Zero Pronouns Identification based on Span Prediction

Sei Iwata[1], **Taro Watanabe**[1], and **Masaaki Nagata**[2]
[1]Nara Institute of Science and Technology
[2]NTT Communication Science Laboratories, NTT Corporation
{iwata.sei.is6,taro}@is.naist.jp
masaaki.nagata.et@hco.ntt.co.jp

Abstract

The presence of zero-pronoun (ZP) greatly affects the downstream tasks of NLP in pro-drop languages such as Japanese and Chinese. To tackle the problem, the previous works identified ZPs as sequence labeling on the word sequence or the linearlized tree nodes of the input. We propose a novel approach to ZP identification by casting it as a query-based argument span prediction task. Given a predicate as a query, our model predicts the omission with ZP. In the experiments, our model surpassed the sequence labeling baseline.

1 Introduction

Pro-drop languages, such as Japanese, Chinese, or Arabic, allow omissions of essential phrases or arguments, e.g., nouns, which could be easily inferred by humans given contexts in a sentence. The omitted argument is called zero-pronoun (ZP), or (small) "pro", which is an instance of empty categories in linguistics.

JA このケーキは美味しい。私は (pro-OBJ) 気に入った.

EN This cake is delicious. I like (it).

In the Japanese example above, the object argument (OBJ) is omitted from the second sentence because Japanese speakers can predict from the context that the OBJ is "it", and the omission is natural for the Japanese speakers.

Downstream tasks involving pro-drop languages could easily suffer from the existence of ZPs. In the machine translation task, it has been reported that supplementing the ZP information when translating from pro-drop languages to non-pro-drop languages improves the performance (Wang et al., 2019).

When identifying a ZP from the sentence where the argument is omitted, the predicate information is the key. The ZP identification is solved in many previous works as a labeling task for input sentence tokens (Aloraini and Poesio, 2020; Song et al., 2020) or nodes in a parse tree (Xiang et al., 2013; Takeno et al., 2015).

In this study, we treat ZP identification as an instance of span prediction tasks inspired by the QA method proposed in Devlin et al. (2019). There are two steps to solve the ZP identification in our approach. 1) Given a predicate as a query, our model extracts each argument, such as subject or object, as the answer from the input sentence. 2) If our model cannot extract any corresponding argument from the input sentence, the model predicts whether or not it is a ZP. In the above example, given a predicate 気に入った "like", our model should predict that the subject argument is 私は "I" in the sentence and the object argument is a ZP. By explicitly providing predicates as queries in this way, our approach allows the model to capture information about the ZP cue from the input sentence, thereby improving the ZP identification performance.

Our contributions are as follows: 1)We proposed a novel approach for ZP identification. 2)The improvement from the sequence labeling baseline was confirmed on two different language datasets.

2 Related work

Most of the researchers considered the ZP detection or ZP identification as a labeling task. Xiang et al. (2013) and Takeno et al. (2015) used parse trees as input and detected empty categories, including ZPs, by labeling a node representing the maximal projection of a predicate, namely IP or VP. Song et al. (2020) proposed jointly learning

Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop, pages 331–336
August 5–6, 2021. ©2021 Association for Computational Linguistics

ZP resolution and ZP identification by treating it as sequence labeling on every word boundary. Aloraini and Poesio (2020) considered word positions before or after each VP node as ZP location candidates and predicted whether the candidate has ZP or not as a binary classification task. To the best of our knowledge, our approach is the first work that formalizes ZP identification as a QA task.

In recent years, approaches for solving various tasks as QA-based span prediction problems have been proposed. Li et al. (2020) made questions corresponding to NER entity tags. Then, their model predicted the entity span giving the question and a sentence as QA tasks to tackle the nested NER problem. In the coreference resolution task, Wu et al. (2020) generated queries based on each mention and extracted the text spans of coreferences as answers to given queries. Nagata et al. (2020) improved the performance of word alignment task by giving a word in the source language sentence as a question and predicting its corresponding word span in the target language sentence.

3 Span-based ZP identification

Treebanks have phrase structure tree information, and in some treebanks, empty categories are also annotated as null terminal nodes (Butler et al., 2012; Xue et al., 2005). However, we focused only on ZP identification, not dealing with other empty categories, such as trace and PRO, in this paper.

We formally define the ZP identification as span-based prediction as follows: Given a tokenized sentence $\boldsymbol{x} = x_1, ..., x_{|\boldsymbol{x}|}$, we denote a span of the sentence as $x_{qs:qe}$ $(1 \leq qs \leq qe \leq |\boldsymbol{x}|)$ that corresponds to the head of predicate of the sentence, i.e., verb or adjective. The task is to identify the span of the sentence $\boldsymbol{x}$ corresponding to the argument required by the predicate $x_{qs:qe}$. When no span is detected, there are three possible cases: (i) the argument is dropped as a kind of ZP; (ii) the argument is not dropped as a ZP, but as another empty category such as trace or PRO; (iii) it is not required by the predicate at all. We grouped the latter two cases into one class, the non-ZP class. Therefore, our model predicts one of the ZP classes or the non-ZP class for the required but omitted argument. The prediction is applied for each grammatical function of the argument, such as SBJ, OBJ, etc.

Our argument span prediction is inspired by BERT fine-tuning for the QA task (Devlin et al., 2019). Inputs follow a BERT style formulated as "[CLS] query [SEP] sentence [SEP]", where [CLS] is a special token to output the classification result and [SEP] denotes the boundary of "query" and "sentence." The query in the input is defined as follows:

$\{\ x_{qs-C:qs-1}$, [Predicate1], $x_{qs:qe}$, [Predicate2], $x_{qe+1:qe+C}\ \}$

where C is the size of the context windows before and after the span $x_{qs:qe}$ in the sentence. [Predicate1] and [Predicate2] [1] are used as boundary markers to specify the start and end of the predicate in the query.

(1)	(ϕ)	大学	へ	着き	まし	た
	(pro)-SBJ	university	at	VB	AX	AXD

' (pro) arrived at the university. '

In the example sentence (1) , there are five words in the tokenized input sentence excluding a null token ϕ [2]. Given "着き" as a predicate with $C = 1$, the query is represented as follows:

{"へ", [Predicate1], "着き", [Predicate2], "まし" }

Given the inputs, our model is expected to predict that SBJ is a required argument belonging to "pro" class and OBJ is a non-ZP argument because the predicate is an intransitive verb.

3.1 Argument Span Prediction

Two independent linear layers are added to BERT for predicting the start and end positions of an argument type for an input predicate. We dealt with three arguments, which are subject, object, and indirect object, for a predicate and added six layers in total.

Using hidden size H, $\boldsymbol{h}_a \in \mathbb{R}^H$ is the embedding of the final BERT encoder layer, corresponding to a token a in the input, and $f^{arg}_{start}(\cdot)$ and $f^{arg}_{end}(\cdot)$ are linear layers to calculate start and end probabilities. Given x_i, the ith word in a sentenece $\boldsymbol{x}$, let $p^{arg}_{start}(x_i) = f^{arg}_{start}(\boldsymbol{h}_{x_i})$ and $p^{arg}_{end}(x_i) = f^{arg}_{end}(\boldsymbol{h}_{x_i})$ denote the probabilities that the ith word is the start and end of the span of arg, argument e.g., SBJ, OBJ, etc.

[1] These words are implemented using unused words in the BERT vocabulary, "[UnusedX]".

[2] ϕ is a null token indicating "pro", which does not appear in the actual input sentence.

The score that the span $x_{i:j}$ is the span of arg is defined as the product of the ith word start probability and the jth word end probability of arg. We define $\hat{\imath}$ and $\hat{\jmath}$ as the start and the end positions that maximize $score_{arg}(i, j)$.

$$score_{arg}(i, j) = p^{arg}_{start}(x_i) \cdot p^{arg}_{end}(x_j) \quad (1)$$

$$(\hat{\imath}, \hat{\jmath}) = \underset{1 \leq i \leq j \leq |\boldsymbol{x}|}{\arg\max}\ score_{arg}(i, j) \quad (2)$$

When there is no arg span in the predicate, we assume its start and end positions equal to that of [CLS] and define the score as follows:

$$score_{null} = p^{arg}_{start}([\text{CLS}]) \cdot p^{arg}_{end}([\text{CLS}]) \quad (3)$$

There are two cases for $score_{null}$ and $score_{arg}(\hat{\imath}, \hat{\jmath})$:

$$score_{null} \leq score_{arg}(\hat{\imath}, \hat{\jmath}) \quad (4)$$

$$score_{null} > score_{arg}(\hat{\imath}, \hat{\jmath}) \quad (5)$$

When Equation 4 holds, our model predicts that the span between the $\hat{\imath}$th and $\hat{\jmath}$th in $\boldsymbol{x}$ is the argument arg for the given predicate. Otherwise, the argument for the given predicate does not exist in $\boldsymbol{x}$ denoted by Equation 5, which implies ZP exists in the argument or the argument is a non-ZP state.

The loss of a single example is calculated by the cross-entropy loss of correct positions i' and j':

$$loss_{span} = \sum_{arg} - \log p^{arg}_{start}(x_{i'}) - \log p^{arg}_{end}(x_{j'}) \quad (6)$$

3.2 ZP classification

The difference between ZP detection and ZP identification is whether there are one or more classes of ZPs for arguments. In the ZP detection task, ZP classification is binary classification whether the argument is either ZP class or non-ZP class. When there are multiple ZP classes to solve the ZP identification task, the ZP classification is a multi-class classification.

To classify, we add an independent layer for each predicted argument type into BERT. The arg class probabilities are as follows:

$$p^{arg}_{class} = softmax(\boldsymbol{h}_{[\text{CLS}]}\mathbf{W}_{arg} + \boldsymbol{b}_{arg}) \quad (7)$$

where $\mathbf{W}_{arg} \in \mathbb{R}^{H \times num_{class}}$, and $\boldsymbol{b}_{arg} \in \mathbb{R}^{num_{class}}$ are parameters. num_{class} is the number of classes including the non-ZP class.

The loss $loss_{label}$ is calculated by cross-entropy function and the correct label probability.

$$loss_{label} = - \log p^{arg}_{class}(index_{correct}) \quad (8)$$

Datasets	Category	Train	Dev	Test
NPCMJ	docs(all)		261	
	sents	29,796	3,724	3,726
	preds	76,892	9,595	9,450
OntoNotes 5.0	docs	1,391	172	166
	sents	32,358	5,435	9,450
	preds	135,241	19,538	16,556

Table 1: Statistics on NPCMJ and OntoNotes5.0. In the "Category" column, "docs", "sents", and "preds" represent documents, senteneces, and predicates, respectively. In NPCMJ, "all" means the total number of documents in train, dev, and test.

Datasets	argument	SBJ	OB1	OB2
NPCMJ	ZP ratio(%)	20.58	3.67	0.24
	ZP number	15,824	2,823	184
OntoNotes 5.0	ZP ratio(%)	21.59	0.05	0.00
	ZP number	29,195	61	1

Table 2: The ratio and the number of ZPs to queries in train datasets of NPCMJ and Chinese subsets OntoNotes.

3.3 Training

The training objective is defined using $loss_{span}$ and $loss_{label}$ in 3.1 and 3.2 as follows:

$$loss_{total} = \alpha loss_{span} + (2 - \alpha) loss_{label} \quad (9)$$

α is a hyperparameter that weights the loss function of each task by taking a value between $0 < \alpha < 2$ [3].

4 Experiments

4.1 Datasets

We take two Datasets: NPCMJ[4] for Japanese ZP identification and OntoNotes5.0[5] for Chinese ZP detection. The dataset statistics are shown in Tables 1 and 2.

NPCMJ is an extension of the Keyaki Treebank (Butler et al., 2012), which contains empty category information including ZP, and has 40,831 sentences with trees in the March 2020 version. ZPs are annotated at the first position of a predicate head phrase (inflectional phrase, IP). In the Japanese experiments, let $x_{qs:qe}$ in a query be a word tagged either with the verb or the adjective that constitutes a predicate.

The verb tags are "VB", "VB0", "VB2", and "AX", and the adjective tags are "ADJN" and

[3] We first run our preliminary experiments by setting $\alpha = 1$, and then, run further experiments using linear interpolation

[4] http://npcmj.ninjal.ac.jp

[5] https://catalog.ldc.upenn.edu/LDC2013T19

"ADJI". The phrase tagged with "-SBJ", "-OB1", or "-OB2", which is at the same depth of the query, is selected as the argument. In training, we used "pro" and its derived tags, i.e., "speaker" and "hearer", as ZP classes for ZP classification.

OntoNotes5.0 is used in the official CoNLL-2012 shared task. The rate of phrase tags of "pro" nodes in train datasets is composed of "-SBJ" with more than 99%, "-OBJ" with less than 0.5%, and others. The phrases tagged with "-SBJ", "-OBJ", or "-IO" are treated as arguments. The head word of the phrase with VP is considered as a predicate, and let the head word be $x_{qs:qe}$ in a query.

In Japanese and Chinese, there are nominal predicate phrases which do not have verbs and copulas. Such phrases were tagged with "-PRD" tags in both datasets, but we did not deal tagged with "-PRD" in this paper.

4.2 Model and Setting

We used NICT BERT Japanese pre-trained model without BPE[6] for NPCMJ, and "bert-base-chinese"[7] models in HuggingFace's Transformers (Wolf et al., 2019) for OntoNotes5. Japanese texts are tokenized by MeCab with Juman dic[8], and Chinese texts are tokenized by BERT Tokenizer, i.e., WordPiece.

The following are the hyperparameters: batch_size = 16, learning_rate = 3e-5, training_epoch = 4, C = 2, $\alpha = 1$ in training objective.

4.3 Baseline

The sequence labeling model with BERT is used as a baseline model, referring to the method of Devlin et al. (2019). The entire sentence is used as input, and the predicate tokens with ZP argument in the sentence are labeled with a particular ZP class using the BIOES format.

For each argument, we use a different model for each argument type prediction.

4.4 Results

We evaluate the results in terms of precision, recall, and F-score. For example, in case the SBJ argument has "pro", one of the ZP classes, it is defined as follows,

$$Precision^{pro}_{SBJ} = \frac{\text{correct number of predicted "pro" SBJ}}{\text{number of predicted "pro" SBJ}}$$

$$Recall^{pro}_{SBJ} = \frac{\text{correct number of predicted "pro" SBJ}}{\text{number of gold "pro" SBJ}}$$

Model	argument	Arg span accuracy	ZP F1	ZP pre	ZP recall
Baseline	SBJ	-	61.5	62.3	60.8
	OB1	-	58.0	62.3	54.2
	ALL	-	**60.9**	62.2	59.6
QAZP	SBJ	90.8	66.0	66.2	65.8
	OB1	88.5	59.7	60.6	59.0
	ALL	89.3	**64.9**	65.4	64.5

Table 3: Argument span accuracy and ZP identification on NPCMJ for each argument. The row of ALL indiciataes the value for SBJ, OB1 and OB2.

label	Model	F1	pre	recall
pro	baseline	60.8	61.3	60.2
	QAZP	65.1	64.2	66.0
speaker	baseline	62.2	66.2	58.8
	QAZP	65.4	68.7	62.5
hearer	baseline	65.1	60.9	70.0
	QAZP	68.7	65.3	72.7

Table 4: ZP identification on NPCMJ for each ZP class. This values are the result for three arguments.

The same calculation applies to the other arguments and the other labels. The accuracy for required arguments that appear in the sentence is evaluated with the accuracy of whether the prediction span matches exactly with the gold span.

Table 3 and Table 4 show the results of ZP identification on NPCMJ for each argument and each ZP class. In Table 3 and Table 4, QAZP indicates our proposal method, and the baseline is left blank because the argument span is not predicted by the baseline. Compared to the baseline, the proposed method outperformed for each argument and each ZP class. The lower F1 value of ZP identification for OB1 in Table 3 can be attributed to the fact that ZPs occur only about 18% as often in OB1 as in SBJ.

Table 5 shows the result of the Chinese ZP detection. Compared to the baseline, the proposed method outperformed for both argument cases. Although it is not directly comparable with (Aloraini and Poesio, 2020) in that their task definition is slightly different and their targets are only anaphoric ZPs, our model achieves about 80% F1 values, which is higher than their F1 of 68.5%.

4.5 Examples

Figure 1 shows the three prediction examples of the baseline and our proposal model, QAZP. Example 1 is the case when the the QAZP's prediction is correct and the baseline's prediction is incorrect. In this example, the model needs to recognize that the SBJ arguments of the two predi-

[6] https://alaginrc.nict.go.jp/nict-bert/index.html

[7] https://huggingface.co/bert-base-chinese/tree/main

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

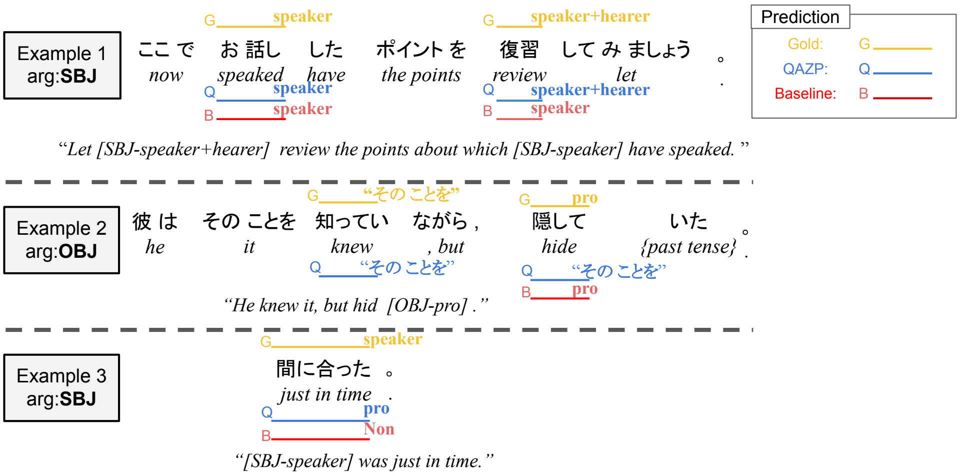

Figure 1: Prediction examples of the baseline and QAZP, our proposal model for three sentences in a Japanese ZP identification task. Each line represents either the prediction of one of the both models, or the Gold data for the argument of a predicate covered by the lines. The first and third examples are predictions for SBJ arguments, and the second example is a prediction result for the OBJ arguments by the baseline and QAZP.

Model	Arg	Arg span accuracy	ZP F1	ZP pre	ZP recall
Baseline	SBJ	-	71.5	72.5	70.5
	ALL	-	**71.4**	72.6	70.3
QAZP	SBJ	88.7	80.6	81.2	80.6
	ALL	88.3	**80.5**	81.0	80.4

Table 5: Argument(Arg) span accuracy and ZP detection on OntoNotes5.0. for "pro" class. The row of ALL indiciataes the value for SBJ, OBJ and IO2 arguments.

cates お 話し "speak" and 復習 "review" are different. While the proposed model predicted a different SBJ argument for each predicate, the baseline predicted the same SBJ item for both predicates. Therefore, we consider that the proposed model is more context-aware than the baseline.

Example 2 is the case when the QAZP's prediction is incorrect and the baseline's prediction is correct. In this example, その ことを "it" is the OBJ argument for the first predicate 知ってい "know", but it is also the referent of the omitted object argument, which is ZP, for the second predicate 隠して "hide". Our model predicted the first predicate 知ってい has その ことを as an OBJ argument. It also predicted the same span その ことを as the OBJ argument for the second predicate 隠して, which results in failing to detect that the OBJ argument is dropped. The reason is that our model predicts an OBJ argument span for each predicate independently. To alleviate such errors, we need to add a constraint to the model that no span in the input sentence can be the argument for more than one predicate at the same time, using Integer Linear Programming as in the method of (Iida and Poesio, 2011).

Example 3 is the case when the predictions of both models are incorrect. In this example sentence, the gold ZP class is the first person "speaker", but it is impossible to identify the ZP without knowing the context before and after the input sentence. We expect our model will capture context information by extending the input unit to multiple sentences instead of a single sentence.

5 Conclusion

We proposed a ZP identification method based on span prediction and evaluate it on Japanese and Chinese datasets. Our model is the first approach to consider ZP detection as a QA task. In experiments, the F1 values of our method were higher than the baseline method using sequence labeling for both Japanese and Chinese.

Future works include to analyze arguments that appeared overtly in tasks such as semantic role labeling. As a setting closer to the real problem, we will use a tagger to create queries instead of using Gold data. The other future work is comparison with a baseline which predicts all arguments at once by sharing the model parameters of BERT as our proposal model. We also consider extending our proposed method to coreference resolution tasks in pro-drop languages.

References

Abdulrahman Aloraini and Massimo Poesio. 2020.

Anaphoric zero pronoun identification: A multilingual approach. In *Proceedings of the Third Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 22–32, Barcelona, Spain (online). Association for Computational Linguistics.

Alastair Butler, Tomoko Hotta, Ruiko Otomo, Kei Yoshimoto, Zhen Zhou, and Hong Zhu. 2012. Keyaki treebank : phrase structure with functional information for japanese. In *In Proceedings of Text Annotation Workshop*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ryu Iida and Massimo Poesio. 2011. A cross-lingual ILP solution to zero anaphora resolution. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 804–813, Portland, Oregon, USA. Association for Computational Linguistics.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020. A unified MRC framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5849–5859, Online. Association for Computational Linguistics.

Masaaki Nagata, Katsuki Chousa, and Masaaki Nishino. 2020. A supervised word alignment method based on cross-language span prediction using multilingual BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 555–565, Online. Association for Computational Linguistics.

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 the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5429–5434, Online. Association for Computational Linguistics.

Shunsuke Takeno, Masaaki Nagata, and Kazuhide Yamamoto. 2015. Empty category detection using path features and distributed case frames. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1335–1340, Lisbon, Portugal. Association for Computational Linguistics.

Longyue Wang, Zhaopeng Tu, Xing Wang, and Shuming Shi. 2019. One model to learn both: Zero pronoun prediction and translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 921–930, Hong Kong, China. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Wei Wu, Fei Wang, Arianna Yuan, Fei Wu, and Jiwei Li. 2020. CorefQA: Coreference resolution as query-based span prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6953–6963, Online. Association for Computational Linguistics.

Bing Xiang, Xiaoqiang Luo, and Bowen Zhou. 2013. Enlisting the ghost: Modeling empty categories for machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 822–831, Sofia, Bulgaria. Association for Computational Linguistics.

Naiwen Xue, Fei Xia, Fu-Dong Chiou, and Marta Palmer. 2005. The penn chinese treebank: Phrase structure annotation of a large corpus. *Natural language engineering*, 11(2):207.