Multi-task Learning for Japanese Predicate Argument Structure Analysis

Hikaru Omori
Tokyo Metropolitan University
omori-hikaru@ed.tmu.ac.jp

Mamoru Komachi
Tokyo Metropolitan University
komachi@tmu.ac.jp

Abstract

An *event-noun* is a noun that has an argument structure similar to a predicate. Recent works, including those considered state-of-the-art, ignore event-nouns or build a single model for solving both Japanese predicate argument structure analysis (PASA) and event-noun argument structure analysis (ENASA). However, because there are interactions between predicates and event-nouns, it is not sufficient to target only predicates. To address this problem, we present a multi-task learning method for PASA and ENASA. Our multi-task models improved the performance of both tasks compared to a single-task model by sharing knowledge from each task. Moreover, in PASA, our models achieved state-of-the-art results in overall F1 scores on the NAIST Text Corpus. In addition, this is the first work to employ neural networks in ENASA.

1 Introduction

Japanese predicate argument structure analysis (PASA) examines semantic structures between the predicate and its arguments in a text. The identification of the argument structure such as "who did what to whom?" is useful for natural language processing that requires deep analysis of complicated sentences such as machine translation and recognizing textual entailment. PASA is a task targeted at predicates such as verbs and adjectives. However, there are also many nouns that have event-related arguments in a sentence. We call these nouns that refer to events *event-nouns*, for example, a verbal noun (*sahen* nouns) such as *houkoku* "report" or a deverbal noun (nominalized forms of verbs) such as *sukui* "rescue."

Figure 1 shows examples of PASA and event-noun argument structure analysis (ENASA). In the NAIST Text Corpus (Iida et al., 2007), both predicates and event-nouns have one of three core

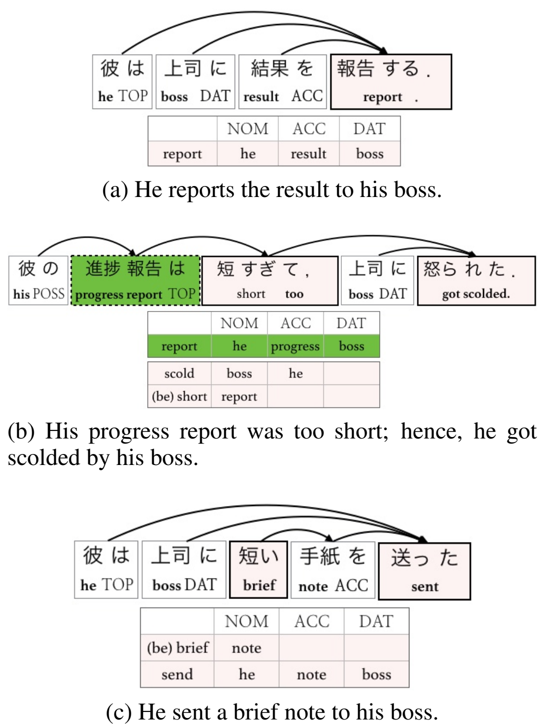

Figure 1: Examples of PASA and ENASA. The edges denote dependency paths.

case roles, *nominative* (NOM), *accusative* (ACC), and *dative* (DAT) as an argument. According to Iida et al. (2007), predicates have almost no argument in the same *bunsetsu*[1] phrase. However, in the case of event-nouns, approximately half of the accusative and dative arguments appear in the same bunsetsu phrase. Accordingly, although PASA and ENASA are semantically highly related, they are syntactically different tasks. However, most previous studies focused on predicates only; hence, there are few studies that focus

[1] Functional chunk in Japanese. It consists of one or more content words (noun, verb, adjective, etc.) followed by zero or more function words (postposition, auxiliary verb, etc.). A verb phrase in Japanese thus cannot bear noun arguments in the same bunsetsu.

Proceedings of NAACL-HLT 2019, pages 3404–3414
Minneapolis, Minnesota, June 2 - June 7, 2019. ©2019 Association for Computational Linguistics

on event-nouns (Komachi et al., 2007; Taira et al., 2008). To identify the semantic units of a sentence and to correctly understand syntactic relations, it is not sufficient to target only PASA.

Thus, we propose a multi-task learning model that effectively leverages ENASA and improves PASA. Our proposed model is based on an end-to-end multilayer bi-directional recurrent neural network (RNN) used in recent works, and the model has networks that distinguish task-independent information and task-specific information.

In summary, the main contributions of this work are the following:

1. This is the first attempt to design a multi-task learning framework for PASA and ENASA, and we show that our models improve the performance of both tasks.

2. Although our model is a simple model that does not consider the interactions between multiple predicates, it achieves a state-of-the-art result on the NAIST Text Corpus (NTC) in PASA by combining syntactic information as one of the features.

3. For ENASA, this is the first work to employ neural networks to effectively incorporate PASA.

2 Related Work

2.1 Japanese PASA and ENASA Approaches

Many machine learning-based methods have been studied in Japanese PASA. Traditional models take pointwise approaches that construct independent models for each core case role (NOM, ACC, DAT). Taira et al. (2008) proposed a supervised model that learns features of each case using decision lists and support vector machines. Imamura et al. (2009) proposed a model that combines a maximum entropy model with a language model trained from large-scale newspaper articles. Hayashibe et al. (2011) designed three models exploiting argument position and type and determined the maximum likelihood output using pairwise comparison.

However, the joint approach that optimizes the scores of all predicate-argument pairs in a sentence simultaneously showed better results than the pointwise approach. Yoshikawa et al. (2011) proposed a model that considers dependency between multiple predicate-argument relations using Markov logic networks. Ouchi et al. (2015) jointly optimized the combinations among multiple predicates and arguments in a sentence using a bipartite graph.

Except for (Taira et al., 2008), these studies focused on the analysis of predicates while there are few studies that focus on event-nouns. Komachi et al. (2007) decomposed ENASA into two tasks: *event-hood* determination and argument identification; they proposed a supervised method using lexico-syntactic patterns. Event-hood determination is the most important characteristic that semantically differentiates ENASA from PASA. It is a task to determine whether a noun refers to an event (e.g., *houkoku* can refer to either “to report” or the outcome of reporting action, “a report”). Since the previous ENASA models adopted the pointwise approach with a single model, they did not explore the effective features in each task. In contrast, our models simultaneously optimize three core case roles. Moreover, the proposed models allow us to distinguish between task-shared and task-specific features using multi-task learning.

2.2 PASA using neural networks

Some neural models have achieved higher performance than traditional machine learning models in Japanese PASA. Shibata et al. (2016) replaced Ouchi et al. (2015)’s scoring function with feed forward neural networks. Matsubayashi and Inui (2017) represented a dependency path between a predicate and its argument with path embeddings and showed that even the local model without multiple predicates can outperform a global model.

Moreover, some end-to-end models have been proposed in Japanese PASA. Ouchi et al. (2017) proposed an end-to-end model based on the model using eight-layer bi-directional long short-term memory (LSTM) proposed by Zhou and Xu (2015) and considered the interaction of multiple predicates simultaneously using a Grid RNN. Matsubayashi and Inui (2018) combined self-attention with Ouchi et al. (2017)’s model to directly capture interaction among multiple predicate-arguments. In particular, the model improved the performance of arguments that have no syntactic dependency with predicates and achieved a state-of-the-art result on Japanese PASA.

2.3 Semantic Role Labeling

Semantic role labeling (SRL) is a similar task to Japanese PASA. Recently, several end-to-end models using neural networks showed high performance in English SRL (Zhou and Xu, 2015; He et al., 2017; Tan et al., 2018). Strubell et al. (2018) proposed a multi-task learning model that jointly learned dependency parsing, part-of-speech tagging, predicate detection, and SRL based on multi-head self-attention. Ouchi et al. (2018) proposed a span-based SRL model using bi-directional LSTMs and achieved state-of-the-art results. The authors scored all possible spans for each label and selected correct spans satisfying constraints when decoding. In terms of the event-noun research, Gerber and Chai (2010) used pointwise mutual information (PMI) as a feature for 10 event-nouns with high frequency and identified semantic roles using a logistic regression model.

There were several LSTM models that also achieved high accuracy gains in Chinese SRL (Wang et al., 2015; Roth and Lapata, 2016; Sha et al., 2016; Marcheggiani et al., 2017; Qian et al., 2017). For event-nouns, Li et al. (2009) showed that combining effective features in verbal SRL with nominal SRL can improve results. Although the authors did not demonstrate that verbal SRL also improves performance in combination with nominal SRL, we show that our model improves performance in both PASA and ENASA.

3 Japanese PASA and ENASA

3.1 Task Description

Japanese predicate (event-noun) argument structure analysis is a task to extract arguments for certain predicates (event-nouns) and assign three case labels, NOM, ACC and DAT (Iida et al., 2007). Arguments are divided into four categories (Taira et al., 2008) according to the positions with their predicates (event-nouns).

Dep Arguments depend on their predicate (event-noun), or a predicate (event-noun) depends on its arguments.

Zero Arguments and their predicate (event-noun) are in the same sentence, but the arguments are omitted by zero anaphora. Therefore, they have no direct dependency.

Inter-zero Zero anaphoric arguments and their predicate (event-noun) are not in the same sentence.

Bunsetsu Arguments and their event-noun are in the same bunsetsu.

A sentence $w = w_1, w_2, \cdots, w_T$ and a predicate (event-noun) $p = p_1, p_2, \cdots, p_q$ are given as input. Iida et al. (2006), Imamura et al. (2009), and Sasano and Kurohashi (2011) also analyze Inter-zero, which is a difficult task because the whole document must be searched. Following existing research (Ouchi et al., 2015, 2017; Matsubayashi and Inui, 2017, 2018; Taira et al., 2008), we only focus on three categories where arguments and their predicate (event-noun) are in the same sentence. In addition, we exclude the Bunsetsu category from the PASA evaluation following Ouchi et al. (2017) and Matsubayashi and Inui (2018).

3.2 End-to-end Single Model

Our single model is based on an end-to-end approach (Zhou and Xu, 2015; Ouchi et al., 2017; Matsubayashi and Inui, 2018). Additionally, we add new features. Figure 2 shows the network architecture of our base model.

3.2.1 Input Layer

Each word $w_t \in [w_1, \cdots, w_T]$ is converted to a feature representation $\mathbf{x}_t \in [\mathbf{x}_1, \cdots, \mathbf{x}_T]$ at the input layer. We use six types of features. The feature representation $\mathbf{x}_t$ is defined as follows:

$$\mathbf{x}_t = \mathbf{x}_t^{\text{as}} \oplus \mathbf{x}_t^{\text{posi}} \oplus \mathbf{x}_t^{\text{dep}} \oplus \mathbf{x}_t^{\text{type}} \oplus \mathbf{x}_p^{\text{task}} \quad (1)$$

where ($\oplus$) indicates concatenation of vectors.

Argument Structure Predicate (event-noun) w_p and argument candidates w_t are converted to the vectors $\mathbf{x}_t^{\text{as}} \in \mathbb{R}^{2d_{\text{w}}}$ by the word embedding matrix.

Position This is a feature that represents the positional relation between w_p and w_t. The feature is calculated by subtracting the word index of argument candidates from the word index of predicates (event-nouns). We use two types of units to represent relative position: word unit p_t^{word} and bunsetsu unit p_t^{bunsetsu}, which are converted to the word positional vector $\mathbf{p}_t^{\text{word}} \in \mathbb{R}^{d_{\text{p}}}$ and the bunsetsu positional vector $\mathbf{p}_t^{\text{bunsetsu}} \in \mathbb{R}^{d_{\text{p}}}$, respectively, by the word and bunsetsu positional embed-

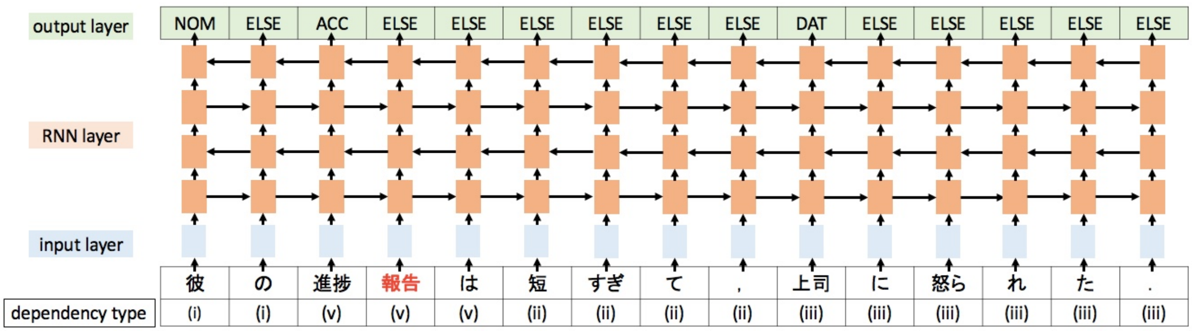

Figure 2: End-to-end single model.

ding matrices. We concatenate these two vectors and obtain the positional vectors $\mathbf{x}_t^{\mathrm{posi}} \in \mathbb{R}^{2d_{\mathrm{p}}}$.

Dependency This is a feature that represents the dependency relation between w_p and w_t. We set five types of dependency relations:

i). Argument candidates depend on the predicate (event-noun).

ii). The predicate (event-noun) depends on the argument candidates.

iii). No dependency relations between the predicate (event-noun) and argument candidates.

iv). The predicate and candidate arguments are in the same bunsetsu.

v). The event-noun and candidate arguments are in the same bunsetsu.

The dependency relation d_t is converted to the dependency vector $\mathbf{x}_t^{\mathrm{dep}} \in \mathbb{R}^{d_{\mathrm{d}}}$ by the dependency relation embedding matrix. The dependency type in Figure 2 shows how to make dependency features in Figure 1b as an example. We define the dependency type from the syntactic information annotated in the NTC.

In previous work, dependency features are used differently from our study. Imamura et al. (2009) used a binary feature that represents whether or not there is a dependency relation between the predicate and its arguments. We employ more fine-grained relation types to adapt to event-nouns. Matsubayashi and Inui (2017) represented the interactions between a predicate and its arguments using path embedding. In contrast, we define different types for a predicate and event-noun to distinguish event-nouns from predicates and learn embeddings to find the associated latent structures.

Event-hood Type This is a binary feature to flag all predicates (event-nouns) in a sentence inspired by Matsubayashi and Inui (2018). The purpose of this feature is to prevent predicates from becoming arguments and to help some event-nouns become arguments. The event-hood type vector $\mathbf{x}_t^{\mathrm{type}} \in \mathbb{R}^2$ of a candidate indicates [0,1] if the candidate is a predicate, [1,0] if the candidate is an event-noun, and [0,0] otherwise. The predicate and event-noun are annotated in the NTC.

Task Label This is a binary feature vector $\mathbf{x}_p^{\mathrm{task}} \in \mathbb{R}^1$ that indicates 1 if the task is predicate argument structure analysis; otherwise, 0.

3.2.2 RNN Layer

We use the gated recurrent unit (GRU) (Cho et al., 2014) for RNN. The RNN layers are made up of L layers of stacked bi-directional GRU. Additionally, we apply the residual connections (He et al., 2016) following Ouchi et al. (2017); Matsubayashi and Inui (2018). At each time step t, the hidden state $\mathbf{h}_t^l \in \mathbb{R}^{d_{\mathrm{h}}}$ in the $l \in [1, \cdots, L]$-th layer is calculated as follows:

$$\mathbf{h}_t^l = \begin{cases} g^l(\mathbf{h}_t^{l-1}, \mathbf{h}_{t-1}^l) & (l = \mathrm{odd}) \\ g^l(\mathbf{h}_t^{l-1}, \mathbf{h}_{t+1}^l) & (l = \mathrm{even}) \end{cases} \quad (2)$$

where $g^l(\cdot)$ denotes the l-th layer GRU function. In addition, $\mathbf{h}_t^0 = \mathbf{x}_t$.

3.2.3 Output Layer

In the output layer, we input each hidden state $\mathbf{h}_t^L$. Then, we obtain the output vector $\mathbf{o}_t$ using the softmax function:

$$\mathbf{o}_t = \mathrm{softmax}(\mathbf{W}_{\mathrm{o}}\mathbf{h}_t^L + \mathbf{b}_{\mathrm{o}}) \quad (3)$$

where $\mathbf{W}_{\mathrm{o}} \in \mathbb{R}^{4 \times d_{\mathrm{h}}}$ is the parameter matrix, and $\mathbf{b}_{\mathrm{o}} \in \mathbb{R}^4$ is the bias term. The output vector represents the probability for each argument candidate

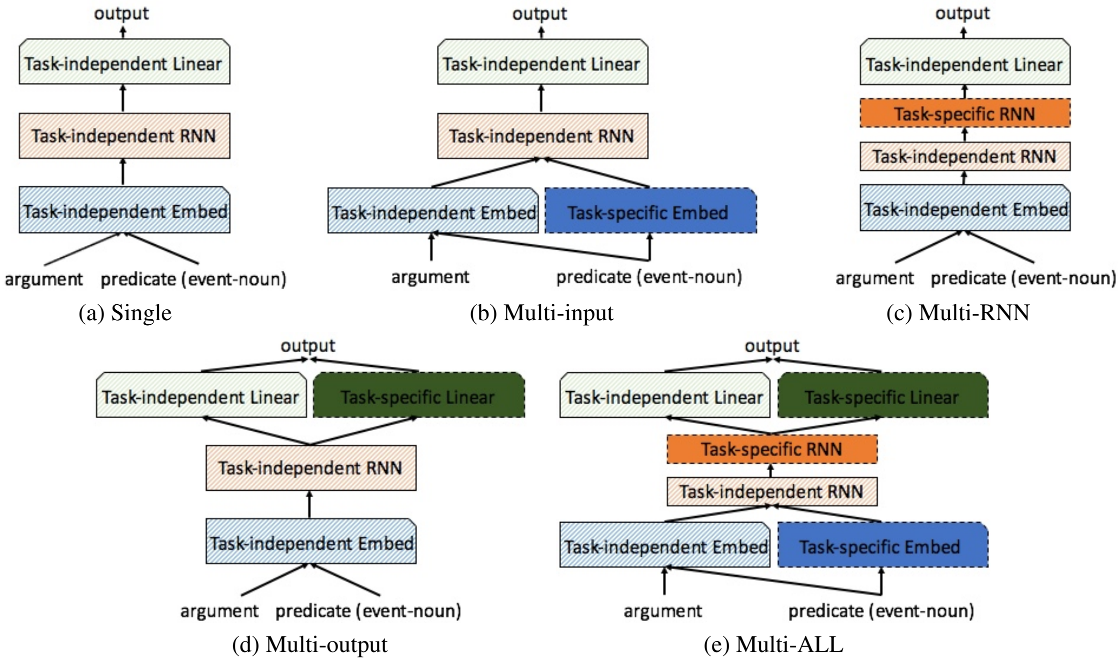

Figure 3: Proposed models: (a) Single, (b) Multi-input, (c) Multi-RNN, (d) Multi-output, (e) Multi-ALL.

over four labels, [NOM, ACC, DAT, ELSE]. ELSE denotes that the candidate argument does not have a case label. In testing, the maximum probability label is selected as the output label. We train the model using the cross-entropy loss function.

4 Multi-task Model

Multi-task learning has been successfully applied to various natural language processing tasks (Collobert et al., 2011; Søgaard and Goldberg, 2016; Luong et al., 2016; Hashimoto et al., 2017; Liu et al., 2017; Stoyanov et al., 2018; Marasovic and Frank, 2018; Strubell et al., 2018). One of the advantages of multi-task learning is that it learns better representation, which is robust against task-dependent noise by increasing training data. In this paper, we introduce multi-task learning to PASA and ENASA for the first time. We propose three methods to extend the end-to-end single model to the multi-task learning model in the input layer, RNN layer, and output layer. Figure 3 shows the proposed models. Our final model combines all three methods (Figure 3e).

4.1 Multi Input Layer

Even if the surface form is the same, the contexts are different for predicates and event-nouns. For example, the event-noun *houkoku* "report" in Figure 1b has an argument in the same bunsetsu unlike predicates. Moreover, the event-noun also has a nominative argument role for the predicate *mijikai* "short". Therefore, given this, we prepare a task-specific word embedding matrix that addresses the task-specific distribution of words. The predicate is converted to PASA-specific vectors $\mathbf{x}_t^{\mathrm{p}} \in \mathbb{R}^{d'_{\mathrm{w}}}$ by the PASA-specific predicate embedding matrix. Similarly, the event-noun is converted to ENASA-specific vectors $\mathbf{x}_t^{\mathrm{n}} \in \mathbb{R}^{d'_{\mathrm{w}}}$ by the ENASA-specific event-noun embedding matrix. These matrices are randomly initialized and can be learned during training.

The feature vector $\overline{\mathbf{x}}_t$ is defined as follows:

$$\overline{\mathbf{x}}_t = \begin{cases} \mathbf{x}_t \oplus \mathbf{x}_t^{\mathrm{p}} & (\mathrm{PASA}) \\ \mathbf{x}_t \oplus \mathbf{x}_t^{\mathrm{n}} & (\mathrm{ENASA}) \end{cases} \quad (4)$$

4.2 Multi RNN Layer

Previous work (Søgaard and Goldberg, 2016; Hashimoto et al., 2017) proposed hierarchical multi-task learning models that exploited features obtained from easy tasks for difficult tasks. These studies showed that performance improves when low-layer RNN representations are trained in easy tasks and high-layer RNN are leveraged for difficult tasks. Therefore, we construct a network that hierarchically overlaps a task-specific RNN on a task-independent RNN. Lower RNN layers learn task-independent knowledge representations. Then, the task-specific RNN adjusts the representations for each task. At each time step t, the hidden state $\mathbf{m}_t^{l'} \in \mathbb{R}^{d'_\mathrm{h}}$ in the $l' \in [1, \cdots, L']$-th layer is calculated as follows:

$$\mathbf{m}_t^{l'} = \begin{cases} g^{l'}(\mathbf{m}_t^{l'-1}, \mathbf{m}_{t-1}^{l'}) & (l' = \mathrm{odd}) \\ g^{l'}(\mathbf{m}_t^{l'-1}, \mathbf{m}_{t+1}^{l'}) & (l' = \mathrm{even}) \end{cases} \quad (5)$$

$$g^{l'}(\cdot) = \begin{cases} g_\mathrm{p}^{l'}(\cdot) & (\mathrm{PASA}) \\ g_\mathrm{n}^{l'}(\cdot) & (\mathrm{ENASA}) \end{cases} \quad (6)$$

where $g^{l'}(\cdot)$, $g_\mathrm{p}^{l'}(\cdot)$, and $g_\mathrm{n}^{l'}(\cdot)$ denote the l'-th layer GRU functions. In addition, $\mathbf{m}_t^0 = \mathbf{h}_t^L$.

4.3 Multi Output Layer

The position of arguments is different with respect to predicates and event-nouns. For example, predicates seldom have arguments in the same bunsetsu. In contrast, event-nouns often have arguments in the same bunsetsu, compound nouns, for example. Therefore, it is intuitive and natural to divide the output layer into task-independent and task-specific layers. The task-specific output vectors are calculated as follows:

$$\mathbf{o}_t^\mathrm{p} = \mathbf{W}_\mathrm{o}^\mathrm{p}\mathbf{h}_t + \mathbf{b}_\mathrm{o}^\mathrm{p} \quad (7)$$

$$\mathbf{o}_t^\mathrm{n} = \mathbf{W}_\mathrm{o}^\mathrm{n}\mathbf{h}_t + \mathbf{b}_\mathrm{o}^\mathrm{n} \quad (8)$$

$$\mathbf{g}_t = \sigma(\mathbf{W}_\mathrm{g}\mathbf{h}_t + \mathbf{b}_\mathrm{g}) \quad (9)$$

where $\mathbf{W}_\mathrm{o}^\mathrm{p}, \mathbf{W}_\mathrm{o}^\mathrm{n}, \mathbf{W}_\mathrm{g} \in \mathbb{R}^{4 \times d_\mathrm{h}}$ are the parameter matrices, and $\mathbf{b}_\mathrm{o}^\mathrm{p}, \mathbf{b}_\mathrm{o}^\mathrm{n}, \mathbf{b}_\mathrm{g} \in \mathbb{R}^4$ are the bias terms. $\mathbf{h}_t$ is the hidden state of the last layer. We combine task-specific output vectors $\mathbf{o}_t^\mathrm{p}, \mathbf{o}_t^\mathrm{n}$ with task-independent output vector $\mathbf{o}_t$ by the gate $\mathbf{g}_t$.

$$\mathbf{c}_t = \begin{cases} \mathbf{g}_t \odot \mathbf{o}_t + (1 - \mathbf{g}_t) \odot \mathbf{o}_t^\mathrm{p} & (\mathrm{PASA}) \\ \mathbf{g}_t \odot \mathbf{o}_t + (1 - \mathbf{g}_t) \odot \mathbf{o}_t^\mathrm{n} & (\mathrm{ENASA}) \end{cases} \quad (10)$$

$$\overline{\mathbf{o}}_t = \mathrm{softmax}(\mathbf{c}_t) \quad (11)$$

where ($\odot$) denotes the element-wise product. The output vector $\overline{\mathbf{o}}_t$ represents the probability of [NOM, ACC, DAT, ELSE].

5 Experiments

5.1 Dataset and Setting

We use NTC 1.5 for our experiments. We divide the dataset into training, development, and test sets in the same way as Taira et al. (2008). We use morphological and syntactic information, such as the word boundaries, the bunsetsu boundaries and the dependency relations provided in the NTC.

For the development and test sets, if there are two or more arguments annotated with the same case label in a sentence, we set an argument that only has a dependency relation with a predicate as a correct answer and assign the ELSE label to other arguments. If there is no dependency relation, we set an argument with the shortest distance $|w_p - w_t|$ as a correct answer. If the distance is equal, an argument on the left side of a predicate is considered a correct answer.

In NTC 1.5, if there is a predicate phrase, such as "verbal noun + *suru*," *suru* is annotated as a predicate word. We consider the verbal noun as the predicate word at the preprocessing step to match the surface of a predicate with that of an event-noun. Take the predicate *houkoku-suru* "to report" and an event-noun *houkoku* "report" as an example. Although w_p before preprocessing are *suru* and *houkoku*, w_p are unified to *houkoku* after preprocessing.

the dimension of word embeddings d_w	300
the dimension of position embeddings d_p	16
the dimension of dependency embeddings d_d	16
the dimension of hidden states d_h	300
the number of GRU layers L	4
the dimension of task-specific word embeddings d'_w	16
the dimension of task-specific hidden states d'_h	300
the number of task-specific GRU layers L'	2
dropout rate	0.4
batch size	8
gradient clipping	4

Table 1: Hyperparameters.

5.2 Hyperparameters

We use pre-trained embeddings[2] for the initial values of the word embedding matrix. The initial values of the other embedding matrices are sampled according to a uniform distribution of [-0.25,0.25]. We convert words appearing more than once in the training set into word vectors and the remaining words into the unknown word vector. We adopt AdaDelta ($\epsilon = 10^{-6}$, $\rho = 0.95$) as the optimization method. We set the number of epochs to 20 and evaluate the model with the highest F1 scores on the development set. Table 1 shows the hyperparameters.

5.3 Results

We evaluate each model with the NTC 1.5 test. The experimental results for the argument structure analysis of predicates and event-nouns are shown in Tables 2 and 3.

[2] http://www.asahi.com/shimbun/medialab/word_embedding

Method	ALL	SD	*Dep* ALL	*Dep* NOM	*Dep* ACC	*Dep* DAT	*Zero* ALL	*Zero* NOM	*Zero* ACC	*Zero* DAT
Ouchi+ 17	81.42		88.17	88.75	93.68	64.38	47.12	50.65	32.35	7.52
M&I 17	83.50	±0.17	89.89	91.19	95.18	61.90	51.79	54.69	41.8	17
M&I 18	83.94	±0.12	90.26	90.88	94.99	67.57	**55.55**	**57.99**	**48.9**	**23**
Single	83.62	±0.17	90.09	90.45	94.84	69.77	51.87	54.73	43.48	11.40
Multi-input	83.88	±0.11	90.27	90.65	95.12	69.86	53.01	55.82	44.68	10.77
Multi-RNN	83.91	±0.23	90.17	90.58	95.07	67.94	53.31	55.85	45.71	9.97
Multi-output	83.77	±0.20	90.13	90.68	94.89	68.16	53.93	56.73	43.79	9.45
Multi-ALL	83.82	±0.10	90.15	90.68	95.06	67.56	53.50	56.37	45.36	8.70
Multi-RNN+DEP	84.55	±0.11	90.69	91.28	95.25	70.07	51.56	54.29	42.67	1.85
Multi-output+DEP	84.73	±0.11	90.82	**91.46**	95.29	70.69	52.29	55.14	42.15	1.81
Multi-ALL+DEP	**84.75**	±0.16	**90.88**	91.40	**95.37**	**71.02**	52.35	55.10	42.54	2.32
M&I 17 (ens. of 5)	84.07		90.24	91.59	95.29	62.61	53.66	56.47	44.7	16
M&I 18 (ens. of 10)	85.34		91.26	91.84	95.57	70.8	**58.07**	**60.21**	**52.5**	**26**
Multi-RNN+DEP (ens. of 5)	85.85		91.61	92.11	**95.87**	72.63	53.41	55.96	46.10	0
Multi-output+DEP (ens. of 5)	85.83		91.52	92.12	95.69	72.72	54.35	57.02	45.95	0
Multi-ALL+DEP (ens. of 5)	**86.01**		**91.63**	**92.15**	95.80	**72.95**	54.99	57.84	45.20	0

Table 2: F1 scores on the PASA test set. Single is a base model without multi-task learning.

Method	ALL	SD	*Dep* ALL	*Dep* NOM	*Dep* ACC	*Dep* DAT	*Zero* ALL	*Zero* NOM	*Zero* ACC	*Zero* DAT	*Bunsetsu* ALL	*Bunsetsu* NOM	*Bunsetsu* ACC	*Bunsetsu* DAT
Taira+ 08 on NTC 1.4				68.01	62.46	56.05		36.19	20.46	6.62		78.93	77.96	58.13
Single	66.21	±0.15	74.64	76.06	74.54	51.28	46.05	49.67	33.36	13.63	78.24	76.67	81.75	48.55
Multi-input	67.89	±0.42	75.62	76.63	75.78	**57.17**	49.07	52.81	36.95	19.39	**79.35**	77.31	**83.31**	51.03
Multi-RNN	67.96	±0.44	75.86	76.90	76.33	54.46	48.67	52.18	**38.47**	18.89	79.08	77.24	82.89	50.93
Multi-output	67.96	±0.17	76.25	**77.18**	**76.90**	54.97	48.74	52.48	36.09	**19.64**	79.02	77.00	83.04	50.60
Multi-ALL	**68.00**	±0.41	75.90	77.16	76.05	53.00	**49.66**	**53.37**	37.64	14.46	79.05	77.32	82.61	**51.83**
Multi-ALL+DEP	67.68	±0.39	**75.95**	**77.18**	76.11	55.26	47.57	51.21	35.14	15.65	79.06	**77.44**	82.66	51.10
Multi-ALL (ens. of 5)	**71.14**		**78.63**	**79.66**	**78.83**	58.29	**52.49**	**56.41**	**39.02**	16.42	**81.90**	**80.25**	**85.21**	**56.29**
Multi-ALL+DEP (ens. of 5)	69.90		77.86	78.89	78.16	**58.46**	49.36	53.10	36.36	17.23	81.16	79.74	84.57	52.99

Table 3: F1 scores on the ENASA test set.

Predicate Argument Structure Analysis The first set of rows in Table 2 shows the results of previous models. Ouchi+ 17 is the model from the Multi-Seq model in (Ouchi et al., 2017). M&I 17 is the model in (Matsubayashi and Inui, 2017). M&I 18 is the model from the MP-POOL-SELFATT model in (Matsubayashi and Inui, 2018).

The second set of rows in Table 2 shows the results of the proposed models. These models do not use the dependency feature. Compared with the single model, all multi-task learning models improved the overall F1 scores. Among them, Multi-RNN improved the overall F1 score from the single model by 0.29 points. In previous work, Ouchi et al. (2017); Matsubayashi and Inui (2018) see improvements of 0.27 and 0.55 F1 points in their baseline models by considering multiple predicate-argument interactions. Therefore, we show that multi-task learning with ENASA achieved comparable effects as these studies in PASA.

The third set of rows shows the results of proposed models using all features including the dependency feature. Multi-ALL+DEP achieved the best F1 score among all the models including previous state-of-the-art models. In particular, the dependency feature was effective for Dep arguments. On the other hand, the performance for Zero arguments was poor. This result suggests that the dependency feature causes the model to optimize mainly for Dep arguments since Dep arguments are more numerous than Zero arguments.

The fourth set of rows shows the results of ensemble models. Overall, our proposed model outperformed the previous ensemble model by 0.67 points in the overall F1 score. Moreover, our models are simple models that independently analyze each predicate in a sentence. Nevertheless, our models achieved higher results than Ouchi et al. (2017); Matsubayashi and Inui (2018). Although recent works have researched the method whereby multiple predicate-argument interactions are considered simultaneously, how to use syntactic in-

社会党 の 山花 貞夫 ・ 新 民主 連合 会長 [NOM] は 十 七 日 午前 、 新た に 結成 [PRED] する
Social Party of Yamahana Sadao ・ new democratic federation president NOM 1 7 day a.m. , newly organize
新 会派 [ACC] に 参加 する 同 党 所属 国会 議員 二 十 四 人 の 「 日本 社会党 ・ 護憲
new faction ACC participate the said party belonging Diet member 2 1 4 people of " Japan Social Party ・ constitution protection
民主 連合 」 から の 離党 届 を 森井 忠良 国対 委員 長 に 提出 し た 。
democratic federation " from secession notification ACC Morii Tadayoshi National Diet committee chair DAT submitted .

New Democratic Federation President Yamada Sadao, a member of Social Democratic Party of Japan, submitted to Mori Tadayoshi, National Diet Committee Chairperson, a notice of resignation from a party "Japan Social Party-Pro-constitution Democratic Federation" from 24 Diet members of SDP who will join a newly organized new faction in the morning of the 17th.

(a) predicate: 結成 "organize," NOM: 会長 "president," ACC: 会派 "faction."

問題 の 行方 [ACC] を 左右 [PRED] する カギ [NOM] は 「 電子 レビュー 」 の よう な もの かも しれ ない 。
Problem of whereabouts ACC determine key NOM " electronic review " like may .

The key to determine whereabouts of the problem may be like an "electronic review."

(b) predicate: 左右 "determine," NOM: カギ "key," ACC: 行方 "whereabouts."

事態 [ACC] 打開 [EN] の カギ [NOM] を 握る の は 、 セルビア 人 勢力 に 影響 力 を 持つ ロシア 及び 新 ユーゴスラビア だろ う 。
situation break of key ACC hold NOM , Serbs peopel Influence DAT affect power ACC have Russia and new Yugoslavia will .

Russia and New Yugoslavia, which are affecting Serb influence, will hold the key to break the situation.

(c) event-noun: 打開 "break," NOM: カギ "key," ACC: 事態 "situation."

行政 、 銀行 、 農協 系統 の 経営 トップ [NOM] が 責任 [ACC] 回避 [PRED] する の は なぜ か 。
administration , bank , agricultural coorperative system of management top NOM responsibility avoid NOM why .

Why do top managements of administration, bank and agricultural cooperative avoid responsibility?

(d) predicate: 回避 "avoid," NOM: トップ "top," ACC: 責任 "responsibility."

私 は 、 紙幣 を 改めて 見る と 、 なるほど 千 円 札 の 左下 [DAT] に 直径 五 ミリ ほど の 丸 [ACC] が 一つ 押さ [PRED] れ て ある 。
I NOM , bill ACC again look when , I see 1000 yen bill of lower left DAT diameter 5 mm about of circle NOM one push PASSIVE

When I look at the bill again, I see, a circle about 5 mm in diameter is pushed at the lower left of 1000 yen bill.

(e) predicate: 押す "push," ACC: 丸 "circle," DAT: 左下 "lower left."

ロージズ 博士 は 、 アポ E [NOM] が 発病 [DAT] に 果たす [PRED] 役割 [ACC] の 解明 [ACC/FALSE] に 取り組ん で おり 、
PERSON doctor NOM , Apolipoprotein E NOM pathogenesis DAT play role of solution DAT tackle ,
「 発病 を 予防 する 薬 の 開発 を めざし て いる 」 と いう 。
" onset ACC prevent drug of develop ACC aim " .

Dr. PERSON is working on elucidating the role Apolipoprotein E plays in the pathogenesis, and says "We aim to develop drugs to prevent onset".

(f) predicate: 果たす "play," NOM: E, ACC: 役割 "role," DAT: 発病 "pathogenesis."

Figure 4: Examples of analysis errors on the PASA test set

formation in the end-to-end model is a subject for future work.

Event-noun Argument Structure Analysis The first set of rows in Table 3 shows the results of a previous model in event-noun argument structure analysis. Taira+ 08 is the model from (Taira et al., 2008). Since its scores are from NTC 1.4, the model cannot be directly compared to our models. Compared with the single model, all multi-task models improved the overall F1 scores. However, Multi-ALL+DEP compared unfavorably with Multi-ALL even though it was the best PASA architecture. Therefore, this implies that the dependency type feature between the predicate and its argument is not effective in ENASA.

5.4 Analysis

In Figure 4, we compare the PASA results from test sets for each model. In Examples (a), (b) and (d), the single model failed to predict correct arguments but the Multi-RNN model correctly predicted arguments. In Example (a), the single model incorrectly predicted that arguments do not exist in this sentence. Comparing the training set of each task, although the number of event-nouns is approximately one-third of the number of predicates, the number of *kessei* 結成 "organize (event-nouns)" is approximately twice the number of *kessei* 結成 "organize (predicates)." Accordingly, we showed that the Multi-RNN model effectively leverages the information of event-nouns using multi-task learning.

In Example (b), the single model incorrectly predicted that the NOM argument does not exist, but the multi-RNN predicted the correct arguments. Comparing the training set, there is *sayuu* 左右 "determine (predicate)" but not *sayuu* 左右 "determine (event-noun)." However, there are some *kagi* カギ "key (arguments of predicates)" in the PASA training set, and there is one *kagi* カギ "key (argument of event-noun)" in the ENASA training set (Example (c)). Moreover, in Example (c), *dakai* 打開 "break (event-noun)" depends on *kagi* カギ "key" like *sayuu* 左右 "determine (predicate)" in Example (b); however, no predicate depends on *kagi* カギ "key" in the training set. Accordingly, the Multi-RNN model also leverages the arguments of event-nouns and the positional relations between event-nouns and their arguments.

Example (d) is an interesting case in which a predicate *kaihi* 回避 "avoid" and its argument *sekinin* 責任 "responsibility" are located in the same bunsetsu. Although this argument type (Bunsetsu) is excluded from the evaluation target in PASA, it is common as a compound noun in ENASA. Therefore, the single model wrongly predicted that the ACC argument does not exist, but multi-RNN was able to predict the answer using the specific knowledge of event-nouns.

In contrast, in Example (e), the single model correctly predicted the answer, but the multi-RNN model failed to predict the correct arguments. Multi-RNN incorrectly predicted that the DAT argument does not exist in this sentence. However, *ni* に, a postpositional particle located after an argument, often indicates a dative case. Nevertheless, multi-RNN often predicted a wrong DAT argument by ignoring *ni* に. Therefore, for DAT analysis, the information of event-nouns adversely affects PASA.

In Example (f), the Multi-ALL+DEP model correctly predicted the answer, but the Multi-ALL model failed. Specifically, Multi-ALL+DEP correctly predicted that the ACC argument is *yakuwari* 役割 "role," which is dependent on *hatasu* 果たす "play." However, the Multi-ALL incorrectly predicted that the ACC argument is *kaimei* 解明 "solution." Similarly, Multi-ALL without syntactic information made many mistakes, including attributive modification, such as Figure 1c. Table 4 shows the results of the two PASA models for attributive modification instances. Multi-ALL+DEP considerably outperformed Multi-ALL for all cases using dependency features. Therefore, these results suggest that the dependency type feature is effective for PASA with respect to attributive modifications.

	ALL	NOM	ACC	DAT
Multi-ALL	80.31	83.37	72.16	19.48
Multi-ALL+DEP	81.83	84.67	74.41	28.31

Table 4: F1 scores on the PASA test set with respect to attributive modifications.

6 Conclusion

We design a multi-task learning model for predicate and event-noun argument structure analysis. The experiment results show that the multi-task models outperform the single-task model on the NAIST Test Corpus for both tasks. Moreover, our model achieves a state-of-the-art result for PASA. In addition, this is the first work to employ neural networks for ENASA. In future work, we plan to consider multiple predicates and event-nouns.

References

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 *Proceedings of EMNLP*, pages 1724–1734.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537.

Matthew Gerber and Joyce Chai. 2010. Beyond NomBank: A study of implicit arguments for nominal predicates. In *Proceedings of ACL*, pages 1583–1592.

Kazuma Hashimoto, caiming xiong, Yoshimasa Tsuruoka, and Richard Socher. 2017. A joint many-task model: Growing a neural network for multiple NLP tasks. In *Proceedings of EMNLP*, pages 1923–1933.

Yuta Hayashibe, Mamoru Komachi, and Yuji Matsumoto. 2011. Japanese predicate argument structure analysis exploiting argument position and type. In *Proceedings of IJCNLP*, pages 201–209.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of CVPR*, pages 770–778.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and what's next. In *Proceedings of ACL*, pages 473–483.

Ryu Iida, Kentaro Inui, and Yuji Matsumoto. 2006. Exploiting syntactic patterns as clues in zero-anaphora resolution. In *Proceedings of COLING/ACL*, pages 625–632.

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.

Kenji Imamura, Kuniko Saito, and Tomoko Izumi. 2009. Discriminative approach to predicate-argument structure analysis with zero-anaphora resolution. In *Proceedings of ACL-IJCNLP*, pages 85–88.

Mamoru Komachi, Ryu Iida, Kentaro Inui, and Yuji Matsumoto. 2007. Learning based argument structure analysis of event-nouns in Japanese. In *Proceedings of PACLING*, pages 120–128.

Junhui Li, Guodong Zhou, Hai Zhao, Qiaoming Zhu, and Peide Qian. 2009. Improving nominal SRL in Chinese language with verbal SRL information and automatic predicate recognition. In *Proceedings of EMNLP*, pages 1280–1288.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-task learning for text classification. In *Proceedings of ACL*, pages 1–10.

Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. In *Proceedings of ICLR*.

Ana Marasovic and Anette Frank. 2018. SRL4ORL: Improving opinion role labeling using multi-task learning with semantic role labeling. In *Proceedings of NAACL*, pages 583–594.

Diego Marcheggiani, Anton Frolov, and Ivan Titov. 2017. A simple and accurate syntax-agnostic neural model for dependency-based semantic role labeling. In *Proceedings of CoNLL*, pages 411–420.

Yuichiroh Matsubayashi and Kentaro Inui. 2017. Revisiting the design issues of local models for Japanese predicate-argument structure analysis. In *Proceedings of IJCNLP*, pages 128–133.

Yuichiroh Matsubayashi and Kentaro Inui. 2018. Distance-free modeling of multi-predicate interactions in end-to-end Japanese predicate argument structure analysis. In *Proceedings of COLING*, pages 94–106.

Hiroki Ouchi, Hiroyuki Shindo, Kevin Duh, and Yuji Matsumoto. 2015. Joint case argument identification for Japanese predicate argument structure analysis. In *Proceedings of 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 *Proceedings of ACL*, pages 1591–1600.

Hiroki Ouchi, Hiroyuki Shindo, and Yuji Matsumoto. 2018. A span selection model for semantic role labeling. In *Proceedings of EMNLP*, pages 1630–1642.

Feng Qian, Lei Sha, Baobao Chang, LuChen Liu, and Ming Zhang. 2017. Syntax aware LSTM model for semantic role labeling. In *Proceedings of the 2nd Workshop on Structured Prediction for Natural Language Processing*, pages 27–32.

Michael Roth and Mirella Lapata. 2016. Neural semantic role labeling with dependency path embeddings. In *Proceedings of ACL*, pages 1192–1202.

Ryohei Sasano and Sadao Kurohashi. 2011. A discriminative approach to Japanese zero anaphora resolution with large-scale lexicalized case frames. In *Proceedings of IJCNLP*, pages 758–766.

Lei Sha, Sujian Li, Baobao Chang, Zhifang Sui, and Tingsong Jiang. 2016. Capturing argument relationship for Chinese semantic role labeling. In *Proceedings of EMNLP*, pages 2011–2016.

Tomohide Shibata, Daisuke Kawahara, and Sadao Kurohashi. 2016. Neural network-based model for Japanese predicate argument structure analysis. In *Proceedings of ACL*, pages 1235–1244.

Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of ACL*, pages 231–235.

Veselin Stoyanov, Heng Ji, Ying Lin, and Shengqi Yang. 2018. A multi-lingual multi-task architecture for low-resource sequence labeling. In *Proceedings of ACL*, pages 799–809.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of EMNLP*, pages 5027–5038.

Hirotoshi Taira, Sanae Fujita, and Masaaki Nagata. 2008. A Japanese predicate argument structure analysis using decision lists. In *Proceedings of EMNLP*, pages 523–532.

Zhixing Tan, Mingxuan Wang, Jun Xie, Yidong Chen, and Xiaodong Shi. 2018. Deep semantic role labeling with self-attention. In *Proceedings of AAAI*, pages 4929–4936.

Zhen Wang, Tingsong Jiang, Baobao Chang, and Zhifang Sui. 2015. Chinese semantic role labeling with bidirectional recurrent neural networks. In *Proceedings of EMNLP*, pages 1626–1631.

Katsumasa Yoshikawa, Masayuki Asahara, and Yuji Matsumoto. 2011. Jointly extracting Japanese predicate-argument relation with Markov logic. In *Proceedings of IJCNLP*, pages 1125–1133.

Jie Zhou and Wei Xu. 2015. End-to-end learning of semantic role labeling using recurrent neural networks. In *Proceedings of ACL-IJCNLP*, pages 1127–1137.