Entity-Centric Joint Modeling of Japanese Coreference Resolution and Predicate Argument Structure Analysis

Tomohide Shibata[†‡] and **Sadao Kurohashi**[†‡]
[†]Graduate School of Informatics, Kyoto University
Yoshida-honmachi, Sakyo-ku, Kyoto, 606-8501, Japan
[‡]CREST, JST
4-1-8, Honcho, Kawaguchi-shi, Saitama, 332-0012, Japan
{shibata, kuro}@i.kyoto-u.ac.jp

Abstract

Predicate argument structure analysis is a task of identifying structured events. To improve this field, we need to identify a salient entity, which cannot be identified without performing coreference resolution and predicate argument structure analysis simultaneously. This paper presents an entity-centric joint model for Japanese coreference resolution and predicate argument structure analysis. Each entity is assigned an embedding, and when the result of both analyses refers to an entity, the entity embedding is updated. The analyses take the entity embedding into consideration to access the global information of entities. Our experimental results demonstrate the proposed method can improve the performance of the inter-sentential zero anaphora resolution drastically, which is a notoriously difficult task in predicate argument structure analysis.

1 Introduction

Natural language often conveys a sequence of events like "who did what to whom", and extracting structured events from the raw text is a kind of touchstone for machine reading. This is realized by a combination of coreference resolution (called *CR*, hereafter) and predicate argument structure analysis (called *PA*, hereafter).

The characteristics and difficulties in the analyses vary among languages. In English, there are few omissions of arguments, and thus PA is relatively easy, around 83% accuracy (He et al., 2017), while CR is relatively difficult, around 70% accuracy (Lee et al., 2017).

On the other hand, in Japanese and Chinese, where arguments are often omitted, PA is a difficult task, and even state-of-the-art systems only achieve around 50% accuracy. Zero anaphora resolution (ZAR) is a difficult subtask of PA, detecting a zero pronoun and identifying a referent of the zero pronoun. As the following example shows, CR in English (identifying the antecedent of *it*) and ZAR in Japanese (identifying the omitted nominative argument) are similar problems.

(1) a. John bought a car last month.
It was made by Toyota.

b. ジョンは 先月 車を 買った。
John-TOP last month a car-ACC bought.
(ϕが) トヨタ製だった。
(ϕ-NOM) Toyota made-COPULA.

Note that CR such as the relation between "the company" and "Toyota" is also difficult in Japanese.

According to the argument position relative to the predicate, ZAR is classified into the following three types:

- *intra-sentential* (*intra* in short): an argument is located in the same sentence with the predicate
- *inter-sentential* (*inter* in short): an argument is located in the preceding sentences, such as "車" for "トヨタ製だった" (Toyota made-COPULA) in sentence (1b)
- *exophora*: an argument does not appear in a document, such as *author* and *reader*

Among these three types, the analysis of *inter* is extremely difficult because there are many candidates in preceding sentences, and clues such as a dependency path between a predicate and an argument cannot be used.

This paper presents a joint model of CR and PA in Japanese. It is necessary to perform them together because PA (especially inter-sentential

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers), pages 579–589
Melbourne, Australia, July 15 - 20, 2018. ©2018 Association for Computational Linguistics

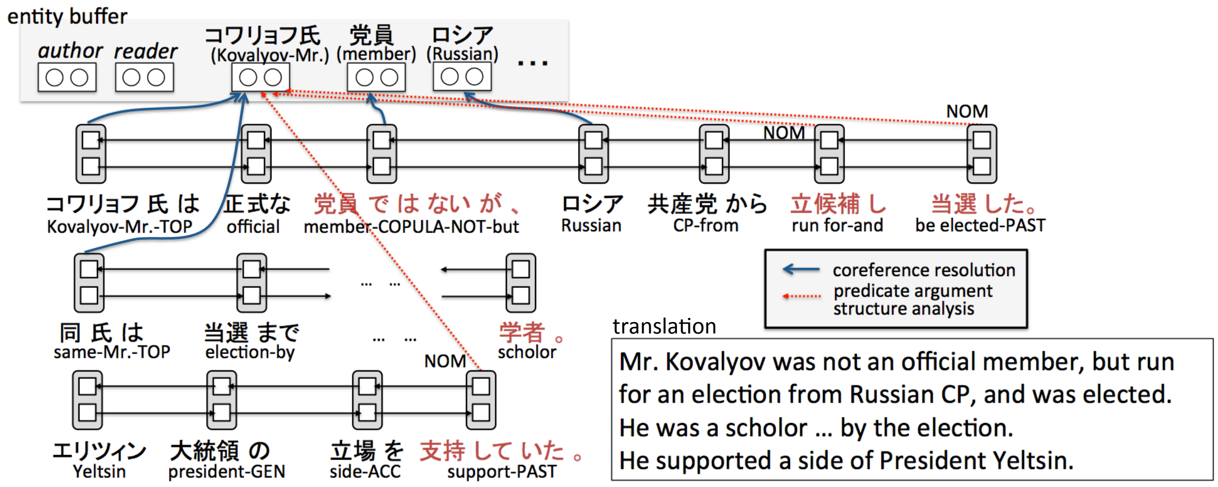

Figure 1: An overview of our proposed method. The phrases with red represent a predicate.

ZAR) needs to identify salient entities, which cannot be identified without performing CR and PA simultaneously. Our results support this claim, and suggest that the status quo of PA-exclusive research in Japanese is an insufficient approach.

Our work is inspired by (Wiseman et al., 2016), which described an English CR system, where entities are represented by embeddings, and they are updated by CR results dynamically. We perform Japanese CR and PA by extending this idea. Our experimental results demonstrate the proposed method can improve the performance of the inter-sentential zero anaphora resolution drastically.

2 Related Work

Predicate Argument Structure Analysis. Early studies have handled both *intra*- and *inter*-sentential anaphora (Taira et al., 2008; Sasano and Kurohashi, 2011), and Hangyo et al. (2013) present a method for handling *exophora*. Recent studies, however, focus on only *intra*-sentential anaphora (Ouchi et al., 2015; Shibata et al., 2016; Iida et al., 2016; Ouchi et al., 2017; Matsubayashi and Inui, 2017), because the analysis of *inter*-sentential anaphora is extremely difficult. Neural network-based approaches (Shibata et al., 2016; Iida et al., 2016; Ouchi et al., 2017; Matsubayashi and Inui, 2017) have improved its performance.

Although most of studies did not consider the notion *entity*, Sasano and Kurohashi (2011) consider an entity, and its salience score is calculated based on simple rules. However, they used gold coreference links to form the entities, and reported the salience score did not improve the performance. In contrast, we perform CR automatically, and capture the entity salience by using RNNs.

For Chinese, where zero anaphors are often used, neural network-based approaches (Chen and Ng, 2016; Yin et al., 2017) outperformed conventional machine learning approaches (Zhao and Ng, 2007).

Coreference Resolution. CR has been actively studied in English and Chinese. Neural network-based approaches (Wiseman et al., 2016; Clark and Manning, 2016b,a; Lee et al., 2017) outperformed conventional machine learning approaches (Clark and Manning, 2015). Wiseman et al. (2016) and Clark and Manning (2016b) learn an entity representation and integrate this into a mention-based model. Our work is inspired by Wiseman et al. (2016), which learn the entity representation by using Recurrent Neural Networks (RNNs). Clark and Manning (2016b) adopt a clustering approach for the entity representation. The reason why we do not use this is that if we take a clustering approach in our setting, zero pronouns need to be first identified before clustering, and thus, it is hard to perform CR and PA jointly. Lee et al. (2017) take an end-to-end approach, aiming at not relying on hand-engineering mention detector (consider all spans as potential mentions). In used Japanese evaluation corpora, since the basic unit for the annotations and our analyses (CR and PA) is fixed, we do not need consider all spans.

In Japanese, CR has not been actively studied other than Iida et al. (2003); Sasano et al. (2007)

since the use of zero pronouns is more common and problematic.

Semantic Role Labeling. Japanese PA is similar to Semantic Role Labeling (SRL) in English. Neural network-based approaches have improved the performance (Zhou and Xu, 2015; He et al., 2017). In these approaches, an appropriate argument for a predicate is searched among mentions in a text. The notion *entity* is not considered.

Other Entity-Centric Study. There are several studies that consider the notion *entity* in other areas: text comprehension (Kobayashi et al., 2016; Henaff et al., 2016) and language modeling (Ji et al., 2017).

3 Japanese Preliminaries

Before presenting our proposed method, we describe the basics of Japanese predicate argument structure and its analysis.

Since the word order is relatively free among arguments in Japanese, an argument is followed by a case marking postposition. The postpositions が (*ga*), を (*wo*), and に (*ni*) indicate nominative (NOM), accusative (ACC) and dative (DAT), respectively. In the double nominative construction such as "私が英語が上手だ" (My English is good), "英語" (English) is regarded as NOM, and "私" (I), the outer nominative is regarded as NOM2. This paper targets these four cases.

PA is tightly related to a dependency structure of a sentence. Considering the relation between a predicate and its argument, and a necessary analysis can be classified into the following three categories (see example sentence (2) below).

(2) ジョン は 買った パン を 食べた
John-TOP bought bread–ACC ate.

Overt case: When an argument with a case marking postposition has a dependency relation with a predicate, PA is not necessary. In example (2), since "パン を" (bread-ACC) has a dependency relation with "食べた" (ate), it is obvious that "食べた" takes "パン" as its ACC argument.

Case analysis: When a topic marker は (*wa*) is attached to an argument, the case marking postposition disappears, and the analysis of identifying the case role becomes necessary. The analysis is called *case analysis*. In the example, although "ジョン は" (John-TOP) has a dependency relation with "食べた" (ate), the analysis of identifying NOM is necessary. The same phenomenon happens when a relative clause is used. When an argument is modified by a relative clause, we do not know its case role to the predicate in the relative clause. In the example, although "パン" has a dependency relation with "買った" (bought), the analysis of identifying ACC is necessary.

Zero anaphora resolution (ZAR): Some arguments are not included in the phrases with which a predicate has a dependency relation. While pronouns are mostly used in English, they are rarely used in Japanese. This phenomenon is called *zero anaphora*, and the analysis of identifying an argument (referent of the zero pronoun) is called *zero anaphora resolution* (ZAR). In the example, although "買った" takes "ジョン" as its NOM argument, they do not have a dependency relation, and thus zero anaphora resolution is necessary.

When dependency relations are identified by parsing, what Japanese PA has to do is case analysis and zero anaphora resolution.

Each predicate has a set of required cases, but not all the four cases. For example, "買う" (buy) takes NOM and ACC, but neither DAT nor NOM2. PA for "買う" in sentence (2) has to find John as NOM, but also has to judge that it does not take DAT and NOM2 arguments.

Another difficulty lies in that a predicate takes a case, but in a sentence it does not take a specific argument. For example, in the sentence "it is difficult to bake a bread", NOM of "bake" is not a specific person, but means "anyone" or "in general". In such cases, PA has to regard arguments as *unspecified*.

4 Overview of Our Proposed Method

An overview of our proposed model is described with a motivated example (Figure 1). Our model equips an *entity buffer* for entity management. At first, it contains only special entities, *author* and *reader*.

In Japanese CR and PA, a basic phrase, which consists of one content word and zero or more function words, is adopted as a basic unit. When an input text is given, the contextual representations of basic phrases are obtained by using Convolutional Neural Network (CNN) and Bidirectional LSTM. Then, from the beginning of the text, CR is performed if a target phrase is a noun phrase, and PA is performed if a target phrase is a predicate phrase. Both of these analyses take into consideration not only the mentions in the text but also the entities in the entity buffer.

In CR, when a mention refers to an existing entity, the entity embedding in the entity buffer is updated. In Figure 1, "同氏" (said person) is analyzed to refer to "コワリョフ氏" (Mr.Kovalyov), and the entity embedding of "コワリョフ氏" is updated. When a mention is analyzed to have no antecedent, it is registered to the entity buffer as a new entity.

In PA, when a predicate has no argument for any case, its argument is searched among any mentions in the text, *author* and *reader*. In the same way as CR, PA takes into consideration not only the mentions but also entities in the entity buffer, and updates the entity embedding.

In Figure 1, the predicate "立候補し" (run for) has no NOM argument. Our method finds "コワリョフ氏" as its NOM argument, and then updates its entity embedding. As mentioned before, the entity embedding of "コワリョフ氏" is updated by the coreference relation with "同氏" in the second sentence. In the third sentence, the predicate "支持していた" (support) has also no NOM argument, and "コワリョフ氏" is identified as its NOM argument, because the frequent reference implies its salience.

5 Base Model

5.1 Input Encoding

Conventional machine learning techniques have extracted features from a basic phrase, which require much effort on feature engineering. Our method obtains an embedding of each basic phrase using CNN and bi-LSTM as shown in Figure 2.

Suppose the i-th basic phrase bp_i consists of $|bp_i|$ words. First, the embedding of each word is represented as a concatenation of word (lemma), part of speech (POS), sub-POS and conjugation embeddings. We append start-of-phrase and end-of-phrase special words to each phrase in order to better represent prefixes and suffixes. Let $W^i \in \mathbb{R}^{d\times(|bp_i|+2)}$ be an embedding matrix for bp_i where d denotes the dimension of word representation.

The embedding of the basic phrase is obtained by applying CNN to the sequence of words. A feature map $\boldsymbol{f}^i$ is obtained by applying a convolution between W^i and a filter H of width n. The m-th element of $\boldsymbol{f}^i$ is obtained as follows:

$$\boldsymbol{f}^i[m] = \tanh(\langle W^i[*, m : m+n-1], H\rangle), \qquad (1)$$

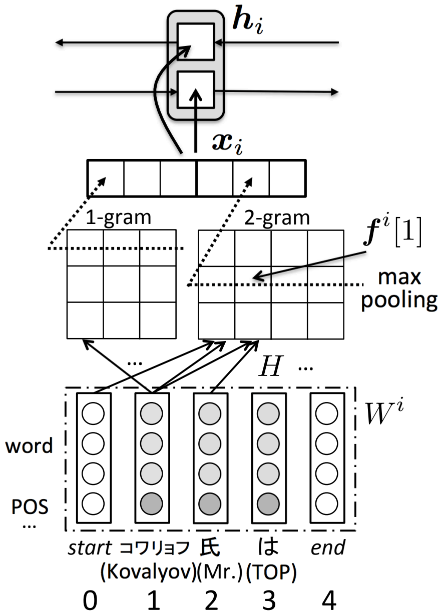

Figure 2: Basic phrase embedding obtained with CNN and Bi-LSTM.

where $W^i[*, m : m+n-1]$ denotes the m-to-($m+n-1$)-th column of W^i, and $\langle A, B\rangle = \mathrm{Tr}(AB^{\mathrm{T}})$ is the Frobenius inner product. Then, to capture the most important feature for a given filter in bp_i, the max pooling is applied as follows:

$$x^i = \max_m \boldsymbol{f}^i[m]. \qquad (2)$$

The process described so far is for one filter. The multiple filters of varying widths are applied to obtain the representation of bp_i. When we set h filters, $\boldsymbol{x}_i$, the embedding of the i-th basic phrase, is represented as $[x^i_1, \cdots, x^i_h]$.

The embeddings of basic phrases are read by bi-LSTM to capture their context as follows:

$$\begin{aligned}\overrightarrow{\boldsymbol{h}}_i &= \overrightarrow{LSTM}(\boldsymbol{x}_i, \overrightarrow{\boldsymbol{h}}_{i-1}),\\ \overleftarrow{\boldsymbol{h}}_i &= \overleftarrow{LSTM}(\boldsymbol{x}_i, \overleftarrow{\boldsymbol{h}}_{i+1}),\end{aligned} \qquad (3)$$

and the contextualized embedding of the i-th basic phrase is represented as a concatenation of the hidden layers of forward and backward LSTM.

$$\boldsymbol{h}_i = [\overrightarrow{\boldsymbol{h}}_i; \overleftarrow{\boldsymbol{h}}_i] \qquad (4)$$

This process is performed for each sentence. Since CR and PA are performed for a whole document D, the indices of basic phrases are reassigned from the beginning to the end of D in a consecutive order: $D = \{\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_i, \cdots\}$.

To handle exophora, *author* and *reader* are assigned a unique trainable embedding, respectively.

5.2 Coreference Resolution

We adopt a mention-ranking model that assigns each mention its highest scoring candidate antecedent. This model assigns a score $s^m_{CR}(ant, m_i)$ to a target mention m_i and its candidate antecedent ant[1]. The candidate antecedents include i) mentions preceding m_i, ii) *author* and *reader*, and iii) $\texttt{NA}_{\texttt{CR}}$ (no antecedent). $s^m_{CR}(ant, m_i)$ is calculated as follows:

$$s^m_{CR}(ant, m_i) = W_2^{CR} ReLU(W_1^{CR} \boldsymbol{v}^{CR}_{input}), \quad (5)$$

where W_1^{CR} and W_2^{CR} are weight matrices, and $\boldsymbol{v}^{CR}_{input}$ is an input vector, a concatenation of the following vectors:

- embeddings of m_i and ant
- exact match or partial match between strings of m_i and ant
- sentence distance between m_i and ant. The distance is binned into one of the buckets [0, 1, 2, 3+].
- whether a pair of m_i and ant has an entry in a synonym dictionary.

When a candidate antecedent is $\texttt{NA}_{\texttt{CR}}$, the input vector is just the embedding of a target mention m_i, and the same neural network with different weight matrices calculates a score.

The following margin objective is trained:

$$\mathcal{L}_{CR} = \sum_i^{N_m} \max_{ant \in \mathcal{ANT}(m_i)} (1 + s^m_{CR}(ant, m_i) - s^m_{CR}(\hat{t}_i, m_i)), \quad (6)$$

where N_m denotes the number of mentions in a document, $\mathcal{ANT}(m_i)$ denotes the set of candidate antecedents of m_i, and $\hat{t}_i$ denotes the highest scoring true antecedent of m_i defined as follows:

$$\hat{t}_i = \mathop{\mathrm{argmax}}_{ant \in \mathcal{T}(m_i)} s^m_{CR}(ant, m_i), \quad (7)$$

where $\mathcal{T}(m_i)$ denotes the set of true antecedents of m_i.

5.3 Predicate Argument Structure Analysis

When a target phrase is a predicate phrase, PA is performed. For each case of a predicate, PA searches an appropriate argument among candidate arguments: i) basic phrases located in the sentence including the predicate and preceding sentences, ii) *author* and *reader*, iii) unspecified, and iv) $\texttt{NA}_{\texttt{PA}}$ which means the predicate takes no argument of for the case.

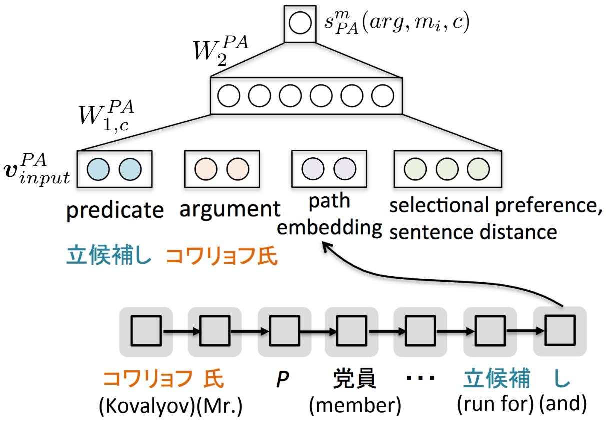

Figure 3: A neural network for PA.

The probability that the predicate m_i takes an argument arg for case c is defined as follows:

$$P(c = arg|m_i) = \frac{\exp(s^m_{PA}(arg, m_i, c))}{\sum\limits_{\substack{carg \in \\ \mathcal{ARG}(m_i)}} \exp(s^m_{PA}(carg, m_i, c))}, \quad (8)$$

where $\mathcal{ARG}(m_i)$ denotes the set of candidate arguments of m_i, and a score $s^m_{PA}(arg, m_i, c)$ is calculated by a neural network as follows (Figure 3):

$$s^m_{PA}(arg, m_i, c) = W_2^{PA} \tanh(W_{1,c}^{PA} \boldsymbol{v}^{PA}_{input}), \quad (9)$$

where $W_{1,c}^{PA}$, W_2^{PA} are weight matrices, and $\boldsymbol{v}^{PA}_{input}$ is an input vector, a concatenation of the following vectors:

- embeddings of m_i and arg[2]
- path embedding: the dependency path between a predicate and an argument is an important clue. Roth and Lapata (2016) learn a representation of a lexicalized dependency path for SRL. An LSTM reads words[3] from an argument to a predicate along with a dependency path, and the final hidden state is adopted as the embedding of the dependency path.[4] For case analysis, the direct dependency relation between a predicate and its argument can be represented as the path embedding.

[1] The superscript m of $s^m_{CR}(ant, m_i)$ represents a *mention*-based score, which contrasts with an *entity*-based score introduced in Section 6.

[2] An embedding for $\texttt{NA}_{\texttt{PA}}$ is assigned a trainable one.

[3] We add special words {Parent, Child}, which indicate a dependency direction between basic phrases.

[4] When an argument is an *inter* or *exophora*, the path embedding is set to be a zero vector.

- selectional preference: selectional preference is another important clue for PA. A selectional preference score is learned in an unsupervised manner from automatic parses of a raw corpus (Shibata et al., 2016).
- sentence distance between m_i and arg. The distance is binned in the same way as CR.

The objective is to minimize the cross entropy between predicted and true distributions:

$$\mathcal{L}_{PA} = -\sum_{i}^{N_p} \sum_{c} \log P(c = \widehat{arg} | p_i), \quad (10)$$

where N_p denotes the number of predicates in a document, and $\widehat{arg}$ denotes a true argument.

6 Entity-Centric Model

While the base model performs *mention*-based CR and PA, our proposed model performs *entity*-based analyses as shown in Figure 1.

6.1 Entity Embedding Update

The entity embeddings are managed in an entity buffer. First, let us introduce time stamp i for the entity embedding update. Time i corresponds to the analysis for the i-th basic phrase in a document. If an entity is referred to by the analysis, its embedding is updated. Let $\boldsymbol{e}_i^{(k)}$ be the embedding of an entity k at time i (after the entity embedding is updated).

In CR, following Wiseman et al. (2016), when a target phrase m_i refers to the entity k, $\boldsymbol{e}_i^{(k)}$ is updated as follows:

$$\boldsymbol{e}_i^{(k)} \leftarrow LSTM_e(\boldsymbol{h}_i, \boldsymbol{e}_{i-1}^{(k)}) \quad (11)$$

where $LSTM_e$ denotes an LSTM for the entity embedding update. When an antecedent is $\texttt{NA}_{\text{CR}}$, a new entity embedding is set up, initialized by a zero vector. The entity buffer maintains K LSTMs (K is the number of entities in a document), and their parameters are shared.

The proposed method updates the entity embedding not only in CR but also in PA. When the referent of a zero pronoun of case c of predicate p_i is entity k, the entity embedding is updated by using the predicate embedding $\boldsymbol{h}_i$ multiplied by a weight matrix W_c for case c as follows:

$$\boldsymbol{e}_i^{(k)} \leftarrow LSTM_e(W_c \boldsymbol{h}_i, \boldsymbol{e}_{i-1}^{(k)}). \quad (12)$$

In both CR and PA, the embeddings of entities other than the referred entity k are not updated ($\boldsymbol{e}_i^{(l)} \leftarrow \boldsymbol{e}_{i-1}^{(l)} (l \neq k)$).

6.2 Use of Entity Embedding in CR and PA

Both CR and PA are allowed to take the entity embeddings into consideration. In CR, let z_{ant} denote the id of an entity to which the candidate antecedent ant belongs. The *entity*-based score s_{CR}^e is calculated as follows:

$$s_{CR}^e(ant, m_i) = \begin{cases} \boldsymbol{h}_i^{\mathrm{T}} \boldsymbol{e}_{i-1}^{(z_{ant})} & (ant \neq \texttt{NA}_{\text{CR}}) \\ g_{NA}(m_i) & (ant = \texttt{NA}_{\text{CR}}). \end{cases} \quad (13)$$

The intuition behind the first case is that the dot-product of $\boldsymbol{h}_i$, the embedding of the target mention, and $\boldsymbol{e}_{i-1}^{(z_{ant})}$, the embedding of the entity that ant belongs to indicates the plausibility of their coreference. $g_{NA}(m_i)$ is defined as follows:

$$g_{NA}(m_i) = \boldsymbol{q}^{\mathrm{T}} \tanh(W_{NA} \left[\begin{smallmatrix} \boldsymbol{h}_i \\ \sum_k \boldsymbol{e}_{i-1}{}^{(k)} \end{smallmatrix} \right]), \quad (14)$$

where $\boldsymbol{q}$ is a weight vector, and W_{NA} is a weight matrix. The intuition is that whether a target phrase is $\texttt{NA}_{\text{CR}}$ can be judged from $\boldsymbol{h}_i$, the embedding of the target mention itself, and the sum of all the current entity embeddings. s_{CR}^e is added to s_{CR}^m, and the training objective is the same as the one described in Section 5.2.

In PA, the entity embedding corresponding to a candidate argument arg[5] is just added to the input vector $\boldsymbol{v}_{input}^{PA}$ described in Section 5.3, and *mention*- and *entity*-based score $s_{PA}^{m+e}(arg, m_i, c)$ is calculated in the same way as $s_{PA}^m(arg, m_i, c)$. The training objective is again the same as the one in Section 5.3.

In Wiseman et al. (2016), the oracle entity assignment is used for the entity embedding update in training, and the system output is used in a greedy manner in testing. Since the performance of PA is lower than that of English CR, there might be a more significant gap between training and testing. Therefore, scheduled sampling (Bengio et al., 2015) is adopted to bridge the gap: in training, the oracle entity assignment is used with probability ϵ_t (at the t-th iteration) and the system output otherwise. Exponential decay is used: $\epsilon_t = k^t$ (we set $k = 0.75$ for our experiments).

7 Experiments

7.1 Experimental Setting

The two kinds of evaluation sets were used for our experiments. One is the KWDLC (Kyoto Uni-

[5] When arg is $\texttt{NA}_{\text{PA}}$, the entity embedding is set to a zero vector.

versity Web Document Leads Corpus) evaluation set (Hangyo et al., 2012), and the other is Kyoto Corpus. KWDLC consists of the first three sentences of 5,000 Web documents (15,000 sentences) and Kyoto Corpus consists of 550 News documents (5,000 sentences). Word segmentations, POSs, dependencies, PASs, and coreferences were manually annotated (the closest referents and antecedents were annotated for zero anaphora and coreferences, respectively). Since we want to focus on the accuracy of CR and PA, gold segmentations, POSs, and dependencies were used. KWDLC (Web) 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; Kyoto Corpus (News) was divided into 360 documents (3,210 sents.) for training, 98 documents (971 sents.) for development, and 100 documents (967 sents.) for testing.

The evaluation measure is an F-measure, and the evaluation of both CR and PA was relaxed using a gold coreference chain, which leads to an entity-based evaluation. We did not use the conventional CR evaluation measures (MUC, B^3, CEAF and CoNLL) because our F-measure is almost the same as MUC, which is a link-based measure, and the other measures considering singletons get excessively high values[6], and thus they do not accord with the actual performance in our setting.[7]

7.2 Implementation Detail

The dimension of word embeddings was set to 100, and the word embeddings were initialized with pre-trained embeddings by Skip-gram with a negative sampling (Mikolov et al., 2013) on a Japanese Web corpus consisting of 100M sentences. The dimension of POS, sub-POS and conjugation were set to 10, respectively, and these embeddings were initialized randomly. The dimensions of the hidden layer in all the neural networks were set to 100. We used filter windows of 1,2,3 with 33 feature maps each for basic phrase CNN. Adam (Kingma and Ba, 2014) was adopted as the optimizer. F measures were averaged over four runs.

Checkpoint ensemble (Chen et al., 2017) was adopted, where the k best models were taken in terms of validation score, and then the parameters from the k models were averaged for testing. This method requires only one training process. In our experiments, k was set to 5, and the maximum number of epochs was set to 10.

We used a single-layer bi-LSTM for the input encoding (Section 5.1); preliminary experiments with stacked stacked bi-directional LSTM with residual connections were not favorable. Although we tried to use the character-level embedding of each word obtained with CNN, as the same way in the basic phrase embedding from the word sequences, the performance was not improved. The synonym dictionary used for CR (Section 5.2) was constructed from an ordinary dictionary and Web corpus, and has about 7,300 entries (Sasano et al., 2007).

7.3 Experimental Result

The following three methods were compared:

- Baseline: the method described in Section 5.
- "+entity (CR)": this method corresponds to (Wiseman et al., 2016). Entity embedding is updated based on the CR result, and CR takes the entity embedding into consideration.
- "+entity (CR,PA)" (**proposed method**): entity embedding is updated based on PA as well as CR result, and the CR and PA take the entity embedding into consideration.

The performance of CR and PA (case analysis and zero anaphora resolution (ZAR)) is shown in Table 1. The performance of CR and case analysis was almost the same for all the methods. For ZAR, "+entity (CR,PA)" improved the performance drastically.

CR surely benefits from the entity salience. Since entity embeddings are updated based on system outputs, its performance matters. The performance of Japanese CR is lower than that of English CR. Therefore, we assume there are improved/worsen examples, and our CR performance did not improve significantly. The performance of ZAR also matters. However, the performance of ZAR in our baseline model is extremely low, and thus there are few worsen examples and

[6] In Japanese, since zero pronouns are often used, there are many singletons. In example sentences (1) of the Introduction section, while "a car" and "It" form one cluster in English sentences (1-a), "a car" is a singleton in Japanese sentences (1-b) because a zero pronoun is used in the second sentence.

[7] For the Web evaluation set, the F-measure of our proposed method is 0.685, and the conventional evaluation measures are as follows; MUC: 69.1, B^3: 97.2, CEAF: 95.7, and CoNLL: 87.3.

	Web			News		
method	coreference resolution	case analysis	zero anaphora resolution (ZAR)	coreference resolution	case analysis	zero anaphora resolution (ZAR)
Baseline	0.661	0.887	0.516	**0.543**	**0.896**	0.278
+entity (CR)	0.666	0.890	0.518	0.539	0.894	0.275
+entity (CR,PA)	**0.685**	**0.892**	**0.581**	0.541	0.895	**0.356**

Table 1: Performance (F-measure) of coreference resolution, case analysis and zero anaphora resolution.

		Web					News				
case	method	case analysis	zero anaphora resolution (ZAR)				case analysis	zero anaphora resolution (ZAR)			
			all	*intra*	*inter*	*exophora*		all	*intra*	*inter*	*exophora*
NOM	Baseline	0.942	0.575	0.466	0.083	0.695	0.939	0.316	0.455	0.042	0.261
	+entity (CR)	**0.945**	0.579	0.475	0.117	0.693	**0.940**	0.315	0.452	0.037	0.239
	+entity (CR,PA)	**0.945**	**0.646**	**0.508**	**0.502**	**0.721**	**0.940**	**0.390**	**0.486**	**0.256**	**0.357**
	# of arguments	(1,461)	(2,009)	(338)	(393)	(1,278)	(905)	(1,016)	(451)	(388)	(177)
ACC	Baseline	0.853	0.268	0.368	0.119	0.000	**0.679**	**0.053**	**0.093**	0.000	0.000
	+entity (CR)	0.855	0.254	0.357	0.108	0.000	0.631	0.025	0.048	0.000	0.000
	+entity (CR,PA)	**0.857**	**0.343**	**0.413**	**0.282**	0.000	0.651	0.016	0.028	0.000	0.000
	# of arguments	(299)	(224)	(106)	(105)	(13)	(105)	(97)	(41)	(56)	(0)
DAT	Baseline	**0.498**	0.432	0.115	0.016	0.581	**0.308**	0.183	**0.039**	0.000	0.367
	+entity (CR)	0.445	0.422	0.119	0.016	0.574	0.223	0.162	0.005	0.000	0.334
	+entity (CR,PA)	0.411	**0.465**	**0.133**	**0.126**	**0.600**	0.292	**0.328**	0.030	**0.005**	**0.566**
	# of arguments	(101)	(576)	(86)	(149)	(341)	(26)	(286)	(82)	(89)	(115)
NOM2	Baseline	0.478	0.216	**0.259**	0.000	0.245	**0.098**	0.000	0.000	0.000	0.000
	+entity (CR)	0.501	0.212	0.226	0.000	0.257	0.069	0.000	0.000	0.000	0.000
	+entity (CR,PA)	**0.526**	**0.283**	0.240	**0.112**	**0.341**	0.092	0.000	0.000	0.000	0.000
	# of arguments	(110)	(140)	(29)	(28)	(83)	(13)	(37)	(17)	(13)	(7)
all	Baseline	0.887	0.516	0.400	0.074	0.654	**0.896**	0.278	0.394	0.032	0.291
	+entity (CR)	0.890	0.518	0.405	0.093	0.654	0.894	0.275	0.396	0.027	0.265
	+entity (CR,PA)	**0.892**	**0.581**	**0.439**	**0.399**	**0.681**	0.895	**0.356**	**0.417**	**0.204**	**0.432**
	# of arguments	(1,971)	(2,949)	(559)	(675)	(1,715)	(1,049)	(1,436)	(591)	(546)	(299)

Table 2: Performance of case analysis and zero anaphora resolution for each case, and each argument position for zero anaphora resolution. The underlined values indicate the proposed method outperforms the baseline by a large margin.

a number of improved examples. Therefore, ZAR can benefit from the entity representation obtained by both CR and PA.

Table 2 shows performance of case analysis and zero anaphora resolution for each case, and each argument position. *Unspecified* was counted for *exophora*. Both for the News and Web evaluation sets, the performance for *inter* arguments of zero anaphora resolution, which was extremely difficult in the baseline method, was improved by a large margin by our proposed method.

7.4 Ablation Study

To reveal the importance of each clue for CR and PA, each clue was ablated. Table 3 shows the result on the development set. We found that, the path embedding was effective for PA, and the string match was effective for CR. The sentence distance for both CR and PA was effective for News, but not for Web since the Web evaluation corpus consists of three-sentence documents.

7.5 Comparison with Other Work

It is difficult to compare the performance of our method with other studies directly because there are no studies handling both CR and PA. The comparisons with other studies are summarized as follows:

- Shibata et al. (2016) proposed a neural-network based PA. Their target was *intra* and *exophora* for three major cases (NOM, ACC and DAT), and the performance was 0.534 on the same Web corpus as ours. The performance of our proposed method for the same three cases was 0.626. Furthermore, since their model assumes a static PA graph, their model is difficult to be extended to handle CR.
- Ouchi et al. (2017) proposed a grid-type RNN model for capturing the multi-predicate interaction. Their target was only *intra* on the NAIST text corpus (News), and the performance was 47.1. Since the NAIST text

	coreference resolution				zero anaphora resolution (ZAR)			
	Web		News		Web		News	
	F1	Δ	F1	Δ	F1	Δ	F1	Δ
Our proposed model	0.633		0.613		0.512		0.361	
CR								
- string match	0.212	-0.420	0.184	-0.429	0.474	-0.038	0.348	-0.013
- sentence distance	0.643	+0.011	0.588	-0.025	0.505	-0.007	0.343	-0.018
- synonym dictionary	0.643	+0.010	0.613	0.000	0.510	-0.002	0.348	-0.013
PA								
- path embedding	0.643	+0.010	0.625	+0.012	0.459	-0.054	0.268	-0.093
- selectional preference	0.638	+0.005	0.316	-0.297	0.507	-0.005	0.173	-0.188
- sentence distance	0.647	+0.014	0.606	-0.007	0.516	+0.004	0.327	-0.034

Table 3: Ablation study on the development set. The cells shaded gray represent they are not directly affected from the ablation, but from the counterpart analysis result.

corpus contains a lot of annotation errors as pointed out in Iida et al. (2016), we did not conduct our experiments on the NAIST text corpus.

- Iida et al. (2003) reported an F-measure of about 0.7 on News domain. The possible reason why our performance on News (0.541) is lower than theirs is that their basic unit is a compound noun while our basic unit is a noun, and thus our setting is difficult in comparison with theirs.

Since we handle *inter* as well as *intra* and *exophora* arguments in PA, together with CR, we can say that our experimental setting is more practical in comparison with other studies.

7.6 Error Analysis

In example (3), although the NOM argument of the predicate "通院ですよ!" (go to hospital) is *author*, our method wrongly classified it as *unspecified.*

(3) 毎日のように 通院ですよ！ 私自身は
every day go to hospital! I myself-TOP
とても 健康なんですけど。
very healthy.
((I) go to hospital every day!
(I am) very healthy, though.)

In the second sentence, our method correctly identified the antecedent of "私" (I) as *author*, and the NOM of "健康なんですけど" (healthy) as "私" (I). Our method adopts the greedy search so that it cannot exploit this handy information in the analysis of the first sentence. The global modeling using reinforcement learning (Clark and Manning, 2016a) for a whole document is our future work.

In example (4), although the NOM argument of "装飾されています" (be decorated) in the second sentence is "ドレス" (dress) in the first sentence, our method wrongly classified it as $\mathtt{NA}_{\mathtt{PA}}$.

(4) 大変 印象的な ドレスです。
very impressive dress-COPULA.
オーガンジーの 上に ラインを 描くように
organdie-GEN top-DAT line-ACC draw-as
小さな ビーズで 装飾されています。
small bead-INS decorated
((This is) a very impressive dress.
(The dress) is decorated by small beads as they draw a line on its organdy.)

"オーガンジー" (organdie) has a bridging relation to "ドレス", which might help capture the salience of "ドレス". The bridging reference resolution is our next target and must be easily incorporated into our model.

8 Conclusion

This paper has presented an entity-centric neural network-based joint model of coreference resolution and predicate argument structure analysis. Each entity has its embedding, and the embeddings are updated according to the result of both of these analyses dynamically. Both of these analyses took the entity embedding into consideration to access the global information of entities. The experimental results demonstrated that the proposed method could improve the performance of the inter-sentential zero anaphora resolution drastically, which has been regarded as a notoriously difficult task. We believe that our proposed method is also effective for other pro-drop languages such as Chinese and Korean.

Acknowledgment

This work was supported by JST CREST Grant Number JPMJCR1301, Japan.

References

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. *CoRR* abs/1506.03099. `http://arxiv.org/abs/1506.03099`.

Chen Chen and Vincent Ng. 2016. Chinese zero pronoun resolution with deep neural networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 778–788. `http://www.aclweb.org/anthology/P16-1074`.

Hugh Chen, Scott Lundberg, and Su-In Lee. 2017. Checkpoint ensembles: Ensemble methods from a single training process. *CoRR* abs/1710.03282. `http://arxiv.org/abs/1710.03282`.

Kevin Clark and Christopher D. Manning. 2015. Entity-centric coreference resolution with model stacking. In *Association for Computational Linguistics (ACL)*.

Kevin Clark and Christopher D. Manning. 2016a. Deep reinforcement learning for mention-ranking coreference models. In *Empirical Methods on Natural Language Processing (EMNLP)*.

Kevin Clark and Christopher D. Manning. 2016b. Improving coreference resolution by learning entity-level distributed representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, pages 643–653. `https://doi.org/10.18653/v1/P16-1061`.

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*. Faculty of Computer Science, Universitas Indonesia, Bali,Indonesia, pages 535–544. `http://www.aclweb.org/anthology/Y12-1058`.

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*. Association for Computational Linguistics, Seattle, Washington, USA, pages 924–934. `http://www.aclweb.org/anthology/D13-1095`.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and what's next. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Mikael Henaff, Jason Weston, Arthur Szlam, Antoine Bordes, and Yann LeCun. 2016. Tracking the world state with recurrent entity networks. *CoRR* abs/1612.03969. `http://arxiv.org/abs/1612.03969`.

Ryu Iida, Kentaro Inui, Hiroya Takamura, and Yuji Matsumoto. 2003. Incorporating contextual cues in trainable models for coreference resolution. In *In Proceedings of the EACL Workshop on The Computational Treatment of Anaphora*. pages 23–30.

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 *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 1244–1254. `https://aclweb.org/anthology/D16-1132`.

Yangfeng Ji, Chenhao Tan, Sebastian Martschat, Yejin Choi, and Noah A. Smith. 2017. Dynamic entity representations in neural language models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 1831–1840. `http://www.aclweb.org/anthology/D17-1195`.

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

Sosuke Kobayashi, Ran Tian, Naoaki Okazaki, and Kentaro Inui. 2016. Dynamic entity representation with max-pooling improves machine reading. In *Proceedings of the NAACL HLT 2016*.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 188–197. `http://www.aclweb.org/anthology/D17-1018`.

Yuichiroh Matsubayashi and Kentaro Inui. 2017. Revisiting the design issues of local models for japanese predicate-argument structure analysis. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Asian Federation of Natural Language Processing, Taipei, Taiwan, pages 128–133. `http://www.aclweb.org/anthology/I17-2022`.

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*, Curran Associates, Inc., pages 3111–3119.

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)*. Association for Computational Linguistics, Beijing, China, pages 961–970. `http://www.aclweb.org/anthology/P15-1093`.

Hiroki Ouchi, Hiroyuki Shindo, and Yuji Matsumoto. 2017. Neural modeling of multi-predicate interactions for Japanese predicate argument structure analysis. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, pages 1591–1600. `http://aclweb.org/anthology/P17-1146`.

Michael Roth and Mirella Lapata. 2016. Neural semantic role labeling with dependency path embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 1192–1202. `http://www.aclweb.org/anthology/P16-1113`.

Ryohei Sasano, Daisuke Kawahara, and Sadao Kurohashi. 2007. Improving coreference resolution using bridging reference resolution and automatically acquired synonyms. In *DAARC*.

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*. Asian Federation of Natural Language Processing, Chiang Mai, Thailand, pages 758–766. `http://www.aclweb.org/anthology/I11-1085`.

Tomohide Shibata, Daisuke Kawahara, and Sadao Kurohashi. 2016. Neural network-based model for Japanese predicate argument structure analysis. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 1235–1244. `http://www.aclweb.org/anthology/P16-1117`.

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*. Association for Computational Linguistics, Honolulu, Hawaii, pages 523–532. `http://www.aclweb.org/anthology/D08-1055`.

Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. 2016. Learning global features for coreference resolution. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, pages 994–1004. `http://www.aclweb.org/anthology/N16-1114`.

Qingyu Yin, Yu Zhang, Weinan Zhang, and Ting Liu. 2017. Chinese zero pronoun resolution with deep memory network. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, pages 1320–1329. `https://www.aclweb.org/anthology/D17-1136`.

Shanheng Zhao and Hwee Tou Ng. 2007. Identification and resolution of Chinese zero pronouns: A machine learning approach. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Association for Computational Linguistics, Prague, Czech Republic, pages 541–550. `http://www.aclweb.org/anthology/D/D07/D07-1057`.

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)*. Association for Computational Linguistics, Beijing, China, pages 1127–1137. `http://www.aclweb.org/anthology/P15-1109`.