ENIAC-ML: Environmentally Interactive Active Meta-learning for Zero-Shot Relation Triplet Extraction

Anonymous ACL submission

Abstract

Zero-shot relation triplet extraction (ZeroRTE) task aims to extract unseen relations and corresponding entities from the text. Existing methods conflate the Relation Extraction (RE) and Named Entity Recognition (NER) subtasks. Moreover, some methods introduce synthetic data or information that contains noise, resulting in failures on ZeroRTE. We propose a novel meta-learning approach named **En**vironmentally **I**nteractive **AC**tive **M**eta-**L**earning (ENIAC-ML) that can mimic human processing on ZeroRTE. We decompose ZeroRTE into RE and NER subtasks and train the model using a pipelined approach. We further develop an active meta-learning approach that can acquire knowledge by interacting with an agent in the environment, autonomously determine the focus of learning, and mitigate the impact of noise in external information. The experimental results demonstrate that ENIAC-ML surpasses existing methods on Fewrel and Wiki-ZSL datasets. Our code is available at `https://anonymous.4open.science/r/ENIAC-ML-E0FF`.

1 Introduction

The objective of relation triplet extraction (RTE) is to extract triplets in the form of (head entity, tail entity, relation label) from unstructured text and is essential for several applications (Xu et al., 2016). To delve into the generalization of the RTE task, Chia et al. (2022) investigate it in a zero-shot setting, henceforth zero-shot relation triplet extraction (ZeroRTE). As shown in Figure 1, a model should extract all relation triplets mentioned in the text, and ZeroRTE model is trained solely on seen relation classes and needs to generalize to unseen relation classes in zero-shot scenario.

Several works have attempted to address ZeroRTE. RelationPrompt (Chia et al., 2022) utilizes synthetic data of unseen relations for training. However, the model heavily relies on synthetic data that

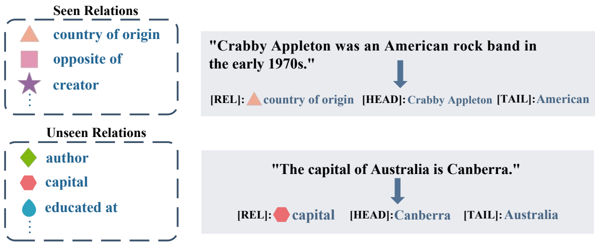


Figure 1: Example of the ZeroRTE task. No overlap between training and testing relations.

may contain noise. Kim et al. (2023) develop ZETT based on template filling and successfully extracts the triplets without the help of synthetic data. Its core idea is to retain pre-trained knowledge by limiting the model's output, thereby enhancing the model's generalization ability. These methods have the disadvantage of not optimizing for the model's generalization and lack the ability to learn generalized knowledge from the data. Although existing large language models (LLMs) have excellent language understanding performance, they still cannot address ZeroRTE tasks well (Li et al., 2024).

The key to tackling ZeroRTE is the improvement of generalization. We believe that task-invariant knowledge can be effectively explored by constructing a set of meta-tasks from the training data. This meta-knowledge can be then modeled through meta-learning frameworks, enabling the model to capture transferable patterns. In this paper, we identified two critical factors governing the generalization through a theoretical analysis: (1) the diversity of meta-tasks and (2) the computational complexity of meta-knowledge representation.

To improve the diversity, we firstly combine the human way of thinking[1] to decouple the RTE task into relation extraction and named entity recognition (NER). Then, we design a metric-based meta-learning module for specifically modeling meta-knowledge between meta-tasks. Lastly,

[1] See appendix for detailed illustration of human thinking.

our framework further optimizes model performance through an innovative active learning mechanism that enables effective engagement with external environments. Specifically, we propose a novel **En**vironmentally **I**nteractive **AC**tive **M**eta-**L**earning (ENIAC-ML) framework containing a **m**etric-based **p**ipelined **m**eta-**l**earning Module (MPML) and an **E**nvironmentally **I**nteractive **A**ctive **l**earning module (EIAL). **MPML** contains a pipelined task design and a metric-based meta-learning approach. Experimental results demonstrate MPML improves the model's generalization capabilities. **EIAL** module actively absorb information such as relations descriptions and entity features from the environment. This module enables the model to actively interact with the environment based on a prototype-based uncertainty during the inference process, eliminating non-critical information or data noise from interfering with model training. It is demonstrated that EIAL is superior to existing data augmentation methods.

In summary, our contribution are as follows.

- For the first time, we explain the determinants that affect the generalization of ZeroRTE task through theoretical analysis, identifying task diversity and meta-knowledge module as the key factors.
- Based on the cognitive mechanisms of human on ZeroRTE, we propose a novel pipelined meta-learning framework and obtain satisfying improvements.
- For the first time, we introduce an uncertainty-driven process for acquiring external knowledge from environment. It can reduce the impact of non-essential information and data noise. Extensive experiments demonstrate its superiority.

2 Problem formulation

Zero-shot Relation Triplet Extraction (ZeroRTE) is formally defined as: Given a dataset $D = (S, T, R)$ where T represents relation triplets of sentences S, and R is a set of relation classes, D is split into a seen $D_S = (S_S, T_S, R_S)$ and an unseen $D_U = (S_U, T_U, R_U)$ with disjoint relation labels ($R_S \cap R_U = \emptyset$), the goal of ZeroRTE model is to train on D_S and generalize to D_U for relation triplet extraction. Each triplet (e_{head}, e_{tail}, r) consists of head/tail entities and a relation $r \in R$.

3 Methodology

3.1 Theoretical analysis

Inspired by Shu et al. (2023), we analyzed the upper bound of meta-learning and tried to design our model based on the analysis.

Assumption 1 (Bounded Inputs) $\mathcal{X} \subset \mathcal{B}(0, R)$, *for* $R > 0$, *where* $\mathcal{B}(0, R) = \{x \in \mathbb{R}^d : \|x\| \leq R\}$.

Assumption 2 (Bounded and Lipschitz Loss Function) *The loss function* $l(\cdot, \cdot)$ *is B-bounded, and* $l(\cdot, y)$ *be L-Lipschitz for any* $y \in \mathcal{Y}$.

Assumption 3 (Task diversity) *Given the meta learning module* $\mathcal{H}$, *it holds that*

$$R_\eta(\hat{h}) - R_\eta(h^*) \leq \alpha \left(R_{train}(\hat{h}, \hat{f}) - R_{train}(h^*, f^*) \right) + \beta, \tag{1}$$

where h represents the model parameters corresponding to the 'learning method' of meta-learning, and f represents the model parameters corresponding to the 'specific task', which together constitute the parameters of the meta-learning model, h^* is the optimal model corresponding to the 'learning method' parameter, $\hat{h}$ represents the learner obtained by minimizing the empirical risk of the training data. η represents the task distribution, and R represents the risk, so that in Assumption 3, the left-hand represents the difference between the empirical and theoretical risk. The right-hand is the upper bound on the risk spread. All the above assumptions are usually satisfied.

Theorem 1 *If Assumptions 1-3 hold, for any* $\delta > 0$, *with probability at least* $1 - \delta$, *we have*

$$R_{test}(\hat{h}, \hat{f}) - R_{test}(h^*, f^*) \lesssim \alpha \left(R_{train}(\wedge) - R_{train}(*) \right) + \tilde{O}\left(\sqrt{\frac{C(f)}{m_\mu}} \right) + \mathbb{E}_{\mu \sim \eta} d_{\mathcal{F}}(\mu^s, \mu^q), \tag{2}$$

$$R_{train}(\hat{h}, \hat{f}) - R_{train}(h^*, f^*) \lesssim \tilde{O}\left(\sqrt{\frac{C(h)}{\sum_{t=1}^{T} n_t}} + \frac{1}{T} \sum_{t=1}^{T} \sqrt{\frac{C(f)}{m_t}} + \frac{1}{T} \sum_{t=1}^{T} d_{\mathcal{F}}(\mu_t^s, \mu_t^q) \right). \tag{3}$$

The final effectiveness of a meta-learning model is determined by the effectiveness of the model on the test task. Based on Assumptions 1-3, we derive an upper bound on the risk of the test task. Where R_{train} is the risk of the training task. $\tilde{O}$ denotes an expression that hides polylogarithmic factors in all problem parameters. $C(\cdot)$ measures the intrinsic complexity of the function class (e.g.,

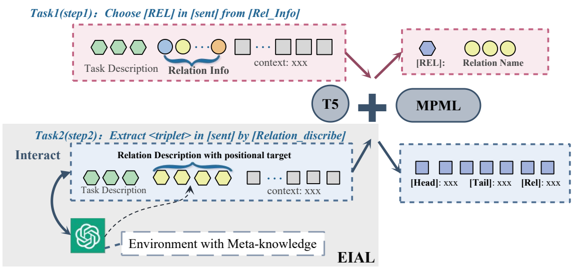


Figure 2: Overview of the proposed framework. The first step is to select true relations contained in the sentence from the candidate relations, and the second step generates triplets based on the relation descriptions.

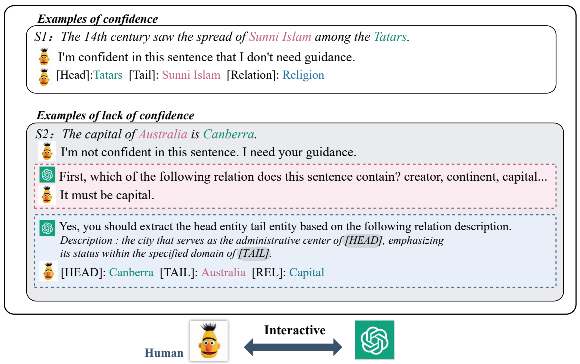


Figure 3: Example of uncertainty-guided active learning. When human or models are not confident enough in solving a task, they can seek more advanced knowledge for help.

VC dimension). m_μ is the sample size of the test task. $d_{\mathcal{F}}(\mu^s, \mu^q)$ denotes the discrepancy divergence between support and query data with respect to their sampled probability distributions μ^s and μ^q imposed on the hypothesis class $\mathcal{F}$.

As shown in Equation (3), the risk upper bound for training task contains: the complexity of independent learning methods h (the first term), the complexity of learning the task-specific model f (the second term), and the distribution shift between support and query sets (the third term). m_t and n_t are the sizes of support set and query set for the t-th task, respectively. T is the number of tasks.

Note that the leading term capturing the complexity of learning methods h decays in terms of the number of task ($\sum_{t=1}^{T} n_t$). And the second term above is of the order $1/T \sum_{t=1}^{T} \mathcal{O}(1/\sqrt{m_t}) \leq \mathcal{O}(1/\sqrt{m})$, $m = \min\{m_1, \cdots, m_T\}$. The third term is only relevant to the division of support sets and query sets. This suggests that increasing task diversity can enhance the model's final performance. Traditional meta-learning methods, however, do not explicitly model learning strategies but embed this knowledge within the entire model. Accordingly, we analyze two approaches to reduce the upper bound of meta-learning and improve model generalization:

- Increasing the diversity of meta-tasks.
- Explicitly modeling the learning methods (meta-knowledge) as modules or functions.

We simulated human problem-solving patterns for the ZS-RTE task through environmental interaction to increase task diversity and incorporated metric-based meta-learning to explicitly model "learning methods" as modular components.

Definition 1 Environment refers to a scenario where the model can interact and access task-related knowledge, such as human experience or relation definitions. Our approach investigates the paradigm of language model-environment interaction, demonstrating that a deeper understanding of relation triplets can enhance the model's performance in ZeroRTE tasks.

3.2 Model overview

As depicted in Figure 2, ENIAC-ML adopts a pipelined inference process involving Relation Extraction (RE) and Named Entity Recognition (NER) subtasks. It consists of two key modules: MPML (Metric-based Pipelined Meta-learning Module) and EIAL (Environmentally Interactive Active Learning Module). MPML defines the subtask formats, including model inputs and outputs, and integrates metric-based meta-learning with backbone models. This architecture separates knowledge between subtasks, enhancing meta-task diversity. Meanwhile, the metric-based meta-learning module explicitly models meta-knowledge, reducing the risk upper bound and improving generalization for the ZS-RTE task.

The EIAL module implements a meta-knowledge interaction component. While external information can enhance model performance, overreliance on it may impair generalization Chia et al. (2022); Gong and Eldardiry (2024). To address this, we introduce a novel approach for acquiring external information while preventing the model from learning non-generalizable knowledge.

As depicted in Figure 3, our proposed ENIAC-ML mimics humans' cognitive processes when dealing with an RTE task. Humans begin by iden-

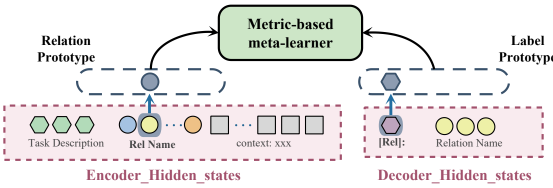


Figure 4: Structure of the MPML module.

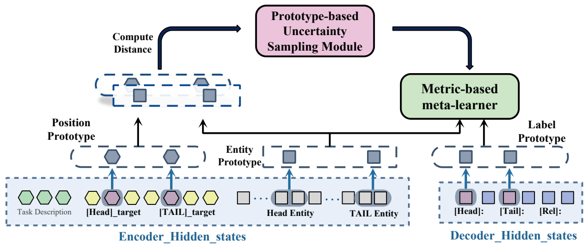


Figure 5: Structure of the EIAL module.

tifying relations contained within the text. When encountering unfamiliar relation categories, they may refer to external sources for supplementary information. They then complete the extraction of head and tail entities by integrating identified relations, personal knowledge, and external information. We posit that the conventional method of utilizing synthetic data for model training lacks the generalizable nature of human interaction.

Building on prior research, we aim to enhance model learning through interactions in environments enriched with high-level bootstrap knowledge. Alt et al. (2019) and Li et al. (2024) indicate that larger models, such as GPT-3.5 and LLaMA, achieve superior performance in Zero-RE tasks, suggesting a deeper understanding of entity relations. By integrating these insights with human problem-solving approaches, we utilize GPT-3.5 to generate high-quality relation descriptions that include meta-knowledge on the locations of head and tail entities. This improves the entity extraction capabilities of models with fewer parameters. Here is an example of a relation description obtained from the agent in the environment (See Appendix B.1 for more descriptions of relations):

"*Date of birth: The time or moment when [HEAD] was born or came into existence, emphasizing its origin or creation within the specified domain or context of [TAIL].*"

3.3 Pipelined Meta-learning Module

3.3.1 Pipelined training

To avoid confusion between RE and NER subtasks and increase the diversity of meta-tasks, we decompose ZeroRTE into two subtasks and train the model using a pipelined approach. We introduce distinct task prompts to differentiate between the subtasks and drive the process.

The first step of the pipeline aims to enable the model to extract unseen relation labels corresponding to sentences. In the second step of the pipeline, we introduce a description of the relation that contains the location of the head and tail entities. Based on the high-level understanding of relations, the model can extract triplets of unseen relations more accurately. The inputs for the two steps are shown below:

$$I_1 = [pad]\, \mathcal{P}_1\, [pad]\, \mathcal{R}_m\, [pad]\, \mathcal{S}\, [eos]\,, \tag{4}$$

$$I_2 = [pad]\, \mathcal{P}_2\, [pad]\, r : \mathcal{D}\, [pad]\, \mathcal{S}\, [eos]\,, \tag{5}$$

where, $\mathcal{R}_m = [Rel\_Info] : \{r_1, r_2, ..., r_m\}$, $\mathcal{P}_1$ and $\mathcal{P}_2$ represent the task prompt of the two steps respectively. m denotes the number of unseen relations, $\mathcal{R}$ encapsulates the relation information containing the names of m relations, and $\mathcal{S}$ signifies the sentence to be processed. $\mathcal{D}$ represents the relation description generated by GPT-3.5 corresponding to r. [pad] and [eos] are special tokens employed by T5 to signify the separation of different segments and the termination of an input. When m=3, an example of an input might be :

- I_1:*<Task1>, Choose [REL] in [SENT] from [Rel_Info]. [Rel_Info]: creator, continent, capital. [SENT]: The capital of Australia is Canberra.*
- I_2:*<Task2>, Extract <triplet> in [SENT] by [Relation_description]. [Relation_description]: capital, the city that serves as the administrative center of [HEAD], emphasizing its status within the specified domain of [TAIL]. [SENT]: The capital of Australia is Canberra.*

We use designated task guidance tokens g_1, g_2 to indicate to the model what step of the pipeline it is at, thus avoiding confusion between the two tasks. The model needs to generate g_1, g_2 at a specific step before generating anything else, where g_1 is "[REL]:" and g_2 is "[HEAD], [TAIL], [REL]". For the running example, the outputs O_1 and O_2 obtained from the two steps are:

- O_1: *[REL]: capital*
- O_2: *[HEAD] Canberra, [TAIL] Australia, [REL] capital.*

Following the existing generative approach, I_1 and I_2 are fed into the encoder of the T5 model, which is based on the Transformer architecture (Vaswani et al., 2017), to obtain the embedding, subsequently fed into the decoder. Finally, our generative model generates results in a predefined order. The training of the generative model aims to maximize the likelihood $L(D)$ in the training set D. The likelihood of our generative model is as follows.

$$\mathcal{L}_{p_1}(S_i) = \prod_{r \in R_i} P((g_1, r) \mid \mathcal{P}_1, \mathcal{R}_m, S_i), \tag{6}$$

$$\mathcal{L}_{p_2}(S_i) = \prod_{(h,t,r) \in T_i} P((g_2, h, t, r) \mid \mathcal{P}_2, r, \mathcal{D}, S_i). \tag{7}$$

3.3.2 Metric-based meta-learning method

Metric-based meta-learning (MEML) methods learn metric-based connections underlying various tasks. These methods typically map input samples to an embedding space and then learn an effective metric space that can quickly find suitable solutions based on similarity metrics within that space when encountering similar samples from similar tasks. We introduced MEML to design a **m**etric-based **p**ipeline **m**eta-**l**earning module (MPML). It has strong generalization ability without generating additional training data, reducing the training cost and making training more flexible.

As shown in Figures 4 and 5, we design a novel feature mapping process, which includes the entity and relation prototype output from the encoder module, a label prototype output from the decoder module and a matching network for predicting matches between both prototypes. We consider the special tokens g_1, g_2 as the label prototype.

We map the relation prototypes encoded by the encoder in step 1, the entity prototypes in step 2, and the labeled prototypes output by the decoder, respectively, to a unified vector space through a linear transformation and predict whether the label prototypes and the corresponding prototypes match using a matching network. The losses of the matching network for each of the two steps of MPML are:

$$\mathcal{L}_{M_1}(S_i) = \sum_{i=1}^{|\mathcal{D}|} CE\left(G(i), MLP\left(E_i^R \odot E_i^L\right)\right), \tag{8}$$

$$\mathcal{L}_{M_2}(S_i) = \sum_{i=1}^{|\mathcal{D}|} CE\left(G(i), MLP\left(E_i^E \odot E_i^L\right)\right), \tag{9}$$

where *G(i)* is the ground-truth of the *i*-th sentence in training set $\mathcal{D}$. The MLP layer measures whether the relation prototype embedding E_i^R or entity prototype embedding E_i^E matches the label prototype embedding E_i^L. $\odot$ denotes the concatenation operation. CE is the cross-entropy loss. The training loss for the entire MPML is:

$$\mathcal{L}_{MPML} = \sum_{i=1}^{|D|} \sum_{j=1}^{n} (\mathcal{L}_{p_j}(S_i) + \mathcal{L}_{M_j}(S_i)), \tag{10}$$

where n is the number of steps included in the pipeline. In the main experiment n is taken to be 2. $\mathcal{L}_{p_j}(S_i)$ correspond to (6) and (7).

We explicitly model the meta-knowledge that the model needs to learn, which reduce the risk upper bound. The model additionally learns a spatial metric pattern about the ZS-RTE task, rather than a single "input-output" distribution. When encountering unseen samples, the model solves the problem based on this "prototype matching" pattern, improving generalizability.

3.4 EIAL Module

Since we introduce a relation description that involves numerous similarity metrics of varying importance after mapping the input samples into the embedding space, it is difficult for the model to efficiently learn the space of metrics associated with Step 2. Inspired by active learning, we designed the **E**nvironmentally **I**nteractive **A**ctive **l**earning Module (EIAL) to enable the model to concentrate on the most pertinent information.

As shown in Figure 5, we introduced a set of position prototypes output by the encoder module. We simulate human learning in active interaction with the environment and then propose two hypotheses: (1) Not all samples benefit training. (2) Not all parts of sentence benefit training.

For Hypothesis 1, we posit that when the model is sufficiently confident in the task analysis of a given sample, additional metric meta-learning is unnecessary. Therefore, we designed the prototype-based uncertainty sampling module to ascertain the model's confidence in the task of the current sample. Since our model operates in metric space, we design a strategy different from traditional uncertainty modeling in active learning.

$$\mathcal{L}_{sample} = \sum_{i=1}^{|\mathcal{D}|} CE\left(G(i), MLP\left(E_i^P \odot E_i^E\right)\right), \tag{11}$$

$$\textit{confidence} = \begin{cases} 1 & \text{if } \mathcal{L}_{sample} \le \alpha \\ 0 & \text{if } \mathcal{L}_{sample} > \alpha \end{cases}, \tag{12}$$

where $\mathcal{L}_{sample}$ represents the matching loss between the position prototype and the entity prototype. Based on (12), the model is considered confident about the current sample when $\mathcal{L}_{sample}$ is less than α, where α is a hyperparameter. In this case, the model does not need to perform metric meta-learning for the current sample. The confidence level can be interpreted as whether the head entity and tail entity in the input sentence can be used as the words in the positions of the head entity and tail entity in the relation description.

$$\mathcal{L} = \mathcal{L}_{p_2}(S_i) + \beta \cdot \textit{confidence} \cdot \mathcal{L}_{M_2}(S_i). \quad (13)$$

The underpinnings of Hypothesis 2 were already explained in the MPML section. With the introduction of the EIAL module, the loss in the second step is denoted as $\mathcal{L}$ in (13). β is a hyperparameter, and *confidence* is the result of the prototype-based uncertainty sampling module.

4 Experiments

4.1 Experimental setup

4.1.1 Datasets

As follows, we evaluate our model on two public datasets: FewRel (Han et al., 2018) and Wiki-ZSL (Chen and Li, 2021). The detailed data statistics are shown in Appendix B.2.

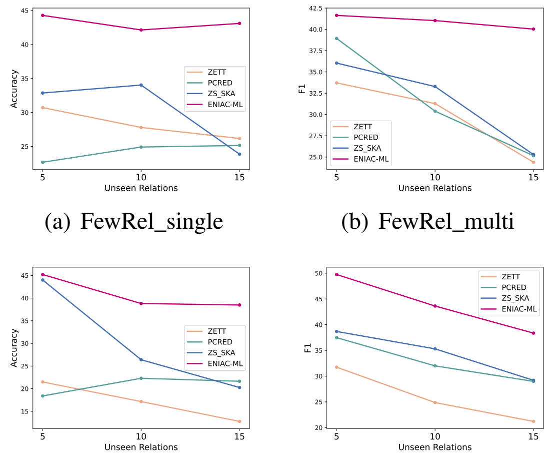
Table 1: Statistics of FewRel and Wiki-ZSL.

Dataset	#Samples	#Entities	#Relations	Sent_len
FewRel	56,000	72,954	80	24.95
Wiki-ZSL	94,383	77,623	113	24.85

4.1.2 Experimental settings

1) We follow the setup of (Chia et al., 2022) for training and evaluation: We maintain disjoint relation types across training, validation, and test splits. 2) We evaluate different methods under varying settings for the size of unseen relation types ($m \in \{5, 10, 15\}$). 3) To mitigate experimental noise, we repeat experiments using different data folds wherein relation types are split with varying random seeds: $\{0, 1, 2, 3, 4\}$. Table 2 also presents the statistics for each dataset and setting.

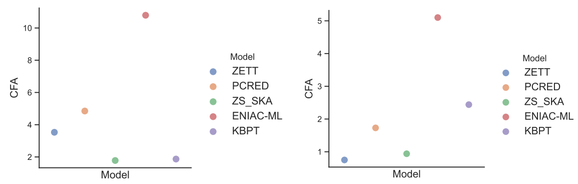
We utilize T5-base (Zong et al., 2021), which comprises 220 million parameters, as our pretrained generative model. The learning rates for the generative model parameters and other parameters are set to 3×10^{-5} and 6×10^{-4}, respectively, and the batch size for training is set to 1. Both α and β in (12) and (13) of the experiment were experimented in the range of 0.1-0.9, and we ended up using $\alpha = 0.1$ and $\beta = 0.2$ to arrive at the final result.



(a) FewRel_single (b) FewRel_multi

(c) Wiki-ZSL_single (d) Wiki-ZSL_multi

Figure 6: Experimental results of different models for single and multi ZeroRTE tasks on FewRel and Wiki-ZSL datasets, respectively.



(a) $\mathcal{CFA}$ on FewRel (b) $\mathcal{CFA}$ on Wiki-ZSL

Figure 7: $\mathcal{CFA}$ for different models in FewRel and Wiki-ZSL respectively. Higher value mean higher accuracy and stability of the model.

4.1.3 Evaluation Metrics

To evaluate the performance of our model, we adhere to the same evaluation metrics as RelationPrompt (Chia et al., 2022) for a fair comparison. We separately report the scores for sentences containing a single triplet and those containing multiple triplets to maintain consistency with previous studies. For single triplet extraction, we employ Accuracy (Acc.) as the evaluation metric; for multiple triplet extraction, we utilize the Micro-F1 score (F1) as the evaluation metric, additionally reporting precision (Pre.) and recall (Rec.) scores. All scores are averaged across five data folds.

4.2 Baseline methods

For more information about baseline, please refer to Appendix B.3

4.3 Main Results

The results of ZeroRTE on two datasets are presented in Table 2. We highlight key observations as follows.

Table 2: Main Results. The best scores are in bold, and the second-best ones are underlined.

Labels	Model	Single-triplet		Multi-triplet					
		Wiki-ZSL	**Fewrel**	**Wiki-ZSL**			**Fewrel**		
		Acc.	Acc.	Pre.	Rec.	F1	Pre.	Rec.	F1
m=5	1) TabSeq	14.47	11.82	43.68	3.51	6.29	15.23	1.91	3.40
	2) RelPrompt	16.64	22.27	29.11	31.00	30.01	20.80	24.32	22.34
	3) KBPT	17.85	24.19	32.45	31.64	32.04	23.15	23.13	24.28
	4) ZETTbase	21.49	30.71	35.89	28.38	31.74	38.14	30.58	33.71
	5) PCRED	18.40	22.67	38.14	36.84	37.48	43.91	34.97	38.93
	6) GPT-3.5	17.19	30.10	12.49	21.71	15.85	37.76	**60.87**	**44.21**
	7) MICRE	27.74	37.53						
	8) ZS-SKA	44.00	32.86	**66.70**	27.24	38.68	**57.50**	26.24	36.04
	9) ENIAC-ML	**45.21**	**44.28**	47.40	**52.56**	**49.74**	36.97	47.71	41.64
m=10	1) TabSeq	9.61	12.54	45.31	3.57	6.40	28.93	3.60	6.37
	2) RelPrompt	16.48	23.18	30.20	32.31	31.19	21.59	28.68	24.61
	3) KBPT	20.45	26.58	32.47	33.69	33.17	24.35	27.28	26.46
	4) ZETTbase	17.16	27.79	24.49	26.99	24.87	30.65	32.44	31.28
	5) PCRED	22.30	24.91	27.09	39.09	32.00	30.89	29.90	30.39
	6) GPT-3.5	14.44	23.32	8.82	17.78	11.79	26.97	**48.71**	32.56
	7) MICRE	24.64	34.77						
	8) ZS-SKA	26.40	34.03	**45.38**	29.27	35.30	**60.48**	23.22	33.28
	9) ENIAC-ML	**38.81**	**42.14**	39.85	**48.32**	**43.62**	36.29	47.21	**41.03**
m=15	1) TabSeq	9.20	11.65	**44.43**	3.53	6.39	19.03	1.99	3.48
	2) RelPrompt	16.16	18.97	26.19	32.12	28.85	17.73	23.20	20.08
	3) KBPT	20.31	22.46	32.15	29.39	30.74	19.61	25.55	22.19
	4) ZETTbase	12.78	26.17	19.45	23.31	21.21	22.50	27.09	24.39
	5) PCRED	21.64	25.14	25.37	33.80	28.98	27.00	23.55	25.16
	6) GPT-3.5	11.01	16.41	7.13	17.21	10.08	20.72	39.30	25.09
	7) MICRE	22.23	32.42						
	8) ZS-SKA	20.26	23.86	31.23	27.20	29.19	**37.29**	19.13	25.29
	9) ENIAC-ML	**38.48**	**43.10**	35.16	**42.63**	**38.36**	36.12	**45.05**	40.03

ENIAC-ML outperforms the latest baseline, ZS-SKA, which has a comparable number of parameters, in both single and multiple triplet extraction tasks. On Wiki-ZSL and FewRel, ENIAC-ML achieves accuracy gains of 1.21% to 19.24% for single triplets and F1 score improvements of 5.6% to 15.43% for multiple triplets. While its precision is not always the highest, its balanced precision and recall result in superior F1 scores. Compared to LLaMA (7B), ENIAC-ML (220M) achieves 15.96% and 8.27% higher accuracy on Wiki-ZSL and FewRel, respectively (Li et al., 2024). We attribute LLaMA's limited performance to its generalized training, which is not optimized for ZeroRTE tasks. This advantage is attributed to LLaMA's generalized training, which is not optimized for ZeroRTE tasks. Our results show that integrating meta-knowledge and active learning enables ENIAC-ML to adapt quickly and generalize effectively, even with fewer parameters. By decoupling ZS-RTE through a human-inspired approach and modeling meta-knowledge independently via metric-based meta-learning, we reduce the meta-learning risk upper bound and enhance model generalization.

As m increases, model effectiveness tends to diminish or fluctuate due to a rise in unseen relations and judgment errors. Figure 6 shows that in single triplet extraction, ENIAC-ML maintains the highest average accuracy without significant decay or fluctuation as m increases. Specifically, ENIAC-ML's accuracy decay rates on Wiki-ZSL and FewRel are 8% and 1%, respectively, compared to ZS-SKA's 32% and 13%. Similar trends are observed for F1 scores in multiple triplet extraction. To evaluate both average accuracy and fluctuation, we introduce the Combined Fluctuation Average ($\mathcal{CFA}$) metric, defined as follows:

$$\mathcal{CFA} = \frac{\psi}{\vartheta} \tag{14}$$

where ψ represents the average precision (single triplet extraction) or average F1 score (multiple triplet extraction) of the model on a given dataset, and ϑ represents the average rate of variation of the accuracy or F1 score as m varies. See Appendix B.4 for a more detailed derivation of this formula. The larger the value of $\mathcal{CFA}$, the more accurate and stable the model. The $\mathcal{CFA}$ distributions for the other baseline models are shown in Figure 7. This demonstrates that ENIAC-ML has not only mastered the comprehension of specific samples but also the meta-knowledge of the ZeroRTE task by the concept of meta-learning, "learning to learn". And thereby explaining its superior performance in terms of accuracy, stability, and generalization.

	Relation	Text	RelationPrompt Prediction	ZETT Prediction	ENIAC-ML Prediction
S1:	Religion	The 14th century saw the spread of Sunni Islam among the Tatars.	Head entity: Tatars Tail entity: Sunni Islam Relation: Religion	Head entity: Tatars Tail entity: Sunni Islam Relation: Religion	Head entity: Tatars Tail entity: Sunni Islam Relation: Religion
S2:	Distributed by	Netflix published in September 2014 the documentary Print the Legend about Makerbot history .	Head entity: Makerbot history Tail entity: Netflix Relation: Distributed by	Head entity: Print the Legend Tail entity: Netflix Relation: Distributed by	Head entity: Print the Legend Tail entity: Netflix Relation: Distributed by
S3:	Location	He then finished 15th at the 1956 World Championships in Garmisch - Partenkirchen, Germany .	Head entity: Garmisch - Partenkirchen Tail entity: Germany Relation: Location	Head entity: He Tail entity: Garmisch - Partenkirchen Relation: Location	Head entity: 1956 World Championships Tail entity: Garmisch - Partenkirchen Relation: Location

Figure 8: Case study from three models. ENIAC-ML demonstrated better accuracy.

4.4 Ablation Study

As illustrated in Table 3, we assessed the performance of each component on FewRel and Wiki-ZSL datasets using the accuracy. NoGen represents the RelationPrompt without additional generated data. In contrast, the PT5 model employs a pipelined approach: it first identifies relations within the sentence and then extracts the corresponding head and tail entities based on these relations. This method mitigates the confusion between the two subtasks of ZeroRTE, leading to enhanced model accuracy.

In addition, the accuracy of the model with the introduction of the relation description which contains the position of the head and tail entities is also substantially improved compared to PT5. It can be inferred that this relation description with positional information enhances the model's generalization. This is a testament to the importance of modeling how humans learn.

Furthermore, we validate the effectiveness of MPML and EIAL, respectively. MPML demonstrates that employing metric-based meta-learning to bridge the distance between label prototypes and entity prototypes can effectively enhance the model's judgment of head, tail entities and relations. EIAL demonstrates that selectively enhancing sample learning based on the model's confidence reduces training overhead and boosts accuracy. Integrating all components into ENIAC-ML yields significant performance gains over existing ZeroRTE models.

Table 3: The ablation experiment results.

PT5	Discription	MPML	EIAL	FewRel Acc.	Wiki-ZSL Acc.
RelationPrompt (NoGen)				11.49	9.05
✓				22.75	20.77
✓	✓			39.91	38.22
✓	✓	✓		40.97	38.86
✓	✓		✓	41.03	39.02
✓	✓	✓	✓	43.17	40.83

4.5 Case Study

To analyze active meta-learning in our framework, we compare the relation triplets extracted by RelationPrompt, ZETT, and ENIAC-ML for three sentences (denoted as S1, S2, and S3). The results are shown in Figure 8.

In S2, RelationPrompt correctly identifies the relation but predicts incorrect head and tail entities due to its reliance on synthetic training samples that may not include relevant data. In contrast, both ZETT and ENIAC-ML extract the correct triplet. ZETT achieves this by leveraging the 'Distributed by' template, while ENIAC-ML benefits from metric-based meta-learning, which reduces the feature-space distance between 'Print the Legend' and '[HEAD]'.

For S3, RelationPrompt again fails to predict the correct head and tail entities. ZETT does not explicitly model entity position information, leading to confusion in head and tail entity location. ENIAC-ML, however, employs active meta-learning to detect prediction uncertainty in different sentence components and applies metric-based meta-learning to samples requiring additional training. This allows ENIAC-ML to accurately recognize and predict the positions of head and tail entities

5 Conclusion

This paper investigates the determinants influencing the generalization of ZeroRTE via a theoretical analysis, identifying task diversity and the meta-knowledge module as key factors. The optimal performance was achieved by using the task decomposition paradigm that mimics human thinking, metric-based meta learning, and active environmental interaction module. These methods achieve state-of-the-art performance while offering valuable insights for future research on generalization.

6 Limitation

While this work represents a significant improvement over previous ZeroRTE tasks, it is necessary to acknowledge the limitations of this work.

Firstly, we found that the accuracy of relation judgment in the first step significantly affects the accuracy of the final result. This is because the additional environmental information added in the second step pertains to the description of the relation extracted in the first step. If the relations in the first step are extracted incorrectly, the erroneous descriptions of the relations in the second step will interfere with the model's judgment of the head and tail entities. Secondly, the evaluation metrics used in this study may only reflect task-specific performance and may not comprehensively measure the model's usability and efficiency in real-world applications.

To address these limitation, future research should implement a protective warning module in the second step, ensuring that if the model lacks confidence in the extracted relation, it refrains from adding the corresponding description. And the future work could consider incorporating additional evaluation dimensions (e.g., computational resource consumption, inference time, etc.) to provide a more comprehensive assessment of the model's usability and efficiency in real-world applications.

References

Christoph Alt, Marc Hübner, and Leonhard Hennig. 2019. Fine-tuning pre-trained transformer language models to distantly supervised relation extraction. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1388–1398. Association for Computational Linguistics.

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*, pages 3470–3479. Association for Computational Linguistics.

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 Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 45–57. Association for Computational Linguistics.

Lianzhai Duan. 2024. Research on relation extraction method based on active learning. *International Journal of Computer and Information System (IJCIS)*, 5(2):92–101.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR.

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*, pages 13143–13156. ELRA and ICCL.

Qian Guo, Yi Guo, and Jin Zhao. 2024. Kbpt: knowledge-based prompt tuning for zero-shot relation triplet extraction. *PeerJ Computer Science*, 10:e2014.

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*, pages 2605–2616. Association for Computational Linguistics.

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*, pages 4803–4809. Association for Computational Linguistics.

Bosung Kim, Hayate Iso, Nikita Bhutani, Estevam Hruschka, Ndapa Nakashole, and Tom M. Mitchell. 2023. Zero-shot triplet extraction by template infilling. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, IJCNLP 2023 -Volume 1: Long Papers, Nusa Dua, Bali, November 1 - 4, 2023*, pages 272–284. Association for Computational Linguistics.

Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. 2015. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille.

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*.

Yoonho Lee and Seungjin Choi. 2018. Gradient-based meta-learning with learned layerwise metric and subspace. In *International Conference on Machine Learning*, pages 2927–2936. PMLR.

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. *arXiv preprint arXiv:2404.17807*.

Huaiyu 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 *International conference on machine learning*, pages 3825–3834. PMLR.

Wanli Li and Tieyun Qian. 2023. Generative meta-learning for zero-shot relation triplet extraction. *arXiv preprint arXiv:2305.01920*.

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*, pages 757–763. Association for Computational Linguistics.

Alex Nichol, Joshua Achiam, and John Schulman. 2018. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*.

Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. 2019. Meta-learning with latent embedding optimization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Seungmin Seo, Byungkook Oh, Jeongbeom Jeong, Donghyun Kim, Kyong-Ho Lee, Dong-Hoon Shin, and Yeonsoo Lee. 2023. Active learning for cross-sentence n-ary relation extraction. *Information Sciences*, 645:119328.

Burr Settles. 2009. Active learning literature survey.

Jun Shu, Deyu Meng, and Zongben Xu. 2023. Learning an explicit hyper-parameter prediction function conditioned on tasks. *The Journal of Machine Learning Research*, 24(1):8818–8891.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. *Advances in neural information processing systems*, 29.

Jue Wang and Wei Lu. 2020. Two are better than one: Joint entity and relation extraction with table-sequence encoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1706–1721.

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, et al. 2023. Zero-shot information extraction via chatting with chatgpt. *arXiv preprint arXiv:2302.10205*.

Kun Xu, Siva Reddy, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2016. Question answering on freebase via relation extraction and textual evidence. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Liang Xu, Changxia Gao, and Xuetao Tian. 2024. Domain-control prompt-driven zero-shot relational triplet extraction. *Neurocomputing*, 574:127270.

Qi Ye, Tingting Cai, Xiang Ji, Tong Ruan, and Hong Zheng. 2023. Subsequence and distant supervision based active learning for relation extraction of chinese medical texts. *BMC Medical Informatics and Decision Making*, 23(1):34.

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*, pages 646–653. Association for Computational Linguistics.

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*, pages 27075–27098. PMLR.

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. 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*, pages 50–61. Association for Computational Linguistics.

Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli. 2021. Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*.

A Related Work

A.1 Meta-learning

Meta-learning can improve the training of machine learning models and thus has attracted significant interest in recent years. The conventional categorizations of meta-learning methods (Lee and Choi, 2018) categorize them into three groups: optimization-based, metric-based, and model-based methods.

The optimization-based methods (Rusu et al., 2019; Finn et al., 2017; Nichol et al., 2018) focus on incorporating optimization within the learning process to achieve an optimized initialization of model parameters. The metric-based methods (Koch et al., 2015; Vinyals et al., 2016; Snell et al., 2017) aim to learn an appropriate distance metric for few-shot classification and have been successfully applied to various few-shot and zero-shot tasks (Han et al., 2021; Liu et al., 2022). The model-based methods (Zhmoginov et al., 2022; Li et al., 2019; Ye and Ren, 2021) involve task specifications to directly generate or modulate model weights.

Following this line, TGM method (Li and Qian, 2023) pioneered meta-learning for generative models in the ZeroRTE. A task-aware generative model combined with three generative meta-learning approaches significantly improved over previous state-of-the-art models. Inspired by this method, we combine meta-learning with our pipelined framework.

A.2 Active learning

In supervised learning problems, labeling is expensive, and labels are difficult to obtain in large quantities. For certain specific tasks, only industry experts can accurately label the samples. In this context, Active learning seeks to save resources by selectively labeling fewer data to train better-performing models (Settles, 2009).

Various active learning algorithms have been implemented for RE tasks (Duan, 2024). Seo et al. (2023) propose an active learning method for a cross-sentence n-ary relation extraction (ANRE), which allows models to be trained on a small amount of labeled data initially. The proposed methods acquire newly labeled train data iteratively and improve the model. Ye et al. (2023) propose a method of active learning based on subsequences and distant supervision. The method annotates by selecting information-rich subsequences as sampling units. Uncertainty can provide useful information to the model, so we try to introduce the uncertainty mechanism into ZeroRTE task.

A.3 Zero-shot relation triplet extraction

Relation triplet extraction has been studied for a long time. Recently, PURE (Zhong and Chen, 2021) proposes a simple and effective pipelined RTE method that refers to the decomposition of the task into RE and NER. In PURE, the results of NER are used to assist RE in a pipeline manner. They demonstrate the importance of learning distinct contextual representations for entities and relations. However, the performance of the pipeline model on the ZeroRTE task remains unexplored.

Most of existing ZeroRTE methods model NER and RE jointly. RelationPrompt (Chia et al., 2022) is a prominent approach for extracting entire triplets in a zero-shot setting. It trains a generator and uses the generated samples to synthesize data for unseen relations. However, its performance is severely constrained by the quality of the synthetic data. Guo et al. (2024) proposed KBPT, a method incorporating prior knowledge from ontological schemas, which enhances semantic representations. ZS-SKA (Gong and Eldardiry, 2024) implements data augmentation through word-level sentence translation to generate instances containing unseen relations from training instances containing seen relations and use the generated data as a training set for unseen relations. However, the reliance of these methods on synthesized data increases the training costs and potentially compromises accuracy. In contrast, Kim et al. (2023) view relation extraction as a template-filling problem, thus eliminating the need for additional training data. Their ZETT approach fine-tuned the T5 model to obtain the ranking scores for potential triplets. However, evaluating and scoring many unseen relations is exceedingly time-consuming, and lexical variants that appear in the relation descriptions and resemble the words in the sentence interfere with the model's judgments.

This paper makes a first attempt to introduce the pipeline design into ZeroRTE and explore a better form of it.

B Experiments

B.1 Relation description

We used GPT-3.5 to generate relation descriptions with information about the location of the head and tail entities, and Figure 9 shows a few examples. The descriptions corresponding to all other relations we show in the json file of the code.

B.2 Datasets

We evaluate our model on two public datasets: FewRel and Wiki-ZSL. FewRel is a standard benchmark dataset designed primarily for the few-shot relation extraction task. It is created using distant supervision and has been additionally filtered by humans. Wiki-ZSL is generated with distant supervision from Wikipedia articles and the Wikidata knowledge is a subset of Wiki-KB, targeting zero-shot relation extraction. We use dataset versions released by Relationprompt, which have been transformed for zero-shot triplet extraction. The detailed data statistics are shown in Table 4.

B.3 Baseline methods

We compare ENIAC-ML with the following methods: **1) TableSequence** (Wang and Lu, 2020) is a joint learning model employing two distinct encoders to simultaneously perform RE and NER. It uses data from (Chia et al., 2022) to train the models and then report the results; **2) RelationPrompt** (Chia et al., 2022) comprises a relation generator and a relation triplet extractor. **3) KBPT** (Guo et al., 2024) incorporate prior knowledge from ontological schemas and employ a generative prompt model to synthesize training samples for unseen relational types. **4) ZETT** (Kim et al., 2023) treats zero-shot relational triplet extraction as a template-filling task and employs a generative model to predict the subject and object of each relation. **5) PCRED** (Lan et al., 2022) directly utilizes the semantics of unseen relations, thereby incurring no additional data or training costs, instead of leveraging PLMs to generate training samples for unseen relations. **6) ChatIE** (Wei et al., 2023) employs GPT-3.5 for the zero-shot RTE task. This method first defines the relations to be extracted and then generates relation triplets in the sentence. Xu et al. (2024) applied it to the Fewrel and Wiki-ZSL datasets. **7) MICRE** (Li et al., 2024) introduces a novel in-context training framework based on LLMs for zero- and few-shot RE, using in-context learning techniques to enhance few-shot prompting performance on unseen RE tasks. We use the experimental results of LLaMa on ZeroRTE from this study. **8) ZS-SKA** (Gong and Eldardiry, 2024) first implements data augmentation through word-level sentence translation to generate augmented instances with unseen relations from training instances with seen relations.

B.4 Definition of CFA

In this section we detail the $\mathcal{CFA}$ calculation process. As (15), (16) and (17) shown, Where m represents the number of unseen relations, assumed to grow in units of 5, i.e., $m \in \{5, 10, 15, ...\}$, while q is taken to be 5. $\frac{m}{q}$ represents the number of groups into which the dataset is divided based on m. For single-triplet extraction tasks, a_i represents the accuracy of a model on a particular set of tasks (For different unseen relations), and for multi-triplet extraction tasks, a_i represents the F1 score. ψ represents the average precision or average F1 score of the model on a given dataset, and ϑ represents the volatility of the precision or F1 score. The combined fluctuation average indicator $\mathcal{CFA} = \frac{\psi}{\vartheta}$.

$$\psi = \frac{\sum_{i=1}^{\frac{m}{q}} a_i}{\frac{m}{q}} \tag{15}$$

$$\vartheta = \frac{\sum_{i=1}^{\frac{m}{q}-1} \frac{|a_{i+1} - a_i|}{a_{i+1}}}{\frac{m}{q} - 1} \tag{16}$$

$$\mathcal{CFA} = \frac{\psi}{\vartheta} \tag{17}$$

C Model Complete Input-Output Presentation

As shown in Figure 10, these are three examples of complete inputs and outputs of ENIAC-ML.

D Human thinking.

In real-world scenarios, humans tackling Zero-Shot Relation Triple Extraction (ZeroRTE) tasks typically adopt a structured and interactive approach to comprehend and solve the problem. Initially, they engage with knowledgeable individuals, such as domain experts, or consult external materials, such as textbooks, databases, or online resources, to gain a thorough understanding of the semantics and nuances of relation types embedded within sentences. This step is crucial because it allows them to build a foundational understanding of how different relations are expressed and contextualized in text. Once they have grasped the semantics of the relation types, they leverage this knowledge to make informed judgments about the entities involved, identifying the head and tail entities that correspond to the extracted relations.

Relation		Description of relation with targets
R1	nominated for	The status of [HEAD] being proposed or suggested as a potential recipient of recognition or an award, emphasizing its candidacy within the specified domain or context of [TAIL].
R2	spouse	The marital relationship between [HEAD] and their partner, indicating the legally or socially acknowledged bond within the specified family or context of [TAIL].
R3	cast member	The actor or performer who appears in [HEAD], emphasizing its role or function within the specified domain or context of [TAIL].
R4	date of birth	The time or moment when [HEAD] was born or came into existence, emphasizing its origin or creation within the specified domain or context of [TAIL].

Figure 9: Relation description with head and tail entity location information generated by GPT-3.5.

Text	Step1 Input	Step2 Input	Final Output
He then finished 15th at the 1956 World Championships in Garmisch - Partenkirchen, Germany .	**<Task1>**, Choose [REL] in [SENT] from [Rel_Info]. **[Rel_Info]**: creator, continent, capital, location, sibling **[SENT]**: He then finished 15th at the 1956 World Championships in Garmisch - Partenkirchen, Germany.	**<Task2>**, Extract <triplet> in [SENT] by [Relation_description]. **[Relation_description]**: location, the adherence of [HEAD] to a specific set of spiritual, emphasizing its connection to a particular belief system within the specified domain or context of [TAIL].	**T1**: [REL] location **T2**: [HEAD] 1956 World Championships [TAIL] Garmisch - Partenkirchen [REL] location
There is another Italian river named "Arrone" that flows into the Tyrrhenian Sea .	**<Task1>**, Choose [REL] in [SENT] from [Rel_Info]. **[Rel_Info]**: occupant, country, director, genre, child **[SENT]**: There is another Italian river named "Arrone" that flows into the Tyrrhenian Sea.	**<Task2>**, Extract <triplet> in [SENT] by [Relation_description]. **[Relation_description]**: country, [TAIL] has defined boundaries, governments, and [HEAD] is a geographic, political marker or other component thereof.	**T1**: [REL] country **T2**: [HEAD] Arrone [TAIL] Italian [REL] country
Shortly after the release of the GP32 in 2001, its maker Game Park began to design their next handheld.	**<Task1>**, Choose [REL] in [SENT] from [Rel_Info]. **[Rel_Info]**: producer, country, manufacturer, use, family **[SENT]**: Shortly after the release of the GP32 in 2001, its maker Game Park began to design their next handheld.	**<Task2>**, Extract <triplet> in [SENT] by [Relation_description]. **[Relation_description]**: manufacturer, the role played by [HEAD] in the creation of various items, emphasizing their involvement in the production process within the specified domain of [TAIL].	**T1**: [REL] manufacturer **T2**: [HEAD] GP32 [TAIL] Game Park [REL] manufacturer

Figure 10: Complete inputs and outputs of ENIAC-ML.

	Samples	Entities	Relation Labels				Average Length
			Total	Train	Validation	Test	
Wiki-ZSL	94383	77623	113	103	5	5	24.85
				98	5	10	
				93	5	15	
FewRel	56000	72954	80	70	5	5	24.95
				65	5	10	
				60	5	15	

Table 4: Statistics of FewRel and Wiki-ZSL.