 2017	83.50	±0.17	86.00	81.15	89.89	51.79	60.17	49.4	38	23
Ensemble Models	MP-POOL-SELFATT (10 models)	**85.34**		**87.90**	**82.93**	**91.26**	**58.07**	**64.89**	**57.5**	**47**	**33**
	M&I 2017 (5 models)	84.07		86.09	82.15	90.24	53.66	61.94	51.8	40	24

Table 1: F_1 scores on the NTC 1.5 test set. *Dep* and *Zero* denote instances where the dependency distance between the predicate and argument is one and more than one, respectively. M&I 2017 is the model of Matsubayashi and Inui (2017).

Model A \ Model B	F_1 (%)	SD	BASE	ATT-POOL	SELFATT	POOL	MP	POOL-SELFATT	MP-SELFATT
BASE	83.39	±0.13							
ATT-POOL	83.48	±0.24	0.18						
SELFATT	83.56	±0.22	**0.03**	0.22					
POOL	83.56	±0.16	**0.014**	0.21	0.53				
MP	83.67	±0.22	**0.003**	**0.048**	0.16	0.12			
POOL-SELFATT	83.76	±0.17	**4.3E-5**	**0.004**	**0.023**	**0.0084**	0.16		
MP-SELFATT	83.79	±0.22	**1.0E-4**	**0.0046**	**0.021**	**0.0096**	0.13	0.39	
MP-POOL-SELFATT	83.94	±0.12	**5.4E-6**	**5.4E-6**	**2.2E-4**	**2.7E-5**	**0.0013**	**0.013**	**0.046**

Table 2: *p-values* in one-sided permutation test using 10 overall F_1 scores for each model. The bold values indicate that an average F_1 score of model A outperforms that of model B at the 5% significance level.

5.4 Results

Impact of Extensions

The first two sets of rows in Table 1 compare the impact of each component of our extension. The effects of incorporating the interaction layer can be seen in the comparisons of the BASE model with the SELFATT, POOL, ATT-POOL, and POOL-SELFATT models. Among the four proposed extensions, POOL-SELFATT, an integration of POOL and SELFATT, achieved the best performance (83.76 in F_1), gaining 0.37 points in overall F_1 from BASE. Also, the significance tests in Table 2 show that the POOL and SELFATT models significantly outperform the BASE model, and the POOL-SELFATT model makes a further significant gain from the POOL and SELFATT models. This indicates that POOL and SELFATT work complementarily with each other, and combining them makes a further improvement from each individual extension. Recall that SELFATT is designed to capture long-distance dependencies over a single predicate-argument structure, whereas POOL is expected to capture argument sharing across multiple predicates. These results provide empirical support to the hypotheses behind our design of the interaction layer.

The MP model, where the input layer is extended to represent the positions of all the predicates in a sentence, significantly outperforms the BASE model by 0.28 points in overall F_1. This result suggests the importance of position information regarding the neighboring predicates in identifying the arguments of a given predicate. Furthermore, the MP-POOL-SELFATT model, which is a combination of MP and POOL-SELFATT, resulted in a further 0.27-point improvement and consequently achieved the best overall F_1 of 83.94 as a single model.

Following Matsubayashi and Inui (2017), we also assess F_1 values at different dependency distances. The results are shown in the right half of Table 1. From the table, we can see that MP-POOL-SELFATT

Model	ALL	Dep				Zero			
		ALL	NOM	ACC	DAT	ALL	NOM	ACC	DAT
MP-POOL-SELFATT	**83.94**	**90.26**	90.88	94.99	**67.57**	**55.55**	**57.99**	**48.9**	**23**
Ouchi et al. (2015)	79.23	86.07	88.13	92.74	38.39	44.09	48.11	24.4	4.8
Ouchi et al. (2017)	81.42	88.17	88.75	93.68	64.38	47.12	50.65	32.4	7.5
M&I 2017	83.50	89.89	**91.19**	**95.18**	61.90	51.79	54.69	41.8	17
MP-POOL-SELFATT (ens.)	**85.34**	**91.26**	**91.84**	**95.57**	**70.8**	**58.07**	**60.21**	**52.5**	**26**
M&I 2017 (ens. of 5)	84.07	90.24	91.59	95.29	62.61	53.66	56.47	44.7	16

Modified NTC 1.5 (Iida et al., 2016)

Model	Zero NOM
Ouchi et al. (2015)	**57.3**
Iida et al. (2015)	41.1
Iida et al. (2016)	52.5

(Note) Results on a dataset different from our experiments

Table 3: F_1 scores of each argument label on the NTC 1.5 test set.

improves F_1 from BASE by 0.9–1.4 points consistently across all the distance categories other than *Dep*.

Comparison to Related Work

The third set of rows in Table 1 shows the reported performance of related studies. Grid RNN of Ouchi et al. (2017) is a state-of-the-art end-to-end model, designed to capture interactions among multiple predicate-argument relations. A comparison between their model and the proposed models was somewhat tricky because our replication of Grid RNN did not reproduce the reported performance on the same dataset (see the row of GRID in Table 1). Unlike the results reported in Ouchi et al. (2017), the GRID model in our experiment did not clearly outperform the model without the grid architecture, i.e., the Base model. We first suspected that this might have resulted from the difference in dimensionality d_r of RNN hidden states: $d_r = 32$ in Ouchi et al. (2017), whereas $d_r = 256$ in our experiments. Specifically, we speculated that the base model with a low dimensionality left a larger margin for improvement and incorporating the Grid architecture derived positive effects. We thus trained our GRID model with Ouchi et al. (2017)'s settings ($d_r = 32$ and $K = 8$) and the best performing hyperparameters; however, we were not able to reproduce the reported gain from Grid RNN (see the row of "GRID ($d_r = 32$, $K = 8$)" in Table 1).[2] This might be an indication of the difficulty in capturing multi-predicate interactions by threading deep bi-RNNs with RNNs, as we discussed in Section 1.

Another previous state-of-the-art model was proposed by Matsubayashi and Inui (2017) (M&I 2017). This model extends a feedforward NN with dependency path embeddings and other new features to capture long-distance dependencies in a single PAS. The row "M&I 2017" in Table 1 shows the reported performance of their model.[3] The performance of M&I 2017 is comparable with the performance of our SELFATT model. This result provides another piece of empirical evidence that the self-attention mechanism has a comparably positive effect in incorporating dependency path information for capturing long-distance dependencies in a single PAS.

Overall, the proposed methods of using the pooling and attention mechanisms for capturing interactions across predicates and arguments gained considerable improvement and achieved state-of-the-art accuracy, significantly outperforming the previous state-of-the-art models. The last set of rows in Table 1 shows the results of the ensemble models. A model that predicts arguments with the average score of the 10 MP-POOL-ATT models further improves the overall F_1 by 1.4 points from that of a single model, achieving state-of-the-art accuracy for NTC 1.5.

Table 3 shows the F_1 score for each case label. In a comparison of the single models, although our MP-POOL-ATT model slightly degrades the scores of NOM and ACC on the *Dep* cases compared to the state-of-the-art model (M&I 2017), it greatly improves the scores for DAT and the *Zero* cases. Regarding the ensemble models, MP-POOL-ATT improves the scores for all cases.

Iida et al. (2015) and Iida et al. (2016) report Japanese subject anaphora resolution systems, designed to predict only *Zero* NOM arguments. It is not straightforward to directly compare their results with ours due to the differences in the experimental settings. However, our best performing model outperforms the

[2]We discussed this negative result, including the implementation details, with one of the authors of Ouchi et al. (2017). However, we could not find a plausible reason for the results.

[3]For the purpose of a strict comparison with Ouchi et al. (2017), we re-evaluate the model of Matsubayashi and Inui (2017) by excluding the instances for which the argument is in the same *bunsetsu* phrase as the predicate; this is the same setting as that in Ouchi et al. (2017). We have reported the new results in Tables 1 and 3.

(1) 背筋 を [伸ばし $_{\text{PRED}}$] 少 [考 $_{\text{NOM\_FALSE}}$] の 後 、 応じる [谷川 $_{\text{NOM\_GOLD}}$] 。
spine ACC stretch little thinking of after (nominal) , respond Tanigawa .
'[Tanigawa $_{\text{NOM\_GOLD}}$], responding after [stretching $_{\text{PRED}}$] his spine and [thinking $_{\text{NOM\_FALSE}}$] briefly.'

(2) 大学 教授 [ら $_{\text{NOM\_GOLD}}$] が 地下 に 潜って 、 ファクス で 連絡 を 取り 合い 、 地方 の 組織 の 活動 を [支えて $_{\text{PRED}}$] いる 。 [*NONE*$_{\text{NOM\_FALSE}}$]
university professor PLURAL NOM underground DAT dive , fax by contacting ACC take each-other , district of organization of activities ACC support PROGRESSIVE
'The university [professors $_{\text{NOM\_GOLD}}$] went into hiding and are [supporting $_{\text{PRED}}$] the activities of the local organizations, contacting each other by fax. [NONE $_{\text{NOM\_FALSE}}$]'

(3) 中央 [省庁 $_{\text{NOM\_FALSE}}$] が [職員 $_{\text{NOM\_GOLD}}$] に 対し 「 夜 の 接待 は [受ける $_{\text{PRED}}$] な 」 と 通達 すれば 済む こと だ 。
central ministries NOM staff DAT against " night in entertainment TOP accept NEG$_{\text{IMPERATIVE}}$ " as notification VERBALIZER$_{\text{CONDITIONAL}}$ finish NOMINALIZER COPULA.
'It is sufficient enough if the central [ministries $_{\text{NOM\_FALSE}}$] tell the [staff$_{\text{NOM_GOLD}}$] "Do not [accept $_{\text{PRED}}$] a business dinner."'

(4) 十三 日 午後 に は 盆栽 作家 、 木村 正彦 [氏 $_{\text{NOM\_GOLD}}$] に よる 実技 の デモンストレーション が [行わ $_{\text{PRED}}$] れる 。 [*NONE*$_{\text{NOM\_FALSE}}$]
13 day afternoon DAT TOP bonsai artist , Kimura Masahiko Mr. {by- -by} techniques of demonstration NOM perform PASSIVE .
'On the afternoon of 13th, a practical demonstration by the bonsai artist [Mr. $_{\text{NOM\_GOLD}}$] Masahiko Kimura will be [performed $_{\text{PRED}}$]. [NONE $_{\text{NOM\_FALSE}}$]'

Figure 4: Examples of prediction errors. In Example (1), only SELFATT failed to predict the answer. In Example (2), only MP-POOL-SELFATT correctly predicted the answer. In Examples (3) and (4), none of the systems predict the answers correctly.

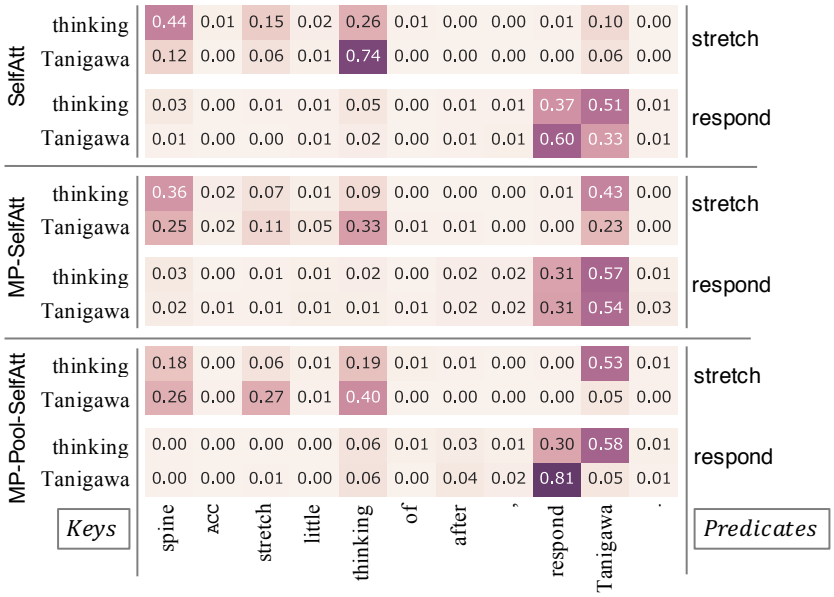


Figure 5: Attention weights of proposed models for Example (1).

model of Ouchi et al. (2015), which is then reported to outperform both Iida et al. (2015) and Iida et al. (2016) in their experimental settings.

5.5 Detailed Analysis

To analyze the behavior of our proposed models in detail, we show some prediction examples of the SELFATT, MP-SELFATT, and MP-POOL-ATT models in the development set with the weights in the attention layers in Figures 4-7.

In Figure 4, Examples (1) and (2) are the instances for which only SELFATT failed to predict the answer and for which only MP-POOL-SELFATT correctly predicted the answer, respectively. For these examples, the weights in the attention layers behave similarly. Figure 5 shows the weights for Example (1). In this sentence, the correct NOM of *stretch*, *Tanigawa*, is also NOM of *respond*, which is relatively easy to predict. SELFATT, which is designed to capture dependencies over a single predicate-argument structure, failed to predict NOM of *stretch* most likely because the answer *Tanigawa* is distant from the target predicate with its limited syntactic clues. Conversely, MP-POOL-SELFATT and MP-SELFATT successfully predicted the answer by taking the answer token *Tanigawa* into account when computing the

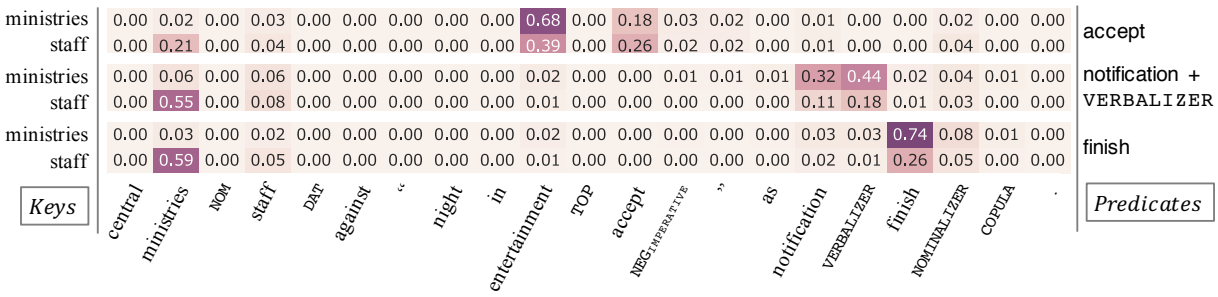


Figure 6: Attention weights of MP-POOL-SELFATT for Example (3).

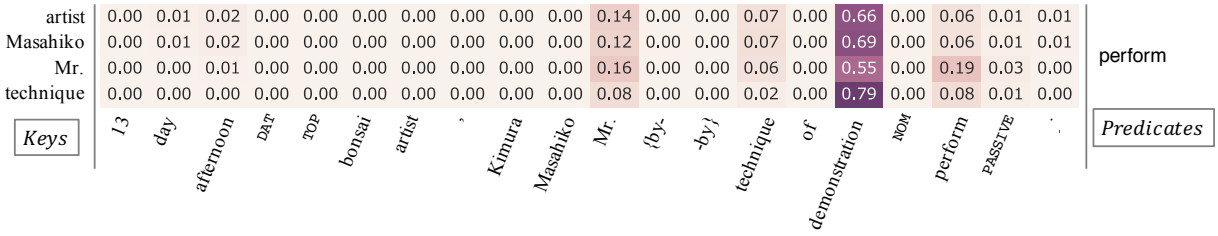


Figure 7: Attention weights of MP-POOL-SELFATT for Example (4).

score of the counter candidate *thinking*. MP-SELFATT, the model that incorporates the other predicate positions into SELFATT, significantly increases the weight for the answer token. MP-POOL-SELFATT, which explicitly integrates the predictions for the other predicates, further increases the weight for the answer token. This example demonstrates that the proposed extensions successfully predict a correct argument by considering the relation to the argument in another predicate where the syntactic relation between the predicate and argument is much clearer and thus the argument relation is relatively easy to predict. Due to space limitations, we cannot show the weights for Example (2), but the same also holds for that example. MP-POOL-SELFATT focuses on professors, which is the "easy-to-predict" NOM argument of dive, when the model computes the scores of this token for *take* and, consequently, *support*. SELFATT and MP-SELFATT assign smaller weights to that token for *take* and even smaller weights for *support*, which is far from the answer token.

Examples (3) and (4) are the instances where all the three models failed to predict the answers. Figure 6 illustrates the attention weights in MP-POOL-SELFATT for Example (3). To solve this example, the model is expected to understand that NOM of *accept* should be the same as the persons who received the order from the *ministries*. However, MP-POOL-SELFATT could not acquire this kind of dialog-level knowledge and pays little attention to the correct argument *staff* when the model computes the score of the wrong answer *ministries* for NOM of *accept*.

In Example (4), NOM of the nominal predicate *demonstration* can be a clue for predicting NOM of *perform*. However, the models currently do not predict the arguments of nominal predicates and therefore cannot capture the relationships between these two sufficiently (Figure 7). This example suggests one of our future directions: the joint prediction of verbal and nominal predicates.

6 Related Work

End-to-End Models in SRL End-to-end approaches to SRL have been widely explored recently, and many state-of-the-art results have been achieved (Zhou and Xu, 2015; He et al., 2017; Marcheggiani and Titov, 2017; Tan et al., 2018). Following these advanced models, we adopted a stacked bi-RNN as our base model.

Methods for Dealing with Long-Distance Dependencies in End-to-End Models In SRL studies, Marcheggiani and Titov (2017) proposed a variant of deep bi-RNN models that connects the intermediate representations of the predictions for the words in syntactic dependency relations on top of the deep RNN.

Very recently, aiming to directly connect the related words, Tan et al. (2018) stacked self-attention layers, each of which followed a feedforward layer, in a manner similar to the method of Vaswani et al. (2017), which was originally applied to an encoder-decoder model.

Self-attention has been successfully applied to several NLP tasks, including textual entailment, sentiment analysis, summarization, machine translation, and language understanding (Paulus et al., 2017; Shen et al., 2018; Lin et al., 2017; Vaswani et al., 2017). Techniques using pooling have been applied to merge intermediate expressions in predictions in the tasks where related tokens are often at long distance such as coreference resolution and machine reading (Clark and Manning, 2016; Kobayashi et al., 2016). One major contribution of this study is its novel idea of using these techniques for capturing long-distance dependencies for modeling interactions among multiple predicate-argument relations.

Approaches to Capturing Multi-Predicate Interactions For Japanese, Ouchi et al. (2015) jointly identified arguments of multiple predicates by modeling argument interactions with a bipartite graph. Iida et al. (2015) constructed a *subject-shared predicate network* and deterministically propagated the predicted subjects to other predicates. Shibata et al. (2016) adapted a NN framework to Ouchi et al. (2015)'s model using a feedforward network. For an end-to-end neural model, Ouchi et al. (2017) used a Grid RNN to capture multiple predicate interactions. Through experiments, we demonstrated that our proposed models outperformed these models in terms of the overall F_1 on a standard benchmark corpus.[4]

To the best of our knowledge, there are few previous studies related to SRL considering multiple predicate interactions for languages other than Japanese. Yang and Zong (2014) performed a discriminative reranking in the role classification of shared arguments. Lei et al. (2015) proposed an SRL model based on the dimensionality reduction on a tensor representation to capture meaningful interactions between the argument, predicate, corresponding features, and role label. It is not straightforward to compare these methods with our models; however, it is an intriguing future issue to consider how well the techniques devised for Japanese PAS analysis work for other languages.

Other Approaches to Argument Omission In order to perform robust prediction for arguments with fewer syntactic clues, several previous studies have explored various types of selectional preference scores that consider the semantic relations between a predicate and its arguments (Iida et al., 2007; Imamura et al., 2009; Komachi et al., 2010; Sasano and Kurohashi, 2011; Shibata et al., 2016). This direction of research is orthogonal to our approach, suggesting that the models could be further improved by being combined with these extra features.

7 Conclusion

In this study, we have proposed new Japanese PAS analysis models that integrate prediction information of arguments in multiple predicates. We extended the end-to-end style model using a deep bi-RNN and introduced the components that consider the multiple predicate interactions into the input and last layers. As a result, we achieved a new state-of-the-art accuracy on the standard benchmark data.

Our detailed analysis showed that the proposed models successfully predict the correct arguments by using the information of the *"easy-to-predict"* arguments in other predicates. In addition, the error analysis suggests that jointly predicting the arguments of verbal and nominal predicates may further improve the performance. An intriguing issue we plan to address next is how to extend the proposed interaction layer to cross-sentential interactions of PASs.

Acknowledgements

We are grateful to the anonymous reviewers for their useful comments and suggestions. We thank Hiroki Ouchi for his help in checking our re-implementation. We also thank Shun Kiyono and Kento Watanabe for valuable discussions. This work was partially supported by JSPS KAKENHI Grant Numbers 15H01702 and 15K16045.

[4] Shibata et al. (2016) evaluated the model on a different dataset and hence, it is difficult to compare the results directly.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *ICLR*, pages 1–15.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *EMNLP*, pages 1724–1734.

Kevin Clark and Christopher D. Manning. 2016. Improving Coreference Resolution by Learning Entity-Level Distributed Representations. In *ACL*, pages 643–653.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *CVPR*, pages 770–778.

Luheng He, Kenton Lee, Mike Lewis, Luke Zettlemoyer, and Paul G Allen. 2017. Deep Semantic Role Labeling: What Works and What's Next. In *ACL*, pages 473–483.

Ryu Iida, Kentaro Inui, and Yuji Matsumoto. 2007. Zero-anaphora Resolution by Learning Rich Syntactic Pattern Features. *ACM Transactions on Asian Language Information Processing*, 6(4):1:1–1:22.

Ryu Iida, Mamoru Komachi, Naoya Inoue, Kentaro Inui, and Yuji Matsumoto. 2010. Annotating Predicate-Argument Relations and Anaphoric Relations: Findings from the Building of the NAIST Text Corpus. *Natural Language Processing*, 17(2):25–50.

Ryu Iida, Kentaro Torisawa, Chikara Hashimoto, Jong-Hoon Oh, and Julien Kloetzer. 2015. Intra-sentential Zero Anaphora Resolution using Subject Sharing Recognition. In *EMNLP*, pages 2179–2189.

Ryu Iida, Kentaro Torisawa, Jong-Hoon Oh, Canasai Kruengkrai, and Julien Kloetzer. 2016. Intra-Sentential Subject Zero Anaphora Resolution using Multi-Column Convolutional Neural Network. In *EMNLP*, pages 1244–1254.

Ryu Iida, Mamoru Komachi, Naoya Inoue, Kentaro Inui, and Yuji Matsumoto. 2017. NAIST Text Corpus: Annotating Predicate-Argument and Coreference Relations in Japanese. In *Handbook of Linguistic Annotation*, pages 1177–1196. Springer.

Kenji Imamura, Kuniko Saito, and Tomoko Izumi. 2009. Discriminative Approach to Predicate-Argument Structure Analysis with Zero-Anaphora Resolution. In *ACL-IJCNLP*, pages 85–88.

Sosuke Kobayashi, Ran Tian, Naoaki Okazaki, and Kentaro Inui. 2016. Dynamic Entity Representation with Max-pooling Improves Machine Reading. In *NAACL-HLT*, pages 850–855.

Mamoru Komachi, Ryu Iida, Kentaro Inui, and Yuji Matsumoto. 2010. Argument Structure Analysis of Event-nouns Using Lexico-syntactic Patterns of Noun Phrases. *Journal of Natural Language Processing*, 17(1):141–159.

Tao Lei, Yuan Zhang, Lluis Marquez, Alessandro Moschitti, and Regina Barzilay. 2015. High-Order Low-Rank Tensors for Semantic Role Labeling. In *NAACL-HLT*, pages 1149–1159.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A Structured Self-attentive Sentence Embedding. In *ICLR*, pages 1–15.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies. *Transactions of the Association for Computational Linguistics*, 4(1):521–535.

Diego Marcheggiani and Ivan Titov. 2017. Encoding Sentences with Graph Convolutional Networks for Semantic Role Labeling. In *EMNLP*, pages 1507–1516.

Yuichiroh Matsubayashi and Kentaro Inui. 2017. Revisiting the Design Issues of Local Models for Japanese Predicate-Argument Structure Analysis. In *IJCNLP*, pages 128–133.

Hiroki Ouchi, Hiroyuki Shindo, Kevin Duh, and Yuji Matsumoto. 2015. Joint Case Argument Identification for Japanese Predicate Argument Structure Analysis. In *ACL-IJCNLP*, pages 961–970.

Hiroki Ouchi, Hiroyuki Shindo, and Yuji Matsumoto. 2017. Neural Modeling of Multi-Predicate Interactions for Japanese Predicate Argument Structure Analysis. In *ACL*, pages 1591–1600.

M. Palmer, P. Kingsbury, and D. Gildea. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1):71–106.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A Deep Reinforced Model for Abstractive Summarization. In *ICLR*, pages 1–12.

Ryohei Sasano and Sadao Kurohashi. 2011. A Discriminative Approach to Japanese Zero Anaphora Resolution with Large-scale Lexicalized Case Frames. In *IJCNLP*, pages 758–766.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. 2018. DiSAN: Directional Self-Attention Network for RNN/CNN-Free Language Understanding. In *AAAI-18*, pages 5446–5455.

Tomohide Shibata, Daisuke Kawahara, and Sadao Kurohashi. 2016. Neural Network-Based Model for Japanese Predicate Argument Structure Analysis. In *ACL*, pages 1235–1244.

Hirotoshi Taira, Sanae Fujita, and Masaaki Nagata. 2008. A Japanese Predicate Argument Structure Analysis using Decision Lists. In *EMNLP*, pages 523–532.

Zhixing Tan, Mingxuan Wang, Jun Xie, Yidong Chen, and Xiaodong Shi. 2018. Deep Semantic Role Labeling with Self-Attention. In *AAAI-18*, pages 4929–4936.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *NIPS*, pages 5998–6008.

Haitong Yang and Chengqing Zong. 2014. Multi-Predicate Semantic Role Labeling. In *EMNLP*, pages 363–373.

Jie Zhou and Wei Xu. 2015. End-to-end Learning of Semantic Role Labeling Using Recurrent Neural Networks. In *ACL*, pages 1127–1137.

Appendix A: Implementation Details

Hyperparameters The hyperparameters were selected to obtain a maximum F_1 on the development set. The dimension of the word embeddings d_w was set to 256. The dimension of the hidden state of the GRUs d_r was set to 256 from $\{128, 256, 512\}$ and the number of the GRU layers was set to 10 from $\{6, 8, 10, 12\}$. The dropout rate of the GRUs was set to 0.1 from $\{0.0, 0.1, 0.2\}$. The dimensions of the outputs of the nonlinear transformations f, g and $h_{i,t}$ were set to 1024 from $\{512, 768, 1024\}$. We set the batch size of the training data as the number of predicates in each sentence. We employed the negative log likelihood as the training loss and an Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1e-08$. During the training, we halved the learning rate when the F_1 score on the development set did not improve after four epochs, and restarted training with the parameters that obtained the maximum F_1 score. We repeated this process and terminated the training when the new learning rate was less than 1/16 of the initial value. The initial learning rate of each model was selected from $\{0.00002, 0.00005, 0.0001, 0.0002, 0.0005\}$. The output threshold for each label $\theta_c \in [0.0, 1.0]$ was searched in increments of 0.01 to maximize the F_1 score in the training data.

Preprocessing As initial word embeddings, we used vectors obtained via the same procedure as the one proposed by Matsubayashi and Inui (2017) using Japanese Wikipedia articles. These vectors were fine-tuned in the training. Following their approach, we used part of speech (PoS) vectors for words that were not contained in the lexicon of the Wikipedia articles. We used the CaboCha parser v0.68 [5] with the JUMAN dictionary for word segmentation and PoS tagging of NTC.

[5] https://taku910.github.io/cabocha/