Pseudo Zero Pronoun Resolution Improves
Zero Anaphora Resolution

Ryuto Konno[1*]   **Shun Kiyono**[2,3]   **Yuichiroh Matsubayashi**[3,2]
Hiroki Ouchi[4,2]   **Kentaro Inui**[3,2]

[1]Recruit Co., Ltd.   [2]RIKEN   [3]Tohoku University   [4]Nara Institute of Science and Technology

ryuto_konno@r.recruit.co.jp
shun.kiyono@riken.jp   y.m@tohoku.ac.jp
hiroki.ouchi@is.naist.jp   inui@tohoku.ac.jp

Abstract

Masked language models (MLMs) have contributed to drastic performance improvements with regard to zero anaphora resolution (ZAR). To further improve this approach, in this study, we made two proposals. The first is a new pretraining task that trains MLMs on anaphoric relations with explicit supervision, and the second proposal is a new finetuning method that remedies a notorious issue, the pretrain-finetune discrepancy. Our experiments on Japanese ZAR demonstrated that our two proposals boost the state-of-the-art performance, and our detailed analysis provides new insights on the remaining challenges.

1   Introduction

In pronoun-dropping languages such as Japanese and Chinese, the semantic arguments of predicates can be omitted from sentences. As shown in Figure 1, the semantic subject of the predicate *used* is omitted and represented by ϕ, which is called **zero pronoun**. This pronoun refers to *the criminal* in the first sentence. This way, the task of recognizing the antecedents of zero pronouns is called **zero anaphora resolution** (ZAR). This study focuses on Japanese ZAR.

ZAR is a challenging task because it requires reasoning with commonsense knowledge about the semantic associations between zero pronouns and the local contexts of their preceding antecedents. As shown in Figure 1, to identify the omitted semantic subject of *used*, the model should know the semantic relationship between *the criminal's weapon* and *a hammer*, namely, *a hammer is likely to be used as a weapon for murder and thus was used by the criminal*, is crucial. We hereinafter refer to such knowledge as **anaphoric relational knowledge**.

A conventional approach to acquire anaphoric relational knowledge is to collect predicate–argument

---

*Work done while at Tohoku University.

被害者の   部屋 から **犯人の**   凶器が   見つかった。
*victim-GEN room from **criminal**-GEN weapon-NOM was found.*

ϕ-NOM ハンマーを 使用した 模様。
ϕ-NOM *hammer-ACC used   seem.*

The **criminal**'s weapon was found in the victim's room.
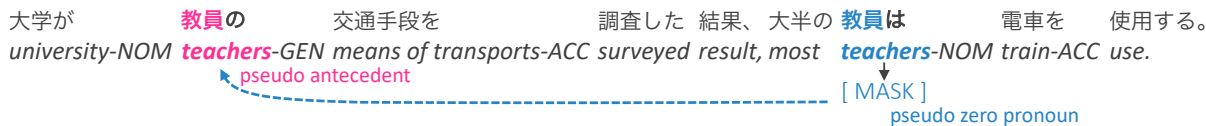It seems that ϕ used a hammer.

Figure 1: Example of argument omission in Japanese.

pairs from large-scale raw corpora and then, use them as features (Sasano et al., 2008; Sasano and Kurohashi, 2011; Yamashiro et al., 2018), or using selectional preference probability (Shibata and Kurohashi, 2018) in machine learning models. A modern approach is to use masked language models (MLMs) (Devlin et al., 2019), which is effective in implicitly capturing anaphoric relational knowledge. In fact, recent studies used pretrained MLMs and achieved drastic performance improvements in the tasks that require anaphoric relational knowledge, including Japanese ZAR (Joshi et al., 2019; Aloraini and Poesio, 2020; Song et al., 2020; Konno et al., 2020).

To get more out of MLMs, in this paper, we propose a new training framework that pretrains and finetunes MLMs specialized for ZAR. The idea is two-fold.

First, we design a new pretraining task that trains MLMs with explicit supervision on anaphoric relations. Many current pretraining tasks adopt a form of the Cloze task, where each MLM is trained by predicting the original token filling the [MASK] token. Although this task provides each MLM with no supervision on anaphoric relations, the MLM implicitly learns about them. In contrast, our new task, called the **pseudo zero pronoun resolution** (PZERO), provides supervision on anaphoric relations. PZERO assumes that when the same noun phrases (NPs) appear multiple times in a text, they are coreferent. From this assumption, we mask one of such multiple-occurring NPs as a **pseudo zero**

大学が　　　　教員の　　　　交通手段を　　　　調査した 結果、大半の 教員は　　　　電車を　　使用する。
university-NOM ***teachers*-GEN** *means of transports-ACC* *surveyed result, most* ***teachers*-NOM** *train-ACC* *use.*

pseudo antecedent

[MASK]
pseudo zero pronoun

The university has surveyed **teachers**' means of transport and found that most **teachers** use the train.

Figure 2: Example of our new pretraining task, PZERO. The second *teachers* is regarded as a pseudo zero pronoun and is masked, and the first *teachers* is its pseudo antecedent and should be selected to fill the mask.

pronoun and consider the other NPs as its **pseudo antecedents**.[1] As shown in the example in Figure 2, the NP, *teachers*, appears twice. The second is masked as a pseudo zero pronoun, and the first is regarded as its pseudo antecedent. Then, given the masked zero pronoun, an MLM is trained to select its (pseudo) antecedent from the candidate tokens in the context. The explicit supervision on such pseudo anaphoric relations allows MLMs to more effectively learn anaphoric relational knowledge.

Second, we address the issue called pretrain-finetune discrepancy (Yang et al., 2019). Generally, some part of an MLM is changed for finetuning on a target task, e.g., discarding the pretrained parameters at the last layer or adding randomly-initialized new parameters. Such changes in the architecture are known to be obstacles to the adaptation of pretrained MLMs to target tasks. To solve this issue, we design a new ZAR model that takes over all the pretrained parameters of an MLM to the ZAR task with minimal modification. This realizes a smoother adaptation of the anaphoric relational knowledge acquired during pretraining to ZAR.

Through experiments on Japanese ZAR, we verify the effectiveness of PZERO and the combination of PZERO and our new ZAR model. Also, our analysis offers insights into the remaining challenges for Japanese ZAR. To sum up, our main contributions are as follows:

- We propose a new pretraining task, PZERO, that provides MLMs with explicit supervision on anaphoric relational knowledge;
- We design a new ZAR model[2] that makes full use of pretrained MLMs with minimal architectural modifications;
- Our empirical results show that both the proposed methods can improve the ZAR performance and achieve state-of-the-art F_1 scores;
- Our analysis reveals that the arguments far

from predicates and the arguments of predicates in the passive voice are hard to predict.

2 Japanese Zero Anaphora Resolution

Japanese ZAR is often treated as a part of the predicate-argument structure analysis, which is the task of identifying semantic arguments for each predicate in a text. In the NAIST Text Corpus (NTC) (Iida et al., 2017), a standard benchmark dataset that we used in our experiments, each predicate is annotated with the arguments filling either of the three most common argument roles: the nominative (NOM), accusative (ACC), or dative (DAT) roles. If an argument of a predicate is a syntactic dependant of the predicate, we say that the argument is a syntactically dependent argument (DEP) and is relatively easy to identify. If an argument of a predicate is omitted, in contrast, we say that the argument position is filled by zero pronouns. This study is focused on recognizing such zero pronouns and identifying antecedents.

The ZAR task is classified into the following three categories according to the positions of the arguments of a given predicate (i.e., the antecedent of a given zero pronoun):

Intra-Sentential (*intra*): the arguments **within the same sentence** where the predicate appears.
Inter-Sentential (*inter*): the arguments **in the sentences preceding** the predicate.
Exophoric: the arguments (entities) that exist **outside the text**. These are categorized into one of three types: *author*, *reader*, and *general*.[3]

The identification of inter-sentential and exophoric arguments is an especially difficult task (Shibata and Kurohashi, 2018). For inter-sentential arguments, a model has to search the whole document. For exophoric arguments, a model has to deal with entities outside the document. Because of this difficulty, many previous studies have exclusively focused on the intra-sentential task. In this paper, not

---

[1]In addition to antecedents, we deal with postcedents. We use the term "antecedents" to refer both concepts for brevity.

[2]Our code is publicly available: https://github.com/Ryuto10/pzero-improves-zar

[3]The definitions of *Author* and *reader* correspond to those in Hangyo et al. (2013). *General* refers to the rest of *exophoric*.

only the intra-sentential task, we also treat inter-sentential and exophoric tasks as the same task formulations, as in previous studies.

3 Pseudo Zero Pronoun Resolution

3.1 Motivation and Task Formulation

The proposed PZERO is a pretraining task for acquiring anaphoric relational knowledge necessary for solving ZAR. PZERO is *pseudo* since it is assumed that all the NPs with the same surface form have anaphoric relationships. This assumption provides a large-scale dataset from raw corpora. Although the assumption seems to be too strong, an empirical evaluation confirmed that the pretraining task was effective (Section 6).

The task is defined as follows: Let $\boldsymbol{X}$ be a given input token sequence $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T)$ of length T, where one of the tokens is [MASK]. Here, $\boldsymbol{x} \in \mathbb{R}^{|\mathcal{V}|}$ is a one-hot vector and $\mathcal{V}$ is a vocabulary set. The task is to *select* the token(s) corresponding to the original NP of [MASK] from the input tokens. All the NPs with the same surface form as the masked NP are the answers of this task.

The most naive approach for masking NP is replacing all the tokens in the NP with the same number of [MASK] tokens. However, this approach is not appropriate for acquiring anaphoric relational knowledge, as the model can simply use a superficial clue, that is, the number of [MASK] tokens, to predict the original NP. Instead, we replace all the tokens in the NP with a **single** [MASK] token. Then, we formulate the task objective as predicting the last token in the original NP. This formulation is consistent with that of Japanese ZAR; when the argument consists of multiple tokens, the very last token is annotated as an actual argument.

3.2 Preparing Pseudo Data

To create training instances for PZERO, we first extract n consecutive sentences from raw text and split them into a subword sequence. We then insert [SEP] tokens as sentence separators (Devlin et al., 2019). Subsequently, we prune tokens from the beginning of the sequence and then prepend [CLS] at the beginning. As a result, the sequence consists of at most $T_{\max}$ subword tokens, which is the maximum input size of our model, as shown in Section 3.3. Then, for each NP in the last sentence, we search for corresponding NPs with the same surface form in this sequence. Upon finding such NPs, we replace the selected NP in the last

sentence with a single mask token and collect this sequence as a training instance.

3.3 Pretraining Model

Our model for PZERO closely resembles that of the transformer-based MLM (Devlin et al., 2019). Given an input sequence $\boldsymbol{X}$, each token $\boldsymbol{x}_t \in \{0, 1\}^{|\mathcal{V}|}$ is mapped to an embedding vector of size D, namely, $\boldsymbol{e}_t \in \mathbb{R}^D$ as follows:

$$\boldsymbol{e}_t = \boldsymbol{e}_t^{\text{token}} + \boldsymbol{e}_t^{\text{position}}. \quad (1)$$

Here, an embedding vector $\boldsymbol{e}_t^{\text{token}} \in \mathbb{R}^D$ is obtained by computing $\boldsymbol{e}_t^{\text{token}} = \boldsymbol{E}^{\text{token}} \boldsymbol{x}_t$, where $\boldsymbol{E}^{\text{token}} \in \mathbb{R}^{D \times |\mathcal{V}|}$ is a token embedding matrix. Similarly, an embedding vector $\boldsymbol{e}_t^{\text{position}} \in \mathbb{R}^D$ is obtained from the position embedding matrix $\boldsymbol{E}^{\text{position}} \in \mathbb{R}^{D \times T_{\max}}$ and a one-hot vector for position t. $T_{\max}$ represents the predefined maximum input length of the model.

Then, the transformer layer encodes the input embeddings $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_T$ into the final hidden states $\boldsymbol{H} = (\boldsymbol{h}_1, \ldots, \boldsymbol{h}_T)$. Given each hidden state $\boldsymbol{h}_t \in \mathbb{R}^D$ of the t-th token, we calculate the score $s_t \in \mathbb{R}$, which represents the likelihood that the token is a correct answer, by taking the dot product between the hidden state of the candidate token $\boldsymbol{h}_t$ and mask token $\boldsymbol{h}_{\text{mask}}$:

$$s_t = (\boldsymbol{W}_1 \boldsymbol{h}_t + \boldsymbol{b}_1)^\top \cdot (\boldsymbol{W}_2 \boldsymbol{h}_{\text{mask}} + \boldsymbol{b}_2), \quad (2)$$

where $\boldsymbol{W}_1$ and $\boldsymbol{W}_2 \in \mathbb{R}^{D \times D}$ are parameter matrices, and $\boldsymbol{b}_1, \boldsymbol{b}_2 \in \mathbb{R}^D$ are bias terms.

We train the model to maximize the score of the correct tokens. Specifically, we minimize the Kullback-Leibler (KL) divergence $\mathcal{L} = \text{KL}(\boldsymbol{y} \| \text{softmax}(\boldsymbol{s}))$, where $\boldsymbol{s} = (s_1, \ldots, s_T)$. $\boldsymbol{y} \in \mathbb{R}^T$ is the probability distribution of the positions of all the correct tokens. In this vector, the values corresponding to the positions of the correct tokens are set as $1/n$ and otherwise 0, where n is the number of correct tokens.

4 ZAR Models

Supposing that our model obtains anaphoric relational knowledge for ZAR by **pretraining** on PZERO, we design a ZAR model, as it can best utilize such knowledge during **finetuning**. In this section, we first describe an argument selection model (Section 4.1), which is considered the most straightforward adaptation of a pretrained model for ZAR. Then, we propose a novel model that addresses the pretrain-finetune discrepancy (Section 4.2).
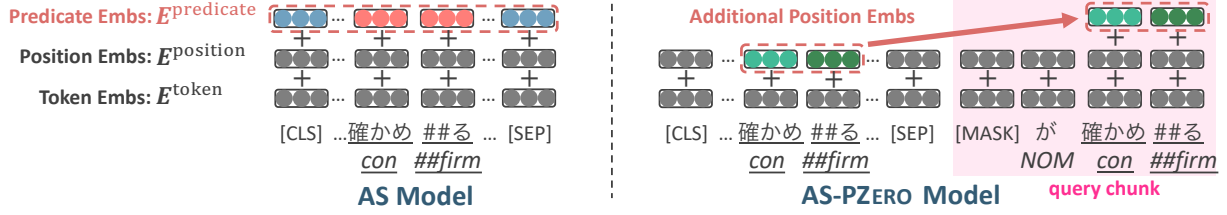
Figure 3: Input layer of AS and AS-PZERO. Their differences are that (1) a query chunk exists for AS-PZERO, and (2) the position of the target predicate is informed via different embedding types: $\boldsymbol{E}^{\text{predicate}}$ and $\boldsymbol{E}^{\text{position}}$.

4.1 Argument Selection with Label Probability: AS

The argument selection model, hereinafter AS, is a model inspired by the model of Kurita et al. (2018). From the recent standard practice of the pretrain-finetune paradigm (Devlin et al., 2019), we add a classification layer on top of the pretrained model.

The model takes an input sequence $\boldsymbol{X}$, which is created in a similar manner to that described in Section 3.1. $\boldsymbol{X}$ consists of multiple sentences and is pruned to contain $T_{\max}$ tokens at maximum. The target predicate is in the last sentence, and the [CLS] and [SEP] tokens are included. Also, the model takes two natural numbers p_{start} and p_{end} as inputs, where $1 \leq p_{\text{start}} \leq p_{\text{end}} \leq T$. These represent the position of the target predicate.

The model selects a filler token for each argument slot following a label assignment probability over $\boldsymbol{X}$: $\operatorname{argmax}_t P(t|\boldsymbol{X}, l, p_{\text{start}}, p_{\text{end}})$, where $l \in \{\text{NOM}, \text{ACC}, \text{DAT}\}$. We regard [CLS] (i.e., $\boldsymbol{x}_1$) as a dummy token representing the case that the argument filler does not exist in the input sequence. The model selects the dummy token in such cases.

The operation on the input layer of the model is shown on the left-hand side of Figure 3. First, each token $\boldsymbol{x}_t \in \{0,1\}^{|\mathcal{V}|}$ in a given input sequence $\boldsymbol{X}$ is mapped to an embedding vector $\boldsymbol{e}_t \in \mathbb{R}^D$ using the pretrained embedding matrices $\boldsymbol{E}^{\text{token}}$ and $\boldsymbol{E}^{\text{position}}$, and another new embedding matrix $\boldsymbol{E}^{\text{predicate}} \in \mathbb{R}^{D \times 2}$, as follows:

$$\boldsymbol{e}_t = \boldsymbol{e}_t^{\text{token}} + \boldsymbol{e}_t^{\text{position}} + \boldsymbol{e}_t^{\text{predicate}}, \quad (3)$$

where $\boldsymbol{e}_t^{\text{token}}$ and $\boldsymbol{e}_t^{\text{position}}$ are the same as in Equation 1. Moreover, $\boldsymbol{e}_t^{\text{predicate}}$ is an embedding vector computed from $\boldsymbol{E}^{\text{predicate}}$, p_{start}, and p_{end}. This vector represents whether the token in position t is a part of the predicate or not (He et al., 2017).

Second, we apply a pretrained transformer to encode each embedding $\boldsymbol{e}_t$ into the final hidden state $\boldsymbol{h}_t \in \mathbb{R}^D$. The probability distribution of assigning the label l over the input tokens $\boldsymbol{o}_l =$

$(o_{l,1}, ..., o_{l,T}) \in \mathbb{R}^T$ is then obtained by the soft-max layer:

$$o_{l,t} = \frac{\exp(\boldsymbol{w}_l^{\mathsf{T}} \boldsymbol{h}_t + b_l)}{\sum_t \exp(\boldsymbol{w}_l^{\mathsf{T}} \boldsymbol{h}_t + b_l)}, \quad (4)$$

where $\boldsymbol{w}_l \in \mathbb{R}^D$ and $b_l \in \mathbb{R}$. Finally, from the probability distribution $\boldsymbol{o}_l$, the model selects the token with the maximum probability as the argument of the target predicate.

When the model selects the dummy token as an argument, we further classify the argument into the following four categories: $z \in \{\text{author}, \text{reader}, \text{general}, \text{none}\}$. Here, none shows no slot filler for this instance. The other three categories author, reader, and general represent that there is a certain filler entity but do not appear in the context (*exophoric*). For this purpose, we calculated a probability distribution over the four categories $\boldsymbol{o}_l^{\text{exo}} = (o_{l,\text{author}}^{\text{exo}}, o_{l,\text{reader}}^{\text{exo}}, o_{l,\text{general}}^{\text{exo}}, o_{l,\text{none}}^{\text{exo}}) \in \mathbb{R}^4$ by applying a softmax layer to the hidden state of the dummy token $\boldsymbol{h}_1$ as follows:

$$o_{l,z}^{\text{exo}} = \frac{\exp(\boldsymbol{w}_{l,z}^{\mathsf{T}} \boldsymbol{h}_1 + b_{l,z})}{\sum_z \exp(\boldsymbol{w}_{l,z}^{\mathsf{T}} \boldsymbol{h}_1 + b_{l,z})}, \quad (5)$$

where $\boldsymbol{w}_{l,z} \in \mathbb{R}^D$, and $b_{l,z} \in \mathbb{R}$. Then the model selects the category with the maximum probability.

In the training step, we assign a gold label to the last token of an argument mention. If there are multiple correct answers in the coreference relations in the context, we assign gold labels to all these mentions. We prepare a probability distribution $\boldsymbol{y} \in \mathbb{R}^T$ of gold labels over the input token in a manner similar to that in Section 3.3. The models are then trained to assign high probabilities to gold arguments.

4.2 Argument Selection as Pseudo Zero Pronoun Resolution: AS-PZERO

One potential disadvantage of the AS model is that it may suffer from pretrain-finetune discrepancy.

That is, AS does not use the pretrained parameters, such as $\boldsymbol{W}_1$, $\boldsymbol{W}_2$, $\boldsymbol{b}_1$, and $\boldsymbol{b}_2$ in Equation 2, but is instead finetuned with randomly-initialized new parameters, such as w_l and b_l in Equation 4. To make efficient use of the anaphoric relational knowledge acquired during pretraining, we resolve this discrepancy. Inspired by studies addressing such discrepancies (Gururangan et al., 2020; Yang et al., 2019), we propose a novel model for fine-tuning; argument selection as pseudo zero pronoun resolution (AS-PZERO).

The underlying idea of AS-PZERO is to solve ZAR as PZERO. We use the network structure pretrained on PZERO as it is. Thus, the parameters w_l and b_l are no longer required. To do this, we modify the input sequence $\boldsymbol{X}$ for ZAR and reformulate the ZAR task as PZERO. Specifically, we prepare a short sentence, called a *query chunk*, and append it to the end of the input sequence $\boldsymbol{X}$. The query chunk represents a target predicate-argument slot whose filler is a single [MASK] token, so ZAR can be resolved by selecting the antecedent of the [MASK] token.

Let $\boldsymbol{X}'$ denote the modified input of AS-PZERO. The input layer of the model is shown on the right-hand side of Figure 3. The query chunk consists of a [MASK] token, a token representing a target argument label (i.e., NOM, ACC, or DAT), and a target predicate. For example, when the number of tokens in the target predicate is represented as $T_{\text{predicate}} = p_{\text{end}} - p_{\text{start}} + 1$, the length of $\boldsymbol{X}'$ is $T + 2 + T_{\text{predicate}}$. The modified input sequence is represented as $\boldsymbol{X}' = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{T+2+T_{\text{predicate}}})$.[4]

Given a modified input sequence $\boldsymbol{X}'$ and the start and end positions of the target predicate $p_{\text{start}}, p_{\text{end}} \in \mathbb{N}$, an input token $\boldsymbol{x}_t \in \{0, 1\}^{|V|}$ is mapped to a token embedding $\boldsymbol{e}_t \in \mathbb{R}^D$ as follows:

$$\boldsymbol{e}_t = \boldsymbol{e}_t^{\text{token}} + \boldsymbol{e}_t^{\text{position}} + \boldsymbol{e}_t^{\text{addposi}}, \qquad (6)$$

where $\boldsymbol{e}_t^{\text{addposi}}$ is an additional position embedding, which informs the model about the position of the target predicate. This information is intended to be used for distinguishing the target predicate from the multiple predicates appearing with an identical surface form in the input sequence. Specifically, for the target predicate in the query chunk, $\boldsymbol{e}_t^{\text{addposi}}$ is the same as the position embedding of the target predicate in the original sequence $\boldsymbol{X}$. Otherwise,

---

$\boldsymbol{e}_t^{\text{addposi}}$ is zero:

$$\boldsymbol{e}_t^{\text{addposi}} = \begin{cases} \boldsymbol{e}_{t'}^{\text{position}} & (p_{\text{start}} \leqq t' \leqq p_{\text{end}}) \\ \boldsymbol{0} & (\text{otherwise}), \end{cases} \quad (7)$$

where $t' = t - (T + 3) + p_{\text{start}}$. For example, as shown in Figure 3, the position embeddings of the target predicate (*con* and *##firm*) are added to those in the query chunk. Thus, we can avoid using the extra embedding matrix $\boldsymbol{E}^{\text{predicate}}$ in Equation 3.

We encode the embeddings with the transformer layer, and then use Equation 2 for the remaining computation of AS-PZERO to fill out the [MASK] token with the argument of the target predicate. If the score of the dummy token ($\boldsymbol{x}_1$) is highest, the model computes *exophoric* scores as described in Section 4.1 using Equation 5.

5 Experimental Settings

PZERO Dataset Japanese Wikipedia corpus (Wikipedia) is the origin of the training data of PZERO.[5] All the NPs in the corpus are PZERO targets. To detect NPs, we parsed Wikipedia using the Japanese dependency parser *Cabocha* (Kudo and Matsumoto, 2002) and applied a heuristic rule based on part-of-speech tags. We used $n = 4$ consecutive sentences to develop the input sequence $\boldsymbol{X}$. From 17.4M sentences in Wikipedia, we obtained 17.3M instances as training data for PZERO.

ZAR Dataset For the ZAR task, we used NAIST Text Corpus 1.5 (NTC) (Iida et al., 2010, 2017), which is a standard benchmark dataset of this task (Ouchi et al., 2017; Matsubayashi and Inui, 2018; Omori and Komachi, 2019; Konno et al., 2020). We used the training, development, and test splits proposed by Taira et al. (2008). The numbers of *intra-sentential*, *inter-sentential*, and *exophoric* for the training/test instances were 18068/6159, 11175/4081, and 13676/3826, respectively. The NTC details are shown in Appendix A. The evaluation script corresponds to that of Matsubayashi and Inui (2018).

Model Our implementation is based on the Transformers library (Wolf et al., 2020). We used the pretrained parameters of the bert-base-japanese model as the initial parameters of our pretraining models.

We trained our model using an Adam optimizer (Kingma and Ba, 2015) with warm-up steps. As a loss function, we used cross-entropy for the

---

ID	Method	ZAR *intra*	DEP	All
(a)	M&I	55.55	90.26	83.94 ± 0.12
(b)	O&K	53.50	90.15	83.82 ± 0.10
(c)	Konno et al. (2020)	64.15	92.46	86.98 ± 0.13
(d)	AS	69.32	93.65	88.87 ± 0.12
(e)	AS-PZERO	**69.91**	**93.83**	**89.06** ± 0.11

Table 1: F_1 scores on the NTC test set on ***intra-sentential*** setting. M&I: Matsubayashi and Inui (2018). O&K: Omori and Komachi (2019).

Cloze task and prediction of *exophoric* and used KL divergence for the rest. The details of the hyperparameter search are in Appendix B. Regarding the experiments on ZAR, we trained each model using five random seeds and reported the average score.

6 Results and Analysis

We have two distinct goals in this experiment, that is, to investigate the effectiveness of (1) pretraining on PZERO and (2) finetuning on AS-PZERO. To achieve these goals, we first compare our AS and AS-PZERO models with previous studies to ensure that our models are strong enough in a conventional experimental setting, i.e., the *intra-sentential* setting (Section 6.1). Then we investigate (1) and (2) based on *inter-sentential* setting (Section 6.2).

6.1 Intra-sentential Experiment

In this setting, the input sequence consists of a **single sentence**, and only the *intra*-zero and DEP arguments are targets of the evaluation. As mentioned in Section 2, most of the previous studies on Japanese ZAR use this setting (Matsubayashi and Inui, 2018; Omori and Komachi, 2019; Konno et al., 2020). Thus, we can strictly compare our results with those of other studies in this setting.

We finetuned AS and AS-PZERO from a pretrained MLM. The results in Table 1 show that both the AS and AS-PZERO models already outperformed the previous state-of-the-art models in *intra*-zero and DEP (Konno et al., 2020) with large margins. This improvement is due to the difference in the use of the pretrained MLM; given a pretrained MLM, we finetuned its entire parameters whereas Konno et al. (2020) used it as input features. Additionally, our pretrained MLM was trained better than theirs.

6.2 Inter-sentential Experiment

In this setting, the input sequence consists of **multiple sentences**: a sentence containing a target predicate and preceding sentences in the document. The *intra-sentential*, *inter-sentential*, *exophoric*, and DEP arguments are the evaluation targets.

We investigate the effectiveness of the proposed PZERO and AS-PZERO. For the experiment, we initialized the parameters of the transformer-based model with the pretrained MLM (pretrain 1) and further pretrained the model on Cloze and PZERO with the same number of updates. This resulted in having two pretrained models (pretrain 2 & 3). Then, we created models of all the possible combinations from {pretrain 1, 2, 3} and {AS, AS-PZERO}, resulting in the six models shown in Table 2.

(I) Do inter-sentential contexts help intra-sentential argument identification? We first investigate the impact of inter-sentential context on the performances of *intra*-zero and DEP by comparing the models (f) and (g) in Table 2 and the models (d) and (e) in Table 1. Here, note that model architectures of (f) and (g) are identical to those of (d) and (e), respectively. In addition, the evaluation instances of the *intra*-zero and DEP categories are the same for all four models. The differences are that the models (f) and (g) have broader contexts (inter-sentential contexts), i.e., multiple preceding sentences as inputs, and extra training signals from both the *inter*-zero and *exophoric* instances. A comparison of these four models shows that (f) and (g) have better performance than (d) and (e) in *intra*-zero and DEP. This result indicates that inter-sentential contexts are important clues even for identifying intra-sentential argument relations. This result is consistent with those of Guan et al. (2019) and Shibata and Kurohashi (2018), which discussed a method for utilizing inter-sentential contexts as clues for resolving semantic relations in target sentences.

(II) Does pretraining on PZero improve the performance of AS? As shown in Table 2, the comparison between the models pretrained on PZERO (j) and Cloze (h) shows that PZERO outperforms Cloze, especially in *inter*-zero argument (44.98 → 46.37). As discussed in Section 2, *inter*-zero is challenging because there are multiple answer candidates across the sentences. The improvement in *inter*-zero implies that the model effectively learns

ID	PT Task Cloze	Further PT Task Cloze	Further PT Task PZERO	FT Model AS	FT Model AS-PZERO	ZAR All	ZAR intra	ZAR inter	ZAR exophoric	DEP	All
(f)	✔			✔		62.27 ± 0.42	71.55	44.30	64.04	94.44	82.97
(g)	✔				✔	62.47 ± 0.53	71.09	45.20	64.41	94.46	83.03
(h)	✔	✔		✔		62.54 ± 0.47	71.82	44.98	63.94	**94.51**	83.10
(i)	✔	✔			✔	62.85 ± 0.19	71.52	45.97	64.55	94.49	83.18
(j)	✔		✔	✔		63.06 ± 0.19	71.96	46.37	64.42	94.43	83.26
(k)	✔		✔		✔	**64.18 ± 0.23**	**72.67**	**48.41**	**65.40**	94.50	**83.65**

Table 2: F_1 scores in the NTC test set with the ***inter-sentential*** setting. The bold values indicate the best results in the same column. PT and FT are abbreviations of pretraining and finetuning. The improvement of (k) over (h) is statistically significant in all the categories of ZAR F_1 ($p < 0.05$) with a permutation test.

ID	PT Task Cloze	Further PT Task Cloze	Further PT Task PZERO	FT Model AS	FT Model AS-PZERO	intra Precision	intra Recall	intra F_1	inter Precision	inter Recall	inter F_1
(h)	✔	✔		✔		**76.52**	67.67	71.82 ± 0.21	55.49	37.88	44.98 ± 1.05
(i)	✔	✔			✔	75.59	67.87	71.52 ± 0.22	56.31	38.91	45.97 ± 0.42
(j)	✔		✔	✔		75.85	68.46	71.96 ± 0.38	55.92	39.61	46.37 ± 0.34
(k)	✔		✔		✔	76.06	**69.58**	**72.67 ± 0.32**	**57.63**	**41.74**	**48.41 ± 0.35**

Table 3: The NTC test set results of the ***inter-sentential*** setting. The bold values indicate the best results in the same column group. PT and FT are abbreviations of pretraining and finetuning.

anaphoric relational knowledge through the pretraining on PZERO.

(III) Does pretraining on PZero improve the performance of AS-PZero? The performance comparison between the models (j) and (k) demonstrates the effectiveness of the combination of PZERO and AS-PZERO. The model (k) achieved the best result in all categories except for DEP. This indicates that AS-PZERO has successfully addressed the pretrain-finetune discrepancy and that it effecively used the anaphoric relational knowledge learned from PZERO.

Table 3 shows the precision and recall of the models (h)-(k) for the *intra*-zero and *inter*-zero arguments. The model (k) achieved the best recall performance in both categories and indicated that the proposed PZERO contributes mainly to the improvement in recall.

6.3 Analysis

We analyze the source of the improvement in recall as observed in Table 3. Table 4 shows our analysis of the *intra/inter-sentential* arguments from three aspects **I–III** and compares the detailed results of the baseline model (h) and our model (k).

(I) Number of gold antecedents in input This number determines the difficulty of the anaphora resolution. This is because the saliency of the entity is an important clue, i.e., ZAR is difficult when the argument appears only once in an input. Our model improved the performance of such difficult instances by a large margin (65.87 → 69.12 in *intra*-zero and 35.96 → 39.57 in *inter*-zero).

(II) Position of the argument relative to the target predicate The distance between a predicate and its argument determines the difficulty of ZAR. According to (3), (4), and (5), the performance of *inter-sentential* decreased as the predicate was farther from its argument's last surface-form appearance. Interestingly, the performance of the two models was comparable in (5), which is the case that the arguments are more than two sentences away from the target predicate. This result indicates that the proposed method is not effective for these instances. The error analysis on these instances revealed the fact that even though the argument did not appear explicitly, it was semantically present throughout the context as omitted arguments of the multiple predicates, all pointing to the same entity. This suggests that combining our proposed model with a model that propagates ZAR results through relevant contexts (Shibata and Kurohashi, 2018) can further improve ZAR performance.

(III) Voice of the target predicate Identifying the arguments of the predicate in the nonactive voice clause is difficult because of *case alternation*;

ID	Type of instances	*intra* recall		# of	*inter* recall		# of
		Model (h)	Model (k)	instances	Model (h)	Model (k)	instances
(I) Number of gold antecedents in input							
(1)	Only one	65.87	69.12	2001	35.96	39.57	1218
(2)	More than one	73.78	74.26	1247	53.1	54.1	872
(II) Position of the argument relative to the target predicate							
(3)	One sentence before	-	-	0	48.86	51.96	1099
(4)	Two sentences before	-	-	0	37.7	42.6	411
(5)	More than two sentences before	-	-	0	40.5	40.3	516
(6)	Out of input sequence	-	-	0	0.0	0.0	64
(III) Voice of the target predicate							
(7)	Active	70.80	72.62	2918	45.02	47.42	1877
(8)	Passive	51.8	57.9	309	25.2	29.1	206
(9)	Causative	55	50	20	60	60	7
(10)	Causative & Passive	100	100	1	-	-	0
	All	68.9	**71.09**	3248	43.11	**45.65**	2090

Table 4: Recall scores for each type of instance in the NTC development set. Models (h) and (k) are from Table 2.

semantic subjects and objects appear in the other syntactic positions. Table 4 shows that both models perform worse in (8) and (9) than in (7). Also the case alternation is different for every predicate. Thus, the model had to learn each behavior from training data and raw corpus. However, acquiring such information is not in the scope of PZERO.

6.4 Discussion on Pseudo Data Generation

In this study, we generated pseudo data for PZERO by exploiting the strong assumption that all the NPs with the same surface form have anaphoric relationships (Section 3.1). The advantage of our method is its high scalability in data collection; we can obtain a large amount of pseudo instances from raw corpora. Our empirical evaluation showed that our assumption is effective, however more sophisticated methods could be considered. Our future work includes analyzing the noise in pseudo data, i.e., NPs with the same surface but *no* anaphoric relationships, and its effect on the model performance.

7 Related Work

Anaphoric Relational Knowledge Our proposed pretraining task for acquiring anaphoric relational knowledge is related to script knowledge acquisition (Chambers and Jurafsky, 2009). Script knowledge models chains of typical events (predicates and their arguments). Between events, some arguments are shared and represented as variables, such as purchase X → acquire X, which can be regarded as a type of anaphoric relational knowl-

edge. While script knowledge only deals with shared arguments as anaphoric (coreferring) phenomena, anaphoric relational knowledge is not limited to them. In the sentence of Figure 1, the word *criminal* is not an argument of the predicate and is ignored in script knowledge, whereas it is within the scope of this work. Thus, it can be said that this work deals with broader anaphoric phenomena.

Zero Anaphora Resolution (ZAR) ZAR has been studied in multiple languages, such as Chinese (Yin et al., 2018), Japanese (Iida et al., 2016), Korean (Han, 2006), Italian (Iida and Poesio, 2011), and Spanish (Palomar et al., 2001). ZAR faces a lack of labeled data, which is a major challenge, and the traditional approach to overcome this is to use large-scale raw corpora. Several studies have employed these corpora as a source of knowledge for ZAR, e.g., case-frame construction (Sasano et al., 2008; Sasano and Kurohashi, 2011; Yamashiro et al., 2018) and selectional preference probability (Shibata et al., 2016). Furthermore, semi-supervised learning approaches, such as pseudo data generation (Liu et al., 2017) and adversarial training (Kurita et al., 2018), have been proposed. However, the use of pretrained MLM has been the most successful approach (Konno et al., 2020), and we sought to improve the pretraining task to better acquire anaphoric relational knowledge.

Pseudo Zero Pronoun Resolution (PZERO) Several studies have created training instances in a similar way as in PZERO. For example, Liu et al. (2017) casted the ZAR problem

as a reading comprehension problem, such that the model chose an appropriate word for the [MASK] from the vocabulary set. The difference is that, unlike their work, we filled the [MASK] by selecting a token from the given sentences. Also, Kocijan et al. (2019) created similar training data for Winograd Schema Challenge (Levesque, 2011). While we considered replacing arbitrary NPs with [MASK], they exclusively replaced the personal name. We expect that our approach is more suitable for ZAR because arguments are not necessarily personal names.

Pretrain-finetune Discrepancy Addressing the discrepancy between pretraining and finetuning is one of the successful approaches for improving the use of pretrained MLMs. For example, Gururangan et al. (2020) addressed the discrepancy with respect to the domain of the training dataset. Furthermore, Yang et al. (2019) indicated that [MASK] is used during the pretraining of MLM but never during finetuning. They improved a model architecture to mitigate such discrepancies. Therefore, inspired by these studies, we designed a finetuning model (AS-PZERO) that is suitable for a model pretrained on PZERO and demonstrated its effectiveness.

Prompt-based Learning Our use of query chunk in AS-PZERO can be seen as a prompt-based learning approach (Radford et al., 2019; Brown et al., 2020), which has been actively studied (Liu et al., 2021). In a typical prompt-based learning with a pretrained MLM, a model is trained to replace the masked token with a token from a predefined vocabulary (Schick et al., 2020; Schick and Schütze, 2021a,b; Gao et al., 2021). Our model is pretrained on PZERO, which is a task to select a pseudo antecedent from the preceding context. Thus, we designed AS-PZERO as a model to select the argument from the input sentences using a prompt-based approach to avoid the pretrain-finetune discrepancy.

8 Conclusion

In this study, we proposed a new pretraining task, PZERO, which aims to explicitly teach the model anaphoric relational knowledge necessary for ZAR. We also proposed a ZAR model to remedy the pretrain-finetune discrepancy. Both the proposed methods improved the performance of Japanese ZAR, leading to a new state-of-the-art performance. Our analysis suggests that the hard subcategories of ZAR; distant arguments and passive predicates are still challenging.

References

Abdulrahman Aloraini and Massimo Poesio. 2020. Cross-lingual Zero Pronoun Resolution. In *Proceedings of LREC*, pages 90–98.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam Mc-Candlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Proceedings of NeurIPS*.

Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised Learning of Narrative Schemas and their Participants. In *Proceedings of ACL-IJCNLP*, pages 602–610.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL*, pages 4171–4186.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making Pre-trained Language Models Better Few-shot Learners. In *Proceedings of ACL-IJCNLP*, pages 3816–3830.

Chaoyu Guan, Yuhao Cheng, and Hai Zhao. 2019. Semantic Role Labeling with Associated Memory Network. In *Proceedings of NAACL*, pages 3361–3371.

Suchin Gururangan, Ana Marasovic, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In *Proceedings of ACL*, pages 8342–8360.

Na-Rae Han. 2006. *Korean zero pronouns: Analysis and resolution*. Ph.D. thesis, University of Pennsylvania.

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

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

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

Ryu Iida and Massimo Poesio. 2011. A Cross-Lingual ILP Solution to Zero Anaphora Resolution. In *Proceedings of 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 *Proceedings of EMNLP*, pages 1244–1254.

Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel Weld. 2019. BERT for Coreference Resolution: Baselines and Analysis. In *Proceedings of EMNLP-IJCNLP*, pages 5803–5808.

Diederik Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of ICLR*.

Vid Kocijan, Ana-Maria Cretu, Oana-Maria Camburu, Yordan Yordanov, and Thomas Lukasiewicz. 2019. A Surprisingly Robust Trick for the Winograd Schema Challenge. In *Proceedings of ACL*, pages 4837–4842.

Ryuto Konno, Yuichiroh Matsubayashi, Shun Kiyono, Hiroki Ouchi, Ryo Takahashi, and Kentaro Inui. 2020. An Empirical Study of Contextual Data Augmentation for Japanese Zero Anaphora Resolution. In *Proceedings of COLING*, pages 4956–4968.

Taku Kudo and Yuji Matsumoto. 2002. Japanese Dependency Analysis using Cascaded Chunking. In *Proceedings of CoNLL*.

Shuhei Kurita, Daisuke Kawahara, and Sadao Kurohashi. 2018. Neural Adversarial Training for Semi-supervised Japanese Predicate-argument Structure Analysis. In *Proceedings of ACL*, pages 474–484.

Hector J. Levesque. 2011. The Winograd Schema Challenge. In *Proceedings of AAAI*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *arXiv preprint arXiv:2107.13586*.

Ting Liu, Yiming Cui, Qingyu Yin, Wei-Nan Zhang, Shijin Wang, and Guoping Hu. 2017. Generating and Exploiting Large-scale Pseudo Training Data for Zero Pronoun Resolution. In *Proceedings of ACL*, pages 102–111.

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.

Hikaru Omori and Mamoru Komachi. 2019. Multi-Task Learning for Japanese Predicate Argument Structure Analysis. In *Proceedings of NAACL*, pages 3404–3414.

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.

Manuel Palomar, Antonio Ferrández, Lidia Moreno, Patricio Martínez-Barco, Jesús Peral, Maximiliano Saiz-Noeda, and Rafael Muñoz. 2001. An Algorithm for Anaphora Resolution in Spanish Texts. *Computational Linguistics*, 27(4):545–567.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. Technical report, OpenAI.

Ryohei Sasano, Daisuke Kawahara, and Sadao Kurohashi. 2008. A Fully-Lexicalized Probabilistic Model for Japanese Zero Anaphora Resolution. In *Proceedings of 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 *Proceedings of IJCNLP*, pages 758–766.

Timo Schick, Helmut Schmid, and Hinrich Schütze. 2020. Automatically Identifying Words That Can Serve as Labels for Few-Shot Text Classification. In *Proceedings of COLING*, pages 5569–5578.

Timo Schick and Hinrich Schütze. 2021a. Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference. In *Proceedings of EACL*, pages 255–269.

Timo Schick and Hinrich Schütze. 2021b. It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners. In *Proceedings of NAACL-HLT*, pages 2339–2352.

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.

Tomohide Shibata and Sadao Kurohashi. 2018. Entity-Centric Joint Modeling of Japanese Coreference Resolution and Predicate Argument Structure Analysis. In *Proceedings of ACL*, pages 579–589.

Linfeng Song, Kun Xu, Yue Zhang, Jianshu Chen, and Dong Yu. 2020. ZPR2: Joint Zero Pronoun Recovery and Resolution using Multi-Task Learning and BERT. In *Proceedings of ACL*, pages 5429–5434.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of EMNLP demo*, pages 38–45.

Souta Yamashiro, Hitoshi Nishikawa, and Takenobu Tokunaga. 2018. Neural Japanese Zero Anaphora Resolution using Smoothed Large-scale Case Frames with Word Embedding. In *Proceedings of PACLIC*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Proceedings of NeurIPS*, pages 5754–5764.

Qingyu Yin, Yu Zhang, Weinan Zhang, Ting Liu, and William Yang Wang. 2018. Zero Pronoun Resolution with Attention-based Neural Network. In *Proceedings of COLING*, pages 13–23.

Dataset		dep	intra	inter	exophoric
Training	NOM	36934	12219	7843	11511
	ACC	24654	2136	948	128
	DAT	5744	465	294	60
Development	NOM	7424	2665	1812	1917
	ACC	5055	445	177	32
	DAT	1612	138	101	28
Test	NOM	14003	4993	3565	3717
	ACC	9407	906	371	55
	DAT	2493	260	145	54

Table 5: Statistics of NAIST Text Corpus 1.5

A   Statistics of NAIST Text Corpus 1.5

We used NAIST Text Corpus 1.5 (NTC) (Iida et al., 2010, 2017) for ZAR task. Table 5 shows the number of instances in NTC.

B   Hyperparameter Search on Validation Set

Table 6 shows a complete list of hyper-parameters used in this study. For both pretraining and fine-tuning, maximum learning rate and loss function are the target of the hyperparameter search. All the candidates of learning rates and loss functions are presented in Table 7. We used Nvidia Tesla V100 for the entire experiment.

Pretraining on Cloze   For the hyperparameter search of *Cloze* task, we adopted the hyperparameters that achieves the lowest perplexity value. We adopted 1.0×10^{-4} for maximum learning rate. We used the development set that we created from Japanese Wikipedia.

Pretraining on PZERO   For the hyperparameter search of PZERO; the parameters were determined by the validation performance on PZERO and ZAR.[6] We eventually employed the parameters with the highest F_1 on *inter* arguments.

Finetuning on ZAR   For the hyperparameter search of ZAR; the hyperparameters that achieve the highest overall ZAR F_1 were used. Here, we finetuned pretrained MLM without any further pretraining. Table 8 shows the result of our search process.

C   Heuristics for Extracting Noun Phrases from Raw Text

In order to extract noun phrases (NPs) from Japanese Wikipedia, we first parsed the corpus us-

ing Japanese dependency parser *Cabocha* (Kudo and Matsumoto, 2002). The parser divides the sentences into a phrase (Japanese "bunsetsu"). Note that each bunsetsu consists of a sequence of words. We then extracted the NPs as follows:

1. Choose a phrase that (1) contains noun(s) and (2) does not contain verb(s).
2. Scan the phrase from the end, and keep eliminating words until a noun appears.
3. Scan the phrase from the beginning, and keep eliminating words until a word other than a symbol appears.
4. The remaining words are regarded as a noun phrase. If the remaining words contain symbols, alphabet, or numbers only, then the words are not discarded.

Table 9 shows the statistics of Japanese Wikipedia and the number of PZERO instances generated from this process.

D   Performance on Validation Set

We report the performance on development set of NTC for model (d) to model (k), in Table 10 and Table 11. Here, each model ID follows that of Table 1 and Table 2.

E   Number of Parameters of each Model

We report total number of parameters of AS and AS-PZERO in Table 12.

---

[6]The task formulation of PZERO and AS-PZERO are quite similar. Thus we can evaluate the model, which is pretrained on PZERO, directly on ZAR without finetuning.

Configurations	Values
Optimizer	Adam (Kingma and Ba, 2015) ($\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1 \times 10^{-8}$)
Hidden State Size (D)	768 (defined in `bert-base-japanese`)
$T_{\max}$	512 (defined in `bert-base-japanese`)

Further Pretraining	
Mini-batch Size	2,048
Max Learning Rate	1.0×10^{-4} (Cloze task), 2.0×10^{-5} (PZERO)
Learning Rate Schedule	Inverse square root decay
Warmup Steps	5,000
Number of Updates	30,000
Loss Function	Cross entropy (Cloze task) and KL divergence (PZERO)
MLM's Mask Position	Random for each epoch

Finetuning	
Mini-batch Size	256
Max Learning Rate	1.0×10^{-4} (AS) and 1.0×10^{-5} (AS-PZERO)
Learning Rate Schedule	Same as described in Appendix A of Matsubayashi and Inui (2018)
Number of Epochs	150
Stopping Criterion	Same as described in Appendix A of Matsubayashi and Inui (2018)
Loss Function	KL divergence (AS) and KL divergence (AS-PZERO) and Cross entropy (prediction of *exophoric*)

Table 6: List of hyper-parameters

Pretraining Task	Finetuning Model	Loss Function
Cloze	-	{**Softmax Cross Entropy**}
PZERO	-	{Sigmoid Cross Entropy, **KL Divergence**}
-	AS	{Sigmoid Cross Entropy, **KL Divergence**}
-	AS-PZERO	{Sigmoid Cross Entropy, **KL Divergence**}

		Maximum Learning Rate
Cloze	-	$\{\mathbf{1.0 \times 10^{-4}}, 5.0 \times 10^{-5}, 2.0 \times 10^{-5}, 1.0 \times 10^{-5}, 5.0 \times 10^{-6}\}$
PZERO	-	$\{1.0 \times 10^{-4}, 5.0 \times 10^{-5}, \mathbf{2.0 \times 10^{-5}}, 1.0 \times 10^{-5}, 5.0 \times 10^{-6}\}$
-	AS	$\{\mathbf{1.0 \times 10^{-4}}, 5.0 \times 10^{-5}, 2.0 \times 10^{-5}, 1.0 \times 10^{-5}\}$
-	AS-PZERO	$\{1.0 \times 10^{-4}, \mathbf{5.0 \times 10^{-5}}, 2.0 \times 10^{-5}, 1.0 \times 10^{-5}\}$

Table 7: Candidates of hyper-parameters. **Bold** value indicates the adopted values.

Loss Function	Learning Rate	ZAR				DEP	All
		All	*intra*	*inter*	*exophoric*		
Argument Selection as PZERO (AS-PZERO)							
Sigmoid Cross Entropy	1.0×10^{-4}	61.66	71.10	45.29	61.98	94.64	83.07
	5.0×10^{-5}	62.00	71.52	45.25	62.80	94.80	83.25
	2.0×10^{-5}	62.28	71.26	47.25	62.15	94.47	83.23
	1.0×10^{-5}	62.20	71.17	48.02	61.57	94.54	83.22
KL Divergence	1.0×10^{-4}	63.30	72.40	47.33	64.21	94.82	83.74
	$\mathbf{5.0 \times 10^{-5}}$	**63.73**	**72.55**	**48.39**	**64.33**	**94.99**	**83.97**
	2.0×10^{-5}	62.52	71.78	46.98	62.96	94.91	83.52
	1.0×10^{-5}	61.60	70.38	47.29	61.44	94.45	82.86
Argument Selection with Label Probability (AS)							
KL Divergence	$\mathbf{1.0 \times 10^{-4}}$	**62.77**	**72.68**	45.65	**63.15**	**95.03**	**83.70**
	5.0×10^{-5}	62.28	71.87	**46.65**	62.39	94.96	83.48
	2.0×10^{-5}	61.73	71.58	45.99	61.56	94.52	82.97
	1.0×10^{-5}	61.57	71.60	45.57	61.22	94.73	83.04
Sigmoid Cross Entropy	1.0×10^{-4}	59.25	69.02	41.30	60.42	94.36	82.05
	5.0×10^{-5}	57.51	66.88	40.34	58.47	93.90	81.20
	2.0×10^{-5}	59.29	68.87	42.33	60.16	94.24	82.07
	1.0×10^{-5}	59.40	68.90	43.28	59.77	94.47	82.22

Table 8: F_1 scores on the NTC **development set** for hyper-parameter search. Bold value indicates the best results in the same column.

	Training	Development
Documents	1,121,217	300
Sentences	17,436,975	3,622
Instances of PZERO	17,353,590	3,236

Table 9: Statistics of Japanese Wikipedia

ID	Method	ZAR *intra*	DEP	All
(d)	AS	70.41	94.21	89.48±0.10
(e)	AS-PZERO	**70.66**	**94.23**	**89.50**±0.08

Table 10: F_1 scores on the NTC development set on *intra-sentential* setting.

ID	PT Task Cloze	Further PT Task Cloze	PZERO	FT Model AS	PZERO	ZAR All	intra	inter	exophoric	DEP	All
(f)	✔			✔		62.97±0.27	72.67	46.84	63.33	94.99	83.73
(g)	✔				✔	63.34±0.16	72.61	47.21	64.04	94.96	83.83
(h)	✔	✔		✔		63.42±0.29	72.85	48.26	63.21	**95.02**	83.92
(i)	✔	✔			✔	63.63±0.29	72.57	48.70	63.82	94.95	83.92
(j)	✔		✔	✔		63.15±0.45	72.95	47.48	62.84	94.96	83.78
(k)	✔		✔		✔	**64.59±0.17**	**74.03**	**49.60**	**64.12**	95.00	**84.27**

Table 11: F_1 scores on the NTC 1.5 development set on ***inter-sentential*** setting. Bold value indicates the best results in the same column. PT and FT are abbreviations of pretraining and finetuning.

Model	Number of Parameters
AS	111,223,312
AS-PZERO	112,395,268

Table 12: Number of parameters of the models

Method	ZAR				DEP	All
	All	*intra*	*inter*	*exophoric*		
Finetuning						
AS	62.27±0.42	71.55	44.30	64.04	94.44	82.97
AS-PZERO	62.47±0.53	71.09	45.20	64.41	94.46	83.03
10K Further Pretraining + Finetuning						
Cloze + AS	62.52±0.37	71.40	44.79	64.55	94.48	83.08
Cloze + AS-PZERO	62.42±0.30	71.04	45.31	64.15	94.49	83.04
PZERO + AS	62.63±0.41	71.63	45.14	64.47	94.39	83.07
PZERO + AS-PZERO	**63.52**±0.19	**72.07**	**47.43**	**64.95**	**94.58**	**83.46**
30K Further Pretraining + Finetuning						
Cloze + AS	62.54±0.47	71.82	44.98	63.94	**94.51**	83.10
Cloze + AS-PZERO	62.85±0.19	71.52	45.97	64.55	94.49	83.18
PZERO + AS	63.06±0.19	71.96	46.37	64.42	94.43	83.26
PZERO + AS-PZERO	**64.18**±0.23	**72.67**	**48.41**	**65.40**	94.50	**83.65**

Table 13: F_1 scores on the NTC 1.5 **test set**. Bold value indicates the best results in the same column.