Neural Japanese Zero Anaphora Resolution using Smoothed Large-scale Case Frames with Word Embedding

Souta Yamashiro **Hitoshi Nishikawa** **Takenobu Tokunaga**
Tokyo Institute of Technology / W8-73, 2-12-1 Ookayama, Meguro, Tokyo 152-8552, Japan
{yamashiro.s.aa@m, {hitoshi, take}@c}.titech.ac.jp

Abstract

This paper presents a Japanese zero anaphora resolution model which deals with both intra- and inter-sentential zero anaphora. Solving inter-sentential anaphora needs to consider a large number of antecedent candidates beyond the sentence boundaries, which is a crucial obstacle for training the model and resolving the anaphora. To cope with this problem, we propose an effective candidate pruning method using case frame information. Also, we introduce a local single-attention RNN for inter-sentential anaphora resolution, allowing the model to consider the distant context from the target predicate. We evaluated the proposed models with a Japanese balanced corpus and confirmed the effectiveness of the candidate pruning by showing 0.056 point increase of accuracy.

1 Introduction

Zero anaphora resolution is the task to detect omitted arguments (zero anaphors) of the predicate in a given text and to identify their antecedents. The antecedents might or might not appear in the text. In the latter case (exophora), the antecedents exist outside of the text, e.g. the writer of the text. In the former case, they appear within the same sentence as the predicate (intra-sentential anaphora) or appear in the preceding sentences (inter-sentential anaphora)[1].

(1) *Tikaku no syôtengai-ni*$_{\text{DAT}}$ (*watasi-ga*$_{\text{NOM}}$) *osyarena tatemono-wo*$_{\text{ACC}}$ *mikakeru*$_{v_1}$ *yôninatta.*
Kafeteria-ga$_{\text{NOM}}$ *tokuni ôku, kongetu mo* (*kafeteria-ga*$_{\text{NOM}}$) (*Tikaku no syôtengai-ni*$_{\text{DAT}}$) *ôpunsiteiru*$_{v_2}$.

(I_{NOM}) see$_{v_1}$ fashionable buildings$_{\text{ACC}}$ in the nearby shopping district$_{\text{DAT}}$ recently. There are many cafeterias$_{\text{NOM}}$ in particular, and (a cafeteria$_{\text{NOM}}$) has opened$_{v_2}$ this month (in the nearby shopping disctict$_{\text{DAT}}$).

In the example (1), the nominative argument of v_1 (see) and the nominative argument and dative argument of v_2 (open), which are enclosed by the parentheses, are omitted from the sentences. The nominative argument of "open" is "cafeteria" which appears in the same sentence (intra-sentential zero anaphora) and the dative argument is "the nearby shopping district" which appears in the previous sentence (inter-sentential zero anaphora). On the other hand, the nominative argument of v_1 (see) is the writer of this text who is not explicitly mentioned in the text (exophora).

This study focuses on zero anaphora resolution of Japanese texts, but we observe such pronoun-dropping phenomenon in other languages as well, e.g. Chinese, Italian, Turkish and so on. There have been many studies on the task similar to the Japanese zero anaphora resolution in other languages (Iida and Poesio, 2011; Rello et al., 2012; Chen and Ng, 2016; Yin et al., 2017).

The zero anaphora resolution is one of the active research areas in the Japanese language processing as it is crucial for improving the performance of various natural language processing applications such as automatic text summarisation (Yamada et al., 2017), information extraction (Sudo et al., 2001) and machine translation (Kudo et al.,

[1] We do not deal with cataphora in this study.

32nd Pacific Asia Conference on Language, Information and Computation
Hong Kong, 1-3 December 2018
Copyright 2018 by the authors

2014). Therefore it has been extensively studied as an urgent problem to be solved (Sasano and Kurohashi, 2011; Hangyo et al., 2013; Ouchi et al., 2017; Matsubayashi and Inui, 2017). Our contribution in this study is twofold: proposing a method for both intra- and inter-sentential zero anaphora of the Japanese language and evaluating the method with a large-scale balanced corpus.

The past research evaluated their system with NAIST Text Corpus (NTC) (Iida et al., 2007) that consists of newspaper articles; therefore the evaluation is skewed regarding text genres. When considering real applications, we need a zero-anaphora resolution method that is robust against the difference in text genres. We use Balanced Corpus of Contemporary Written Japanese (BCCWJ) (Maekawa et al., 2014) for evaluation. BCCWJ consists of about 100 million words that were systematically sampled from several sources such as newspaper articles, novels, magazines, white papers, QA texts on the internet and blog texts. We use the core data set of BCCWJ consisting of about two million words that are annotated with co-reference relations and predicate-argument relations for nominative, dative and accusative cases.

Most past research on the Japanese zero anaphora resolution (Iida et al., 2016; Shibata et al., 2016; Ouchi et al., 2017; Matsubayashi and Inui, 2017) has targeted only intra-sentential anaphora. As a reason to focus on intra-sentential zero anaphora, Ouchi et al. (2017) pointed out a search space problem. The system needs to consider antecedent candidates in the entire text for the inter-sentential anaphora. It makes the search space larger than that for the intra-sentential anaphora. Matsubayashi and Inui (2017) introduced a recurrent neural network (RNN) for intra-sentential zero anaphora resolution which takes an entire sentence as an input. However, if we apply the same method to inter-sentential zero anaphora resolution, we need to input the entire text to the RNN, which makes the RNN training impractical.

However, we can observe quite a lot of inter-sentential anaphora in real texts. Table 1 shows the distribution of the distance between a predicate and its arguments for each case in the BCCWJ core data set. The distance is measured by the number of sentences between the predicate and its argument. The distance zero means intra-sentential anaphora, and the distance more than zero means inter-sentential zero anaphora. We can see that more than the half of zero anaphora are inter-sentential anaphora. Table 2 shows the distribution of the predicate-argument distance for the nominative case across different text genres. We can see the difference in the distribution of intra- and inter-sentential anaphora across the genres. This observation supports the importance of evaluation with different types of texts.

distance	NOM	ACC	DAT	total	%
0	16,621	4,545	2,059	23,225	50.4
1	8,231	1,764	1,113	11,108	24.1
2	3,396	599	430	4,425	9.6
3	1,792	317	227	2,336	5.1
4	1,020	172	126	1,318	2.9
5	690	83	84	857	1.9
6	414	45	51	510	1.1
≥ 7	1,917	217	178	2,312	5.0
total	34,081	7,742	4,268	46,091	

Table 1: Distribution of the argument-verb distance

distance	OW	PB	PN	PM	OC	OY
0	72.3	49.5	51.1	40.3	38.8	49.8
1	15.1	25.0	24.4	23.9	29.1	23.1
2	5.6	9.8	9.6	11.2	13.4	8.7
3	2.4	5.0	4.8	6.7	6.9	4.5
4	0.9	2.7	2.6	4.3	3.9	3.5
5	0.9	1.3	2.3	2.7	2.9	1.6
6	0.4	1.1	1.0	1.7	1.4	1.7
≥ 7	2.5	5.5	4.4	9.3	3.6	7.2

OW: white papers, PB: books, PN: newspapers, PM: magazines, OC: QA tests, OY: Blog texts

Table 2: Distribution of nominative zero anaphora across text genres (%)

Unlike the above studies, Sasano and Kurohashi (2011) and Hangyo et al. (2013) proposed a zero anaphora resolution method for both intra- and inter-sentential anaphora. However, they evaluated their method by using only Web text corpora.

To address the above two issues, we introduce a method to reduce the number of antecedent candidates by using case frame information and evaluate the proposed method by using a large-scale balanced corpus, i.e. BCCWJ. When we deal with intra- and inter-sentential anaphora in a single model, we need to cope with a large search space for the an-

Copyright 2018 by the authors

tecedent. Particularly adopting a machine learning approach, we have a far larger number of negative instances than that of positive instances. The ratio can be one against 1,000 in our case with BCCWJ. Such skewed training data unnecessarily increases computing time and hinder the system generalisation ability. To reduce the unnecessary negative instances, we filter out antecedent candidates by using the case frame information of the target predicate. We achieved 1/1,000 in reduction rate of candidate numbers by the proposed filtering method. Also, we incorporated the RNN into our model with the local attention mechanism (Luong et al., 2015) so that the system can selectively utilise the useful preceding sentences. This study is the first attempt to deal with both intra- and inter-sentential Japanese zero anaphora for three cases: nominative, accusative and dative in a single model, and to evaluate it by using a balanced corpus, BCCWJ.

2 Related Work

2.1 Japanese Zero Anaphora Resolution

Table 3 summarises related work regarding task types, text genres, corpus size, and methods. Hangyo et al. (2013) proposed a method based on ranking SVM for resolving intra- and inter-sentential anaphora and exophora in a Web corpus which they created for their study. The corpus consists of 1,000 text fragments extracted from the first three sentences of Web pages (Hangyo et al., 2012). Shibata et al. (2016) used a feed-forward neural network (FNN) for the analysis of directly dependent arguments and intra-sentential zero anaphora in the Web corpus created by Hangyo et al. (2012). Matsubayashi and Inui (2017) used a combination of an FNN and a recurrent neural network (RNN) to analyse directly dependent arguments and intra-sentential zero anaphora in NTC to show it outperformed the state-of-the-art model for directly dependent arguments and intra-sentential zero anaphora. Sasano and Kurohashi (2011) used a log-linear model to analyse intra and inter-sentential zero anaphora in a Web corpus consisting of 979 sentences and showed it outperformed the state-of-the-art model for intra and inter-sentential zero anaphora. Unlike these past studies, we adopt ranking SVM[2] (Joachims, 2006) and a combination of FNN and RNN to analyse intra- and inter-sentential zero anaphora in BCCWJ.

2.2 Large-scale Case Frames

A case frame represents co-occurrence information of a predicate and its possible arguments organised in case patterns of the predicate and its cases. Organising the case frame based on the case pattern as shown in Table 4 enables us to utilise its lexical preference for resolving anaphora (Sasano et al., 2008; Sasano and Kurohashi, 2011; Hangyo et al., 2013). We adopt Kyoto University Case Frames (KUCF)[3] which were compiled from a large-scale Web corpus by Kawahara and Kurohashi (2006).

2.3 Candidate Reduction

The past studies for inter-sentential zero anaphora resolution adopted criteria for candidate reduction. Sasano and Kurohashi (2011) and Hangyo et al. (2013) collected antecedent candidates from the sentence containing the target predicate and its preceding three sentences. Although antecedents could appear in the sentence beyond the preceding three sentences, Hangyo et al. (2013) reported that they could find 82.9% of the correct arguments of the predicates within the three sentences in NTC. Imamura et al. (2009) collected antecedent candidates from the sentence of the predicate and its previous sentence, reporting that they could find 62.5% of the correct antecedents in NTC, while reducing the number of candidates from 102.2 to 3.2 on average. Ouchi et al. (2015) formulated the predicate-argument structure analysis as a search on a bipartite graph with predicates and their argument candidates. They searched for a local optimum by hill climbing.

3 Proposed Model

Our proposed method consists of two components: a candidate reduction algorithm using word embeddings in the case frame, and a neural network-based model that utilises the word embeddings used for

[2] https://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

[3] http://www.gsk.or.jp/catalog/gsk2008-b/ We used an unpublished version.

Copyright 2018 by the authors

	type				text genre			corpus size	methods		
	dep	intra	inter	exophora	News	Web	etc.	(sentences)	linear	NN	+ att
(Imamura et al., 2009)	o	o	o		o			40,000	o		
(Hangyo et al., 2013)		o	o	o		o		3,000	o		
(Ouchi et al., 2015)	o	o			o			40,000	o		
(Shibata et al., 2016)	o	o				o		15,000		o	
(Iida et al., 2016)		o			o			40,000		o	
(Ouchi et al., 2017)	o	o			o			40,000		o	
(Matsubayashi and Inui, 2017)	o	o			o			40,000		o	
(Sasano and Kurohashi, 2011)		o	o			o		1,000	o		
Present work		o	o		o	o	o	60,000	o	o	o

Table 3: Related work

case frame	NOM	count	ACC	count	DAT	count
open:v$_1$	shop	129	–	–	near	6
	cafe	38	–	–	site	2
	restaurant	14	–	–	place	2
	...	...	–	–	...	...
open:v$_2$	brand	12	shop	59	–	–
	specialist	8	salon	18	–	–
	owner	4	boutique	13	–	–
	...	...	...	...	–	–

Table 4: Example of case frames for "open"

the candidate reduction in the training phase[4]. The method is based on the model by Hangyo et al. (2013).

3.1 Model Structure

Let S_0 be the sentence containing the predicate p to be analysed in the input text t and let its preceding h sentences be $S_{-1}, S_{-2}, \cdots, S_{-h}$. Let $E_p = \{e_1, e_2, \cdots, e_n\}$ be a set of all noun phrases appear in S_0 to S_{-h}. We further extend E_p by adding e_{none} denoting "no zero pronoun" or "exophora". Let $CF_p = \{cf_1^p, cf_2^p, \cdots, cf_m^p\}$ be a case frame set in KUCF corresponding to the predicate p. A case frame cf_l^p contains three case slots corresponding to each case $c \in \{\text{NOM}, \text{ACC}, \text{DAT}\}$, and one of the noun phrases in E_p is a case element corresponding to one case slot. Let $a = \langle \text{NOM} : e_i, \text{ACC} : e_j, \text{DAT} : e_k \rangle$ be the correspondence between case slots and case elements. Let (cf_l^p, a) be this predicate-argument structure candidate and let $\boldsymbol{f}(cf_l^p, a, t)$ be the feature vector representing it. The output of this model is given by Formula (1), where $\boldsymbol{w}$ is a weight vector learned by the training data.

$$cf_l^{p*}, a^* = \underset{cf_l^p, a}{\text{argmax}}\, \boldsymbol{w} \cdot \boldsymbol{f}(cf_l^p, a, t) \quad (1)$$

3.2 Features

A feature vector $\boldsymbol{f}(cf_l^p, a, t)$ consists of a combination of five types of features: *base model features*, *argument embeddings*, *a predicate embedding*, *a mean vector for case frame* (MVC) and *a context embedding*.

Base model features The base model features is represented by a vector ϕ_{BMF} each element of which is a real or binary value. These features have been used for conventional machine learning techniques such as SVM. The base model features ϕ_{BMF} include the probability of surface dependency obtained by the probabilistic case analysis model by Sasano et al. (2008) and the features proposed by Hangyo et al. (2013). The features by Hangyo et al. (2013) are divided into three types: case frame features, predicate features and context features. For instance, one of the case frame features is the probability that the input argument is assigned to the case slot of the case frame.

Argument embeddings Argument embeddings ϕ_e consist of three embeddings corresponding to antecedent candidates e_c for each case c. We used word2vec (Mikolov et al., 2013) to generate the word embedding[5].

[4] https://github.com/yamashiros/Japanese_zero_anaphora

[5] We used a model obtained by learning about 100 million articles acquired from the full body text of Japanese Wikipedia (2016-09-20) with 500 dimensions and 15 windows.

Copyright 2018 by the authors

Predicate embedding A predicate embedding is an embedding of the predicate generated by using word2vec.

Mean vector for case frame In KUCF, each case frame cf_l^p for a predicate p consists of three word lists for each case c as shown in Table 4. For instance, the case frame open:v_1 has "shop", "cafe", and "restaurant" in its nominatie case. Let $W_{cf_l^p(c)}$ be a set of words for case c in case frame cf_l^p. Let ϕ_w be an embedding of word $w \in W_{cf_l^p(c)}$. Let count(cf_l^p, c, w) be the number of occurrence of word w for case c in case frame cf_l^p. We then calculate a weighted mean vector (MVC) $\overline{\phi}_{cf_l^p(c)}$ for each case c in case frame cf_l^p from embedding vectors of words $W_{cf_l^p(c)}$.

$$\overline{\phi}_{cf_l^p(c)} = \frac{\sum_{w \in W_{cf_l^p(c)}} \text{count}(cf, c, w) \cdot \phi_w}{\sum_{w \in W_{cf_l^p(c)}} \text{count}(cf, c, w)} \quad (2)$$

For instance, the case frame open:v_1 has "shop" for its nominative case with 129 occurrences in Table 4. $\overline{\phi}_{\text{open:v}_1\text{(NOM)}}$ is calculated as follows:

$$\overline{\phi}_{\text{open:v}_1\text{(NOM)}} = \frac{129 \cdot \phi_{shop} + 38 \cdot \phi_{cafe} + \cdots}{129 + 38 + \cdots}. \quad (3)$$

We then concatenate $\overline{\phi}_{cf_l^p\text{(NOM)}}$, $\overline{\phi}_{cf_l^p\text{(ACC)}}$, and $\overline{\phi}_{cf_l^p\text{(DAT)}}$ to make $\overline{\phi}_{cf_l^p}$. We use $\overline{\phi}_{cf_l^p}$ to measure the relevance between a and cf_l^p for finding the best pair of them, i.e. selectional preference. We utilise MVCs as for both anaphora resolution phase and candidate reduction phase.

Context embedding Context embedding $c_{cf_l^p,a,t}$ is the output of the RNN with a local single-attention mechanism. The RNN receives a sentence with a target predicate and its preceding h sentences, modeling a context of the target predicate. Given $S_{-h:0}$ as an input, let Enc$(S_{-h:0})$ be the hidden states of the RNN encoder. LocalAtt$(\cdot)$ implements a single-local attention mechanism.

Our attentional model then infers an alignment weight vector based on the concatenation of the other feature vectors and the context embedding $c_{cf_l^p,a,t}$ is computed as the weighted average of the output of the encoder Enc$(S_{-h:0})$, according to the alignment weight vector.

$$c_{cf_l^p,a,t} = \text{LocalAtt}([\phi_{BMF}; \phi_e; \overline{\phi}_{cf_l^p}], \text{Enc}(S_{-h:0})) \quad (4)$$

Intuitively, this mechanism enables our model to identify a distant word from the predicate as an argument in the long context through the alignment vector. We expect that some case frames take the distant nouns from their predicates as their arguments, and this mechanism can directly model such phenomena.

4 Candidate Reduction using Word Embeddings in Case Frame

Naively enumerating all candidates for the predicate-argument structure (cf_l^p, a) makes a huge number, leading to an impractical search space. Following Sasano and Kurohashi (2011), we restrict the search range for the antecedent to at most three sentences before the predicate, i.e. we set the parameter h in Section 3.1 to 3. The distribution of case elements in BCCWJ shown in Table 1 tells us that we can find 89.16% of the antecedents even with this restriction.

As the number of candidates is $O(n^3m)$ where n and m are the numbers of noun phrases in E_p and the case frames for the target predicate respectively, we have still 20,000 candidates of the predicate-argument structure for each predicate in BCCWJ.

4.1 Mean Vector for Predicate

We propose an effective candidate reduction method using two kinds of mean vectors: MVC introduced in Section 3.2 and *mean vector for predicate* (MVP) $\overline{\phi}_{p(c)}$ which is a weighted mean vector of MVC $\overline{\phi}_{cf_l^p(c)}$ over the case frames of the predicate p for each case c. The weight is calculated based on the frequency of each case frame in KUCF. Our candidate reduction method reduces the number of combination of case frame candidates and argument candidates by using the hill climbing method proposed by Ouchi et al. (2015). The purpose of this candidate reduction is not only for efficiency but also for alleviating the imbalance between the number of positive and negative examples in the training data. In our case, a single positive example has 20,000 negative counterparts. We assume that most of the negative examples in the training data do not make much contribution to training.

Copyright 2018 by the authors

Algorithm 1 Candidate reduction algorithm

Input:
a predicate p to be analyzed,
a set of case frames CF_p corresponding to p,
a set of cases $C = \{\text{NOM}, \text{ACC}, \text{DAT}\}$,
a set of nouns E_p appearing within the h preceding sentences.
Output:
optimal cf_l^{p*}, e_c^* for the analyzed p and each case $c \in C$.

1: **for** each case $c \in C$ **do**
2: $e_c^{(0)} \Leftarrow \underset{e \in E_p}{\text{argmax}} \cos(\bar{\phi}_{p(c)}, \phi_e)$
3: **end for**
4:
5: $cf^{(0)} \Leftarrow \underset{cf_l^p \in CF_p}{\text{argmax}} \sum_{c \in C} \text{PSEUDO-SCORE}(cf_l^p, e_c^{(0)})$
6: $t \Leftarrow 0$
7: **repeat**
8: **for** each case $c \in C$ **do**
9: $e_c^{(t+1)} \Leftarrow \underset{e_c \in E_p}{\text{argmax}}\ \text{PSEUDO-SCORE}(cf^{(t)}, e_c)$
10: **end for**
11:
12: $cf^{(t+1)} \Leftarrow \underset{cf_l^p \in CF_p}{\text{argmax}} \sum_{c \in C} \text{PSEUDO-SCORE}(cf_l^p, e_c^{(t+1)})$
13: $t \Leftarrow t + 1$
14: **until** $e_c^{(t)} = e_c^{(t+1)}$ and $cf^{(t)} = cf^{(t+1)}$
15: **return** $cf_l^{p*} \Leftarrow cf^{(t)}, e_c^* \Leftarrow e_c^{(t)}$ for each case $c \in C$
16:
17: **function** PSEUDO-SCORE(cf_l^p, e)
18: $score \Leftarrow 0$
19: **for** each case $c \in C$ **do**
20: $score \Leftarrow score + P(p, cf_l^p, e, c)$
21: $score \Leftarrow score + \cos(\bar{\phi}_{cf_l^p(c)}, \phi_e)$
22: $score \Leftarrow score + 0.5 \times d_{p,e}$
23: **end for** ▷ $d_{p,e}$ is the distance between p and e
24:
25: **return** $score$
26: **end function**

4.2 Algorithm

We show our candidate reduction algorithm in Algorithm 1. Every predicate has more than one case frames, and this means that the predicate has ambiguity in selecting a correct case frame in a given context. Since each MVC reflects the selectional preference of that case frame, the closer the distance between the vector of an argument candidate and the MVC is, the more likely the candidate is to be taken as the argument of the case slot of the case frame. This algorithm searches for the combination of a case frame candidate and argument candidates for the given predicate that minimises the distances between the vectors. However, since KUCF was constructed by an automatic method, there can be wrong classification of case frames, i.e. distinct case frames had been merged into the same case frame and the same case frames had been divided into the different ones. To remedy this problem, we have introduced two kinds of mean vectors into our algorithm: MVC that differentiates the different case frames of the predicate and MVP that concerns only the predicate and does not care about the case frame difference.

First, the algorithm determines the initial argument $e_c^{(0)}$ for each $c \in C$ by calculating the cosine similarity between the word embedding ϕ_e of argument candidates and the MVP $\overline{\phi}_{p(c)}$. At this stage, we do not care about a specific case frame but only the predicate by using MVP. The most similar noun phrase to the case argument of the MVP is utilised as the initial argument of the case (line 1-3). PSUEDO-SCORE of the candidate (line 17-26) provides a score for choosing the optimal initial case frame $cf^{(0)}$ for these initial arguments (line 5). Following Sasano et al. (2008), we consider three factors: (1) probability derived from the combination of the predicate, the case frame, the pair of a case and its argument based on KUCF, (2) cosine similarity between the case frame slot and the argument candidate and (3) the distance between the predicate and the argument candidate regarding the number of sentences between them. We empirically determined the coefficients for these three factors. We alternate the phase of searching for arguments $e_c^{(t+1)}$ for the fixed case frame $cf^{(t)}$ (line 8-10) and the phase of searching for a case frame $cf^{(t+1)}$ for the fixed arguments $e_c^{(t+1)}$ (line 12), until the case frame and the arguments are no longer updated (line 6-15). In this algorithm, the combination of the case frame and arguments with the highest score is returned, but in practice, at most three best argument candidates calculated in each cycle are retained. The final output is all combinations of case frames and arguments saved during the search process.

5 Evaluation Experiment

5.1 Methods

We used Ranking SVM and FNN for learning and compared their results. Following the previous studies (Sasano and Kurohashi, 2011; Hangyo et al., 2013), we first run a morphological analysis, named entity extraction, and syntactic analysis on the entire document. We used JUMAN Ver.7.01[6], KNP

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

Copyright 2018 by the authors

Ver.4.16[7], CaboCha Ver.0.69[8] for this preprocessing.

S0 We implemented an SVM model with the base model features. We used linear SVM^{rank} (Joachims, 2006) for learning to rank. This model learns a discrimination function from positive and negative examples. Its outputs are ones with the highest score given by the discriminant function.

S0′ Our candidate reduction algorithm reduced the number of candidates to approximately 1/1000 while retaining about 70% of correct answers in the reduced candidates. To verify the effect of the proposed candidate reduction method, it would be natural to implement a model without any candidate reduction method. However, we have 20,000 candidates of the predicate-argument structure for each predicate, even if we restrict the search range for the antecedent to at most three sentences before the predicate. The computational complexity of the training is not realistic. Therefore we prepared another simple candidate reduction method. In this simple method, we choose only n nouns preceding the target predicate as argument candidates. We set $n = 5$ because our candidate reduction algorithm leaves five candidates on average. We name a model using ranking SVM with this simple candidate reduction method S0′.

F0 We implemented an FNN model with the base model features. We employed the softmax cross-entropy loss for training in the same way as Matsubayashi and Inui (2017) did. We apply the batch normalization (BN) and the ReLU activation function to each hidden layer.

F1 We extend F0 by adding argument embeddings and predicate embeddings.

F2 We extend F1 by replacing predicate embeddings with MVCs.

F3 We extend F2 by adding the output of the RNN. We employed GRU for our RNN. Figure 1 shows the overall structure of F3.

Table 5 summarises the models and their features. We did not use the MVP as the feature to input.

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

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

features	S0	F0	F1	F2	F3
Base model features	o	o	o	o	o
Argument embeddings			o	o	o
Predicate embedding			o		
MVCs				o	o
Context embedding					o

Table 5: Combination of features

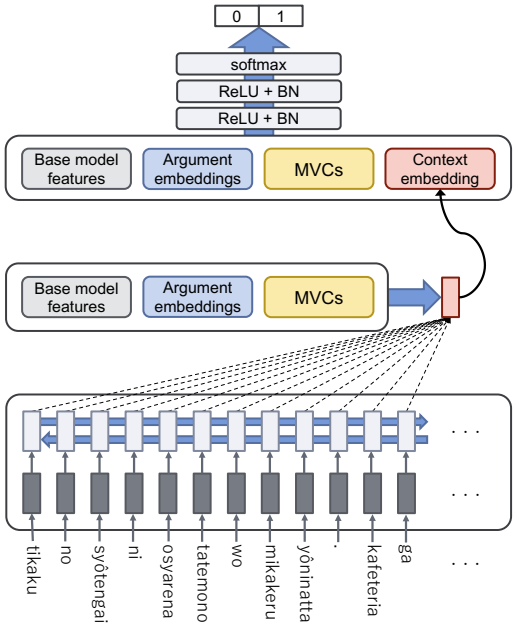

Figure 1: Network structure of our FNN model with attention RNN (model F3)

5.2 Dataset

We used the core data of Balanced Corpus of Contemporary Written Japanese (BCCWJ)[9] (Maekawa et al., 2014) for the experiments. The core data of BCCWJ includes 2,000 documents that are annotated with predicate-argument structures and coreference relations. The core data documents come from six genres: newspapers, magazines, books, white papers, social QAs, and blogs. We divided the core data into approximately 4/5 for training, 1/20 for development, and the rest for testing, making the distribution of genres in each portion as similar as possible. In cases where different noun phrases refer to the same object, i.e. co-reference, we regarded all the phrases referring to the correct entity as the

[9] http://pj.ninjal.ac.jp/corpus_center/bccwj/

Copyright 2018 by the authors

	intra				inter				All			
case	NOM	ACC	DAT	All	NOM	ACC	DAT	All	NOM	ACC	DAT	All
model \ #examples	3,137	1,458	873	5,468	2,359	495	243	3,097	5,496	1,953	1,116	8,565
S0′ (Base)	.490	.712	.725	.589	.032	.016	.140	.038	.331	.584	.632	.435
S0 (Base)	**.575**	**.758**	**.777**	**.661**	.044	.016	.145	.048	.390	**.628**	**.679**	**.491**
F0 (Base)	.523	.736	.775	.623	.054	.019	.151	.057	.356	.610	.677	.462
F1 (Base, Arg, Pred)	.470	.682	.762	.564	**.141**	.041	.138	**.126**	.342	.537	.659	.416
F2 (Base, Arg, MVC)	.563	.707	.773	.641	.103	**.063**	**.154**	.099	.394	.565	.674	.479
F3 (Base, Arg, MVC, Cont)	.562	.726	.757	.641	.096	.032	.147	.090	**.395**	.598	.658	.482

Table 6: Results on BCCWJ (F-measure)

correct answer based on the co-reference information annotated in the corpus. That is, we consider our system outputs as correct antecedent if our system locates any of the antecedents which corefer to the correct one. In this paper, we target only zero-anaphora of verbs, not adjectives nor event nouns.

6 Results and Discussion

Effect of candidate reduction Table 6 shows the experimental results on BCCWJ. Comparing S0′ and S0, we find that S0 is superior to S0′ for all columns in Table 6. We confirmed the statistical significance of the result by conducting a McNemar test at the significance levelt 0.001. This indicates the proposed candidate reduction algorithm works well.

Effect of word embedding and MVC Introducing word embeddings of the arguments and the predicate (F1) into the baseline model (F0) degrades the total accuracy. However, replacing the predicate embedding with the MVC (F2) increases the accuracy in comparison with F0. This indicates that using the case frame information (F2) instead of the predicate information (F1) is more effective.

As described in 3.2, we used word embeddings learnt from the Wikipedia articles, but the word embeddings calculated from corpora of more diverse genre texts and syntactic information (Levy and Goldberg, 2014) might further improve the performance.

Effect of context embedding Introducing the context information using the RNN model with the local attention mechanism (F3) shows some improvement over F2. This suggests that the model succeeded to learn effective preceding context information. Although the overall accuracy of F3 is still lower than that of S0, the FNN models with various features show higher accuracy for the inter-sentential cases.

Additional Results Appendix A. describes a further detailed analysis focusing on the interaction between the result of the dependency analysis and the accuracy of the proposed method. To compare the past research, we also describe the evaluation results with NTC that is popular for evaluating Japanese zero anaphora resolution in Appendix B.

7 Conclusion

This paper proposed a Japanese intra- and inter-sentential zero anaphora resolution model with a candidate reduction algorithm using case frames and word embeddings. Our candidate reduction algorithm enables the model to learn its parameters from a large-scale corpus and we confirmed the FNN models with various features showed higher accuracy for the inter-sentential cases. This study is the first attempt to resolve both intra- and inter-sentential zero anaphora for the three cases simultaneously, and to conduct the evaluation using the large-scale multi-domain balanced corpus, BCCWJ. Our future work includes handling adjectives and event nouns as the target predicate. We will also refine our candidate reduction algorithm by introducing the information whether a candidate noun is already an argument of the other predicates.

Acknowledgments

We would like to thank Dr. Masatsugu Hangyo and Dr. Hiroyuki Ouchi for providing the detailed information of their work (Hangyo et al., 2013; Ouchi et al., 2017).

Copyright 2018 by the authors

References

Chen Chen and Vincent Ng. 2016. Chinese Zero Pronoun Resolution with Deep Neural Networks. In *ACL*, pages 778–788.

Masatsugu Hangyo, Daisuke Kawahara, and Sadao Kurohashi. 2012. Building a Diverse Document Leads Corpus Annotated with Semantic Relations. In *PACLIC*, pages 535–544.

Masatsugu Hangyo, Daisuke Kawahara, and Sadao Kurohashi. 2013. Japanese Zero Reference Resolution Considering Exophora and Author/Reader Mentions. In *EMNLP*, pages 924–934.

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.

Ryu Iida and Massimo Poesio. 2011. A Cross-Lingual ILP Solution to Zero Anaphora Resolution. In *ACL*, pages 804–813.

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.

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

Thorsten Joachims. 2006. Training Linear SVMs in Linear Time. In *Proceedings of the 12th ACM SIGKDD*, pages 217–226.

Daisuke Kawahara and Sadao Kurohashi. 2006. A Fully-Lexicalized Probabilistic Model for Japanese Syntactic and Case Structure Analysis. In *ACL*, pages 176–183.

Taku Kudo, Hiroshi Ichikawa, and Hideto Kazawa. 2014. A joint inference of deep case analysis and zero subject generation for Japanese-to-English statistical machine translation. In *ACL*, pages 557–562.

Omer Levy and Yoav Goldberg. 2014. Dependency-Based Word Embeddings. In *ACL*, pages 302–308.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *EMNLP*, pages 1412–1421.

Kikuo Maekawa, Makoto Yamazaki, Toshinobu Ogiso, Takehiko Maruyama, Hideki Ogura, Wakako Kashino, Hanae Koiso, Masaya Yamaguchi, Makiro Tanaka, and Yasuharu Den. 2014. Balanced corpus of contemporary written Japanese. *Language Resources and Evaluation*, 48(2):345–371.

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

Tomas Mikolov, Kai Chen, Grag Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781*.

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.

Luz Rello, Ricardo Baeza-Yates, and Ruslan Mitkov. 2012. Elliphant: Improved Automatic Detection of Zero Subjects and Impersonal Constructions in Spanish. In *EACL*, pages 706–715.

Ryohei Sasano, Daisuke Kawahara, and Sadao Kurohashi. 2008. A Fully-Lexicalized Probabilistic Model for Japanese Zero Anaphora Resolution. In *COLING*, pages 769–776.

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.

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

Copyright 2018 by the authors

Kiyoshi Sudo, Satoshi Sekine, and Ralph Grishman. 2001. Automatic pattern acquisition for Japanese information extraction. In *HLT*, pages 1–7.

Hiroaki Yamada, Simone Teufel, and Takenobu Tokunaga. 2017. Designing an annotation scheme for summarizing Japanese judgment documents. In *KSE*, pages 275–280.

Qingyu Yin, Yu Zhang, Weinan Zhang, and Ting Liu. 2017. Chinese Zero Pronoun Resolution with Deep Memory Network. In *EMNLP*, pages 1309–1318.

Copyright 2018 by the authors

Appendix A. Interaction between Dependency Relations and Zero Anaphora Resolution

To see how the accuracy of zero anaphora resolution interacts with the dependency relations, we classified the test instances according to the combinations of cases that had been already filled by the dependency analysis. Table 7 shows the accuracy for each combination. The column indicates the determined cases while the row indicates the cases to be filled by the system. For instance, the figure at row "NOM" and column "ACC" indicates the accuracy to identify the "NOM" argument given the "ACC" argument by the result of dependency analysis.

As the accuracy in the shaded cells are low although their number is large, improving the performance for these shaded examples is important to increase the overall accuracy of zero anaphora resolution.

Cases to be filled as zero	Already-filled cases by dependency analysis						
	no args.	NOM	ACC	DAT	NOM, ACC	ACC, DAT	NOM, DAT
exo or none	.495 (794)	.817 (3461)	.586 (1011)	.785 (275)	.697 (2046)	.645 (152)	.724 (76)
NOM	.313 (1645)	-	.282 (1870)	.287 (683)	-	.243 (292)	-
ACC	.257 (416)	.384 (656)	-	.247 (81)	-	-	.750 (1222)
DAT	.505 (111)	.430 (337)	.319 (47)	-	.419 (129)	-	-
NOM, ACC	.112 (492)	-	- .144 (90)	-	-	-	
ACC, DAT	.091 (33)	.057 (35)	-	-	-	-	-
NOM, DAT	.228 (281)	-	.189 (122)	-	-	-	-
NOM, ACC, DAT	.000 (21)	-	-	-	-	-	-

Table 7: Interaction between dependency relations and zero anaphora resolution

Appendix B. Experiment on NAIST Text Corpus (NTC)

We compare our proposed method with Sasano and Kurohashi (2011) and Matsubayashi and Inui (2017) by using NTC. Table 8 shows the task design and Table 9 shows the results of the methods. As shown in Table 8, the task design across the methods is not the same; the comparison of the values in the absolute sense is not appropriate. In the present study, we used BCCWJ as training data and NTC as test data. One reason why we did not use NTC as training data is Sasano and Kurohashi (2011) also used their original Web corpus as training data (as described in Table 8). Another reason is that one of our contributions is to enable the model to learn its parameters from a large-scale corpus.

	task		training corpus			target			
	intra	inter	News	Web	etc.	verbs	adjectives	event nouns	
Matsubayashi and Inui (2017)	o		o			o	o		o
Sasano and Kurohashi (2011)	o	o		o		o	o		
Our methods	o	o	o	o	o	o			

Table 8: Task design of the related work

Copyright 2018 by the authors

	intra				inter				All			
case	NOM	ACC	DAT	All	NOM	ACC	DAT	All	NOM	ACC	DAT	All
number of verbs	11,559	7,472	4,389	23,420	2,810	229	142	3,181	14,369	7,701	4,531	26,601
S0 (Base)	.227	.271	.120	.224	.071	.020	.014	.058	.193	.243	.111	.196
Sasano and Kurohashi (2011)	.395	.175	.089		.244	.066	.026					
Matsubayashi and Inui (2017)	.565	.447	.160	.537								

Table 9: F-measure of experiment results using NTC

Copyright 2018 by the authors