Generative Meta-Learning for Zero-Shot Relation Triplet Extraction

Wanli Li
Huazhong Agricultural University
Wuhan, Hubei, China
liwanli@mail.hzau.edu.cn

Tieyun Qian
Wuhan University
Wuhan, Hubei, China
qty@whu.edu.cn

Yi Song
Huazhong Agricultural University
Wuhan, Hubei, China
songee@webmail.hzau.edu.cn

Zeyu Zhang*
Huazhong Agricultural University
Wuhan, Hubei, China
zhangzeyu@mail.hzau.edu.cn

Jiawei Li*
Huazhong Agricultural University
Wuhan, Hubei, China
lijw@mail.hzau.edu.cn

Zhuang Chen
Central South University
Changsha, Hunan, China
zhchen18@foxmail.com

Lixin Zou
Wuhan University
Wuhan, Hubei, China
zoulixin@whu.edu.cn

Abstract

Zero-shot Relation Triplet Extraction (ZeroRTE) aims to extract relation triplets from texts containing unseen relation types. This capability benefits various downstream information retrieval (IR) tasks. The primary challenge lies in enabling models to generalize effectively to unseen relation categories. Existing approaches typically leverage the knowledge embedded in pre-trained language models to accomplish the generalization process. However, these methods focus solely on fitting the training data during training, without specifically improving the model's generalization performance, resulting in limited generalization capability. For this reason, we explore the integration of bi-level optimization (BLO) with pre-trained language models for learning generalized knowledge directly from the training data, and propose a generative meta-learning framework which exploits the 'learning-to-learn' ability of meta-learning to boost the generalization capability of generative models.

Specifically, we introduce a BLO approach that simultaneously addresses data fitting and generalization. This is achieved by constructing an upper-level loss to focus on generalization and a lower-level loss to ensure accurate data fitting. Building on this, we subsequently develop three generative meta-learning methods, each tailored to a distinct category of meta-learning. Extensive experimental results demonstrate that our framework performs well on the ZeroRTE task. Our code is available at https://github.com/leeworry/TGM-MetaLearning.

*Corresponding authors.

SIGIR '25, Padua, Italy.

ACM ISBN 979-8-4007-1592-1/2025/07
https://doi.org/10.1145/3726302.3729988

CCS Concepts

• **Computing methodologies** → **Information extraction**; **Natural language generation**.

Keywords

Relation Triplet Extraction, Zero-shot Learning, Meta-learning, Pre-trained Language Models

ACM Reference Format:

Wanli Li, Tieyun Qian, Yi Song, Zeyu Zhang, Jiawei Li, Zhuang Chen, and Lixin Zou. 2025. Generative Meta-Learning for Zero-Shot Relation Triplet Extraction. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25), July 13–18, 2025, Padua, Italy*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3726302.3729988

1 Introduction

The purpose of relation triplet extraction (RTE) is to extract triplets of the form (head entity, tail entity, relation label) from unstructured text. For example, given the sentence "Washington is the capital of the U.S.A." in Fig. 1 (a), RTE aims to extract the triplet (head entity: Washington, tail entity: the U.S.A., relation: capital of). RTE can transform unstructured texts into structured knowledge, which is valuable for various downstream information retrieval IR tasks [10, 19].

Existing studies followed standard deep learning and have achieved impressive performance in supervised relation extraction (RE) [32, 34] or semi-supervised RE [20] with sufficient or limited labeled data. Their training process can be formulated in Equation 1.

$$w_\theta^* = \arg\min_{w \in \Psi} \frac{1}{N} \sum_{i=1}^{N} \ell(w(x_i), y_i), \tag{1}$$

where w_θ^* is the optimal weight vector w, parameterized by θ, that minimizes the loss function, x_i and y_i is the input feature vector and corresponding golden label for i-th sample separately, and ℓ can be cross-entropy loss for classification, or negative log-likelihood (NLL) for language models. Although successful in many areas, standard deep learning methods often lack sufficient data in real-world

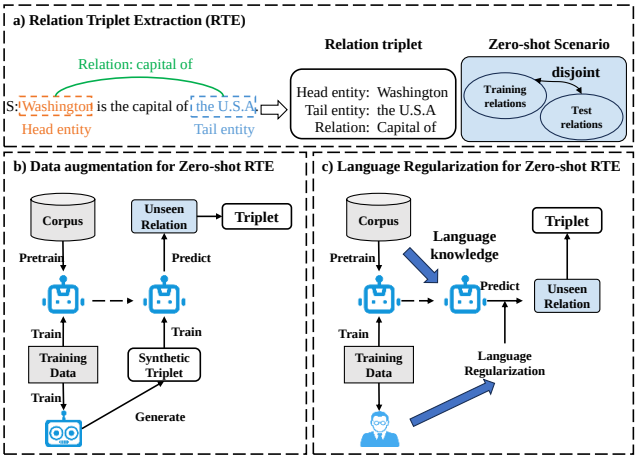


Figure 1: An illustration of the RTE task and the difference between existing methods for ZeroRTE.

scenarios. This scarcity limits the training of models on less common relations, making the enhancement of model generalization to handle unseen relations a critical research challenge.

As a result, many researchers are focusing on the generalization problem in RE, with zero-shot RTE being the most challenging. Along these lines, Chia et al. [2] firstly introduced the ZeroRTE task to extract unseen relations and their corresponding entities. The purpose is to extract relations and entities that have not been seen before, based on training data [1, 15].

Existing ZeroRTE methods treat the generalization issue as a *distribution shift problem*, operating under the assumption that the distributions of the training and test datasets are distinct. As illustrated in Fig. 1 (b, c), existing methods are generally classified into two approaches. One category involves data augmentation techniques that leverage the text generation capabilities of language models to create training data for unseen relations. The other approach involves using regularization methods to constrain model capacity and retain semantic knowledge, thus enhancing its generalization performance.

Based on synthetic data of unseen relations, models can adapt to the new distribution, the process can be formulated as follows:

$$w_\theta^* = \arg\min_{w \in \Psi} \frac{1}{N} \sum_{i \in D^{syn}} \ell(w(x_i), y_i), \tag{2}$$

where D^{syn} is the synthetic data set generated by language models. For example, RelationPrompt adopts a pre-trained generative model (GPT-2) to generate synthetic samples for unseen relations. KBPT uses a prompt model to synthesize training samples [6].

The data augmentation approach has two primary limitations. Firstly, the model's performance is constrained by the quality of the generated text, which can be of low quality since the text generator operates without ever having seen real samples, making the output quality unpredictable and potentially unreliable. Secondly, this method lacks efficiency. The model requires retraining to accommodate new classes, which is time-consuming [37].

Language regularization methods design constraints to force models to retain generic semantic knowledge in order to adapt to unseen relation categories. For example, PCRED employs a knowledge-based prompt to increase performance in ZeroRTE [13]. Kim et al. [11] transforms ZeroRTE into a template-filling paradigm, leveraging the model's pre-existing knowledge to effectively generalize to unseen relations. The process can be formulated as follows:

$$w_\theta^* = \arg\min_{w \in \Psi} \left(\frac{1}{N} \sum_{i=1}^{N} \ell(w(x_i, \mathcal{P}), y_i) + \lambda R(w) \right), \tag{3}$$

where $w(x_i, \mathcal{P})$ denotes the model's prediction based on input data x_i augmented with a prompting phrase p. This prompting phrase p can encapsulate pre-trained knowledge, guiding the model to better leverage such information for predictions. By adjusting the content and structure of the prompting phrase, one can effectively integrate pre-trained knowledge into the model's prediction process, thereby enhancing model performance and generalization capabilities. $R(w)$ serves as a regularization term quantifying the disparity in model parameters w and pre-trained knowledge, with λ controlling the weight of the regularization term during optimization. Tuning λ balances the model's training performance and retention of pre-trained knowledge. Since pre-trained language models contain a vast amount of knowledge unrelated to ZeroRTE, it is challenging to design appropriate regularization terms that effectively activate the relevant capabilities. Moreover, the model's generalization ability relies heavily on pre-training knowledge while neglecting the task-specific knowledge provided in the current training data, resulting in the limited generalization of the model.

The key to improving model generalization performance is to capture as much generalized knowledge as possible from the training data in addition to leveraging pre-trained knowledge. Therefore, the training target becomes learning RTE and learning generalized knowledge. These two issues are intertwined, and balancing them is challenging when using only one single training objective function. In order to balance the two targets, we propose a generative meta-learning framework based on bi-level optimization (BLO) [21]. The standard BLO problem contains two levels of optimization tasks:

$$\min_{x \in \mathcal{X}} F(x, y), s.t. y \in \mathcal{S}(x), \tag{4}$$

where $y \in \mathbb{R}^n$ and $x \in \mathbb{R}^m$ are respectively referred to LL and UL variables, F is the UL objective, and $\mathcal{S}(x)$ is the solution of the LL subproblem. Specifically, the UL subproblem in the context of ZeroRTE pertains to the discovery of a relational triplet extraction pattern conducive to enhancing the model's ability to generalize across novel categories, and the LL subproblem involves the comprehensive acquisition of triple extraction knowledge embedded within the training dataset. In addition, there are various meta-learning techniques that can aid in improving generalization. Therefore, we further explored the combination of three meta-learning techniques including metric learning, gradient optimization, and model architecture adjustments.

In summary, our contributions are as follows.

- This work presents the first application of BLO to enhance the generalization performance of ZeroRTE tasks, thereby establishing ZeroRTE on a novel paradigm.

- We innovatively developed three types of generative meta-learning techniques. These advances further improve the effectiveness of the model.
- Detailed experimental analysis demonstrates the effectiveness of the proposed framework. Additionally, by comparing the performance of different meta-learning methods, we conclude that the design of meta-learning should be consistent with the schema of the pre-trained model.

2 Related Work

2.1 Zero-shot Relation Triplet Extraction

The extraction of the ZeroRTE is a challenging but valuable task in RE, and is first proposed by Chia et al. [2]. In ZeroRTE, the model needs to learn the general knowledge of RTE based on the training data under the known relation categories, and then extract the unseen relations and the corresponding entities.

In addressing this issue, current ZeroRTE research has evolved into two main categories. The first category typically leverages data augmentation to improve the model's generalization. For example, a method called RelationPrompt employs synthetic data of unseen relations [2]. This approach utilizes pre-trained BART [16] and synthetic data derived from GPT-2 [24] to improve generalization specifically on unseen relations. Building on this premise, numerous methods incorporate external knowledge to enrich the quality of synthetic data pertaining to unseen relations [5, 6]. Obviously, these methods cannot adapt to unseen relations without tuning.

Another line emphasizes stressing the incorporation of prior knowledge in the pre-trained model to improve the generalization. For instance, Kim et al. [11] extends ZeroRTE to a template completion task, leveraging the model's knowledge to intuitively adapt to novel relations. However, a key limitation of these approaches lies in the neglect of optimizing model generalization throughout the training process. Therefore, this paper attempts to improve model generalization performance from a BLO perspective.

2.2 Bi-level Optimization

The origin of BLO can be traced to the domain of game theory and is known as Stackelberg competition [21]. BLOs are hierarchical in nature, where the feasible space of the upper-level (UL) problem is constrained by the solution set mapping graph of the lower-level (LL) problem (i.e., the second task is embedded within the first one).

A range of machine learning methods, such as hyper-parameter optimization, adversarial training, deep reinforcement learning, and meta-learning, involve closely interconnected sub-tasks. For instance, adversarial training comprises an UL objective discriminator (distinguishing real samples from generator-generated data) and a LL objective generator (producing samples that the discriminator cannot confidently classify as real or fake). Similarly, deep reinforcement learning includes two objectives: a policy model responsible for action decisions and a value function model evaluating the quality of policies. By employing BLO, complex tasks can be decoupled to enhance model performance.

Our framework must account for the various scenarios in BLO problems, particularly when the LL problem has multiple optimal solutions. It becomes crucial to determine the best solution in such cases and handle them based on an optimistic BLO assumption [3].

2.3 Meta-learning

Meta-learning is a subfield of machine learning that focuses on developing algorithms and models capable of learning how to learn efficiently and effectively. Due to the ability to improve the generalization capacity of machine learning models, it has attracted great interest in recent years. Meta-learning usually consists of two modules. One captures meta-knowledge (common knowledge across tasks), and the other models task-specific knowledge learned by the base learner. The key is to find meta-knowledge in this complex process. From the perspective of BLO, meta-learning can be formulated as follows:

$$\omega^* = \arg\min_{\omega} \sum_{i=1}^{M} l^{meta}(D_{source}^{val(i)}; w_{\theta}^{*(i)}, \omega) \tag{5}$$

$$s.t.\ \theta^{*(i)}(\omega) = \arg\min_{\theta} l^{task}(D_{source}^{train(i)}; w_{\theta}; \omega), \tag{6}$$

where l^{meta} and l^{task} respectively refer to the UL and LL objectives, ω represents the meta knowledge that needs to be learned from different tasks, and θ represents the parameter in the model.

Conventional categorizations of meta-learning methods [14, 33] can be classified to metric-based, model-based, and optimization-based methods. The metric-based methods [12, 27, 31] aim to learn an appropriate distance metric for few-shot classification and have been successfully applied to some few-shot and zero-shot tasks [8, 22]. The model-based methods [18, 35, 36] involve a task specification to directly generate or modulate model weights. The optimization-based methods [4, 23, 26] focus on incorporating optimization within the learning process to learn an optimized initialization of model parameters.

Meta-learning offers a promising approach to addressing various challenges, particularly in the context of generalization. However, its potential remains largely unexplored in the ZeroRTE domain. In this work, we make the first attempt to incorporate BLO and meta-learning into the ZeroRTE task. To this end, we propose a generative meta-learning framework that eliminates the need for generated data and directly learns task-specific meta-knowledge during the training process.

3 Methodology

3.1 Problem Formulation

Definition 1 (**RTE**) Given a piece of text $s = (w_1, w_2, ..., w_l)$, the RTE task aims to extract the relation triplets $T = \{t^1, t^2, ..., t^o\}$ in s. In each $t^i = (\tilde{e}^i_{head}, \tilde{e}^i_{tail}, r^i)$, $\tilde{e}^i_{head}$ and $\tilde{e}^i_{tail}$ are the head and tail entities, respectively, and $r^i \in R$ is the relation between $\tilde{e}^i_{head}$ and $\tilde{e}^i_{tail}$, where $R = \{r_1, ..., r_{|R|}\}$ is a set of predefined relations.

Definition 2 (**ZeroRTE**) Given a seen dataset D_S and an unseen dataset D_U, the goal of ZeroRTE is to extract triplets T_U in D_U by learning knowledge from D_S. In the zero-shot setting, the seen relation set $R_S = \{r_1, ..., r_{|n|}\}$ is disjoint with the unseen relation set $R_U = \{r_{n+1}, ..., r_{|n+m|}\}$, i.e., $R_S \cap R_u = \emptyset$, where n and m are the sizes of the seen and unseen relations, respectively.

3.2 Framework Analysis

Existing ZeroRTE tasks are usually based on various pre-trained generative language models (GLMs), such as BERT, BART, T5, etc,

that are usually based on the Transformer architecture. The challenges of introducing BLO in these methods are as follows:

Challenge 1. **What type of partition can naturally decouple ZeroRTE into upper-level and lower-level sub-problems, facilitating subsequent model optimization?** In the intricate ZeroRTE tasks, the shared meta-knowledge manifests in the unseen tasks, necessitating the UL task to grasp inter-task knowledge, while the LL task focuses on acquiring task-specific insights. This paper approaches model design from the task level, randomly generating a large number of tasks from the training set. LL tasks learn knowledge specific to individual tasks, while UL tasks are responsible for capturing generalizable patterns across tasks. In this process, the model needs to make inferences based on the tasks. Therefore, to effectively capture the input differences between tasks, we design a task-aware generative model.

Challenge 2. **Can the application of different techniques (metric-based, model-based, and optimization-based) on the Transformer architecture further enhance the model's generalization performance?** In addition to BLO, several supplementary methods can also enhance the model's capacity to effectively capture cross-task meta-knowledge. Existing meta-learning studies demonstrate that metric-based approaches, model-based techniques, and optimization-based strategies all contribute to enhancing the model's generalization capabilities. How to smoothly combine these modules with the existing knowledge in the pre-trained model is a challenge. In order to compare the impact of integrating these methods in detail, this paper redesigns the recursive generation process of language models, introduces new processes (metrics, models, optimization), and strives to further improve the generalization potential of GLMs.

3.3 Model Overview

An overview of our proposed generative meta-learning framework is illustrated in Fig.2. To implement BLO in GLMs, we initially craft a task-aware generative model (TGM) capable of assimilating meta knowledge across diverse tasks, as shown in Fig.2 (a). Subsequently, we leverage different meta-learning methods for enhancing the models. Specifically, we introduce three generative meta-learning methods rooted in distinct meta-learning categories: metric-based generative meta-learning (TGM-Metric) based on metric-based meta-learning, model-based generative meta-learning (TGM-Model) based on model-based meta-learning, and optimization-based generative meta-learning (TGM-Optimization) based on optimization-based meta-learning, depicted in Fig. 2 (b)-(d).

3.4 Task-aware Generative Model

To achieve BLO within ZeroRTE, our model must possess the capability to discern between various tasks. As a result, we introduce a novel prompt for GLMs to encode task information. This approach enables the model to extract triplets corresponding to unseen relation labels at a task level rather than a sample level.

The input structure of our generative model comprises two key components. The first component entails the task information, which serves to suggest a series of potential relation categories for the given task. The second part encompasses the current text being processed. The format is as follows:

"*Relation:* R_1, R_2, ..., R_m. *Context: Washington is the capital of the U.S.A.*"

Note that m is the unseen relation number, and R_i is the candidate relation type in one task. For example, when m=5, a task prompt might be "*Relation: sitter, capital of, conflict, elector, direction.*".

In this way, the task-aware generative model will output the relation triplets contained in the input sentence. The predefined format is:

"*Head Entity: Washington, Tail Entity: the U.S.A, Relation: capital of.*"

Notably, the task information prompt can drive the generative model to make selection among candidate relations, which is identical to the 'learning to learn' idea of meta-learning. As a result, it forms a sound basis for the subsequent meta-learning methods. Moreover, we perform multiple tasks in each epoch during training, and the optimization process is conducted across these tasks. This forces the generative model to pay more attention to the task-level information, and thus it can learn general knowledge across different tasks.

The training for the generative model is to maximize the likelihood $L(\mathcal{D})$ in the data set $\mathcal{D}$ as follows.

$$L(\mathcal{D}) = \prod_{i=1}^{|\mathcal{D}|} \prod_{(h,t,r)\in T_i} P((h,t,r)|s_i,\mathcal{P}), \tag{7}$$

where (h,t,r) refers to the (head entity, tail entity, relation), s_i is the i-th input sentence in $\mathcal{D}$, T_i is the annotated relation triplets in s_i, and $\mathcal{P}$ represents the task information in the input.

In bi-level perspective, TGM model contains LL problem and UL problem. It can be formulated as follows.

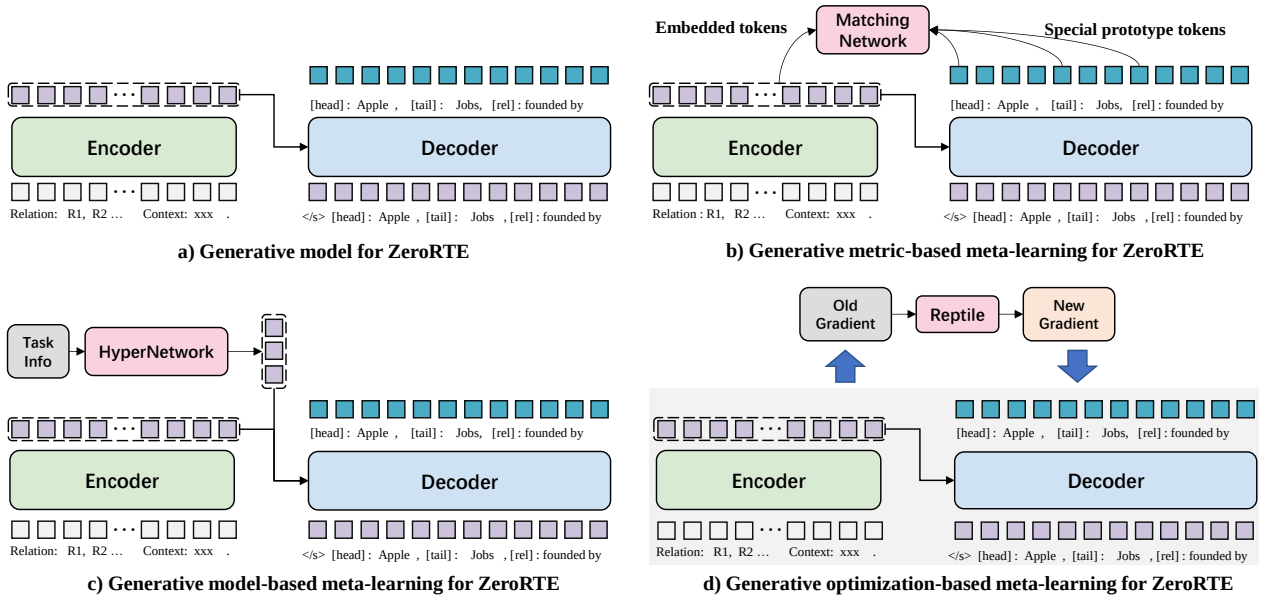
$$L_{\text{LL}} = \prod_{i=1}^{|\mathcal{D}^{train}|} \prod_{(h,t,r)\in T_i} P_{w_\theta}((h,t,r)|s_i,\mathcal{P}), \tag{8}$$

$$L_{\text{UL}} = \prod_{i=1}^{|\mathcal{D}^{val}|} \prod_{(h,t,r)\in T_i} P_{w^*_\theta}((h,t,r)|s_i,\mathcal{P}), \tag{9}$$

where w^*_θ represents the GLM's parameters after LL optimization. To improve the generalization, regularization terms are added to penalize the output that does not conform to the predefined format:

$$L_{\text{TGM}} = L_{\text{UL}} + L_{\text{LL}} + \lambda R(w), \tag{10}$$

Following existing generative methods, we utilize the decoder module in Transformer [30] to generate tokens in a recursive and sequential manner. In this way, our generative model also generates results in a predefined order. The difference is that we take the task prompt into consideration. Given the input sentence s_i and the predefined order of head entity, tail entity, and relation, the likelihood of our generative model is as follows:



a) Generative model for ZeroRTE

b) Generative metric-based meta-learning for ZeroRTE

c) Generative model-based meta-learning for ZeroRTE

d) Generative optimization-based meta-learning for ZeroRTE

Figure 2: An overview of our proposed generative meta-learning framework for zero-shot RTE.

$$
\begin{aligned}
L(s_i) &= \prod_{(h,t,r)\in T_i} P((h,t,r)|s_i,\mathcal{P}) \\
&= \prod_{(h,t,r)\in T_i} P(h|s_i,\mathcal{P}) \cdot P(t|s_i,h,\mathcal{P}) \\
&\cdot P((r|s_i,h,t,\mathcal{P}).
\end{aligned} \tag{11}
$$

Note that the above decoding order is experimentally explored to be optimal, and the effect of different extraction orders on the model is detailed in the Section 4.7. Below we will show how TGM can be integrated into three types of meta-learning methods.

3.5 Metric-based Generative Meta-learning (TGM-Metric)

Metric-based meta-learning (MEML) methods learn the metric-based connections behind objects. They typically map input samples into an embedding space and then use the nearest neighbor or matching mechanism to label query samples based on the connections between their embeddings to those of labeled ones. By this means, MEML methods can naturally generalize to new domains.

Inspired by the prototype-based methods [27, 31], we design a novel feature mapping process with three prototypes output by the decoder module and a matching network for predicting whether the input tokens and the prototypes match. Since existing generative language models follow a recursive and sequential generation manner, we insert three special tokens '[HEAD]', '[TAIL]', and '[REL]' before the relation triplet, to represent the head entity prototype, tail entity prototype, and relation prototype, respectively:

'*[HEAD]: Washington, [TAIL]: the U.S.A, [REL]: capital of.*'

We then map the token embedding encoded by the encoder and the prototypes output by the decoder to one same vector space through a linear transformation, and then predict whether the prototypes and the corresponding token embedding by a matching network (one layer of neural network). The loss for training our TGM-Metric is the combination of the loss of the generative model and that of the matching network:

$$
L_{Metric} = L_{TGM} + \alpha L_{Matching}, \tag{12}
$$

where $\mathcal{L}_{TGM}$ is the loss for the generative model to maximize Eq. 11 and generate the answer that conforms to the proposed format, and $L_{Matching}$ is the loss for the matching network, and α is a trade-off parameter.

$$
L_{Matching} = \sum_{i=1}^{|\mathcal{D}^{train}|} \sum_{j=1}^{|\mathcal{T}|} CE(G_i^j, \text{MLP}(E_i^j \oplus E_i^p)), \tag{13}
$$

where the MLP layer measures whether the t-th token embedding E_i^t matches the p-th prototype embedding E_i^p, and G_i^j is the ground-truth of matching for the j-th token in the i-th sentence with the prototype, $\oplus$ denotes the concatenation, and CE is the cross-entropy loss.

3.6 Model-based Generative Meta-learning (TGM-Model)

Model-based meta-learning (MOML) focuses on improving the generalization ability of the model through an external module responsible for modeling meta knowledge [7, 35]. In MOML, when the external module deals with different meta-learning tasks, the module will generate optimal parameters from the perspective of the task for enhancing the inference process.

Following the MOML scheme, we explore how the external module can improve the generative models. To this end, we first design a prompt generator module for modeling task information. We then input the newly generated parameters as encoded information into the decoder module for adapting to the generation process. By this

means, the task information and the encoder-encoded information are concatenated together and forwarded into the decoder to perform the RTE task, which forms our TGM-Model. The training for our TGM-Model is to maximize the likelihood $L(\mathcal{D})$ in the training set $\mathcal{D}$:

$$L(\mathcal{D}) = \prod_{i=1}^{|\mathcal{D}|} \prod_{(h,t,r)\in T_i} \prod_{j=1}^{|\mathcal{T}_i|} P((h,t,r)|s_i, \text{MLP}(\mathcal{P}_j)), \tag{14}$$

where $\mathcal{P}_j$ represents the j-th task information prompt in the candidate task set $\mathcal{T}$ for the i-th sentence. The BLO process is consistent with the TGM model, in which the module for meta-learning is only optimized in the UL task and not in the LL task. Therefore, its formula is expressed as follows.

$$MLP_{w_\theta^{(\tau+1)}} = \begin{cases} MLP_{w_\theta^{(\tau)}} - \eta \frac{\partial L}{\partial MLP_{w_\theta}}, & \text{if in UL} \\ MLP_{w_\theta^{(\tau)}}, & \text{if in LL} \end{cases}. \tag{15}$$

3.7 Optimization-based Generative Meta-learning (TGM-Optimization)

The optimization-based meta-learning (OBML) methods improve the generalization ability from the perspective of optimization and are typically model-agnostic [4, 23]. OBML aims to find the most generalizable gradient direction among the gradients obtained by different meta-learning tasks. These methods seek to adjust the parameters of the neural network so that it can quickly adapt to different tasks.

All existing ZeroRTE methods adopt the gradient descent for training model. However, it does not consider whether the current gradient direction can improve the generalization ability or may overfit to training tasks. In contrast, our proposed task-aware generative model provides the opportunity to find the optimal gradient among different meta-learning tasks. In this subsection, we further improve our generative model with an OBML method named Reptile [23] which is mathematically similar to the classic MAML [4, 23] but is simple to implement. By this means, we finally get our TGM-Optimization model.

Formally, for an input sentence s_i, the likelihood in our TGM-Optimization is as follows:

$$L(s_i) = \prod_{i=1}^{|\mathcal{D}|} \prod_{(h,t,r)\in T_i} \prod_{j=1}^{|\mathcal{T}|} P((h,t,r)|s_i, \mathcal{P}_j), \tag{16}$$

where $\mathcal{P}_j$ represents the j-th task information prompt.

The optimization process of our TGM-Optimization model can be defined as follows:

$$\Psi \leftarrow \Psi + \epsilon \frac{1}{n} \sum_{i=1}^{n} (\tilde{\Psi}_i - \Psi), \tag{17}$$

where $\tilde{\Psi}_i$ is the updated parameter space on the i-th task which is randomly sampled, n is the task number in each iteration, Ψ is the model parameter space to be optimized, and ϵ is a step-size parameter.

3.8 Inference

In order to preserve the general knowledge contained in the generative model as much as possible, the inference in our framework also takes the form of the generative model. Specifically, TGM-Metric utilizes meta-knowledge derived from metric learning to guide the decoder in generating the correct output. In the TGM-Model, the decoder uses meta-knowledge output by model-based meta-learner to generate the triplets. TGM-Optimization directly generates RTE results during inference. This is because the optimization-based meta-learner only optimizes the gradient during training and does not affect the inference process of the generative model.

4 Experiments

We conduct extensive experiments to verify the effectiveness of our framework and answer the following research questions:

- **RQ1**: Can the incorporation of BLO enhance the generalization capabilities of the current model?
- **RQ2**: What factors influence the integration of the three meta-learning techniques with models?
- **RQ3**: What are the advantages of this approach compared to the large language models (LLMs)?
- **RQ4**: What is the time complexity of the model?

4.1 Experimental Setup

Datasets We evaluate our model on two public datasets. FewRel [9] is a standard benchmark dataset for the few-shot RE task. Wiki-ZSL [1] is generated with distant supervision from Wikipedia articles and the Wikidata knowledge base [28]. The detailed data statistics are shown in Table 1.

Table 1: Statistics for two datasets.

Dataset	#Samples	#Entities	#Relations	Sent_len
FewRel	54,000	72,954	80	24.95
Wiki-ZSL	94,383	77,623	113	24.85

Experimental Settings To make a fair comparison, we directly leverage the data split provided by RelationPrompt [2][1]. In this setting, 5 random seeds are selected for the label selection process, where 5 validation labels from the seen labels are used to select sentences for early stopping and hyperparameter tuning, m unseen labels ($m \in \{5, 10, 15\}$) are selected for testing, and the remaining sentences are treated as the training samples.

We use T5-base [25] as our pre-trained generative language model. The learning rates of the generative model parameters and other parameters are set to 3e-5 and 6e-4, respectively, and the batch size for training is set to 16. We randomly generate t different tasks for each sample to improve the model's ability to capture connections between tasks and samples. For example, when $m = 3$ and $t = 2$, we will have two task prompts like $R_{i1}R_{i2}R_{i3}$ and $R_{j1}R_{j2}R_{j3}$.

In RelationPrompt [2], the test data is divided into two parts according to whether the sample contains a single triplet or multiple triplets. This is unrealistic in practice because we cannot know the number of triplets contained in each sample in advance. Therefore, in our setting, the number of triplets is unknown, and the model

[1] https://github.com/declare-lab/RelationPrompt

Table 2: Comparison results on FewRel. Except for the LLMs based method (MICRE), the best scores are in bold, and the second best ones are underlined. All results are the average scores of 5 runs with the same seeds.

		m=5			m=10			m=15		
Methods	Synthetic Data	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1
(1) TableSequence	✓	9.86	9.33	9.59	12.50	12.02	12.24	12.46	11.92	12.19
(2) RelationPrompt	✓	24.70	24.54	24.69	24.59	24.23	24.39	20.66	20.25	20.45
(3) KBPT	✓	24.14	23.91	24.02	24.35	27.28	26.02	22.11	21.56	21.83
(4) ZS-SKA	✓	36.24	34.77	35.49	**33.06**	**32.85**	**32.95**	**26.11**	23.51	24.74
(5) NoGen-BART	✗	22.90	22.61	22.75	16.54	16.31	16.42	12.16	11.94	12.05
(6) NoGen-T5	✗	25.90	25.58	25.74	16.42	16.19	16.30	13.01	12.75	12.87
(7) ZETT	✗	31.00	30.61	30.8	28.91	27.46	28.17	24.45	23.42	23.92
(8) TGM	✗	36.87	36.43	36.65	27.16	26.76	26.95	23.86	23.40	23.63
(9) TGM-Metric	✗	38.32	37.86	38.09	28.66	28.26	28.46	24.49	24.02	24.25
(10) TGM-Model	✗	38.16	37.70	37.93	27.98	27.57	27.77	24.38	23.89	24.13
(11) TGM-Optimization	✗	**39.40**	**38.91**	**39.15**	30.18	29.77	29.97	25.43	**24.94**	**25.19**

needs to actively decode multiple triplets during testing. This means that our evaluation setup is more critical than RelationPrompt.

Metrics We utilize the Micro-$F1$ score (F1) as the evaluation metric, additionally reporting precision and recall for more analysis. All results are averaged across five data folds with the same seeds.

4.2 Baseline Methods

The compared baselines are listed as follows: 1) **TableSequence** [32] Following RelationPrompt, we adapt the RTE method to ZeroRTE by training on synthetic data generated by RelationPrompt. 2) **RelationPrompt** [2] is trained on the training set and synthetic data generated by a fine-tuned GPT-2 [24]. 3) **KBPT** [6] incorporates prior knowledge from ontological schemas and is trained on synthetic data generated by a generative prompt model. 4) **ZS-SKA** [5] also employs data augmentation through a word-level sentence translation. 5) **NoGen-BART** is based on RelationPrompt and fine-tunes the BART model only on the training data without synthetic data. 6) **NoGen-T5** is based on RelationPrompt but fine-tunes the T5 model only on the training data without synthetic data. 7) **ZETT** [11] treats zero-shot relational triplet extraction as a template filling task and employs ranking methods to extract the relation triplet.

Within these methods, approaches 1-4 leverage data augmentation strategies to calibrate the model using synthetic data associated with unseen relations. These methods necessitate retraining when applied to unseen relations and lack natural generalization into novel domains. On the other hand, methods 5-7 do not rely on synthetic data and can be directly utilized when encountering new relations.

"**NoGen-BART**" and "**NoGen-T5**" respectively fine-tunes the BART-base and T5-base on the training data. It is notable that the parameters of BART-base, GPT-2, and T5-base are 140M, 124M, and 220M, respectively.

We use the source code provided by the authors of TableSequence [2] and RelationPrompt [3]. We re-train them using the optimal hyper-parameters reported in their original papers.

[2]https://github.com/LorrinWWW/two-are-better-than-one
[3]https://github.com/declare-lab/RelationPrompt

4.3 Main Results

We present the comparative outcomes of our proposed models on FewRel and Wiki-ZSL in Table 2 and Table 3 correspondingly. We draw observations based on these results.

To address the question **RQ1**, we compare two models with the same structure, TGM and NoGen-T5. As shown in Table 2 and Table 3, the proposed TGM outperforms NoGen-T5 in all settings on both datasets. Given that the NoGen-T5 model shares the exact same architecture with the proposed TGM model, its superior performance clearly indicates that the BLO process facilitates the GLMs capable of capturing valid and generalizable knowledge across tasks. This finding also suggests that decomposing a complex task like ZeroRTE into UL and LL subtasks and simultaneously optimizing them can significantly enhance generalization performance of the models.

To answer **RQ2**, we compare three generative meta-learning methods with TGM. Our proposed three generative meta-learning methods can further improve the performance of the TGM model, indicating that the meta-learning mechanism further boosts the generalization capability of the basic generative model. After in-depth analysis, we found that the TGM-Metric and TGM-Model introduced new metric-learning modes and meta-learning modules. Although these additions can be integrated with existing semantic knowledge in Transformer, they damaged the semantic inference ability of the model to a certain extent; while the TGM-Optimization completely relies on the BLO mechanism to adjust the gradient direction and does not introduce additional features for model generalization, and it is more consistent with the original semantic capabilities of the pre-trained models. Therefore, we believe that the optimization process should be designed with careful consideration of how well it aligns with the model's original knowledge. This alignment directly influences the robustness of the model's generalization ability.

In order to analyze whether the stronger generalization of TGM-Metric comes from the influence of pre-training knowledge, we retain the structure of the T5-base model during training but reinitialize the model parameters. The experimental results show that the accuracy of all models tend to 0. This means that a large part of the model's capabilities come from pre-training knowledge.

Table 3: Comparison results on Wiki-ZSL. Except for the LLMs based method (MICRE), the best scores are in bold, and the second best ones are underlined. All results are the average scores of 5 runs with the same seeds.

		m=5			m=10			m=15		
Methods	Synthetic Data	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1
(1) TableSequence	✓	15.51	11.86	13.46	10.64	5.51	8.07	8.67	5.55	7.08
(2) RelationPrompt	✓	24.91	20.46	22.39	19.27	16.19	17.57	14.20	12.31	13.48
(3) KBPT	✓	35.45	31.64	33.50	22.41	21.74	23.57	21.02	17.31	19.01
(4) ZS-SKA	✓	**45.27**	**41.68**	**43.40**	**29.88**	**26.05**	**27.83**	**23.67**	**20.39**	**21.93**
(5) NoGen	✗	18.81	15.41	16.87	11.94	10.16	10.96	8.43	6.95	7.62
(6) NoGen-T5	✗	19.21	16.52	17.66	12.25	10.33	11.19	9.05	7.48	8.19
(7) ZETT	✗	26.22	23.76	24.93	21.05	18.99	19.97	18.31	13.99	15.86
(8) TGM	✗	34.40	27.76	30.64	22.02	18.54	20.10	16.93	14.03	15.34
(9) TGM-Metric	✗	37.35	30.17	33.25	24.77	20.96	22.54	20.42	16.86	18.47
(10) TGM-Model	✗	36.72	30.31	33.51	24.09	20.84	22.23	20.10	16.27	17.99
(11) TGM-Optimization	✗	40.67	33.42	36.56	26.09	21.84	23.73	22.10	18.27	19.99

Table 4: Comparison results with LLMs on $m = 5$.

	Acc (FewRel)	Acc (Wiki-ZSL)	Avg.
MICRE(LLaMA)	37.53	27.77	32.65
GPT-4o	11.71	7.35	9.53
TGM-Optimization	38.90	36.23	37.57

4.4 Comparison with LLMs

For **RQ3**, we analyze the outputs of **MICRE** and **GPT-4o**. MICRE [17] based on LLaMA [29] employs in-context learning technique to tackle ZeroRTE. We also show the accuracy of GPT-4o in the Table 4.

It can be seen that a complex system (GPT-4o) will produce uncontrollable and hallucinatory answers when dealing with ZeroRTE. Although we have explicitly restricted the output format of GPT-4o in the prompt, GPT-4o always outputs some extra symbols or sentences. Perhaps, simple models that learn straightforward patterns offer a more efficient way to achieving intelligence.

4.5 Complexity Analysis

For **RQ4**, we conduct a complexity analysis. Both our proposed method and existing baselines utilize the pre-trained GLMs M based on the Transformer structure. The computational complexity primarily depends on the backbone model M, which includes multi-head self-attention, feed-forward networks, layer normalization, and residual connections in each of its n layers.

The computation of l-th layer feed-forward network is formulated as $X^{L+1} = \sigma(X^l W^l)$. Where $\sigma()$ is a non-linear activation function, and W^l is a feature transformation matrix $\in \mathbb{R}^{F_l \times F_{l+1}}$. For simplicity, we assume the features at every layer are size-d. As such, W^l is an $d \times d$ matrix. From the setting, we know $d_v = d$.

We analyze the time complexity of the TGM by three high-level operations:

i. Multi-head Self-Attention ($Softmax(\frac{QK^T}{\sqrt{d}})V$). We assume that it has v head, then the time complexity is $O(vn^2d)$

ii. Feed-forward Network ($\sigma(X^l W^l)$) is a dense matrix multiplication between matrices of size $n \times d$ and $d \times d$. The time complexity is $O(nd^2)$.

iii. layer normalization and residual connections ($LayerNorm(X + MultiHeadAttention(X))$ and $LayerNor(X + Feedforward(X))$) involves computing mean and variance across features, which has a time complexity of $O(nd)$. Residual connections contains a simple element-wise addition and thus has a time complexity of $O(n)$

Thus, the time complexity of TGM is $O(vn^2d + nd^2 + nd + n)$ for one forward propagation, which simplifies to $O(n^2d)$ given $d << n$.Both TGM-Metric and TGM-Model extend TGM with new feed-forward neural networks, leading to a complexity of $O(vn^2d + nd^2 + nd + n + nd^2)$, still simplified to $O(n^2d)$. In TGM-Optimization, we introduce a novel gradient update pattern that does not alter the model structure or inference process, maintaining a complexity of $O(n^2d)$.

In summary, the computational complexity of the proposed framework aligns with that of other frameworks.

4.6 Impacts of the Relation Number in Prompt

Our prompt contains m candidate task information in the prompt to facilitate the model to understand tasks. Following RelationPrompt [2], we assume m is known in training. However, this is impractical in real-world scenarios. Hence we investigate whether our model can cope with an unknown number of candidate relations. For this, we set the candidate relation number as r, and vary it among {2, 5, 10, 15}. The results are shown in Table 5. It is clear that a r smaller than m often results in a better performance [4]. This is natural since the model can focus more on the true task and is not affected by redundant relations in the prompt. However, a too small r like 2 will prevent the generative model from learning general information across tasks and hurts the performance. This finding also holds true for r=3, 4 for m=5, suggesting that a pre-determined number of relations is not necessary. This flexibility represents another attractive feature of our model.

Table 5: Impacts of the relation number r in the prompt.

FewRel	F1(m=5)	F1(m=10)	F1(m=15)
r=2	37.38	26.83	20.34
r=5	39.15	31.49	26.08
r=10	37.84	29.97	25.90
r=15	36.70	28.72	25.19

[4]This infers that we actually do not need to set t=m. Instead, t can be a relatively small number like 5, which is another appealing property of our model.

	Relation	Text	RelationPrompt Prediction	TGM Prediction	TGM-Optimization Prediction
S1:	Owned by	Portugal Telecom is no longer a publicly traded company, since 100 % of its capital is owned by Altice Group.	Head entity: Portugal Telecom Tail entity: Altice Group Relation: owned by	Head entity: Portugal Telecom Tail entity: Altice Group Relation: owned by	Head entity: Portugal Telecom Tail entity: Altice Group Relation: owned by
S2:	Operating system	Siri on the Apple TV has all of the func--tions of Siri on iOS 9; it can also respond to requests specifically for the TV .	Head entity: Siri Tail entity: Apple TV Relation: operating system	Head entity: Siri Tail entity: IOS 9 Relation: operating system	Head entity: Siri Tail entity: IOS 9 Relation: operating system
S3:	Religion	Giacomo Cantelmo (13 June, 1645–11 December, 1702) was a Roman Catholic cardinal from 1690 to 1702.	Head entity: Giacomo Cantelmo Tail entity: cardinal Relation: location	Head entity: Giacomo Cantelmo Tail entity: cardinal Relation: religion	Head entity: Giacomo Cantelmo Tail entity: Roman Catholic Relation: religion

Figure 3: Case study. The orange, blue, and green tokens respectively denote the head entity, tail entity, and relation. Incorrectly extracted tokens are marked in grey.

4.7 Impacts of the Triplet Order

The decoder in Transformer generates tokens in a sequential manner and the generative model needs a predefined triplet order. We assume an HTR order in Sec. 3.4 and compare other two different orders THR and RHT here, where 'H', 'T', and 'R' denotes the head entity, tail entity, and relation, respectively. The results are shown in Table 6.

Table 6: Impacts of different triplet orders on $m = 5$.

	F1 (FewRel)	F1 (Wiki-ZSL)	Avg.
$\textbf{TGM}_{\textbf{HTR}}$	36.65	30.64	33.65
$\textbf{TGM}_{\textbf{THR}}$	37.11	30.08	33.60
$\textbf{TGM}_{\textbf{RHT}}$	35.16	29.57	32.37

It can be seen from Table 6 that, two generative models with the entity first order HTR and THR are better than that with the relation first order RHT. The reason is that in the ZeroRTE task, most entities are meaningful nouns, and the extraction of entities does not change much under different relation categories. In this case, performing the relatively easy entity extraction task first can help the subsequent hard RE task.

4.8 Analysis on Hyper-parameters

We analyse the task number t and the parameter α for balancing the loss of meta learning and that of the GLMs in the TGM-Metric model. Figure 4 shows the impacts of these two hyper-parameters.

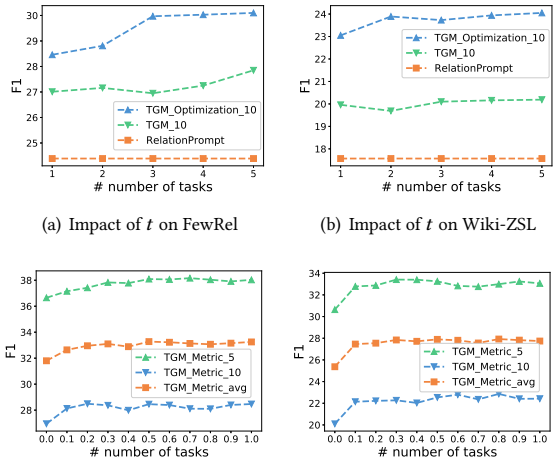
From Fig. 4(a) and 4(b), we find that our two models become stable when t is larger than 3. Note even when t=1, our models perform better than RelationPrompt since our prompt with multiple relations can provide general knowledge across tasks. Fig. 4(c) and 4(d) show that the optimal setting for α is about 0.5 on two datasets.

4.9 Case Study

As shown in Fig. 3, we compare the results of RelationPrompt, TGM, and TGM-Optimization on three sentences (denoted as S1, S2, and S3).

S1 has a phrase 'owned by' which explicitly points out the relation. All three generative models can extract the correct answer for such an easy task.

S2 includes the 'operating system' relation. RelationPrompt predicts the correct relation and a wrong tail entity since the training of its extraction model depends on synthetic samples which may not



(a) Impact of t on FewRel (b) Impact of t on Wiki-ZSL

(c) Impact of α on FewRel (d) Impact of α on Wiki-ZSL

Figure 4: Impacts of the hyper-parameters.

certainly contain 'Apple TV'. In contrast, both our TGM and TGM-Optimization can extract the correct triplet. Though our models do not see samples with 'operating system' relation during training, the relation name has been used as the candidate in the task prompt which will never be chosen as an answer. That is to say, the model has got the implicit knowledge that 'operating system' does not refer to 'Apple TV'. This demonstrates the role of generalizing knowledge across tasks.

S3 contains the 'religion' relation. RelationPrompt extracts wrong tail entity and relation and TGM model guided by the task information extracts the correct head entity but makes mistake on the tail entity. TGM-Optimization obtains the completely correct answer. We believe this is because TGM-Optimization retains the general knowledge about 'religion' via meta-learning, which also proves the value of meta-learning.

5 Conclusion

In this study, we discovered that existing generative language models have limited generalization capabilities in zero-shot learning scenarios. To address this challenge, we propose an innovative generative meta-learning framework that exploits the synergy between BLO and multiple meta-learning strategies. Our approach effectively leverages the meta-knowledge embedded in training datasets, leading to significant improvements.

6 Acknowledgements

This research project was supported in part by Hubei Key Research and Development Program of China under Grant [2024BBB055], the Fundamental Research Funds for the Central Universities, China [Project 2662023XXQD002, 2662023XXQD003, 2662023XXQD004].

References

[1] Chih-Yao Chen and Cheng-Te Li. 2021. ZS-BERT: Towards Zero-Shot Relation Extraction with Attribute Representation Learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (Eds.). Association for Computational Linguistics, 3470–3479. doi:10.18653/v1/2021.naacl-main.272

[2] Yew Ken Chia, Lidong Bing, Soujanya Poria, and Luo Si. 2022. RelationPrompt: Leveraging Prompts to Generate Synthetic Data for Zero-Shot Relation Triplet Extraction. In *Findings of the ACL*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, 45–57. doi:10.18653/v1/2022.findings-acl.5

[3] Stephan Dempe. 2020. Bilevel optimization: theory, algorithms, applications and a bibliography. *Bilevel optimization: advances and next challenges* (2020), 581–672.

[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 2017, Sydney, NSW, Australia, 6-11 August 2017 (Proceedings of Machine Learning Research, Vol. 70)*. PMLR, 1126–1135. http://proceedings.mlr.press/v70/finn17a.html

[5] Jiaying Gong and Hoda Eldardiry. 2024. Prompt-based Zero-shot Relation Extraction with Semantic Knowledge Augmentation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, Nicoletta Calzolari, Min-Yen Kan, Véronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue (Eds.). ELRA and ICCL, 13143–13156. https://aclanthology.org/2024.lrec-main.1151

[6] Qian Guo, Yi Guo, and Jin Zhao. 2024. KBPT: knowledge-based prompt tuning for zero-shot relation triplet extraction. *PeerJ Comput. Sci.* 10 (2024), e2014. doi:10.7717/PEERJ-CS.2014

[7] David Ha, Andrew M. Dai, and Quoc V. Le. 2017. HyperNetworks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=rkpACe1lx

[8] Jiale Han, Bo Cheng, and Wei Lu. 2021. Exploring Task Difficulty for Few-Shot Relation Extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*. 2605–2616. doi:10.18653/v1/2021.emnlp-main.204

[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, Brussels, Belgium, October 31 - November 4, 2018*. 4803–4809. doi:10.18653/v1/d18-1514

[10] Xuming Hu, Junzhe Chen, Shiao Meng, Lijie Wen, and Philip S. Yu. 2023. SelfLRE: Self-refining Representation Learning for Low-resource Relation Extraction. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, Hsin-Hsi Chen, Wei-Jou (Edward) Duh, Hen-Hsen Huang, Makoto P. Kato, Josiane Mothe, and Barbara Poblete (Eds.). ACM, 2364–2368. doi:10.1145/3539618.3592058

[11] Bosung Kim, Hayate Iso, Nikita Bhutani, Estevam Hruschka, Ndapa Nakashole, and Tom M. Mitchell. 2023. Zero-shot Triplet Extraction by Template Infilling. Association for Computational Linguistics, 272–284. doi:10.18653/V1/2023.IJCNLP-MAIN.18

[12] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. 2015. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, Vol. 2. Lille, 0. http://www.cs.cmu.edu/~rsalakhu/papers/oneshot1.pdf

[13] Yuquan Lan, Dongxu Li, Yunqi Zhang, Hui Zhao, and Gang Zhao. 2022. PCRED: Zero-shot relation triplet extraction with potential candidate relation selection and entity boundary detection. *arXiv preprint arXiv:2211.14477* (2022).

[14] Yoonho Lee and Seungjin Choi. 2018. Gradient-Based Meta-Learning with Learned Layerwise Metric and Subspace. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018 (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer G. Dy and Andreas Krause (Eds.). PMLR, 2933–2942. http://proceedings.mlr.press/v80/lee18a.html

[15] Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-Shot Relation Extraction via Reading Comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*. Association for Computational Linguistics, Vancouver, Canada, 333–342. doi:10.18653/v1/K17-1034

[16] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 7871–7880. doi:10.18653/v1/2020.acl-main.703

[17] Guozheng Li, Peng Wang, Jiajun Liu, Yikai Guo, Ke Ji, Ziyu Shang, and Zijie Xu. 2024. Meta In-Context Learning Makes Large Language Models Better Zero and Few-Shot Relation Extractors. *CoRR* abs/2404.17807 (2024). doi:10.48550/ARXIV.2404.17807 arXiv:2404.17807

[18] Huai-Yu Li, Weiming Dong, Xing Mei, Chongyang Ma, Feiyue Huang, and Bao-Gang Hu. 2019. LGM-Net: Learning to Generate Matching Networks for Few-Shot Learning. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 3825–3834. http://proceedings.mlr.press/v97/li19c.html

[19] You Li, Xupeng Zeng, Yixiao Zeng, and Yuming Lin. 2024. Enhanced Packed Marker with Entity Information for Aspect Sentiment Triplet Extraction. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, Grace Hui Yang, Hongning Wang, Sam Han, Claudia Hauff, Guido Zuccon, and Yi Zhang (Eds.). ACM, 619–629. doi:10.1145/3626772.3657734

[20] Hongtao Lin, Jun Yan, Meng Qu, and Xiang Ren. 2019. Learning Dual Retrieval Module for Semi-supervised Relation Extraction. In *The World Wide Web Conference, WWW*. 1073–1083.

[21] Risheng Liu, Jiaxin Gao, Jin Zhang, Deyu Meng, and Zhouchen Lin. 2021. Investigating bi-level optimization for learning and vision from a unified perspective: A survey and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 12 (2021), 10045–10067.

[22] Yang Liu, Jinpeng Hu, Xiang Wan, and Tsung-Hui Chang. 2022. A Simple yet Effective Relation Information Guided Approach for Few-Shot Relation Extraction. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, 757–763. doi:10.18653/v1/2022.findings-acl.62

[23] Alex Nichol, Joshua Achiam, and John Schulman. 2018. On First-Order Meta-Learning Algorithms. *CoRR* abs/1803.02999 (2018). arXiv:1803.02999 http://arxiv.org/abs/1803.02999

[24] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

[25] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* 21 (2020), 140:1–140:67. http://jmlr.org/papers/v21/20-074.html

[26] Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. 2019. Meta-Learning with Latent Embedding Optimization. (2019). https://openreview.net/forum?id=BJgklhAcK7

[27] Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical Networks for Few-shot Learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 4077–4087. https://proceedings.neurips.cc/paper/2017/hash/cb8da6767461f2812ae4290eac7cbc42-Abstract.html

[28] Daniil Sorokin and Iryna Gurevych. 2017. Context-Aware Representations for Knowledge Base Relation Extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, Martha Palmer, Rebecca Hwa, and Sebastian Riedel (Eds.). Association for Computational Linguistics, 1784–1789. doi:10.18653/v1/d17-1188

[29] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *CoRR* abs/2302.13971 (2023). doi:10.48550/ARXIV.2302.13971 arXiv:2302.13971

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 5998–6008. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[31] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. 2016. Matching Networks for One Shot Learning. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*. 3630–3638. https://proceedings.neurips.cc/paper/2016/hash/90e1357833654983612fb05e3ec9148c-Abstract.html

[32] Jue Wang and Wei Lu. 2020. Two are Better than One: Joint Entity and Relation Extraction with Table-Sequence Encoders. In *EMNLP*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, 1706–1721. doi:10.18653/v1/2020.emnlp-main.133

[33] Huaxiu Yao, Xian Wu, Zhiqiang Tao, Yaliang Li, Bolin Ding, Ruirui Li, and Zhenhui Li. 2020. Automated Relational Meta-learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. https://openreview.net/forum?id=rklp93EtwH

[34] Deming Ye, Yankai Lin, Peng Li, and Maosong Sun. 2022. Packed Levitated Marker for Entity and Relation Extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, 4904–4917. doi:10.18653/v1/2022.acl-long.337

[35] Qinyuan Ye and Xiang Ren. 2021. Learning to Generate Task-Specific Adapters from Task Description. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 2: Short Papers), Virtual Event, August 1-6, 2021*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, 646–653. doi:10.18653/v1/2021.acl-short.82

[36] Andrey Zhmoginov, Mark Sandler, and Maksym Vladymyrov. 2022. HyperTransformer: Model Generation for Supervised and Semi-Supervised Few-Shot Learning. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (Eds.). PMLR, 27075–27098. https://proceedings.mlr.press/v162/zhmoginov22a.html

[37] Shen Zhou, Yongqi Li, Xin Miao, and Tieyun Qian. 2024. An Ensemble-of-Experts Framework for Rehearsal-free Continual Relation Extraction. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, 1410–1423. doi:10.18653/V1/2024.FINDINGS-ACL.83