A Two-Agent Game for Zero-shot Relation Triplet Extraction

Anonymous ACL submission

Abstract

Relation triplet extraction is a fundamental task in natural language processing that aims to identify semantic relationships between entities in text. It is particularly challenging in the zero-shot setting, i.e., zero-shot relation triplet extraction (ZeroRTE), where the relation sets between training and test are disjoint. Existing methods deal with this task by integrating relations into prompts, which may lack sufficient understanding of the unseen relations. To address these limitations, this paper presents a novel Two-Agent Game (TAG) approach to deliberate and debate the semantics of unseen relations. TAG consists of two agents, a generator and an extractor. They iteratively interact in three key steps: attempting, criticizing, and rectifying. This enables the agents to fully debate and understand the unseen relations. Experimental results demonstrate consistent improvement over ALBERT-Large, BART, and GPT3.5[1], without incurring additional inference costs in all cases. Remarkably, our method outperforms strong baselines by a significant margin, achieving an impressive 6%-16% increase in F1 scores, particularly when dealing with FewRel with five unseen relations[2].

1 Introduction

Relation triplet extraction (RTE; Miwa and Bansal, 2016) is a pivotal task in information extraction (Yang et al., 2022), which aims to extract the relation triplets in the form of *(head entity, tail entity, relation label)* within a given sentence. RTE boosts a broad spectrum of downstream applications across domains, such as machine reading comprehension (Qiu et al., 2019) and machine translation (Zhao et al., 2020). Existing methods (Zheng et al., 2017; Wang and Lu, 2020) rely on large amounts of labeled data, limiting the scalability and applicability of these methods.

Sentence	Relation Triplets
He was the father of Kate and Frank.	(Kate, Frank, sibling)
The song is "Hello" by Adele.	(Hello, Adele, performer)

(a) Training samples of seen relations.

Sentence	Relation Triplets
Cynidr was the son of Gwladys.	(Cynidr, Gwladys, mother)
The book is "It" by Stephen King.	(It, Stephen King, writer)

(b) Testing samples of unseen relations.

Figure 1: Examples for zero-shot relation triplet extraction. The relation sets between training and test are disjoint. The head and tail entities are shown in blue and yellow, respectively. The relations are shown in red.

To reduce the overreliance on labeled data, Chia et al. (2022) introduce a challenging task, zero-shot relation triplet extraction (ZeroRTE), where relation sets during the training and test stages are disjoint. For example, in Fig. 1, training samples may belong to the seen relation set {*sibling, performer*}, while test samples may belong to the unseen relation set {*mother, writer*}. Generalizing knowledge from the training set to the test set is critical.

Existing methods deal with ZeroRTE by formulating it into prompts to leverage the power of language models. The main idea is to fine-tune the model on seen relations from the training dataset. Next, the model adapts to unseen relations by integrating the semantics of those unseen relations into the prompt templates (Chia et al., 2022; Lv et al., 2023; Lan et al., 2022; Kim et al., 2022). Recent research based on large language models (LLMs) removes the fine-tuning stage and integrates the semantics of relations via chain-of-thought few-shot prompting (Wadhwa et al., 2023) or multi-turn question answering (Wei et al., 2023b).

Despite achieving favorable results, simply integrating relations into prompts is superficial and may lead to an insufficient understanding of the unseen relations. For example, consider the two rela-

[1] In the following section, we use the term GPT-3.5 to refer to the gpt3.5-turbo model. https://platform.openai.com/docs/guides/gpt/chat-completions-api

[2] Our code will be publicly available.

tions, "*located on terrain feature*" and "*contains administrative territorial entity*". Both deal with location but a subtle difference exists between the two relations. Merely merging them into the prompt without distinguishing their nuances may cause confusion between similar relations, ultimately resulting in a misunderstanding of the relations.

To tackle the above issues, we construct a two-agent game (TAG) to facilitate a process where both agents can engage in deliberation and debate to grasp the semantics of unseen relations. TAG aims to comprehensively understand the nuances of unseen relations to improve the performance in the subsequent test phase. Specifically, TAG consists of two agents: an extractor responsible for extracting relation triplets from sentences and a generator for generating sentences and triplets based on the given relation. They deliberate and debate through three key steps: attempting, criticizing, and rectifying. First, the generator *attempts* to understand the unseen relation by generating sentences and triplets associated with the relation. These initial attempts will serve as the target of debate in the later stages of the game. Subsequently, the extractor *criticizes* the generator on the quality of the synthetic data via evaluating the semantic matching degree between sentences and triplets, with higher log-likelihood values indicating better quality. Finally, we use the results of attempting as data and the results of criticizing as rewards to *rectify* both agents through reinforcement learning. In this way, both agents can learn the preference between synthetic data and tend to produce high-quality data. Through this deliberation and debate process, two agents gradually develop a comprehensive understanding of unseen relations. Experiments demonstrate that TAG achieves consistent improvement across different model scales from ALBERT (Lan et al., 2020) with 18M parameters to large language models like GPT-3.5.

Our contributions are three-fold: (1) We introduce a novel two-agent game (TAG) method for ZeroRTE that facilitates deliberation and debate on unseen relations. This approach enables the generator and extractor agents to enhance their understanding of these relations collaboratively. (2) TAG is a versatile approach that can be applied to different backbone models, i.e., ALBERT, BART, and GPT-3.5. Moreover, during deployment, the generator agent can be omitted without incurring additional computation and storage costs for the original extractor agent. (3) Experimental results demonstrate that TAG consistently improves the performance across different backbone models. This highlights the effectiveness and scalability of our approach.

2 Preliminaries

Before delving into the details, we introduce the task definitions of RTE and ZeroRTE. And then introduce the definitions of the two agents in our approach, the generator and extractor. To make the notations consistent throughout the paper, we define the important ones in Table 5 in the appendix.

2.1 Task Definitions

RTE Given a dataset $\mathcal{D} = \{(s_i, t_i)\}_{i=1}^{|\mathcal{D}|}$, where $s_i \in \mathcal{S}$ represents the i-th input sentence and $t_i \in \mathcal{T}$ represents the corresponding output triplet, Relation Triplet Extraction (RTE) aims to extract relation triplet $t \in \mathcal{T}$ from a sentence $s \in \mathcal{S}$, following the form $t = (e^{head}, e^{tail}, r)$. Here, the head entity e^{head} and the tail entity e^{tail} are represented as token spans or word sequences referring to real-world entities. The relation r belongs to the set $\mathcal{R}$, encompassing a predefined collection of relations between the head and tail entities.

ZeroRTE The objective of ZeroRTE (Chia et al., 2022) is to leverage the knowledge from the seen dataset $\mathcal{D}^s$ and generalize to the unseen dataset $\mathcal{D}^u$. Let $\mathcal{D}^s$ and $\mathcal{D}^u$ represent the training and test datasets, respectively, derived from the original full dataset $\mathcal{D}$. The relation sets during training and test are denoted as $\mathcal{R}^s = \{r_1^s, r_2^s, \ldots, r_n^s\}$ and $\mathcal{R}^u = \{r_1^u, r_2^u, \ldots, r_m^u\}$, where $n = |\mathcal{R}^s|$ and $m = |\mathcal{R}^u|$ indicate their respective sizes. Importantly, it is worth noting that ZeroRTE does have training data $\mathcal{D}^s$; zero-shot refers to the fact that the relation sets for training and test are disjoint, i.e., $\mathcal{R}^s \cap \mathcal{R}^u = \emptyset$.

2.2 Agent Definitions

Generator The generator $\mathcal{G}$ aims to generate a relation-specific sentence s and triplet $t = (e^{head}, e^{tail}, r^s)$ given the relation r^s. To adapt to language models, we define two transformations $E_{in}(r^s)$ and $E_{out}(s, e^{head}, e^{tail})$, mapping the input and output into natural language space. They transform structured information into text sequences using the template defined by E_{in} and E_{out}. In this work, we use a simple transformation function: $E_{in}(r^s)$ = "Relation: r^s" and $E_{out}(s, e^{head}, e^{tail})$ = "Head Entity: e^{head}, Tail Entity: e^{tail}, Context: s". The two transformations linearize each element into natural languages.

We optimize the generator through auto-regressive objective on the training set $\mathcal{D}^s$:

$$\mathcal{L}_G(s,t) = -\log P(out|E_{in}(r^s);\mathcal{G}) \quad (1)$$

where $out = E_{out}(s, e^{head}, e^{tail})$ and $t = (e^{head}, e^{tail}, r^s), (s,t) \in \mathcal{D}^s$.

Extractor The extractor $\mathcal{E}$ aims to extract the structured triplet t of the form $(e^{head}, e^{tail}, r^s)$ given the sentence s. To ensure the generality of the method, we do not impose any restrictions on the model structure of the extractor. It can be an encoder-decoder model like BART (Lewis et al., 2020) or any encoder-only model like ALBERT (Lan et al., 2020). The objective of the extractor is to minimize the negative log-likelihood on the training dataset $\mathcal{D}^s$:

$$\mathcal{L}_E = -\log P(t|s;\mathcal{E}) \quad (2)$$

During inference, the extractor extracts triplet of the highest probability:

$$\hat{t} = \arg\max_{t \in \mathcal{T}} P(t|s;\mathcal{E}) \quad (3)$$

Our method works with extractors of different structures, so we're not going into the extraction steps for each one in detail here. You can find a full description of the extraction process we use during inference in Appendix A.1.

3 Methodology

The above section illustrates how the generator and extractor function. Our challenge is to *make the two agents work collaboratively and improve the overall extraction performance*. To achive this, we propose a two-agent game and collaborate through the following three key steps: (1) *attempting*: the generator attempts to express its understanding of the unseen relation by generating sentences and triplets for the relation; (2) *criticizing*: the extractor criticizes the generator on its synthetic data; (3) *rectifying*: both agents rectify on their extraction task and generation task individually. The cycle of attempting, criticizing, and rectifying goes iteratively to refine the two agents' abilities. The overall procedure is described in Fig. 2.

3.1 Attempting for Understanding

The game begins with an unseen relation, and the generator expresses its understanding of the relation. This understanding is reflected in constructing sentences and triplets related to the relation.

To generate relation-specific sentences and triplets for an unseen relation $r_i^u \in \mathcal{R}^u$, we follow these steps:

- Map the relation to natural language space using the $E_{in}(r_i^u)$ transformation.
- Randomly sample K times from the generator to obtain text sequences $text_{i_k} \sim P(text_{i_k}|E_{in}(r_i^u);\mathcal{G}), k = 1, \ldots, K$.
- Extract the sentences, head entities, and tail entities from the generated text sequences using E_{out}^{-1}, the inverse function of E_{out}, $s_{i_k}, e_{i_k}^{head}, e_{i_k}^{tail} = E_{out}^{-1}(text_{i_k})$.
- Combine the entities and the input relation to form the triplet $t_{i_k} = (e_{i_k}^{head}, e_{i_k}^{tail}, r_i^u)$.

We aggregate the extracted sentences and triplets to create the synthetic dataset $\mathcal{D}_i^{syn} = \{(s_{i_1}, t_{i_1}), \ldots, (s_{i_K}, t_{i_K})\}$. The overall synthetic dataset is the union of data on all relations: $\mathcal{D}^{syn} = \mathcal{D}_1^{syn} \cup \ldots \cup \mathcal{D}_m^{syn}$, with $|\mathcal{D}^{syn}| = mK$, which represents the generator's initial understanding of the relation and serves as the target of debate in the later stages of the game.

3.2 Criticizing for Quality Assessment

After the generator expresses its understanding of the relations, we use an extractor to evaluate whether the generator's comprehension is correct. The motivation is: the generator's synthetic data may contain noise. We introduce the extractor to differentiate the quality of synthetic data.

Intuitively, the probability $P(t_j|s_j;\mathcal{E})$ can be interpreted as the likelihood of triplet t_j given sentence s_j. This can be regarded as a semantic matching degree between t_j and s_j, indicating the quality of (s_j, t_j). Given the synthetic data $(s_j, t_j) \in \mathcal{D}^{syn}$, we compute the log-likelihood of t_j extracted from s_j:

$$ll_j = \log P(t_j|s_j;\mathcal{E}) \quad (4)$$

Using this formula, we can assign a higher log-likelihood to high-quality text-triplet pairs and a lower log-likelihood to low-quality text-triplet pairs, thereby distinguishing the quality of different synthetic data.

Since all log-likelihood values are negative, we normalize[3] them using the following formula to obtain the quality value α_j for (s_j, t_j) and distinguish

[3] "Normalizing reward" is a common operation in RL (van Hasselt et al., 2016) that can make the distribution of rewards more stable, making it easier for the algorithm to converge.

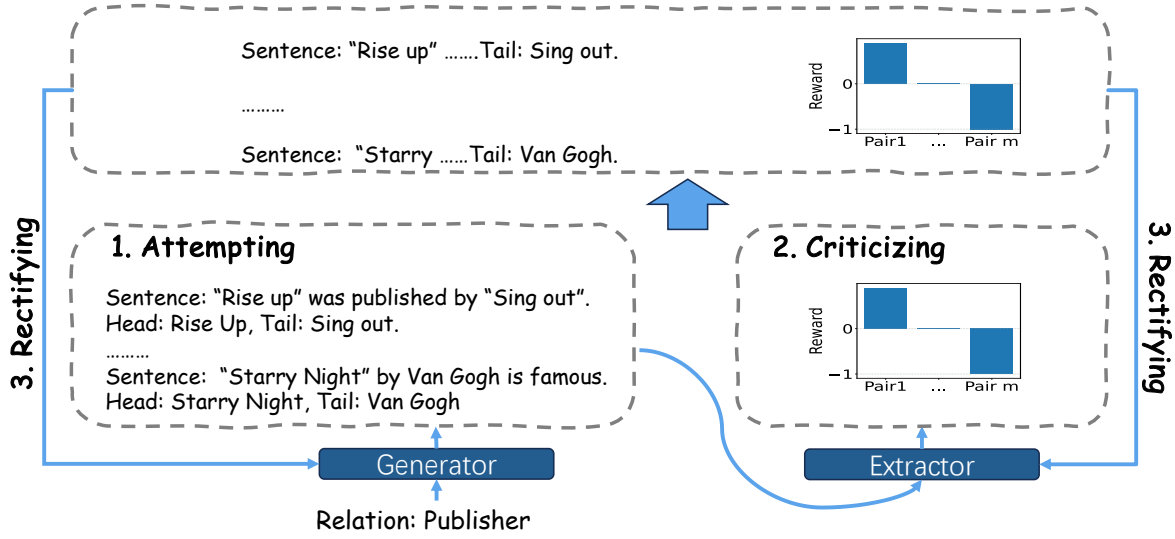


Figure 2: Two-agent game for ZeroRTE. For each iteration, the generator first attempts to understand unseen relations by generating synthetic data. Then the extractor criticizes the generator on the data quality. Finally, both agents rectify their models with reinforcement learning to maximize criticizing rewards on attempting data.

between good and bad data quality based on the sign of the α_j:

$$\alpha_j = \frac{ll_j - \mu}{\delta} \tag{5}$$

where μ and δ are the mean and standard deviation of ll_j on all text-triplet pairs over the dataset $\mathcal{D}^{syn}$.

3.3 Rectifying via Reinforcement Learning

Previous research (Ouyang et al., 2022; Rafailov et al., 2023) demonstrate that reinforcement learning can effectively learn preferences among data. Building on this, we apply a reinforcement learning approach to rectify the model. By providing higher rewards to better-quality text-triplet pairs, the two agents can better understand the semantics of unseen relations through data.

More specifically, we consider the quality value α_j, as defined in Eq. (5), as a reward and aim to maximize the expected reward $\mathbb{E}[\alpha]$ on $\mathcal{D}^{syn}$ through gradient ascent by optimizing the parameters of $\mathcal{E}$ and $\mathcal{G}$ to yield high-quality data:

$$\mathcal{E}_{i+1} \leftarrow \mathcal{E}_i + \gamma_1 \nabla_{\mathcal{E}} \mathbb{E}[\alpha] \tag{6}$$

$$\mathcal{G}_{i+1} \leftarrow \mathcal{G}_i + \gamma_2 \nabla_{\mathcal{G}} \mathbb{E}[\alpha] \tag{7}$$

where γ_1 and γ_2 are the learning rates for the extractor and generator, respectively. The gradients for the extractor $\nabla_{\mathcal{E}}\mathbb{E}[\alpha]$ and generator $\nabla_{\mathcal{G}}\mathbb{E}[\alpha]$ are computed as follows:

$$\nabla_{\mathcal{E}} \mathbb{E}[\alpha] = \mathbb{E}(\alpha_j \nabla_{\mathcal{E}} \log P(t_j | s_j; \mathcal{E})) \tag{8}$$

$$\nabla_{\mathcal{G}} \mathbb{E}[\alpha] = \mathbb{E}(\alpha_j \nabla_{\mathcal{G}} \log P(out_j | in_j; \mathcal{G})) \tag{9}$$

where the expectation is taken over α_j on $\mathcal{D}^{syn}$, $in_j = E_{in}(r_j^u)$, $out_j = E_{out}(s_j, e_j^{head}, e_j^{tail})$.

During the iterative process of attempting, criticizing, and rectifying, the two agents engage in detailed semantic debates, enabling them to discern subtle semantic distinctions, and ultimately refine their extraction and generation tasks.

4 Experiments

To validate the effectiveness of TAG, we conduct experiments on both small and large language models to address the following questions: (1) Can TAG be applied to different extractor model structures? (2) Is TAG effective at different scales of model parameters?

4.1 Experimental Settings

Datasets We evaluate TAG on FewRel (Han et al., 2018), Wiki-ZSL (Chen and Li, 2021), and TACRED (Zhang et al., 2017). The data statistics are shown in Table 3. We follow the same process as Chia et al. (2022) to partition the data into seen and unseen label sets. For each dataset, we set the unseen label size to $m = \{5, 10, 15\}$ and randomly select m relation labels for testing and treat the remaining labels as seen labels during training in the experiments. To reduce the effect of experimental noise, the label selection process is repeated for five random seeds to produce different folds.

Unseen	Model	Single Triplet			Multi Triplet					
		Wiki-ZSL	FewRel	TACRED	Wiki-ZSL			FewRel		
		Acc.	Acc.	Acc.	P.	R.	F1.	P.	R.	F1.
m = 5	TabelSequence	14.47†	11.82†	2.1	43.68†	3.51†	6.29†	15.23†	1.91†	3.40†
	RelationPrompt	16.64†	22.27†	8.34	29.11†	31.00†	30.01†	20.80†	24.32†	22.34†
	TAG + TabelSequence	17.57	18.41	**18.08**	**58.65**	15.53	23.93	36.17	11.54	17.43
	TAG + RelationPrompt	**23.12**	**28.94**	9.59	39.36	37.51	**38.24**	**37.56**	**40.24**	**38.81**
m = 10	TabelSequence	9.61†	12.54†	1.87	45.31†	3.57†	6.40†	28.93†	3.60†	6.37†
	RelationPrompt	16.48†	23.18†	3.26	30.20†	32.31†	31.19†	21.59†	28.68†	24.61†
	TAG + TabelSequence	14.42	19.81	**11.05**	**45.92**	13.98	21.37	**36.10**	8.76	13.90
	TAG + RelationPrompt	**17.24**	**28.16**	3.97	31.37	32.53	**31.88**	31.04	33.49	**32.18**
m = 15	TabelSequence	9.20†	11.65†	0.63	**44.43**†	3.53†	6.39†	19.03†	1.99†	3.48†
	RelationPrompt	16.16†	18.97†	1.57	26.19†	32.12†	28.85†	17.73†	23.20†	20.08†
	TAG + TabelSequence	11.45	16.67	**7.35**	40.09	10.01	15.94	23.46	6.12	9.69
	TAG + RelationPrompt	**16.41**	**22.53**	1.69	26.52	31.34	**29.18**	**25.35**	25.88	**25.59**

Table 1: Comparison of TAG with other small language models. † denotes results from Chia et al. (2022). Best results are highlighted in bold. TAG can achieve consistent improvements when applied to encoder-only (TabelSequence) and encoder-decoder (RelationPrompt) extractors.

Evaluation Metrics Following the work of Chia et al. (2022), we evaluate the results of triplet extraction separately for sentences containing single triplets and multiple triplets. We use the standard micro precision (P.), recall (R.), and F1 metrics commonly used in structured prediction tasks for multiple triplet extraction. On the other hand, evaluating single triplet extraction involves only one possible triplet for each sentence. Hence, the metric of Accuracy (Acc.) is employed. We report the average results across five data folds.

4.2 Experiments on Small Language Models

Baselines We compare our proposed TAG, with competitive baselines in ZeroRTE.

- *TableSequence* (Wang and Lu, 2020) casts the ZeroRTE task as a table-filling problem and uses ALBERT-Large (Lan et al., 2020) to encode the textual information.
- *RelationPrompt* (Chia et al., 2022) uses GPT-2 (Radford et al., 2019) to generate synthetic data for unseen relations and then trains the extractor model BART (Lewis et al., 2020) on the synthetic data from GPT-2.

We choose not to compare with DSP (Lv et al., 2023) because this approach heavily relies on the initial parameters of soft prompts, making it hard to reproduce the results. We set the hyperparameters as those reported in Appendix B because they can attain good performance in our experiments.

Results We report experimental results in Table 1. We omit the multi-triplets for TACRED, which exclusively contains single triplets. From the table, we can observe that TAG consistently improves over TableSequence (encoder-only) and RelationPrompt (encoder-decoder). Compared with RelationPrompt, TAG achieves an absolute F1 improvement of 8.23% and 16.47% on Wiki-ZSL and FewRel in multi-triplet with $m = 5$. Such improvement indicates the effectiveness of TAG across different model architectures. Moreover, TAG can achieve a more balanced precision-recall ratio, leading to better overall F1 results. When comparing TAG + TableSequence to TAG + RelationPrompt on TACRED, we notice a significant performance advantage for the former. This discrepancy is because many of the relations labels in TACRED start with the same token, RelationPrompt solely considers the probability of the first token during relation decoding, so it struggles to differentiate between relations with identical initial tokens.

4.3 Experiments on Large Language Models

Given the strong performance of LLMs across various downstream tasks (Zan et al., 2023), we investigate the performance of GPT-3.5, a highly influential LLM, on the ZeroRTE task. We remove the fine-tuning data $\mathcal{D}^s$ to compare with previous methods based on LLMs and evaluate the performance using the test sets with $m = 5$. Detailed experiment settings can be seen in Appendix B.

Baselines We compare our proposed TAG with GPT-3.5 under different prompting methods.

- *ICL* is an in-context-learning method that directly prompts LLMs, we follow the prompt-

Method	Extractor	Generator	Wiki-ZSL			FewRel			Time
			P.	R.	F1.	P.	R.	F1.	
ICL †	GPT-3.5	-	8.87	**8.68**	8.49	11.35	**12.58**	**11.87**	2.59
ChatIE †	GPT-3.5	-	8.52	8.01	8.15	11.11	10.93	10.99	6.08
RelationPrompt	BART(140M)	GPT-3.5	7.76	6.86	7.28	8.76	8.33	8.54	**0.56**
TAG + RelationPrompt	BART(140M)	GPT-3.5	**10.08**	8.50	**9.21**	**11.75**	10.98	11.35	**0.56**

Table 2: Comparison between TAG and other methods using large language models. † denotes the results we reproduce from ChatIE (Wei et al., 2023b) on Wiki-ZSL and FewRel. Time refers to the model's inference time, measured in seconds per sample. We highlight the best results in bold.

	#Samples	#Entities	#Relations
Wiki-ZSL	94,383	77,623	113
FewRel	56,000	72,954	80
TACRED	21,773	8,958	41

Table 3: Data statistics of Wiki-ZSL, FewRel, and TACRED.

ing method in Wei et al. (2023b).

- *ChatIE* (Wei et al., 2023b) transforms ZeroRTE task into a multi-turn question answering problem with a two-stage framework.
- *RelationPrompt* is the same as described in the previous section with the generation model replaced from GPT-2 to GPT-3.5.

Results We report experimental results in Table 2. The results reveal the following key observations: (1) Based on the results of TAG + RelationPrompt, ICL, and ChatIE, we can conclude that small extractors trained on synthetic data of LLMs can achieve comparable results while reducing the inference time from 78% to 91%. (2) TAG + RelationPrompt's performance on FewRel is lower than ICL's. This is because TAG + RelationPrompt uses BART (140M) as an extractor, which has a much smaller number of model parameters than ICL's GPT-3.5. However, TAG's inference speed is 78% faster than ICL. (3) TAG can still improve the performance of the RelationPrompt by 1.93% and 2.81% in Wiki-ZSL and FewRel, respectively. This underscores the effectiveness of TAG under large language model scales. Moreover, it does not increase the inference cost.

5 Analysis

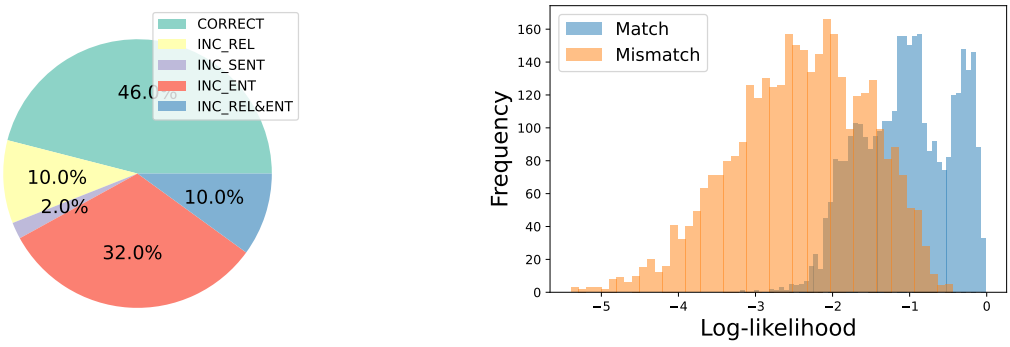
5.1 Analysis on Attempting

To assess whether the generator can understand new relations and produce relation-specific sentences and triplets during the attempting step, we randomly sample 50 outputs from GPT-2 and GPT3.5 and manually evaluate the quality of synthetic data. Specifically, we categorize the generated data into five categories: correct (CORRECT), incorrect sentences (INC_SENT), incorrect entities (INC_ENT), incorrect relations (INC_REL), and cases of incorrect relations and entities (INC_REL&ENT). Detailed settings and generated examples are shown in Appendix B.1. Fig. 3a and Fig. 7 display the analysis of generated samples from GPT-2 and GPT-3.5, showing that the generator accurately produces nearly half of the data. The results suggest that *the generator can capture the semantic information of relation and generalize to unseen relations to some extent.* Consequently, this validates our approach of employing language models as generators for data generation.

5.2 Analysis on Criticizing

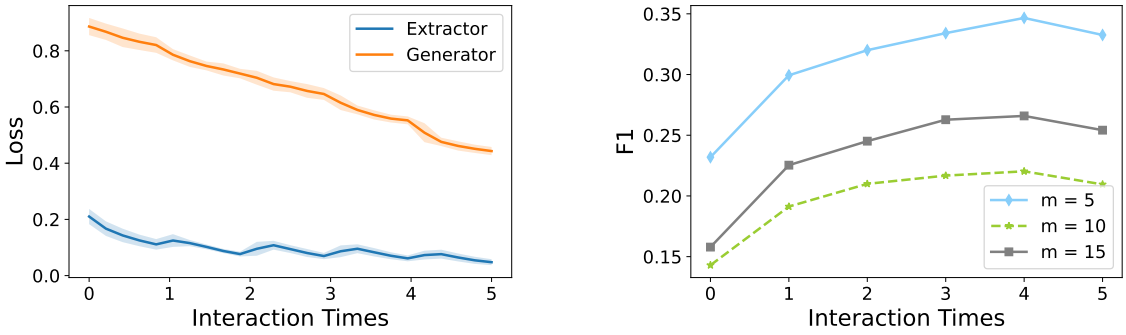
To assess the extractor's ability to differentiate between varying data quality, we use text-triplet pairs from the evaluation set as matching examples and construct mismatching examples by randomly replacing relations, head entities, and tail entities. We calculate the log-likelihood for each pair. Fig. 3b and Fig. 8 show the results from two seminal studies, RelationPrompt (Chia et al., 2022) and TableSequence (Wang and Lu, 2020), respectively. The figure shows a noticeable difference between matching and mismatching examples. The model tends to assign higher log-likelihood values to matching examples and lower values to mismatching examples.

In addition to the manually constructed mismatching data, we also conduct qualitative analysis on the reward for synthetic data in appendix B.3. We can conclude that *the extractor serves as an effective semantic matching evaluator for unseen relation types and, thus has the potential to detect errors from the generator*.



(a) (b)

Figure 3: Analysis on the attempting and criticizing steps. (a) Analysis on the attempting step of GPT-2 on the zero-shot generation task. The results are categorized into correct (CORRECT), incorrect sentences (INC_SENT), incorrect entities (INC_ENT), incorrect relations (INC_REL), and cases of incorrect relations and entities (INC_REL&ENT). (b) Histogram of the match and mismatch text-triplets on the extractor's criticizing log-likelihood: the value is generated by RelationPrompt (Chia et al., 2022), a seminal work in ZeroRTE.



(a) (b)

Figure 4: Stability analysis of reinforcement learning. (a) Loss of the extractor and generator during reinforcement learning. (b) F1 score of the extractor on the validation set of FewRel.

5.3 Stability of Reinforcement Learning

We assess the reinforcement learning stability by analyzing the extractor and generator's training loss dynamics, as well as the extractor's F1 score on the validation set. As shown in Fig. 4a, the loss of the extractor and generator steadily decreases during the training process, indicating that our algorithm can learn a stable policy. Meanwhile, the sharper decrease in the generator's loss is attributed to handling a more complex task than the extractor. Extractor's F1 scores on FewRel validation set (with unseen labels 5-15) are shown in Fig. 4b. TAG exhibits progressive improvement up to 4 iterations, with minimal decreases beyond that point. This suggests a delicate balance between capability enhancement and potential overfitting. In conclusion, reinforcement learning in TAG gradually improves both the extractor and generator but may lead to an over-fitting with excessive training. Here, we set overall interaction times to 5 based on Fig. 4.

In Appendix B.4, we compare the results of Reinforcement Learning (RL) and Maximum Likelihood Estimation (MLE) and discover that RL significantly outperforms MLE, which further demonstrates the effectiveness of our approach.

5.4 Case Study

As presented in Table 4 in the appendix, we compare the outcomes of RelationPrompt and TAG + RelationPrompt in the ZeroRTE task. RelationPrompt tends to confuse the meanings of similar relations like "*located on terrain features*" and "*contains administrative territorial entity*", "*publisher*" and "*distributed by*", whereas TAG distinguishes between them effectively. This showcases TAG's superior ability to understand the nuances of unseen relations through its deliberation and debate process. However, in the fourth example, both

methods incorrectly swap the head and tail entities. This suggests that TAG's entity distinction capability is somewhat weaker, which could be a research direction for the future.

We further analyze the effectiveness of reward value, reinforcement learning, and generated data size, and put them in Appendix B.3, Appendix B.4 and B.5 due to space limitation.

6 Related Work

Zero-shot Relation Triplet Extraction Zero-shot relation triplet extraction (ZeroRTE) is a challenging task that aims to extract relation triplets from unstructured text, where the relation sets between training and test are disjoint. There are many different approaches to ZeroRTE. Conventional methods tackle ZeroRTE by formulating ZeroRTE into prompts to leverage the power of language models. For example, Chia et al. (2022), the seminal work in ZeroRTE, first prompt a generative model to generate synthetic training data for the unseen relations, then they use the synthetic data to further train the extraction model. Lv et al. (2023); Lan et al. (2022); Kim et al. (2022) directly incorporate the semantics of relations through soft prompt[4].

Recent research has shown that large language models (LLMs) can achieve strong performance on downstream tasks (Brown et al., 2020; Kojima et al., 2022; Zan et al., 2023) without tuning the parameters. For the RTE task, Wadhwa et al. (2023) explore chain-of-thought prompting under few-shot settings, Wei et al. (2023b) transforms RTE into a multi-turn question answering task and extract the relation triplets through chatting with LLMs.

However, these methods simply integrate relations into prompts, which is superficial and may lead to an incomplete understanding of the unknown relations.

Multi-agent Game Multi-agent game studies the behavior of multiple language models through debate or cooperation (Talebirad and Nadiri, 2023; Li et al., 2023; Fu et al., 2023; Dasgupta et al., 2023; Wei et al., 2023a). The core idea is the agents can improve each other through debate and cooperation. For example, Fu et al. (2023) improve the negotiation ability by two agents bargaining and one agent criticizing. Talebirad and Nadiri (2023) propose a collaborative multi-agent framework for handling complex tasks more efficiently and effectively. In this paper, we borrow the idea of multi-agent game and construct a two-agent game for ZeroRTE to deliberate and debate the semantics of the relation.

Reinforcement Learning Reinforcement learning is a machine learning paradigm that focuses on how intelligent agents can make decisions in an environment to optimize a given notion of cumulative rewards (François-Lavet et al., 2018). Unlike supervised learning, reinforcement learning does not require labeled data to be presented. Instead, it interacts with the environment to collect information. Reinforcement learning has found applications in various domains such as game playing (Mnih et al., 2015) and recommendation systems (Afsar et al., 2022). In this paper, we regard ZeroRTE as the interaction game between the extractor and generator and employ reinforcement learning to mutually improve the two agents.

7 Conclusion

In this paper, we propose TAG, a two-agent game for ZeroRTE, by introducing a new generator agent to communicate with the original extractor agent. Through iterative processes of attempting, criticizing, and rectifying, the generator and extractor agents engage in a deliberative and collaborative exploration of the semantics of unseen relations, facilitating a comprehensive understanding of these relations. Experimental results demonstrate that TAG consistently enhances performance across different model architectures and scales without incurring additional inference costs. Remarkably, our method outperforms strong baselines by a significant margin, achieving an impressive 6%-16% increase in F1 scores, particularly when dealing with FewRel with five unseen relations. We believe that TAG is a promising approach for ZeroRTE and holds potential for applications in other natural language processing tasks.

Limitations

The proposed method has some limitations with LLMs. Specifically, the criticizing step requires probability values from the extractor, which are difficult to obtain for LLMs that can only be accessed via API. Additionally, the rectifying step necessitates gradient updates to the generator and extractor, which is impractical for LLMs.

[4]Though KnowPrompt (Chen et al., 2022) has been proposed in the literature for relation classification, it differs from our work in both the task and experiment settings.

In order to address these challenges, future work could focus on the following directions: (1) Replacing probability values with textual feedback, which can be readily provided by LLMs; (2) Exploring alternative approaches to rectify LLMs without relying on gradient updates.

References

M Mehdi Afsar, Trafford Crump, and Behrouz Far. 2022. Reinforcement learning based recommender systems: A survey. *ACM Computing Surveys*, 55(7):1–38.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

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

Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 2778–2788, New York, NY, USA. Association for Computing Machinery.

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*, pages 45–57, Dublin, Ireland. Association for Computational Linguistics.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. 2023. Why can GPT learn in-context? language models secretly perform gradient descent as meta-optimizers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4005–4019, Toronto, Canada. Association for Computational Linguistics.

Ishita Dasgupta, Christine Kaeser-Chen, Kenneth Marino, Arun Ahuja, Sheila Babayan, Felix Hill, and Rob Fergus. 2023. Collaborating with language models for embodied reasoning. *arXiv preprint arXiv:2302.00763*.

Vincent François-Lavet, Peter Henderson, Riashat Islam, Marc G Bellemare, Joelle Pineau, et al. 2018. An introduction to deep reinforcement learning. *Foundations and Trends® in Machine Learning*, 11(3-4):219–354.

Yao Fu, Hao Peng, Tushar Khot, and Mirella Lapata. 2023. Improving language model negotiation with self-play and in-context learning from ai feedback. *arXiv preprint arXiv:2305.10142*.

Kelvin Guu, Panupong Pasupat, Evan Liu, and Percy Liang. 2017. From language to programs: Bridging reinforcement learning and maximum marginal likelihood. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1051–1062, Vancouver, Canada. 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*, pages 4803–4809, Brussels, Belgium. Association for Computational Linguistics.

Bosung Kim, Hayate Iso, Nikita Bhutani, Estevam Hruschka, and Ndapa Nakashole. 2022. Zero-shot triplet extraction by template infilling. *arXiv preprint arXiv:2212.10708*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Yuquan Lan, Dongxu Li, 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*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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*, pages 7871–7880, Online. Association for Computational Linguistics.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. Camel: Communicative agents for" mind" exploration of large scale language model society. *arXiv preprint arXiv:2303.17760*.

Bo Lv, Xin Liu, Shaojie Dai, Nayu Liu, Fan Yang, Ping Luo, and Yue Yu. 2023. DSP: Discriminative soft prompts for zero-shot entity and relation extraction. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5491–5505, Toronto, Canada. Association for Computational Linguistics.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Berlin, Germany. Association for Computational Linguistics.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Delai Qiu, Yuanzhe Zhang, Xinwei Feng, Xiangwen Liao, Wenbin Jiang, Yajuan Lyu, Kang Liu, and Jun Zhao. 2019. Machine reading comprehension using structural knowledge graph-aware network. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5896–5901, Hong Kong, China. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.

Yashar Talebirad and Amirhossein Nadiri. 2023. Multi-agent collaboration: Harnessing the power of intelligent llm agents. *arXiv preprint arXiv:2306.03314*.

Hado P van Hasselt, Arthur Guez, Matteo Hessel, Volodymyr Mnih, and David Silver. 2016. Learning values across many orders of magnitude. *Advances in neural information processing systems*, 29.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, Joao Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. 2023. Transformers learn in-context by gradient descent. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 35151–35174. PMLR.

Somin Wadhwa, Silvio Amir, and Byron Wallace. 2023. Revisiting relation extraction in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15566–15589, Toronto, Canada. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

Jimmy Wei, Kurt Shuster, Arthur Szlam, Jason Weston, Jack Urbanek, and Mojtaba Komeili. 2023a. Multi-party chat: Conversational agents in group settings with humans and models. *arXiv preprint arXiv:2304.13835*.

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

Yang Yang, Zhilei Wu, Yuexiang Yang, Shuangshuang Lian, Fengjie Guo, and Zhiwei Wang. 2022. A survey of information extraction based on deep learning. *Applied Sciences*, 12(19).

Daoguang Zan, Bei Chen, Fengji Zhang, Dianjie Lu, Bingchao Wu, Bei Guan, Wang Yongji, and Jian-Guang Lou. 2023. Large language models meet NL2Code: A survey. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7443–7464, Toronto, Canada. Association for Computational Linguistics.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark. Association for Computational Linguistics.

Yang Zhao, Lu Xiang, Junnan Zhu, Jiajun Zhang, Yu Zhou, and Chengqing Zong. 2020. Knowledge graph enhanced neural machine translation via multi-task learning on sub-entity granularity. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4495–4505, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. 2017. Joint extraction of entities and relations based on a novel tagging scheme. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1227–1236, Vancouver, Canada. Association for Computational Linguistics.

A Preliminaries

A.1 Extractor

We base our experiments on two extraction models, RelationPrompt (Chia et al., 2022) and TableSequence (Wang and Lu, 2020). The detailed extraction process is outlined as follows:

- RelationPrompt (Chia et al., 2022) is a sequence-to-sequence model that generates text sequences from *"Head Entity: h, Tail Entity: t, Relation: r"*. Then the generated text sequences are decoded into triplets of the form *(h, t, r)* through regular expressions. It proposes a Triplet Search Decoding method to extract multiple triplets in a sentence. The core concept is enumerating multiple output sequences during generation by considering multiple candidates for the head entity, tail entity, and relation.

- TableSequence (Wang and Lu, 2020) consists of two distinct encoders: a sequence encoder to extract entities and a table encoder to extract relations for entity pairs. The proposed model is capable of adaptively discovering multiple triplets simultaneously in a sentence via the classification results of the table encoder. It formalizes the RTE task as a table-filling task and decodes the triplet from the table.

B Experiments and Analysis

Training Details for Small Language Models We use the pretrained model provided by HuggingFace[5] and run all the experiments on NVIDIA A100 GPU with pytorch. For the hyper-parameters of the baseline models, we follow the original settings in their paper (Chia et al., 2022; Wang and Lu, 2020). We use GPT-2 as the generator in small language models. We set the learning rates for the extractor and generator of TAG + TableSequence as $\gamma_1 = 3 \times 10^{-3}, \gamma_2 = 3 \times 10^{-5}$, respectively. We set the learning rates for the extractor and generator of TAG + RelationPrompt as $\gamma_1 = 3 \times 10^{-5}, \gamma_2 = 3 \times 10^{-5}$, respectively. The only hyper-parameter in TAG is the number of generated data size K. We search K in $\{100, 300, 500, 700\}$. For each value, we conduct experiments with five random data splits and set $K = 500$ by the F1 score on the validation set. The parameter search costs about 50 GPU hours.

Training Details for Large Language Models Due to the model's large size, updating its parameters is challenging. Previous research has shown that prompts have a similar effect to gradient updates (Von Oswald et al., 2023; Dai et al., 2023). Therefore, we have modified the reinforcement learning part of the large model to use data with high rewards as demonstration inputs. For the extractor, we fine-tune the model using the generated output from the generator and the scoring results from the extractor. We show the detailed prompts for ICL, ChatIE, and TAG in Fig. 9, Fig. 10, and Fig. 11, respectively.

B.1 Analysis on Attempting

We evaluate the zero-shot generation ability on two models: GPT-2 and GPT-3.5. For GPT-2, we first fine-tune it on the training set $\mathcal{D}^s$, then evaluate its ability on unseen relation set $\mathcal{R}^u$. We randomly select 50 generated examples and manually assess the quality. We categorized the generated data into five categories:

- CORRECT: The sentence and the triplet are both correct, and the sentence implies the meaning of the triplet.

- INC_SENT: The sentence is grammatically or semantically incorrect, making it impossible to understand.

- INC_ENT: The sentence and the relation are correct, but the corresponding head or tail entity is incorrect.

- INC_REL: The sentence does not preserve the semantics of the corresponding relation.

- INC_REL&ENT: The combination of INC_REL and INC_ENT.

Analysis statistics for GPT-2 and GPT-3.5 are shown in Fig. 3a and Fig. 7. To show the results

[5] https://huggingface.co/

Sentence	Label	RelationPrompt	TAG + RelationPrompt
Caribbean itineraries included the British Virgin Islands , French West Indies , Grenadines , the ABC islands and The Bahamas .	(ABC islands, Caribbean, located on terrain feature)	(ABC islands, British Virgin Islands, contains administrative territorial entity)	(ABC islands, Caribbean, located on terrain feature)
Wiślica is a town in Busko County , Świętokrzyskie Voivodeship , in south - central Poland.	(Świętokrzyskie Voivodeship, Busko County, contains administrative territorial entity)	(Wiślica, Busko County, located on terrain feature)	(Świętokrzyskie Voivodeship, Busko County, contains administrative territorial entity)
In 1933 after the success of the film " Bird of Paradise " (1932) , RKO Pictures tried to reunite the star couple .	(Bird of Paradise, RKO Pictures, distributed by)	(Bird of Paradise, RKO Pictures', 'publisher')	('Bird of Paradise', 'RKO Pictures', 'distributed by')
The first Bundesliga match of the season took place on 11 August which resulted in a 1–1 draw against Bayer 04 Leverkusen.	(Bundesliga, Bayer 04 Leverkusen, participating team)	(Bayer 04 Leverkusen, Bundesliga, participating team)	(Bayer 04 Leverkusen, Bundesliga, participating team)

Table 4: Case Study between RelationPrompt and TAG + RelationPrompt.

Sentence	Triplets	reward
The brewery was purchased in June 2014 by Pheuropa Corporation , a Spanish beer firm owned by Carlos Alberto Moreno Peña , the former owner of Tébéco Aleworks in La Paz .	(Tébéco Aleworks, Carlos Alberto Moreno Peña, owned by)	1.25
Its founder was a young Czech composer and member of the Chamber of Directors of the National Academy of Sciences .	(Chamber of Directors of the National Academy of Sciences, Czech composer, main subject)	-0.08
The venue is located southwest of the city of Waco; it was originally named after former World War II Navy SEAL Brian Dunbar .	(Waco;, Waco;, location)	-5.61

Figure 5: Qualitative analysis of the reward value.

Algorithm 1 The two-agent game framework

Input: Unseen relation set $\mathcal{R}^u$ with $|\mathcal{R}^u| = m$, extractor $\mathcal{E}$ and generator $\mathcal{G}$ after supervised training on $\mathcal{D}^s$, learning rates γ_1, γ_2 for the extractor and generator, respectively.
Output: Trained extractor and generator which can generalize to the unseen relation set $\mathcal{R}^u$

repeat
  ▷ Attempting for Understanding
  for $i = 1, 2, \ldots, m$ **do**
    Select relation r_i^u from $\mathcal{R}^u$
    Sample K text sequences $text_{i_1}, \ldots text_{i_K}$ from $P(text_{i_k}|E_{in}(r_i^u); \mathcal{G})$
    Decode sentences and triplets $(s_{i_1}, t_{i_1}), \ldots (s_{i_K}, s_{i_K})$ from text sequences
    Constructing synthetic data $\mathcal{D}_i^{syn} = \{(s_{i_1}, t_{i_1}), \ldots (s_{i_K}, s_{i_K})\}$
  end for
  Union synthetic data for all relations to construct $\mathcal{D}^{syn} = \mathcal{D}_1^{syn} \cup \ldots \cup \mathcal{D}_m^{syn}$
  ▷ Criticizing for Quality Assessment
  for $(s_j, t_j) \in \mathcal{D}^{syn}$ **do**
    Measure the quality of the jth data using Eq. (4) and Eq. (5).
  end for
  ▷ Rectifying via Reinforcement Learning
  Compute the stochastic gradient of $\mathcal{E}$ and $\mathcal{G}$ using Eq. (8) and Eq. (9).
  Update the model parameters using Eq. (6) and Eq. (7).
until convergence

Notation	Description
$\mathcal{D}$	a dataset
$\mathcal{S}/\mathcal{T}$	the set of sentences/triplets
$\mathcal{R}$	the relation set
s/t	a sentence/triplet
e^{head}/e^{tail}	the head/tail entity
r	the relation
E_{in}/E_{out}	the transformation function for the generator's input/output
$\mathcal{G}/\mathcal{E}$	the generator/extractor model
γ_1/γ_2	learning rates for the generator/ extractor
K	the number of synthetic data for each relation
$text$	sequences generated for the generator
α	quality value
$\mathcal{L}_G/\mathcal{L}_E$	loss function for the generator/extractor

Table 5: Glossary of notations.

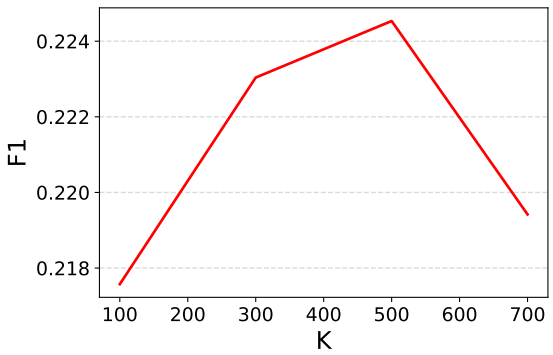


Figure 6: Effect of generated data size on the validation set of FewRel with $m = 5$.

more specifically, we randomly select 10 examples generated by GPT-2 and GPT-3.5, and present them in Table 7 and Table 8, respectively.

B.2 Analysis on Criticizing

We first train the two models on $\mathcal{D}^s$ of seen relations. Next, we calculate the log-likelihood of matching and mismatching data. The results for RelationPrompt and TableSequence are reported in Fig. 3b and Fig. 8, respectively.

B.3 Qualitative Analysis of the Reward

In our study, we employ a qualitative analysis to assess the efficacy of the reward generated by the extractor. Specifically, we scrutinize the reward values assigned to various synthetic data, as depicted in Fig. 5. Notably, the initial instance, boasting the highest reward value, exhibits consistent sentences and triplets. Conversely, the second instance accurately extracts head and tail entities, but the relation is not implied in the sentence, resulting in a moderate reward value. Finally, the last instance extracts incorrect and duplicate head and tail entities, warranting the lowest reward value. This

Unseen	Model	Single Triplet			Multi Triplet					
		Wiki-ZSL	FewRel	TACRED	Wiki-ZSL			FewRel		
		Acc.	Acc.	Acc.	P.	R.	F1.	P.	R.	F1.
m = 5	RelationPrompt	16.64†	22.27†	8.34	29.11†	31.00†	30.01†	20.80†	24.32†	22.34†
	TAG + RelationPrompt	**23.12**	**28.94**	**9.59**	**39.36**	37.51	**38.24**	**37.56**	**40.24**	**38.81**
	MLE + RelationPrompt	18.52	25.28	8.98	34.49	**40.04**	36.94	30.31	36.54	33.00
m = 10	RelationPrompt	16.48†	23.18†	3.26	30.20†	32.31†	31.19†	21.59†	28.68†	24.61†
	TAG + RelationPrompt	**17.24**	**28.16**	**3.97**	**31.37**	**32.53**	**31.88**	**31.04**	**33.49**	**32.18**
	MLE + RelationPrompt	14.21	22.34	3.31	27.54	28.15	27.75	25.04	27.46	26.61
m = 15	RelationPrompt	16.16†	18.97†	1.57	26.19†	**32.12**†	28.85†	17.73†	23.20†	20.08†
	TAG + RelationPrompt	**16.41**	**22.53**	**1.69**	**26.52**	31.34	**29.18**	**25.35**	**25.88**	**25.59**
	MLE + RelationPrompt	12.54	21.09	1.55	23.27	27.12	22.12	23.70	25.41	24.49

Table 6: Comparison of TAG and maximum likelihood estimation (MLE). † denotes results from Chia et al. (2022). The best results are highlighted in bold. TAG can achieve consistent improvements when compared with MLE.

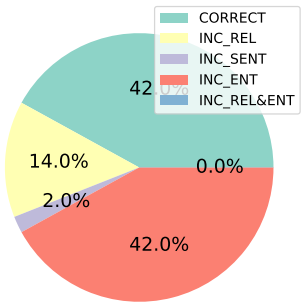


Figure 7: Statistics of the generation results of GPT-3.5 on zero-shot generation task. The results are categorized into correct (CORRECT), incorrect sentences (INC_SENT), incorrect entities (INC_ENT), incorrect relations (INC_REL), and cases of incorrect relations and relations (INC_REL&ENT).

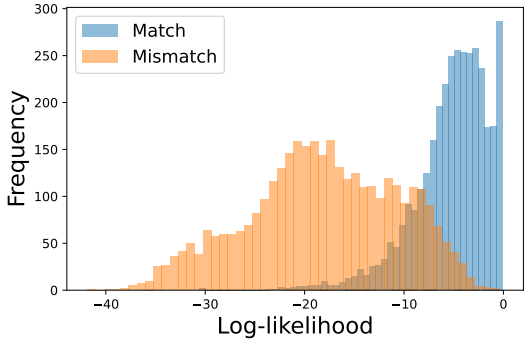


Figure 8: Histogram of the match and mismatch text-triplets on the log-likelihood, the value is generated by TableSequence (Wang and Lu, 2020)

.

analysis leads us to the conclusion that the reward from the extractor effectively evaluates the quality of synthetic data.

B.4 Effect of Reinforcement Learning

We follow Maximum-Likelihood Estimation (MLE) (Guu et al., 2017), i.e., maximizing the likelihood over the whole synthetic dataset, which replaces the update in Eq. (6) and Eq. (7) into:

$$\mathcal{E}_{i+1} = \mathcal{E}_i + \gamma_1 \nabla_{\mathcal{E}} \sum_j \log P(t_j|s_j;\mathcal{E}) \quad (10)$$

$$\mathcal{G}_{i+1} = \mathcal{G}_i + \gamma_2 \nabla_{\mathcal{G}} \sum_j \log P(out_j|in_j;\mathcal{G}) \quad (11)$$

The results are shown in Table 6, we can conclude that our proposed TAG outperforms MLE in all experimental settings. We believe this is because TAG can provide different rewards for data of varying quality, allowing the model to favor high-quality data. In contrast, MLE merely maximizes the probability of synthetic data and cannot differentiate between the quality of different data.

B.5 Effect of Generated Data Size

We explore the impact of generated data size on ZeroRTE performance. F1 scores of FewRel validation set with $m = 10$ and varying generated data sizes (100-700) are shown in Fig. 6. Our findings indicate that TAG's performance improves with increasing generated data up to 500. However, further increasing the data size to 700 does not significantly enhance the final performance.

Input	Output	Results
Relation: competition class	Head Entity: David Kramm, Tail Entity: umpire, Context: David Kramm (December 3 , 1899 – October 9 , 1991) was a U.S. professional baseball umpire .	✔
Relation: competition class	Head Entity: D.J. Haskins, Tail Entity: National Cup, Context: For the first game in the National Cup , D.J. Haskins had been the team 's lone first team defender , but he continued his good form through the entire second half of the season .	✔
Relation: composer	Head Entity: I Am Legend, Tail Entity: Richard L. Bach, Context: I Am Legend would be performed by Richard Wagner under the name of Richard L. Bach in 1824 , but was later shortened to " I Am Legend " and rechristened as " I Am Legend " .	Incorrect Relation
Relation: composer	Head Entity: The Piano Concerto, Tail Entity: George Martin, Context: In addition , a version of the classic opera " The Piano Concerto " was composed by George Martin (whose solo version in " A Christmas Carol " was released on DVD)	✔
Relation: location	Head Entity: National Museum of British Art and Design, Tail Entity: London, Context: The National Museum of British Art and Design in London is currently the headquarters of the Metropolitan Museum of Art , an institution that coordinates with the National Library of Scotland (NLS) .	✔
Relation: location	Head Entity: La Cinares, Tail Entity: La Cinares, Context: The Lotto Divisies at La Cinares won a bronze medal in the women 's freestyle relay at the 1988 Summer Olympics .	Incorrect Entities
Relation: main subject	Head Entity: The World Is Not Enough, Tail Entity: Alan Barnes, Context: The first of the songs recorded for the EP was " The World Is Not Enough " , written by the British songwriters John Waters and Alan Barnes .	Incorrect Relation
Relation: main subject	Head Entity: The Last Tycoon, Tail Entity: The Last Tycoon, Context: In this era , she played the lead role of Vincenzo in the film The Last Tycoon starring Robert De Niro , John Hurt , and Robert Carlyle .	Incorrect Relation & Entities
Relation: operating system	Head Entity: Sony PlayStation 3, Tail Entity: Atari ST Neo, Context: NPD platforms are often found in " Sucker Punch " , which is a platform released for the Atari ST for the Sony PlayStation 3 and Atari ST Neo .	✔
Relation: operating system	Head Entity: Gingerbread, Tail Entity: Apple, Context: For Android , Gingerbread released under the Apple App Store , the Apple - developed version of the OS is bundled with the Gingerbread launcher .	Incorrect Sentence

Table 7: Generated cases for GPT-2 on five unseen relations of FewRel.

Input	Output	Results
Relation: competition class	Head Entity: Figure Skating, Tail Entity: Ice Dance, Context: Figure Skating is a sport that includes Ice Dance as a competition class where skaters perform intricate dance moves on ice.	✔
Relation: competition class	Head Entity: World Chess Championship, Tail Entity: Open, Context: The World Chess Championship is considered the highest competition class in the game, while the Open category allows participation from all players	Incorrect Entities
Relation: composer	Head Entity: "Messiah", Tail Entity: George Frideric Handel,, Context: The famous oratorio "Messiah" was composed by George Frideric Handel, a renowned composer known for his contributions to Baroque music.	✔
Relation: composer	Head Entity: Johann Sebastian Bach., Tail Entity: Brandenburg Concerto No. 3, Context: The composer of Brandenburg Concerto No. 3 was Johann Sebastian Bach.	Incorrect Entities
Relation: location	Head Entity: Central Park, Tail Entity: New York City., Context: Central Park is a famous and beloved location in New York City.	✔
Relation: location	Head Entity: Colosseum, Tail Entity: Rome., Context: The Colosseum is a famous landmark located in Rome.	✔
Relation: main subject	Head Entity: The Theory of Relativity,, Tail Entity: Albert Einstein,, Context: The Theory of Relativity, developed by Albert Einstein, is primarily focused on the study of the fundamental laws of physics	Incorrect Relation
Relation: main subject	Head Entity: Leonardo da Vinci's, Tail Entity: Mona Lisa,, Context: Leonardo da Vinci's famous painting, the Mona Lisa, showcases her enigmatic smile as the main subject.	Incorrect Relation
Relation: operating system	Head Entity: macOS Sierra, Tail Entity: Apple Inc., Context: macOS Sierra is an operating system developed by Apple Inc.	Incorrect Relation
Relation: operating system	Head Entity: Tesla Model 3, Tail Entity: Tesla OS,, Context: The Tesla Model 3 is equipped with the Tesla OS, an advanced operating system.	✔

Table 8: Generated cases for GPT-3.5 on five unseen relations of FewRel.

Prompt
The given sentence is César Gaviria Trujillo Airport is an airport serving the town of Inírida in the Guainía Department of Colombia .
List of given relations: ['located on terrain feature', 'field of work', 'distributed by', 'contains administrative territorial entity', 'spouse']
What relations in the given list might be included in this given sentence? If not present, answer: none. Respond in the form of (head entity1, tail entity1, relation1), (head entity2, tail entity2, relation2), ...
Response
(César Gaviria Trujillo Airport, Inírida, located on terrain feature)

Figure 9: Prompt for ICL.

Prompt for relations
The given sentence is César Gaviria Trujillo Airport is an airport serving the town of Inírida in the Guainía Department of Colombia .
List of given relations: ['field of work', 'contains administrative territorial entity', 'located on terrain feature', 'distributed by', 'spouse']
What relations in the given list might be included in this given sentence? If not present, answer: none. Respond as a tuple, e.g. (relation 1, relation 2, ...)
Response
(contains administrative territorial entity, located on terrain feature)

Prompt for entities
According to the given sentence, the relation between them is contains administrative territorial entity, **find the head and tail entities** and list them all by group if there are groups. If not present, answer: none. Respond in the form of (head entity1, tail entity1), (head entity2, tail entity2), ...
Response
none

Prompt for entities
According to the given sentence, the relation between them is located on terrain feature, **find the head and tail entities** and list them all by group if there are groups. If not present, answer: none. Respond in the form of (head entity1, tail entity1), (head entity2, tail entity2), ...
Response
(César Gaviria Trujillo Airport, terrain feature: Inírida in the Guainía Department of Colombia)

Figure 10: Prompt for ChatIE.

Prompt for generating triplets from a relation
Given a relation, generate the head and tail entities to compose the relation triplet of the form (head entity, tail entity, relation).
For example:
Given the relation composer, we have triplets: (Wolfgang Amadeus Mozart, Symphony No. 40., composer).
Now given the relation: composer, please generate several triplets.
Response
Given the relation composer, here are several triplets:
1. (Johann Sebastian Bach, Mass in B minor, composer)
.....

Prompt for generating sentences from a triplet
Generate a sentence with the given (head entity, tail entity, relation) triplet .
For example:
Given the triplet ('Ludwig van Beethoven', 'Symphony No. 5.', 'composer'), we have sentence: Ludwig van Beethoven is the composer of Symphony No. 5.Now given the triplet: ('Ludwig van Beethoven', 'Symphony No. 9', 'composer').
Now given the triplet: ('Wolfgang Amadeus Mozart', 'Symphony No. 41', 'composer'), please generate the sentence.
Response
Wolfgang Amadeus Mozart composed Symphony No. 41.

.....

Figure 11: Prompt for the LLM-based generator in TAG.