Relation-Guided Few-Shot Relational Triple Extraction

Xin Cong
Institute of Information Engineering, Chinese Academy of Sciences
School of Cyber Security, University of Chinese Academy of Sciences
Beijing, China
congxin@iie.ac.cn

Jiawei Sheng
Institute of Information Engineering, Chinese Academy of Sciences
School of Cyber Security, University of Chinese Academy of Sciences
Beijing, China
shengjiawei@iie.ac.cn

Shiyao Cui
Institute of Information Engineering, Chinese Academy of Sciences
School of Cyber Security, University of Chinese Academy of Sciences
Beijing, China
cuishiyao@iie.ac.cn

Bowen Yu
Institute of Information Engineering, Chinese Academy of Sciences
School of Cyber Security, University of Chinese Academy of Sciences
Beijing, China
yubowen@iie.ac.cn

Tingwen Liu*
Institute of Information Engineering, Chinese Academy of Sciences
School of Cyber Security, University of Chinese Academy of Sciences
Beijing, China
liutingwen@iie.ac.cn

Bin Wang
Xiaomi AI Lab, Xiaomi Inc.
Beijing, China
wangbin11@xiaomi.com

ABSTRACT

In few-shot relational triple extraction (FS-RTE), one seeks to extract relational triples from plain texts by utilizing only few annotated samples. Recent work first extracts all entities and then classifies their relations. Such an entity-then-relation paradigm ignores the entity discrepancy between relations. To address it, we propose a novel task decomposition strategy, Relation-then-Entity, for FS-RTE. It first detects relations occurred in a sentence and then extracts the corresponding head/tail entities of the detected relations. To instantiate this strategy, we further propose a model, RelATE, which builds a dual-level attention to aggregate relation-relevant information to detect the relation occurrence and utilizes the annotated samples of the detected relations to extract the corresponding head/tail entities. Experimental results show that our model outperforms previous work by an absolute gain (18.98%, 28.85% in F1 in two few-shot settings).

CCS CONCEPTS

• **Computing methodologies** → **Information extraction**.

KEYWORDS

Few-shot Learning, Information Extraction, Relational Triple Extraction

ACM Reference Format:
Xin Cong, Jiawei Sheng, Shiyao Cui, Bowen Yu, Tingwen Liu, and Bin Wang. 2022. Relation-Guided Few-Shot Relational Triple Extraction. In *Proceedings of the 45th Int'l ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3477495.3531831

*Corresponding author

SIGIR '22, July 11–15, 2022, Madrid, Spain.

ACM ISBN 978-1-4503-8732-3/22/07.
https://doi.org/10.1145/3477495.3531831

1 INTRODUCTION

Relational Triple Extraction (RTE), as an essential task in Information Extraction, aims to extract entities and classify relations of entity pairs from the unstructured texts. For example, `Hidden Dam is the only storage dam on the Fresno River`, a relational triple (`Hidden Dam`, `Located_in`, `Fresno River`) is extracted from the given text which involves a `Located_in` relation between the head entity, `Hidden Dam`, and the tail entity, `Fresno River`.

Existing methods [6, 8, 14, 22, 24, 26, 32] have achieved great successes by employing the standard fully supervised learning but such a fully supervised paradigm heavily depends on the large-scale human-annotated dataset. As the emerging of knowledge from various domains, new relations, especially those that need professional knowledge to understand, are difficult to be manually annotated on a large scale. Under the circumstance with insufficient annotation resources, existing methods tend to struggle when extracting relational triples of new emerged relations with few annotated samples. Thus, it is critical to study RTE when only few annotated triples are available, a.k.a. few-shot relational triple extraction (FS-RTE).

To solve FS-RTE, previous work [25] follows the conventional entity-then-relation paradigm [13, 27]. It first uses a fully supervised entity extractor trained on large-scale data of known relations to extract all entities and then builds a few-shot relation classifier to classify novel relations of all extracted entity pairs in the few-shot manner [4, 7, 18–21]. However, the **entity discrepancy problem** exists in FS-RTE, which means that entities of the new emerged relations may contain entity types completely different from the known ones since *every relation puts some constraints on the type of head and tail entities* [31]. Hence, the entity extractor, trained on entities of known relations, fails to identify those entities of novel relations. One intuitive way to overcome this problem is to adapt the entity extractor in the few-shot manner by harnessing the few annotated samples. However, it would introduce the **redundant entity pair problem** which means that unrelated entity pairs with non-relation would be extracted unavoidably, misleading the relation classifier.

To solve two shortcomings of the entity-then-relation paradigm, an alternative paradigm is to solve FS-RTE in a unified manner, inspired by Zheng et al. [32]. We could design a unified tagging schema by combining the tags of relation and entity (e.g., *B-Founder-Head*) and then converts FS-RTE into the sequence labeling form. Then, existing few-shot sequence labeling methods [1, 5, 10, 23] can be utilized to solve it. Such a solution seems to be convincing since it models the constraints and dependency between relation and entity based on the tag compositional structures. Unfortunately, this unified paradigm arises the **tag exploding problem**, i.e., combining the tag of relation and entity will introduce too many tags. For example, if using the BIO tagging schema, it will introduce $N \times 4 + 1$ tags for N relations in total (4 for B/I tags of head/tail entity, 1 for O tag). Existing work [19] has indicated that the performance of few-shot models drops significantly when the tag number increases. Hence, the unified paradigm still struggles in FS-RTE.

All aforementioned problems result from the improper task decomposition for FS-RTE. Specifically, FS-RTE can be understood as the joint probability of relational triple extraction $P(h, r, t|x, \mathcal{S})$ where (h, r, t) refers to the relational triple with a head entity h and a tail entity t with their relation r involved in sentence x and $\mathcal{S}$ is the few annotated samples. (1) For the entity-then-relation paradigm, the first solution decomposes the joint probability into conditional probability $P(h, r, t|x, \mathcal{S}) = P(h, t|x)P(r|h, t, x, \mathcal{S})$. Since the entity extraction phase $P(h, t|x)$ does not utilize the samples of novel relations, the entity extractor fails to identify those entities, suffering from the entity discrepancy problem. (2) Introducing the few annotated samples to extract entities $P(h, t|x, \mathcal{S})$ can be viewed as decomposing the joint probability as $P(h, r, t|x, \mathcal{S}) = P(h, t|x, \mathcal{S})P(r|h, t, x, \mathcal{S})$. But it does not consider the relation semantics, resulting in redundant entity pairs. (3) The unified paradigm does not decompose the joint probability $P(h, r, t|x, \mathcal{S})$ so it needs the unified tagging schema to annotate the relation and entity simultaneously, causing the tag exploding problem.

In this paper, we propose a novel task decomposition strategy, **Relation-then-Entity**, for FS-RTE, free from all three problems. As is well known, *the head entity and tail entity should be dependent on a specific relation and relations are usually implied by the context of sentences* [26]. In other words, if one model cannot fully perceive the semantics of relation from sentences, it is unreliable to extract the corresponding head entities and tail entities. Based on this, we decompose the joint probability into $P(h, r, t|x, \mathcal{S}) = P(r|x, \mathcal{S})P(h, t|r, x, \mathcal{S})$: (1) We first judiciously detect the novel relation which a sentence may involve by comparing the sentence semantics with the few annotated samples. (2) Then we extract relation-corresponding head entity and tail entity based on the few annotated triples of the detected relations. We call the former subtask as Relation Detection (RD) and the latter as Relation-specific Entity Extraction (REE). In this manner, only relation-specific entities are extracted avoiding the entity discrepancy and redundant entity pair problem. And through the task decomposition, this paradigm would not use unified tags, which avoids the tag exploding problem.

To instantiate the above paradigm, we propose a model named **RelATE** (Relation-guided Attentive Triple Extractor). It consists of an attentive relation detector for RD, and a relation-guided entity extractor for REE. We further devise a relation-entity hint loss to model the dependency between relation and entity explicitly. We evaluate our method on the public benchmark dataset, *FewRel*. Experimental results show that our proposed method significantly outperforms previous works by a large margin.

2 PROBLEM FORMULATION

We formulate the few-shot relational triple extraction in the typical N-way-K-shot form. For N relations, an small annotated *support set* $\mathcal{S} = \{(r^{(i)}, h^{(i)}, t^{(i)}, x^{(i)})\}_{i=1}^{N \times K}$ is provided with only K triple instances for each relation $r^{(i)}$. Each triple instance is annotated the head entity $h^{(i)}$ and tail entity $t^{(i)}$ in a n-word sentence $x^{(i)} = \{w_1^{(i)}, w_2^{(i)}, \ldots, w_n^{(i)}\}$. FS-RTE aims to extract relational triples from a unlabeled *query set* Q based on the *support set* $\mathcal{S}$. Formally, a $\{\mathcal{S}, Q\}$ pair is called a N-way-K-shot task $\mathcal{T}$.

3 METHODOLOGY

We decompose FS-RTE into two subtasks: 1) Relation Detection which detects the occurred relation in a sentence, then 2) Relation-specific Entity Extraction which extracts the corresponding head/tail entities of the detected relations. Formally, given the training set $\mathcal{D}_{train}$, the overall goal is to predict all triples, i.e., maximizing the joint likelihood of annotated relational triples:

$$\begin{aligned} &\prod_{\{\mathcal{S},Q\}\in\mathcal{D}_{train}} \left[\prod_{x\in Q} P(h, r, t|x, \mathcal{S}) \right] \\ = &\prod_{\{\mathcal{S},Q\}\in\mathcal{D}_{train}} \left[\prod_{x\in Q} P(r|x, \mathcal{S})P(h, t|r, x, \mathcal{S}) \right] \end{aligned} \tag{1}$$

Equation 1 applies the chain rule of probability by exploiting the crucial fact that for a given sentence x, a relation r is implied in the context semantics and any relation r would lead to corresponding head entities h and tail entities t in the sentence. Such a relation-then-entity decomposition strategy are free from all problems claimed in Introduction. The relation-then-entity paradigm can be instantiated in many ways. In this paper, we instantiate it by proposing a novel model, RelATE. Figure 1 illustrates our model and we describe the details of its components below.

3.1 Base Encoder

We first adopt the widely-used BERT [3] to encode the context into real-valued embedding vectors. Given a sentence $x = \{w_1, w_2, \ldots, w_n\}$ from the support set $\mathcal{S}$ or the query set Q, BERT will map all tokens into hidden embedding representations $H \in \mathbb{R}^{n \times d_h}$:

$$H = \{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n\} = \texttt{BERT}(x), \tag{2}$$

where $\mathbf{h}_i \in \mathbb{R}^{d_h}$ is the representation of w_i, d_h is its dimension.

3.2 Attentive Relation Detector

Since relations are implied in the context semantics, we propose an Attentive Relation Detector to integrate the relation-relevant semantic information to detect the relation occurrence. Without knowing the target head/tail entity, we design a dual-level attention: support-level attention and query-level attention to do so.

Support-level Attention is to aggregate the relation-relevant information from the support set to derive the relation prototype.

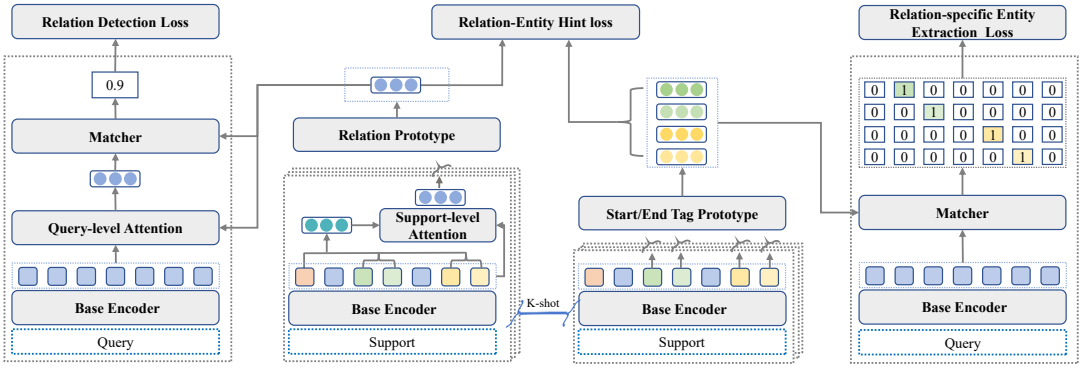

Figure 1: Architecture of our proposed RelATE.

Since the sentence embedding contains the context semantics which may imply a relation and the head/tail entity embedding contain the entity type constraints of a relation, we first gather these three embeddings to derive the raw relation semantic representation. Given the k-th support instance $H^k \in \mathbb{R}^{n \times d_h}$ in the support set, we use the "[CLS]" embedding $\mathbf{h}^k_{cls}$, as the sentence embedding and use the average of the embeddings of the entity span to get the head embedding $\mathbf{h}^k_{head}$ and tail embedding $\mathbf{h}^k_{tail}$ respectively. Then, the raw relation representation $\mathbf{h}^k_{rel}$ is calculated as follows:

$$\mathbf{h}^k_{rel} = \mathbf{h}^k_{cls} + \mathbf{h}^k_{head} + \mathbf{h}^k_{tail} \tag{3}$$

Next, since relation semantics usually scatter in the sentence context, we use the raw relation representation $\mathbf{h}^k_{rel}$ to sift out the relation-specific information through the sentence attentively.

$$\hat{\mathbf{h}}^k_{rel} = \texttt{softmax}(\mathbf{h}^k_{rel}\, H^{k\mathrm{T}}) H^k \tag{4}$$

where $\hat{\mathbf{h}}^k_{rel}$ is the refined relation representation.

Getting the refined relation representation for each support instance, we calculate the prototype for each relation by averaging all the refined representations of that relation in the support set $\mathcal{S}$:

$$\mathbf{c}^{rel}_i = \frac{1}{K} \sum_{k=1}^{K} \hat{\mathbf{h}}^k_{rel}, \quad i = 1, 2, \cdots, N \tag{5}$$

where $\mathbf{c}^{rel}_i$ denotes the relation prototype for relation i.

Query-level Attention is to aggregate the relation-specific information in the query set based on the relation prototypes.

Given a query instance $H^q \in \mathbb{R}^{n \times d_h}$ in the query set, we use each relation prototype $\mathbf{c}^{rel}_i$ to gather relation-specific semantics in the query instance attentively:

$$\hat{\mathbf{h}}^{(q,i)} = \texttt{softmax}(\mathbf{c}^{rel}_i\, H^{q\mathrm{T}}) H^q, \quad i = 1, 2, \cdots, N \tag{6}$$

Finally, getting the relation prototype $\mathbf{c}^{rel}_i$ and the corresponding relation-specific semantic representations $\hat{\mathbf{h}}^{(q,i)}$ of the query instance, a matcher is employed to calculate the probability $p^{rel}_{q,i}$ to detect whether the query instance involves the relation:

$$p^{rel}_{q,i} = \texttt{Matcher}(\hat{\mathbf{h}}^{(q,i)}, \mathbf{c}^{rel}_i), \quad i = 1, 2, \cdots, N \tag{7}$$

The matcher is detailed in Section 3.4.

3.3 Relation-guided Entity Extractor

If a relation is detected, the relation-guided entity extractor is employed to extract the corresponding head/tail entities based on the support instances of the detected relation.

We adopt the span tagging schema [22, 24] to annotate the start/end position of the entity. Assume that relation r is detected, we first select the support instances of relation r from the original support set $\mathcal{S}$ to get relation-specific support set $\mathcal{S}_r$. Then we calculate prototypes of the start/end tags of head/tail entity by averaging all the token representations with that label in $\mathcal{S}_r$:

$$\mathbf{c}^{pos}_i = \frac{1}{|\mathcal{S}_r(i)|} \sum_{w \in \mathcal{S}_r(i)} \mathbf{h}, \quad i \in \{s_h, e_h, s_t, e_t\} \tag{8}$$

where s_h, e_h, s_t, e_t refer to the start/end position of head and tail entity respectively, $\mathbf{c}_i$ denotes the prototype for each span tag, $\mathcal{S}_r(i)$ refers to the token set containing all words in $\mathcal{S}_r$ with tag i, $\mathbf{h}$ represents the corresponding representation of token w in $\mathcal{S}_r(i)$, and $|\cdot|$ is the number of set elements.

Getting the tag prototypes, we employ matcher again to get the probability $p^{ent}_{q,i,j}$ for each token of query to tag head/tail entities:

$$p^{ent}_{q,i,j} = \texttt{Matcher}(\mathbf{h}^q_j, \mathbf{c}^{pos}_i), i \in \{s_h, e_h, s_t, e_t\} \tag{9}$$

where $\mathbf{h}^q_j \in H^q$ and $j = 1, \cdots, n$.

3.4 Matcher

Matcher, shared by two above modules, aims to match the prototype and the query representation to derive the probability by measuring their similarity. It can be implemented variously. In this paper, we implement it as a three-layer neural network.

Given the prototype $\mathbf{c}$ (relation prototype or span tag prototype) and the query representation $\mathbf{h}$ (relation-specific semantic representation or token representation of query instance), we first construct the input $\mathbf{I}$ of the network as follows:

$$\mathbf{I} = [\mathbf{h}; \mathbf{c}; \mathbf{h} - \mathbf{c}; \mathbf{h} + \mathbf{c}; \mathbf{h} \otimes \mathbf{c}] \tag{10}$$

where $\otimes$ means element-wise product, $[\cdot\,;\,\cdot]$ refers to the concatenation operation. We use a two-layer CNN and a followed linear layer to map input $\mathbf{I}$ into a single value to act as the similarity score between the prototype and the query representation.

3.5 Relation-Entity Hint Loss

As is well known, each relation usually puts some constraints on the types of its head and tail entity while the specific types of head entity and tail entity also imply their relation [12, 26, 31]. To model this dependency explicitly to promote two subtasks, we further introduce a relation-entity hint loss.

Getting the relation prototype $\mathbf{c}^{rel}$ and its corresponding entity prototypes $\mathbf{c}^{pos}_{s_h}, \mathbf{c}^{pos}_{e_h}, \mathbf{c}^{pos}_{s_t}, \mathbf{c}^{pos}_{e_t}$, we first utilize the mean pooling over the start/end tag prototype of the head/tail entity respectively to get the entity prototypes which contains the information of the entity types:

$$\begin{aligned} \mathbf{c}^{head} &= \texttt{MeanPool}([\mathbf{c}^{pos}_{s_h}; \mathbf{c}^{pos}_{e_h}]) \\ \mathbf{c}^{tail} &= \texttt{MeanPool}([\mathbf{c}^{pos}_{s_t}; \mathbf{c}^{pos}_{e_t}]) \end{aligned} \tag{11}$$

Then, we minimize the mean square error loss to make the head/tail entity prototypes and relation prototype closer:

$$\mathcal{L}_{hint} = \texttt{MSE}(W_{hint}[\mathbf{c}^{head}; \mathbf{c}^{tail}], \mathbf{c}^{rel}) \tag{12}$$

where $W_{hint} \in \mathbb{R}^{d_h \times 2d_h}$ is a learnable parameter. In this way, the relation prototype and the head/tail prototype can guide the learning process for each other mutually by modeling the dependency of the relation and its head/tail entity types explicitly.

3.6 Objective Function

To train our model, we use the negative log-likelihood loss of two subtasks. We denote them as $\mathcal{L}_{rel}$ and $\mathcal{L}_{entity}$ respectively. Adding our relation-entity hint loss, the final objective function is:

$$\mathcal{L} = \mathcal{L}_{rel} + \mathcal{L}_{entity} + \alpha \mathcal{L}_{hint} \tag{13}$$

where α is the hyperparameter weight.

4 EXPERIMENT

4.1 Experimental Setup

Dataset: Following previous work [25], we conduct experiments on the public benchmark dataset *FewRel* [9], which releases 80 relations and each relation owns 700 triple instances in total. Following Yu et al. [25], we use 50 relations as the training set, 15 relations as the development set and the rest 15 relations as the test set. Note that the relations of the training/dev/test set are non-overlapping. For the detailed statistics of FewRel, please refer to Appendix A.

Evaluation: We follow the metrics in previous work [25] to evaluate the model performance in 5-way-5-shot and 10-way-10-shot settings. Concretely, a relational triple is marked correct if and only if the spans of the head and tail entity are correctly identified and the associated relation is also predicted correctly. We adopt the standard micro F1 score to evaluate the results and report the averages and standard deviations over 5 randomly initialized runs.

Hyperparameter: We use *AdamW* optimizer to train our model with the learning rate of 1×10^{-5} for BERT and 1×10^{-3} for others. The maximum sentence length is set as 128. The coefficient of the relation-entity hint loss is set as 0.2. The filter size of CNN is set as $[3, 3]$ and the channel number is set as $[8, 4]$. All the hyperparameters are tuned on the dev set by grid search. Due to the space limitation, more implementation details are reported in Appendix B including detailed hyperparameter settings, training strategy and computation architecture.

Table 1: Main results: F1 scores (10^{-2}) of different models on the FewRel test set. Bold marks the highest number among all models. Underline marks the second-highest number, and ± marks the standard deviation. * marks statistically significant improvements over the best baseline with $p < 0.01$ under a bootstrap test.

Model	5-Way-5-Shot	10-Way-10-Shot
FT-BERT	4.71 ± 0.96	2.94 ± 0.77
CasRel	2.11 ± 1.03	2.04 ± 0.52
MatchNet	10.13 ± 0.43	4.40 ± 1.02
RelationNet	9.91 ± 0.28	6.65 ± 0.33
Proto	14.18 ± 0.25	6.53 ± 0.60
Proto+Att	18.20 ± 0.46	10.55 ± 0.31
MPE	23.34 ± 0.79	12.08 ± 0.83
FS-MPE	29.27 ± 0.89	20.72 ± 0.92
WPZ	23.61 ± 0.14	23.28 ± 0.33
L-TapNet+CDT	28.23 ± 0.76	26.40 ± 0.31
StructShot	25.94 ± 3.06	20.28 ± 2.43
PA-CRF	<u>34.14</u> ± 0.30	<u>30.44</u> ± 1.15
RelATE	**42.32** ± 0.53	**40.93** ± 0.35

Baseline: We choose several baselines which can be categorized into three paradigms. (1) *Standard Supervised Paradigm* follows the standard fully supervised learning, including FT-BERT [25], CasRel [22]. (2) *Entity-then-Relation Paradigm* first uses a conventional fully-supervised entity extractor to identify all entities and then uses few-shot classifier to classify their relation, including: MatchNet [21, 25], Proto [19, 25], Proto+Att [7, 25], Relation [20, 25], MPE [25]. In addition, FS-MPE is a variant of MPE, which replaces the fully-supervised entity extractor with a few-shot entity extractor using the prototypical network [19]. (3) *Unified Paradigm* adapts several few-shot sequence labeling methods into FS-RTE, including: WPZ [5], L-TapNet+CDT [10], StructShot [23], PA-CRF [1].

4.2 Evaluation Results

Main Results. Table 1 reports the results of our model against other baseline models on the FewRel test set. It can be seen that our method, RelATE, significantly outperforms all competitive baseline models and achieves the state-of-the-art in two few-shot settings.

Comparison with Standard Supervised model. Obviously, all few-shot models exceed FT-BERT and CasRel, proving that the standard fully supervised paradigm is incapable of solving FS-RTE.

Comparison with Entity-then-Relation models. Using conventional entity extractor, the previous work MPE performs lower than our RelATE with a significant gap (18.98% and 28.85% in two settings respectively). Utilizing the support set to extract entities in the few-shot manner, FS-MPE reaches higher performance compared to MPE. But it still cannot catch up with our RelATE with a huge gap (13.05% and 20.21% in two settings respectively). It shows that our relation-then-entity paradigm is more capable of solving FS-RTE than entity-then-relation paradigm.

Comparison with Unified models. Utilizing the few-shot sequence labeling methods to solve FS-RTE, all unified models outperform MPE, indicating that the few-shot sequence labeling methods can

Table 2: Bottleneck analysis: Precision, Recall and F1 score (10^{-2}) are reported on the FewRel test set.

Model	5-Way-5-Shot	10-Way-10-Shot
	P / R / F1	P / R / F1
MPE	21.33 / 35.40 / 24.75	16.81 / 24.31 / 19.87
FS-MPE	36.61 / 48.86 / 40.14	32.47 / 46.17 / 35.58
PA-CRF	54.43 / 52.82 / 53.61	54.29 / 41.66 / 47.14
RelATE	62.02 / 60.35 / 61.18	59.34 / 56.39 / 57.82

extract novel entities since the unified tagging schema model the dependency between relation and entity. However, the best unified model, PA-CRF, still cannot exceed RelATE, which demonstrates the effectiveness of the relation-then-entity paradigm.

Table 1 reports the results of our model against other baseline models on the FewRel test set. It can be seen that our method, RelATE, significantly outperforms all competitive baseline models and achieves the state-of-the-art in both two few-shot scenarios.

Comparison with Standard Supervised model. Obviously, all few-shot-based models exceed FT-BERT and CasRel in two settings, which powerfully proves that the standard supervised paradigm is incapable of solving FS-RTE.

Comparison with Entity-then-Relation models. Using conventional entity extractor, the previous work MPE performs lower than our RelATE with a significant gap (18.98% and 28.85% in two settings respectively). Utilizing the support set to extract entities in the few-shot manner, FS-MPE reaches higher performance compared to MPE. But it still cannot catch up with our RelATE with a huge gap (13.05% and 20.21% in two settings respectively). It indicates that our proposed relation-then-entity paradigm is more capable of solving FS-RTE than entity-then-relation paradigm.

Comparison with Unified models. Utilizing the few-shot sequence labeling methods to solve FS-RTE, all unified models outperform previous work, MPE, indicating that the few-shot sequence labeling methods can extract novel entities since the unified tagging schema model the dependency between relation and entity. However, the best unified model, PA-CRF, still cannot exceed our RelATE, which demonstrates that relation-then-entity paradigm performs better than unified paradigm.

Bottleneck Analysis. To investigate the bottleneck of the entity-then-relation and unified paradigm, we choose the paradigmatic models (MPE, FS-MPE, PA-CRF including RelATE) and evaluate if the spans of the head/tail entities are correct without considering their relation. We conduct experiments on the FewRel test set and report results in Table 2.

Firstly, for the entity-then-relation paradigm, we find that MPE and FS-MPE suffer from the poor entity performance. Concretely, MPE, based on standard supervised entity extractor, owns extremely poor entity extraction performance (24.75% and 19.87%), which powerfully proves that the standard entity extractor cannot extract entities of novel relations due to the entity discrepancy problem. FS-MPE, extracting entities in the few-shot way, has obvious improvements against MPE. It uses the support set to improve the entity extraction performance to some extent but experiences extremely lower precision (36.61% and 32.47%) compared with its recall. Without considering the relation semantics, many redundant entity pairs with non-relation are extracted wrongly, hurting its precision. Hence, its final performance is limited due to the cascading errors. By contrast, the entity extraction performance of RelATE has a absolute gap compared with MPE and FS-MPE, validating that our relation-then-entity paradigm is free from the entity discrepancy and redundant entity pair problem.

Secondly, as the best baseline model of the unified paradigm, PA-CRF achieves higher entity extraction performance than entity-then-relation models, but still cannot exceed our RelATE. Especially, suffering the tag exploding problem, the performance gap between PA-CRF and RelATE enlarges (from 7.57% in 5-way-5-shot to 10.68% in 10-way-10-shot) when the number of relations (N-way) increases. Theoretically, given N relations, there exist total $N \times 4 + 1$ tags (N for relations, 4 for B/I tags of head/tail entity, 1 for O tag) for BIO schema (e.g., $41 = 10 \times 4 + 1$ tags in the 10-way-10-shot setting). Such a tag exploding issue makes PA-CRF struggle to distinguish correct tags since existing few-shot methods experience performance degradation with the number of tags increasing [1, 19]. By contrast, thanks to the reasonable task decomposition, our RelATE only has N+4+1 tags (N for relations, 4 for start/end tag of head/tail entity, 1 for other tokens). In the 10-way-10-shot setting, it only has $15 = 10 + 4 + 1$ tags, which is much less than the unified paradigm (41 tags), alleviating the tag exploding problem.

Overall, we can draw the conclusion that (1) Entity-then-relation paradigm struggles due to the entity discrepancy and redundant entity pair problems. (2) Unified paradigm can alleviate these problems but performs poorly due to the tag exploding problem. (3) Thanks to the reasonable task decomposition Relation-then-Entity, our RelATE can avoid all problems above and achieve better performance.

We further conduct ablation study and case study to validate the strength of our proposed RelATE. Details are listed in Appendix D and E.

5 CONCLUSION

We propose a novel task decomposition strategy, Relation-then-Entity, for few-shot relational triple extraction and further instantiate this strategy as a novel model RelATE. Different from previous work, we first use an attentive relation detector to detect the relation occurrence and then utilize the support set of detected relations to extract corresponding head/tail entities. Therefore, our model can overcome the entity discrepancy, redundant entity pair and tag exploding problem. We conduct extensive experiments on the FewRel dataset, to validate the effectiveness of our proposed decomposition strategy. Experimental results show our model outperforms state-of-the-art baselines over different scenarios.

ACKNOWLEDGEMENTS

We would like to thank all reviewers for their insightful comments and suggestions. This work is supported by the National Key Research and Development Program of China (grant No.2021YFB3100600), the Strategic Priority Research Program of Chinese Academy of Sciences (grant No.XDC02040400) and the Youth Innovation Promotion Association of CAS (Grant No. 2021153).

REFERENCES

[1] Xin Cong, Shiyao Cui, Bowen Yu, Tingwen Liu, Wang Yubin, and Bin Wang. 2021. Few-Shot Event Detection with Prototypical Amortized Conditional Random Field. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*.

[2] Xin Cong, Bowen Yu, Tingwen Liu, Shiyao Cui, Hengzhu Tang, and Bin Wang. 2020. Inductive Unsupervised Domain Adaptation for Few-Shot Classification via Clustering. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2020, Ghent, Belgium, September 14-18, 2020, Proceedings, Part II*.

[3] 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, NAACL-HLT*.

[4] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML*.

[5] Alexander Fritzler, Varvara Logacheva, and Maksim Kretov. 2019. Few-shot classification in Named Entity Recognition Task.

[6] Tsu-Jui Fu, Peng-Hsuan Li, and Wei-Yun Ma. 2019. GraphRel: Modeling Text as Relational Graphs for Joint Entity and Relation Extraction. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL*.

[7] Tianyu Gao, Xu Han, Zhiyuan Liu, and Maosong Sun. 2019. Hybrid Attention-Based Prototypical Networks for Noisy Few-Shot Relation Classification. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI*.

[8] Pankaj Gupta, Hinrich Schütze, and Bernt Andrassy. 2016. Table Filling Multi-Task Recurrent Neural Network for Joint Entity and Relation Extraction. In *26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, COLING*.

[9] Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A Large-Scale Supervised Few-shot Relation Classification Dataset with State-of-the-Art Evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP*.

[10] Yutai Hou, Wanxiang Che, Yongkui Lai, Zhihan Zhou, Yijia Liu, Han Liu, and Ting Liu. 2020. Few-shot Slot Tagging with Collapsed Dependency Transfer and Label-enhanced Task-adaptive Projection Network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

[11] Rohit J. Kate and Raymond J. Mooney. 2010. Joint Entity and Relation Extraction Using Card-Pyramid Parsing. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning, CoNLL*.

[12] Mitchell Koch, John Gilmer, Stephen Soderland, and Daniel S. Weld. 2014. Type-Aware Distantly Supervised Relation Extraction with Linked Arguments. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

[13] Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics, ACL*.

[14] Makoto Miwa and Mohit Bansal. 2016. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL*.

[15] Makoto Miwa and Yutaka Sasaki. 2014. Modeling Joint Entity and Relation Extraction with Table Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP*.

[16] Tsendsuren Munkhdalai and Hong Yu. 2017. Meta Networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML*.

[17] Tapas Nayak and Hwee Tou Ng. 2020. Effective Modeling of Encoder-Decoder Architecture for Joint Entity and Relation Extraction. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI*.

[18] Alex Nichol, Joshua Achiam, and John Schulman. 2018. On First-Order Meta-Learning Algorithms. *CoRR* abs/1803.02999 (2018).

[19] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*. 4077–4087.

[20] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. 2018. Learning to Compare: Relation Network for Few-Shot Learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*.

[21] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. In *Advances in neural information processing systems*. 3630–3638.

[22] Zhepei Wei, Jianlin Su, Yue Wang, Yuan Tian, and Yi Chang. 2020. A Novel Cascade Binary Tagging Framework for Relational Triple Extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL*.

[23] Yi Yang and Arzoo Katiyar. 2020. Simple and Effective Few-Shot Named Entity Recognition with Structured Nearest Neighbor Learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

[24] Bowen Yu, Zhenyu Zhang, Xiaobo Shu, Tingwen Liu, Yubin Wang, Bin Wang, and Sujian Li. 2020. Joint Extraction of Entities and Relations Based on a Novel Decomposition Strategy. In *The 24th European Conference on Artificial Intelligence, ECAI*.

[25] Haiyang Yu, Ningyu Zhang, Shumin Deng, Hongbin Ye, Wei Zhang, and Huajun Chen. 2020. Bridging Text and Knowledge with Multi-Prototype Embedding for Few-Shot Relational Triple Extraction. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING*.

[26] Yue Yuan, Xiaofei Zhou, Shirui Pan, Qiannan Zhu, Zeliang Song, and Li Guo. 2020. A Relation-Specific Attention Network for Joint Entity and Relation Extraction. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI*.

[27] Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2002. Kernel Methods for Relation Extraction. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing, EMNLP*.

[28] Daojian Zeng, Haoran Zhang, and Qianying Liu. 2020. CopyMTL: Copy Mechanism for Joint Extraction of Entities and Relations with Multi-Task Learning. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI*.

[29] Xiangrong Zeng, Shizhu He, Daojian Zeng, Kang Liu, Shengping Liu, and Jun Zhao. 2019. Learning the Extraction Order of Multiple Relational Facts in a Sentence with Reinforcement Learning. 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*.

[30] Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, and Jun Zhao. 2018. Extracting Relational Facts by an End-to-End Neural Model with Copy Mechanism. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL*.

[31] Zhenyu Zhang, Xiaobo Shu, Bowen Yu, Tingwen Liu, Jiapeng Zhao, Quangang Li, and Li Guo. 2020. Distilling Knowledge from Well-Informed Soft Labels for Neural Relation Extraction. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI*.

[32] Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. 2017. Joint Extraction of Entities and Relations Based on a Novel Tagging Scheme. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL*.

A DATASET STATISTICS

	Training Set	Dev Set	Test Set
# Rel.	50	15	15
# Sent./Rel.	700	700	700
# Triple.	35000	10500	10500
# Tok./Sent.	25.02	24.65	25.04

Table 3: Statistics of FewRel Dataset.

Table 3 lists the statistics of FewRel dataset containing the number of relation type (#Rel.), the number of sentence per relation (# Sent./Rel.), the number of relational triple (# Triple.) and the average number of token per sentence (# Tok./Sent.) for train/dev/test set.

B IMPLEMENTATION DETAILS

B.1 Hyperparameter Settings

We employ the *BERT-BASE-UNCASED* [3] as the base encoder. *AdamW* optimizer is used to train our model with the learning rate of $1e-5$ for BERT encoder and $1e-3$ for other modules. We also use the *LinearScheduleWithWarmup* with 100 steps to warm up our model. The maximum sentence length is set as 128. Dropout rate is set as 0.1. Gradient clip-norm is set as 10. The coefficient weight of the relation-entity hint loss is set as 0.2. The filter size of CNN is set as $[3, 3]$ and the channel number is set as $[8, 4]$. All the hyper-parameters are tuned on the validation set by grid search.

B.2 Training Strategy

We follow the widely used few-shot training paradigm, *Episodic Training* [21], to mimic N-way-K-shot scenario in the training phase. In each epoch, we randomly sample N relation types from the training set and each relation type randomly samples K instances as the support set and other M instances as the query set. Our model is trained with 50,000 epochs on the training set and evaluated with 3,000 epochs on the test set following the episodic paradigm.

B.3 Computation Architecture

We run all experiments using PyTorch 1.7.1 on the Nvidia Tesla V100 GPU, Intel(R) Xeon(R) Silver 4110 CPU with 256GB memory on Red Hat 4.8.3 OS.

C RELATED WORK

Relational triple extraction aims to extract relational triples from plain texts. Early works [11, 15] are heavily dependent on manual-designed features. Recently, as neural networks show the effectiveness of extracting features automatically, many neural-based methods are proposed. These methods can be roughly divided into three categories: (1) Table-filling methods [6, 8] build a table for sentences and fill the table cells with entity or relation tags in a specific order. (2) Tagging methods [22, 24, 26, 32] elaborate design a tagging schema to construct connections between entities and relations. (3) Seq2Seq methods [17, 28–30] try to generate the relational triples in a sequence directly. However, all these methods are data-hungry. They require a large amount of labeled data for training while new emerged relations usually have a handful of labeled data. In such a few-shot scenario, the performance of these methods would drop dramatically.

Inspired by the development of Few-Shot Learning [2, 4, 16, 18–21], Yu et al. [25] makes the first attempt to solve few-shot relational triple extraction (FS-RTE). It follows the conventional entity-then-relation paradigm [13, 27] and simply combines a standard supervised entity extractor with a prototype-based few-shot relation classifier. Its standard supervised entity extractor fails to recognize the head/tail entities of novel relations due to the entity discrepancy problem. Replacing the supervised entity extractor with a few-shot entity extractor will still suffer from the redundant entity pair with no relation.

An alternative way is formalizing FS-RTE as a few-shot sequence labeling task. Recently, many efforts have been devoted to solving the few-shot sequence labeling task. Fritzler et al. [5] applied the prototypical networks [19] with vanilla CRF to solve few-shot NER. Hou et al. [10] proposed a collapsed dependency transfer mechanism to learn label dependency of abstract labels. Yang and Katiyar [23] reconstruct the existing NER datasets to adapt to the few-shot NER task for better evaluation. Cong et al. [1] designs a prototypical amortized CRF to estimate the tag-specific transition scores based on the associate label prototypes in the unified manner. All these works can be used to solve FS-RTE but will introduce the tag exploding problem, causing lower performance.

D ABLATION STUDY

Our model contains four main components: the relation-entity hint loss (Hint), the dual-level attention (DualAtt), Matcher, and the span tagging schema (Span). To study the contribution of each component in our model, we run the ablation study on the dev set and the test set of FewRel. From these ablations (see Table 4), we find that: 1) - Hint: To prove the contribution of the relation-entity hint loss, we remove it and train RelATE only with negative log-likelihood loss. Results show that without the relation-entity hint loss, the performance drops slightly. It proves that the relation-entity hint loss can model the dependency between relations and entities to boost the performance. 2) - DualAtt: To study if the dual-level attention is helpful to improve the performance of the relation detection subtask, we remove it and use the [CLS] embedding as the relation representation (replacing $\hat{h}_{rel}$ in Equation 5) directly. The performance degradation shows that without dual-level attention, the relation detector cannot aggregate the relation-specific information to detect the relation correctly. 3) - Matcher: To verify that our matcher contributes to capturing the similarity between prototypes and queries, we remove it and use Euclidean distance [19] instead. The performance decreases significantly, which indicates that the neural-based matcher is good at measuring the similarity between prototypes and queries. 4) - Span: To access the performance influence of span tagging schema, we replace it with the widely-used BIO tagging schema and observed the performance drops.

Table 4: Ablation study: F1 scores (10^{-2}) are reported on both the dev set and the test set.

Model	5-Way-5-Shot		10-Way-10-Shot	
	Dev	Test	Dev	Test
RelATE	48.15	42.32	47.46	40.93
- Hint	46.47	41.55	46.11	39.80
- DualAtt	44.70	40.82	45.44	37.91
- Matcher	35.91	36.31	33.30	33.38
- Span	36.75	37.97	34.56	33.72

(1) *His greatest success was in the leading role in "Peter Grimes", an opera by Benjamin Britten.*	
MPE	(Benjamin Britten, *NULL*, NULL)
RelATE	(Benjamin Britten, *composer*, Peter Grimes)
(2) *South Dakota governor, Dennis Daugaard urged residents in Dakota Dunes to evacuate.*	
FS-MPE	(South Dakota, *head of government*, Dennis Daugaard) (Dakota Dunes, *head of government*, Dennis Daugaard)
RelATE	(South Dakota, *head of government*, Dennis Daugaard)
(3) *The Ohio Connecting Railroad Bridge crosses the Ohio River at the island.*	
PA-CRF	(Ohio Connecting Railroad, *cross*, Ohio River)
RelATE	(Ohio Connecting Railroad Bridge, *cross*, Ohio River)

Figure 2: Cases from FewRel dataset. Orange and blue colors marks the ground truth head entity and tail entity respectively. $(\cdot, \cdot, \cdot)$ refers to the extracted triples and "NULL" means no entity or relation is extracted.

E CASE STUDY

To demonstrate how our proposed decomposition strategy performs better than others intuitively, we use three cases to compare the predictions between our model against three strong baselines. From Figure 2, we can observe that: (1) In Case 1, MPE fails to extract

the tail entity Peter Grimes of relation composer. Because the tail entity of the relation composer should be a Music but such an entity constraint is unseen in the training set. The entity discrepancy causes MPE to fail to extract it. By contrast, since RelATE utilizes the support set of relation composer, it succeeds in recognizing Peter Grimes. (2) In Case 2, FSMPE extracts a redundant entity Dakota Dunes, resulting in an incorrect triple. This phenomenon indicates that the redundant entity pairs will be extracted wrongly without the guide of relation semantics. Since RelATE aggregates the relation semantics to guide the entity extraction, this problem can be avoided. (3) In Case 3, PA-CRF recognizes the boundary of Ohio Connecting Railroad Bridge wrongly. That is because the unified paradigm will introduce too many tags, making it difficult to learn the dependency of so many tags. Thus, the tag sequence could be wrongly decoded. Thanks to the proper task decomposition, RelATE does not introduce extra entity tags so it could extract the head entity correctly.