Boosting Triple Extraction and Relation Normalization with Trained Retrieval and Policy Optimization

1st Yu Bai
School of Computer Science
Shenyang Aerospace University
Shenyang, China
baiyu@sau.edu.cn

2nd Jiansheng Bian
School of Computer Science
Shenyang Aerospace University
Shenyang, China
bianjiansheng@stu.sau.edu.cn

3rd Jianjun Chen
Shenyang Northern Software College
of Information Technology
Shenyang, China
chenjj4003@163.com

***Abstract*—Triple extraction and relation normalization are essential for constructing usable knowledge graphs from text, yet deployable systems often struggle to meet strict usability constraints such as stable formatting, reliable parseability, and consistent relation naming. While prior lightweight large language models (LLMs) pipelines improve robustness through ensembling and definition-aware normalization, they typically rely on heuristic selection and remain misaligned with set-level evaluation metrics. We propose a method that strengthens deployable extraction pipelines along two complementary axes: first, trained retrieval for schema guidance and second, policy optimization for metric-level alignment. We train a schema retriever to embed texts and relations into a relevance-oriented vector space, enabling efficient retrieval of relations likely to appear in the input, including long-tail and low-salience relations. The retriever is fine-tuned from a strong instruction-tuned embedding model using an InfoNCE objective, with positive text-relation pairs and negative relations. We then formulate structured triple generation as a policy optimization problem and fine-tune a lightweight actor with parameter-efficient adapters using Proximal Policy Optimization (PPO) style updates. A hybrid reward is designed, combining a format/parseability signal, a set-level gold-matching score, and a baseline-relative improvement term to encourage gains without regressions. Experiments on standard triple extraction benchmarks show that our approach consistently improves upon a strong baseline across multiple matching criteria, with especially noticeable gains under stricter settings. These results demonstrate that combining trained retrieval with reward-aligned policy optimization is an effective and practical approach to improving both triple quality and consistent relation normalization.**

***Index Terms*—Triple Extraction, Relation Normalization, Schema Retrieval, Reinforcement Learning**

I. INTRODUCTION

Triple extraction and relation normalization are fundamental to knowledge graph construction from text. Given an input sentence or paragraph, the goal is to generate a set of subject–relation–object triples that can support downstream applications such as question answering and structured retrieval [1]. Although recent instruction-following LLMs have made end-to-end extraction more practical, real-world deployment requires outputs that are not only plausible, but also parseable, schema-consistent, and robust under strict set-level evaluation [2].

A major challenge lies in the mismatch between training and evaluation. Most generation-based extractors are optimized with token-level maximum likelihood, while practical extraction quality is measured at the tuple-set level. As a result, models may produce fluent outputs that still violate formatting constraints, miss exact boundaries, or fail to map open relation phrases to predefined schema labels, especially when the schema is large and relation expressions are diverse.

Our previous MEERC framework improves robustness and relation consistency through a deployable lightweight pipeline [3]. However, it still depends on the quality of schema retrieval and does not directly optimize the set-level metrics used in evaluation and deployment. To address these limitations, we propose MEERC-TRPO, which extends MEERC with two complementary components: a trained schema retriever and reward-aligned policy optimization. The trained retriever learns a relevance-oriented embedding space for text–relation matching, providing more reliable schema candidates for normalization. On top of this, we optimize triple generation with PPO-style updates and a hybrid reward that reflects parseability, tuple-level correctness, and improvement over a baseline policy [4]–[6].

Experiments on standard triple extraction benchmarks show that MEERC-TRPO consistently outperforms MEERC under multiple matching criteria, with especially clear gains under stricter settings. These results indicate that combining trained retrieval with policy optimization is an effective way to improve both extraction quality and schema-consistent normalization in lightweight, deployable systems.

Our contributions are as follows: (1) we introduce a trained schema retriever for large-schema triple extraction and normalization; (2) we incorporate PPO-based policy optimization with a hybrid reward tailored to structured generation; and (3) we demonstrate consistent improvements over MEERC while preserving deployment efficiency through parameter-efficient fine-tuning.

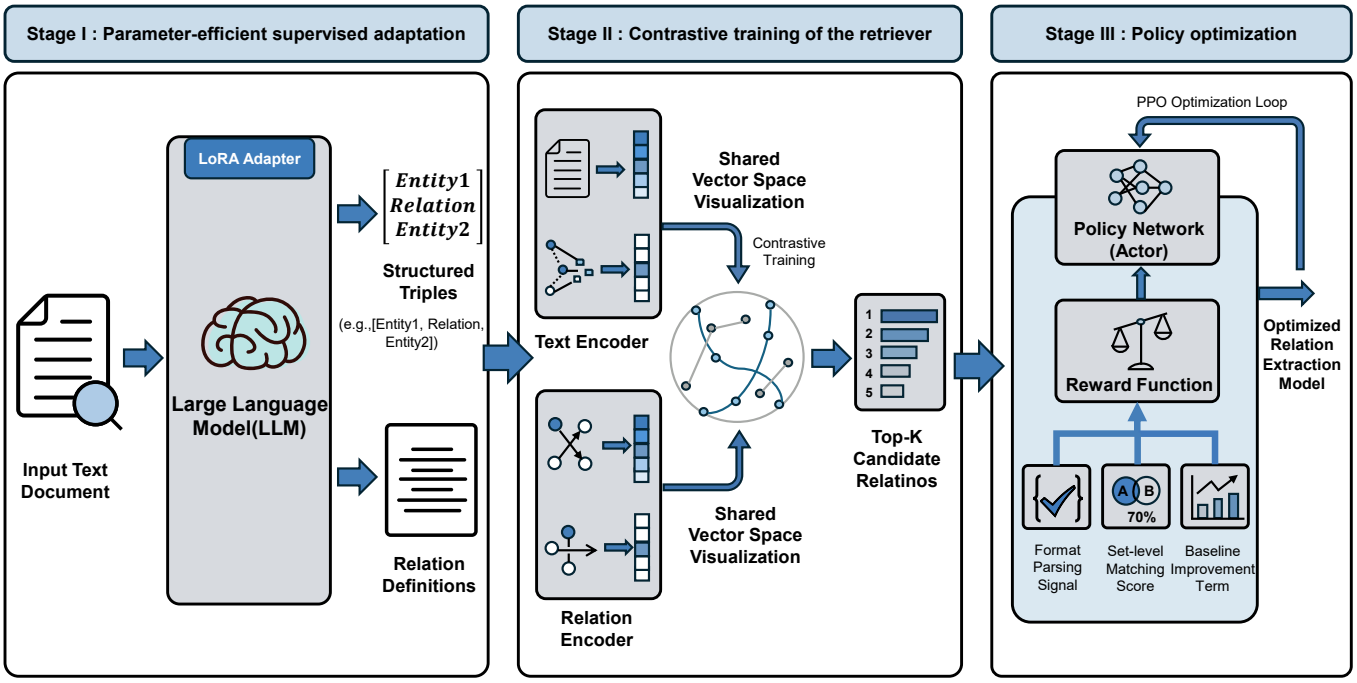

Fig. 1: The overall framework of MEERC-TRPO.

II. RELATED WORK

Generative information extraction formulates structured prediction as sequence generation, allowing models to produce multiple triples in a unified decoding process. REBEL shows that end-to-end generation can achieve competitive relation extraction performance with structured output design [4]. Beyond direct extraction, EDC and our previous MEERC further improve relation normalization through definition-based canonicalization and lightweight deployable pipelines [1], [3]. However, these methods are still mainly trained with maximum likelihood and therefore do not directly optimize the set-level correctness and parseability required in downstream use.

Retrieval-augmented methods provide an effective way to scale schema guidance for large relation spaces. Prior work such as RAG, DPR, and contrastive text embedding models demonstrates the value of relevance-oriented retrieval for knowledge-intensive tasks [2], [7], [8]. Different from using fixed embedding spaces, our method trains a schema retriever specifically for text–relation matching, so that retrieved candidates better reflect relation relevance in the input.

Reinforcement learning offers a natural way to optimize generation against non-differentiable output-level objectives. RLHF-style methods and PPO-based optimization have shown strong effectiveness in aligning language model behavior beyond likelihood training [5], [6]. Related work has also explored RL for improving generation quality and knowledge-related objectives [9]. Building on this direction, we treat structured triple generation as a policy and optimize it with rewards defined on parsed triple sets.

III. PROBLEM DEFINITION

We study open information extraction with schema-guided relation normalization. Given an input sentence or short passage x, the goal is to predict a set of schema-aligned triples

$$T(x) = \{(h, r, t) \mid r \in R\}, \tag{1}$$

where h and t denote the head and tail entities, and R is a predefined relation schema. Since raw extraction may produce free-form relation phrases $\tilde{r}$, the task consists of two stages: generating candidate triples

$$\tilde{T}(x) = \{(h, \tilde{r}, t)\}, \tag{2}$$

and mapping each $\tilde{r}$ to a canonical schema relation $r \in R$.

We evaluate predictions at the triple-set level against gold annotations $\mathcal{T}^*(x)$ using precision, recall, and F_1 under benchmark matching protocols. To support reliable optimization, model outputs are required to follow a deterministic structured format so that each generated output y can be parsed into $T(x)$. This also enables using format validity and set-level matching quality as reward signals in the policy optimization stage.

IV. METHOD

In this section, we present MEERC-TRPO, which improves triple extraction and schema-guided relation normalization through three stages: parameter-efficient supervised adaptation, contrastive schema retriever training, and PPO-based policy optimization. At inference time, the retriever provides a compact top-K candidate relation set for canonicalization, reducing distractors under large schemas. The three stages are described in Section IV-A, Section IV-B, and Section IV-C, respectively (see Fig. 1).

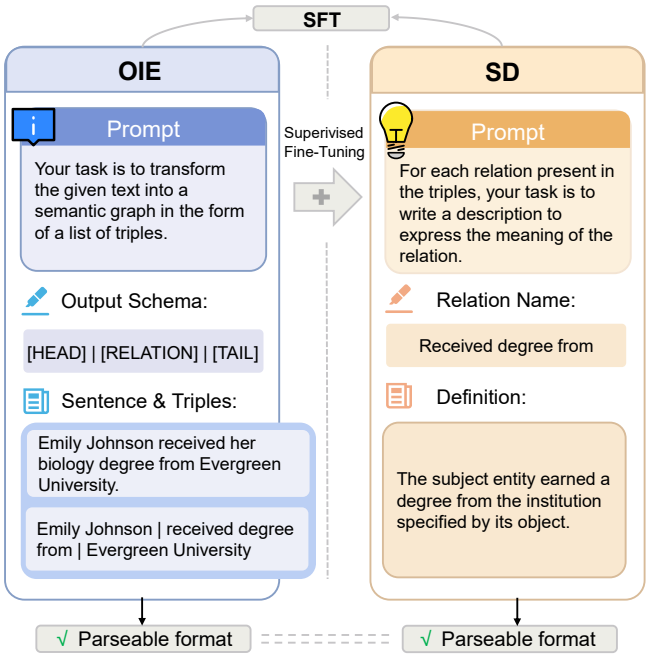

Fig. 2: Instruction-Based SFT Data for OIE and SD.

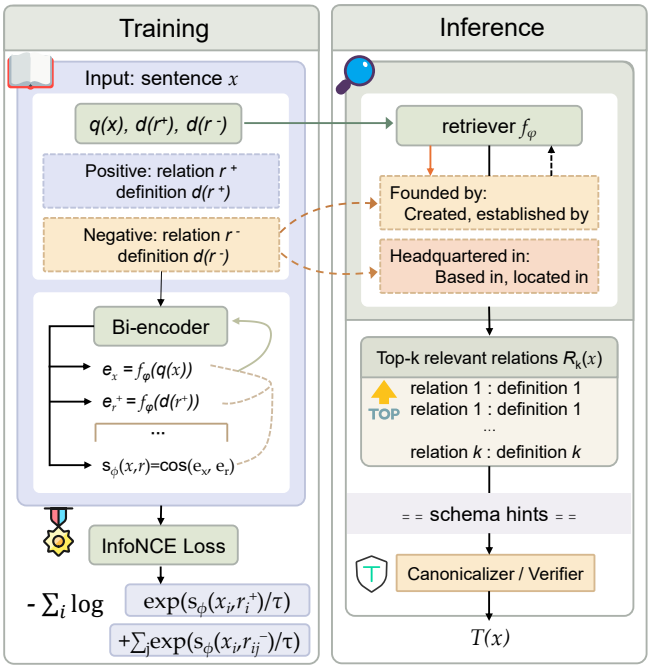

Fig. 3: Schematic of the contrastively trained schema retriever.

A. Parameter-Efficient Supervised Adaptation

We adapt the open information extraction (OIE) model and the schema-definition (SD) generator via parameter-efficient supervised fine-tuning. Starting from an instruction-following LLM, we apply LoRA to obtain task-specific updates while keeping the backbone frozen. For a base model with parameters θ_0, the adapted parameters are

$$\theta = \theta_0 + \Delta\theta, \tag{3}$$

where $\Delta\theta$ is restricted to a low-rank subspace. We optimize the standard negative log-likelihood over instruction–response pairs:

$$L_{\text{SFT}}(\theta) = -\sum_{(u,v)\in D} \log p_\theta(v \mid u), \tag{4}$$

where u denotes the instruction-style input and v the target output.

For OIE, the model takes the input sentence x together with a task instruction and generates a parseable set of raw triples

$$\tilde{T}(x) = \{(h, \tilde{r}, t)\}. \tag{5}$$

For SD, the model maps each relation label to a short textual definition, which is later used by the retrieval and canonicalization modules. Jointly, these two tasks improve both raw triple generation and schema-aware normalization. The instruction formats used for OIE-SFT and SD-SFT are illustrated in Fig. 2.

B. Contrastive training of the retriever

We train a dense schema retriever to select a compact candidate relation set for each input sentence, improving schema-guided canonicalization under large relation spaces. For an input sentence x, we construct an instruction-style query text $q(x)$, and for each schema relation $r \in R$, we construct a textual key $d(r)$ by combining the relation name with its definition from Section IV-A. Retrieval is then performed over these definition-augmented relation representations.

We use a bi-encoder parameterized by φ to map queries and relation texts into a shared embedding space:

$$e_x = f_\varphi\big(q(x)\big), \quad e_r = f_\varphi\big(d(r)\big) \tag{6}$$

and define the relevance score by cosine similarity:

$$s_\varphi(x, r) = \cos\big(e_x, e_r\big). \tag{7}$$

Training instances consist of a positive relation r^+ expressed in x and negatives $\{r_j^-\}$ not expressed in x. The retriever is optimized with the InfoNCE objective:

$$L_{\text{ret}}(\varphi) = -\sum_i \log \frac{\exp\big(s_\varphi(x_i, r_i^+)/\tau\big)}{Z_i} \tag{8}$$

$$Z_i = \exp\big(s_\varphi(x_i, r_i^+)/\tau\big) + \sum_{j=1}^{M} \exp\big(s_\varphi(x_i, r_{ij}^-)/\tau\big). \tag{9}$$

At inference time, we retrieve the top-K relations ranked by $s_\varphi(x, r)$, yielding the candidate set $R_K(x) \subseteq R$. Only relations in $R_K(x)$, together with their definitions, are passed to the canonicalizer/verifier to map each free-form relation phrase $\tilde{r}$ to a schema label $r \in R$. This reduces distractors and improves normalization robustness under large schemas. The overall retrieval and top-K conditioning process is illustrated in Fig. 3.

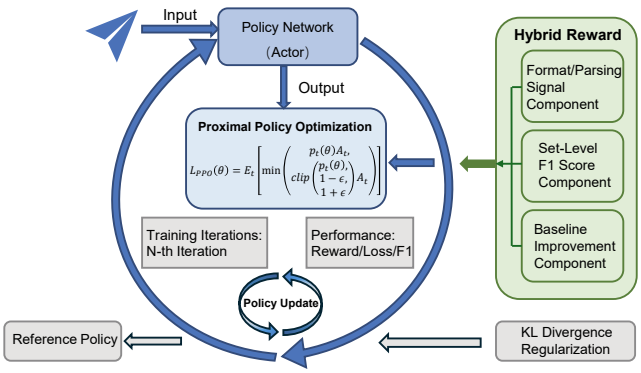

Fig. 4: Overview of the Policy Optimization stage.

C. Policy Optimization with Reward Shaping

We further refine MEERC-TRPO with reinforcement learning to better align generation with the downstream set-level objective. Let $\pi_\theta(y \mid x)$ denote the current policy that generates a structured output y for input x, and let π_{ref} denote the reference policy initialized from the LoRA-SFT model.

1) Reward design.: We define a hybrid reward $R(x, y)$ to encourage both validity and correctness. Specifically, $R_{\text{fmt}}(x, y) \in \{0, 1\}$ indicates whether y can be parsed into a valid triple set, and $R_{\text{set}}(x, y)$ measures the set-level matching quality between the parsed triples and the gold set $\mathcal{T}^*(x)$. To encourage improvement over the supervised baseline, we further define

$$R_\Delta(x, y) = R_{\text{set}}(x, y) - R_{\text{set}}(x, \hat{y}^0), \tag{10}$$

where $\hat{y}^0$ is the output of the reference policy on the same input. The final reward is

$$R(x, y) = \lambda_1 R_{\text{fmt}}(x, y) + \lambda_2 R_{\text{set}}(x, y) + \lambda_3 R_\Delta(x, y), \tag{11}$$

where $\lambda_1, \lambda_2, \lambda_3 \geq 0$ are weighting coefficients.

2) PPO objective.: We optimize the policy using PPO with the clipped surrogate objective

$$L_{\text{PPO}}(\theta) = \mathbb{E}_t\Big[\min\big(\rho_t(\theta)\,\hat{A}_t,\ \text{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon)\,\hat{A}_t\big)\Big], \tag{12}$$

where

$$\rho_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}. \tag{13}$$

To maintain stability, we add KL regularization to the reference policy:

$$L(\theta) = -L_{\text{PPO}}(\theta) + \beta\,\mathbb{E}_x\big[\text{KL}(\pi_\theta(\cdot \mid x) \,\|\, \pi_{\text{ref}}(\cdot \mid x))\big]. \tag{14}$$

This encourages the policy to improve set-level extraction quality while preserving structured and parseable outputs. The PPO training process is illustrated in Fig. 4.

V. EXPERIMENTS

A. Datasets

We evaluate MEERC-TRPO on three public text-to-triples benchmarks with diverse relation schemas:

(1) **WebNLG** [10]: We adopt the test split from the WebNLG+2020 (v3.0) semantic parsing task. This dataset comprises 1,165 text-triple pairs, derived from a reference schema that includes 159 distinct relation types.

(2) **REBEL** [2]: The original test partition of REBEL contains 105,516 entries. To ensure computational feasibility, we randomly sample 1,000 text-triple pairs, resulting in a schema encompassing 200 unique relation types.

(3) **Wiki-NRE** [11]: From the 29,619 entries in the Wiki-NRE test split, we sample 1,000 text-triple pairs, which induce a schema comprising 45 unique relation types.

We select these datasets because their relation schemas are substantially more diverse than those used in many prior LLM-based studies (e.g., ADE [12], ScMEERC [13], and CoNLL04 [14]), and thus better reflect real-world knowledge graph construction scenarios. Our experiments focus on relation extraction and schema-guided relation canonicalization, since relations form the core structure of a knowledge graph and directly determine its utility for downstream applications; however, the MEERC-TRPO framework can be extended to incorporate additional schema components when needed.

In addition to the evaluation benchmarks, we synthesize supervision for Section IV-A and Section IV-B from the TEKGEN dataset [15], a large-scale text–triplets resource constructed by aligning Wikidata triples with Wikipedia sentences. We convert TEKGEN into instruction-format examples for open triple extraction and relation definition generation, and we further derive text–relation relevance pairs for retriever training by treating relations expressed in the aligned triples as positives and sampling non-expressed relations as negatives. TEKGEN is used only for training these components, while all reported results are evaluated on the public benchmarks described above.

B. Implementation

We conduct systematic experiments on three public datasets, using an NVIDIA A800 GPU and the PyTorch framework. All LLMs are open-source instruction-tuned models with fewer than 10 billion parameters, including Llama-3.1-8B, Mistral-7B, and InternLM3-8B. Inference is performed locally by loading model weights. For each dataset, we choose the backbone LLM used for structured triple generation and definition-aware canonicalization based on validation performance under strict set-level matching, and we keep the same backbone for all MEERC-TRPO variants on that dataset to ensure fair comparisons.

C. Baseline

To comprehensively evaluate MEERC-TRPO, we compare it against representative state-of-the-art baselines in structured information extraction and knowledge graph construction:

TABLE I: Main Results of MEERC-TRPO on Public KGC Datasets

Dataset	Method	Partial			Strict			Exact		
		P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
WebNLG	REGEN	75.5	78.8	76.7	71.3	73.5	72.0	71.4	73.8	72.3
	EDC	73.9	76.0	74.6	68.4	69.7	68.8	70.8	72.2	71.3
	MEERC	76.4	78.9	77.6	71.0	72.6	71.8	73.3	76.0	74.6
	MEERC-TRPO	**81.7**	**84.1**	**82.9**	**74.0**	**76.2**	**75.1**	**77.1**	**79.5**	**78.3**
REBEL	GenIE	38.1	39.1	38.5	35.3	36.1	35.6	36.2	36.9	36.4
	EDC	50.3	51.2	50.6	44.8	45.3	44.9	47.1	47.6	47.3
	MEERC	55.5	57.9	56.7	49.1	51.8	50.4	52.0	54.4	53.2
	MEERC-TRPO	**62.8**	**65.0**	**63.9**	**53.7**	**56.1**	**54.9**	**58.1**	**60.7**	**59.4**
Wiki-NRE	GenIE	48.2	48.6	48.4	46.2	46.4	46.3	47.7	47.9	47.8
	EDC	64.5	65.1	64.7	63.6	64.0	63.8	63.8	64.3	64.0
	MEERC	67.8	68.5	68.1	65.3	66.7	65.9	66.1	67.8	66.8
	MEERC-TRPO	**73.5**	**75.9**	**74.7**	**69.1**	**71.7**	**70.4**	**71.4**	**73.6**	**72.5**

TABLE II: Ablation Study Results of MEERC-TRPO Components

Dataset	Method	Partial			Strict			Exact		
		P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
WebNLG	MEERC	76.4	78.9	77.6	71.0	72.6	71.8	73.3	76.0	74.6
	+SFT (LoRA init)	78.6	81.0	79.8	72.3	74.1	73.2	75.0	77.4	76.2
	+SFT + TR (w/o PPO)	80.2	82.5	81.3	73.2	75.2	74.2	76.0	78.4	77.2
	MEERC-TRPO (full)	81.7	84.1	82.9	74.0	76.2	75.1	77.1	79.5	78.3
REBEL	MEERC	55.5	57.9	56.7	49.1	51.8	50.4	52.0	54.4	53.2
	+SFT (LoRA init)	57.4	60.2	58.7	50.1	52.8	51.4	54.1	56.8	55.4
	+SFT + TR (w/o PPO)	60.3	63.0	61.6	51.8	54.3	53.0	56.3	58.9	57.5
	MEERC-TRPO (full)	62.8	65.0	63.9	53.7	56.1	54.9	58.1	60.7	59.4
Wiki-NRE	MEERC	67.8	68.5	68.1	65.3	66.7	65.9	66.1	67.8	66.8
	+SFT (LoRA init)	69.6	71.0	70.3	66.6	68.4	67.5	68.2	69.9	69.0
	+SFT + TR (w/o PPO)	71.8	73.6	72.7	67.9	69.8	68.8	70.0	72.1	70.9
	MEERC-TRPO (full)	73.5	75.9	74.7	69.1	71.7	70.4	71.4	73.6	72.5

(1) **REGEN** [9]: This method is a T5-based sequence-to-sequence model optimized using a reinforcement learning (RL) mechanism. It jointly learns text-to-graph and graph-to-text generation tasks, enabling more effective modeling of the correspondence between graph structures and natural language. REGEN demonstrates strong performance on the WebNLG dataset.

(2) **GenIE** [16]: Based on the pretrained language model BART, GenIE introduces a structure-constrained generation strategy to guide the model in producing triples that conform to predefined schemas. This significantly enhances structural completeness and semantic consistency. GenIE performs well on both the REBEL and Wiki-NRE datasets and is one of the leading schema-aware extraction methods.

(3) **EDC** [17]: EDC is a three-stage structured KGC method that employs a schema retriever trained on large-scale data and leverages the interpretive power of the large-parameter GPT-3.5 model to realize a structured "explain-define-canonicalize" workflow. EDC achieves strong performance across WebNLG, REBEL, and Wiki-NRE.

In addition, we include our prior work MEERC [3] as the primary deployable baseline to quantify improvements brought by trained retrieval and policy optimization.

D. Evaluation

Following previous work [18], we adopt the WEBNLG evaluation script, which computes token-based precision (P), recall (R), and F1 score by comparing the predicted triples against the reference gold standard. Named entity-based evaluation metrics are used to assess performance under three distinct matching conditions:

Partial: for each element of the triple, the candidate string should match at least partially with the reference string, and the element type (subject, predicate, object) is irrelevant.

Strict: for each element of the triple, the candidate must exactly match the reference in both value and type (subject, predicate, or object).

Exact: for each element of the triple, exact match of the candidate string with the reference is required, and the element type (subject, predicate, object) is irrelevant.

E. Result Analysis

Table I reports the main results on WebNLG, REBEL, and Wiki-NRE under Partial, Strict, and Exact matching. MEERC-TRPO consistently achieves the best performance across all three datasets and all matching criteria, showing that the combination of trained schema retrieval and reward-aligned policy optimization improves both extraction quality and schema-consistent canonicalization in a deployable pipeline.

The gains are especially clear under stricter evaluation settings. On WebNLG, Exact F_1 improves from 74.6 to 78.3, indicating better boundary accuracy and output stability. On the larger-schema REBEL benchmark, Exact F_1 rises from 53.2 to 59.4, suggesting that trained retrieval provides more reliable candidate relations for normalization. Similar improvements on Wiki-NRE further demonstrate that the proposed method generalizes across datasets with different schema sizes and relation distributions.

F. Ablation Study

Table II presents an ablation study that progressively adds the main components of MEERC-TRPO to the MEERC baseline. The improvements are cumulative across datasets and matching criteria, indicating that supervised adaptation, trained schema retrieval, and policy optimization address complementary bottlenecks in triple extraction and relation canonicalization.

Adding LoRA-based SFT consistently improves structured generation quality and output stability. Introducing the trained schema retriever yields further gains, especially under stricter criteria where schema confusion becomes more prominent. Finally, PPO-based policy optimization provides an additional boost, showing that reward-aligned optimization improves set-level correctness beyond supervised adaptation and retrieval alone. For example, on Wiki-NRE, Strict F_1 increases from 68.8 to 70.4 and Exact F_1 from 70.9 to 72.5 after adding PPO. Overall, the best performance is achieved when all three components are combined.

VI. Conclusion and Future Work

We presented MEERC-TRPO, a deployable framework for triple extraction and schema-guided relation canonicalization. By combining parameter-efficient supervised adaptation, trained schema retrieval, and PPO-based policy optimization, the proposed method consistently improves performance on three public benchmarks under Partial, Strict, and Exact matching. Future work will further improve retrieval robustness and extend the framework to richer schema settings.

References

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in neural information processing systems*, vol. 33, pp. 9459–9474, 2020.

[2] P.-L. H. Cabot and R. Navigli, "Rebel: Relation extraction by end-to-end language generation," in *Findings of the association for computational linguistics: emnlp 2021*, 2021, pp. 2370–2381.

[3] Y. Bai, J. Bian, Y. Liu, Y. Feng, and J. Chen, "Boosting triple extraction and relation normalization with ensembled lightweight llms," in *2025 IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA)*. IEEE, 2025, pp. 870–877.

[4] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, "Lora: Low-rank adaptation of large language models." *ICLR*, vol. 1, no. 2, p. 3, 2022.

[5] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[6] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *Advances in neural information processing systems*, vol. 35, pp. 27 730–27 744, 2022.

[7] V. Karpukhin, B. Oguz, S. Min, P. S. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, "Dense passage retrieval for open-domain question answering." in *EMNLP (1)*, 2020, pp. 6769–6781.

[8] L. Wang, N. Yang, X. Huang, B. Jiao, L. Yang, D. Jiang, R. Majumder, and F. Wei, "Text embeddings by weakly-supervised contrastive pre-training," *arXiv preprint arXiv:2212.03533*, 2022.

[9] P. Dognin, I. Padhi, I. Melnyk, and P. Das, "Regen: Reinforcement learning for text and knowledge base generation using pretrained language models," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 1084–1099.

[10] T. C. Ferreira, C. Gardent, N. Ilinykh, C. Van Der Lee, S. Mille, D. Moussallem, and A. Shimorina, "The 2020 bilingual, bi-directional webnlg+ shared task overview and evaluation results (webnlg+ 2020)," in *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, 2020.

[11] B. Distiawan, G. Weikum, J. Qi, and R. Zhang, "Neural relation extraction for knowledge base enrichment," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 229–240.

[12] H. Gurulingappa, A. M. Rajput, A. Roberts, J. Fluck, M. Hofmann-Apitius, and L. Toldo, "Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports," *Journal of biomedical informatics*, vol. 45, no. 5, pp. 885–892, 2012.

[13] Y. Luan, L. He, M. Ostendorf, and H. Hajishirzi, "Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 3219–3232.

[14] D. Roth and W.-t. Yih, "A linear programming formulation for global inference in natural language tasks," in *Proceedings of the eighth conference on computational natural language learning (CoNLL-2004) at HLT-NAACL 2004*, 2004, pp. 1–8.

[15] O. Agarwal, H. Ge, S. Shakeri, and R. Al-Rfou, "Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 3554–3565.

[16] M. Josifoski, N. De Cao, M. Peyrard, F. Petroni, and R. West, "Genie: Generative information extraction," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2022, pp. 4626–4643.

[17] B. Zhang and H. Soh, "Extract, define, canonicalize: An llm-based framework for knowledge graph construction," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024, pp. 9820–9836.

[18] I. Melnyk, P. Dognin, and P. Das, "Knowledge graph generation from text," in *Findings of the Association for Computational Linguistics: EMNLP 2022*, 2022, pp. 1610–1622.