MARIO-0.5B: A Multi-Agent Lightweight Model for Real-Time Open Information Extraction in Low-Resource Settings

Donghai Zhang
Knowdee Intelligent
zhangdh@knowdee.com

Shuangtao Yang
Knowdee Intelligent
yangst@knowdee.com

Xiaozheng Dong
Knowdee Intelligent
dongxz@knowdee.com

Wei Song *
Capital Normal University
wsong@cnu.edu.cn

Bo Fu *
Knowdee Intelligent
fubo@knowdee.com

Abstract

Large language models (LLMs) have shown remarkable capabilities in open information extraction. However, their substantial resource requirements often restrict their deployment in resource-constrained industrial settings, particularly on edge devices. The high computational demands also lead to increased latency, making them difficult to apply in real-time applications. In this paper, we introduce MARIO-0.5B, an ultra-lightweight model trained on instruction-based samples in Chinese, English, Korean, and Russian. We also present a novel multi-agent framework, SMOIE, which integrates schema mining, information extraction, reasoning, and decision-making to effectively support MARIO-0.5B. The experimental results show that our framework outperforms large-scale models with up to 70B parameters, reducing computational resources by 140x and delivering 11x faster response times. Moreover, it operates efficiently in CPU-only environments, which makes it well-suited for widespread industrial deployment.

1 Introduction

Open Information Extraction (OIE) is the task of extracting triples (subject, predicate, object) from given text without a fixed schema (Etzioni et al., 2008). Large Language Models (LLMs) have shown success in extracting triples from simple sentences using content-based learning approaches (Wei et al., 2024; Wadhwa et al., 2023).

However, LLMs often fail to capture all relevant triples (Ding et al., 2024), they still struggle with more complex sentences that contain multiple triples or numerous entities and relations. Another challenge is the scale of the parameters. Advanced models such as ChatGPT (Achiam et al., 2024), Qwen (Yang et al., 2025), and DeepSeek (DeepSeek-AI et al., 2024) require substantial computational resources during training and inference, which are typically run on high-performance servers. And heavy computations often result in response delays.

There is a growing demand for models that can operate efficiently in low-resource environments like CPUs to enable deployment on edge devices. However, the substantial resource requirements and response latency of LLMs hinder their use in information extraction. Most current research focuses on large-scale models like ChatGPT, while smaller, lightweight LLMs and agent frameworks have yet to be explored. Inspired by (Yin et al., 2023), which stated that instructions containing label information can stimulate model capabilities, we are interested in this research: whether ultra-small, lightweight models can achieve the performance of their larger counterparts in resource-constrained environments by incorporating potential label information into instructions through collective effects.

In this paper, we introduce MARIO-0.5B, an ultra-lightweight model trained on instruction-following samples in multiple languages, including Chinese, English, Korean, and Russian. We also present a novel multi-agent framework, SMOIE, which integrates schema mining, information extraction, reasoning, and decision-making to effectively support MARIO-0.5B. The framework aims to progressively enrich label information required for OIE via efficient group decision-making, thereby enhancing the reasoning capabilities of the MARIO-0.5B model.

The evaluation results show that the multi-agent framework we proposed achieves a significant 6.9% improvement in the F1 score, outperforms large-scale models with up to 70B parameters and delivers performance comparable to GPT-4o (Achiam et al., 2024). Additionally, it reduces computational resources by 140 times and delivers 11 times faster response times, which enables efficient operation

*Corresponding authors

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 13085–13094

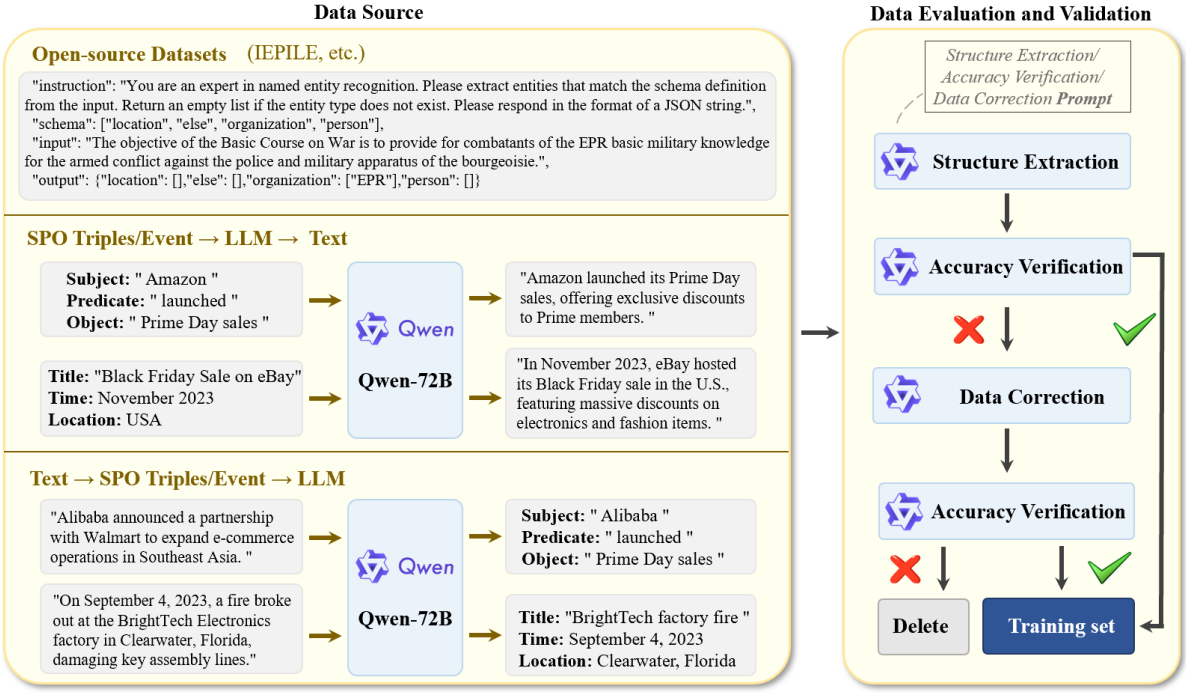

Figure 1: Training data construction and filtering

in a CPU-only environment.

2 Related Work

Early methods for the extraction of entities and relations relied on separate models for each task, leadingo cascading potential errors throughout the process (Brin, 1998). The rise of deep learning introduced new end-to-end models that could handle both tasks simultaneously, sharing information and optimizing performance together (Wang and Lu, 2020; Zhao et al., 2021; Yan et al., 2021).

Recently, LLMs have attracted attention due to their impressive performance across various NLP tasks. With the right prompts, LLMs can match the performance of specially trained models in extraction tasks, even without prior examples or with only a few examples (Gao et al., 2023; Tang et al., 2023; Jeblick et al., 2022). However, these studies have some blind spots: they neither evaluate the models' capability in handling complex sentences with multiple entities and relations, nor explore effective methods for achieving high performance under resource constraints.

3 Supervised Fine-Tuning

3.1 Model Architecture and Training Setup

We introduce MARIO-0.5B, a compact information extraction model containing 500 million parameters, to enable efficient agent operations described in Section 4. Based on the Qwen2.5 series architecture (Team, 2024), this model employs a Transformer Decoder-only framework with 16 attention heads across 24 layers. Through instruction-based supervision, we fine-tuned MARIO-0.5B on more than 2 million samples spanning multiple tasks, including schema construction, information extraction, and reasoning chain generation.

We use LoRA (Hu et al., 2021) for training, setting the rank to 8. The Adam optimizer (Kingma and Ba, 2017) is applied with an initial learning rate of 1.0e-5 and a warm-up ratio of 0.1.

3.2 Data Collection

Common crawl dataset consists of open-source data like IEPile (Gui et al., 2024), which spans multiple fields and languages. It employs a schema-based polling instruction method to make it suitable for open-domain information extraction tasks.

Text2Triples synthesis dataset is generated by extracting triples and events from approximately 500,000 excerpts, sourced from news, Chinese mythology novels, and social media comments.

Triples2Text synthesis dataset is collected by generating texts from given triples. We merge triples from both source datasets above and cluster them based on semantic similarity. For text generation, we iteratively select triples from these clusters to create candidate sets. To ensure topical diversity

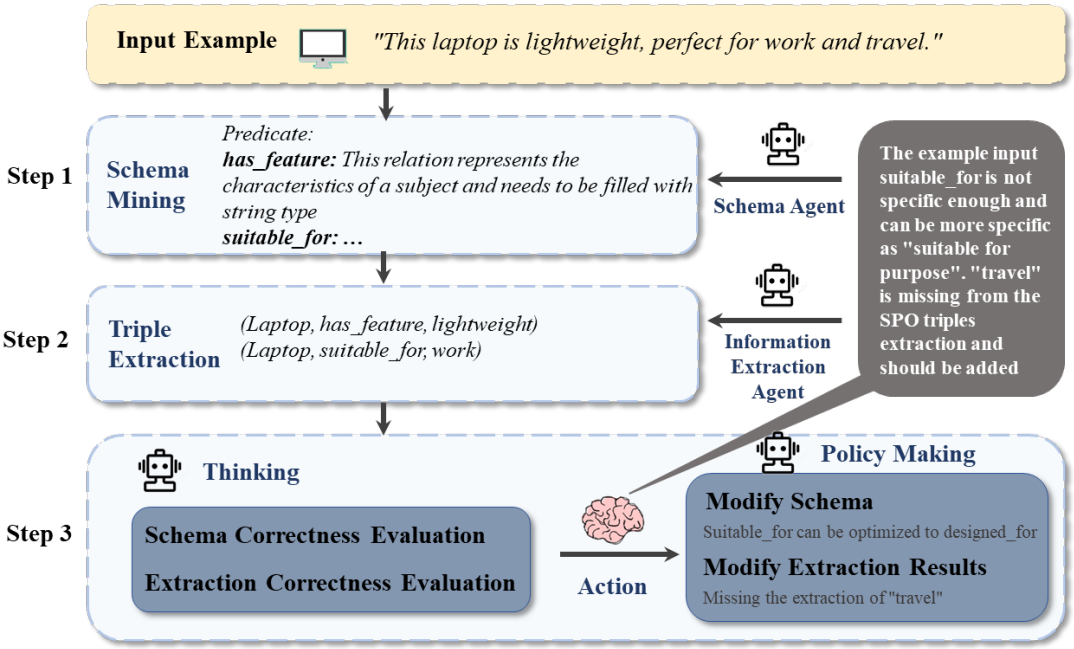

Figure 2: Lightweight multi-agent framework.

within individual text samples, 30% of our sampling process draws triples from different clusters rather than the same. We use carefully designed prompts to guide the (Yang et al., 2025) in generating both Text2Triples and Triples2Text datasets.

3.3 Data Processing

Both crowdsourced and synthetic datasets face quality issues. We propose a data processing pipeline to assess and validate dataset quality, as illustrated in Figure 1. We have designed three prompts for *Structural Extraction*, *Correctness Validation*, and *Error Correction*. For *Structural Extraction*, we add a new dimension, SCHEMA, to the work of (Zhang et al., 2024). Next, we design a *Correctness Validation* prompt (see Appendix A) to detect potential errors in the structures, addressing issues related to precision and recall. Once errors are identified, an *Error Correction* prompt (see Appendix B) is applied to fix them. The corrected data is then reintroduced into the validation process.

4 Multi Agents for OIE

We present a novel multi-agent framework for extracting triples from texts. The agent framework uses MARIO-0.5B as the base model. As shown in Figure 2, this framework employs the agent communication protocols to enhance system stability. These protocols enable several coordinated processes: schema mining, triple extraction, reasoning, and behavioral correction.

4.1 Schema Mining Agent

The predicate defines the specific relations between subject and object in the representation of tuples-based knowledge. We employ a specialized agent to identify these relations by analyzing semantic patterns and context. The agent works through a carefully designed schema generation task to help this process. For example, given the sentence "*This laptop is lightweight, perfect for work and travel.*", our agent identify multiple relations: "*has_feature*" to connect "laptop" with its property, and "*suitable_for*" to link "laptop" with its use cases.

This process is important for creating a clear schema, providing detailed explanations, and judging the expected types of subjects and objects for each relation. While the range of possible types is virtually limitless, the model autonomously makes determinations based on the context. The agent does not seek to normalize schemas, but rather to extract triplets for mutual validation, which strengthens reasoning.

4.2 Triple Extraction Agent

To minimize the confusion of inference, the agent is solely required to follow the schema and adhere to the expected value types, with no additional restrictions. Its ability to follow instructions is further refined during the fine-tuning phase. For a given sample, the agent is tasked with extracting the fol-

Methods	CmiCommentIE			CoRB		
	P	R	F1	P	R	F1
ChunkOIE	-	-	-	-	-	0.536
DetIELSOIE	-	-	-	-	-	0.521
GPT-4o	**0.910**	0.805	**0.854**	**0.751**	**0.533**	**0.623**
Qwen2.5-72B	0.895	0.794	0.842	0.695	0.521	0.595
Qwen2.5-32B	0.793	0.747	0.769	0.553	0.482	0.431
Ours	0.881	**0.812**	0.847	0.679	0.516	0.586

Table 1: Performance comparison for information extraction.

lowing triples: (*laptop*, *has_feature*, *lightweight*), (*laptop*, *suitable_for*, *work*), (*laptop*, *suitable_for*, *travel*).

4.3 Thinking Agent

We use LLMs to make critical evaluations. Our setting utilizes well-crafted instructions that guide the agent in performing a thorough self-assessment of its actions and the information it has gathered. The agent evaluates its previous steps by analyzing the generated schema and extracted triples. We allow for continuous cognitive flow during this analysis phase, rather than implementing fixed checkpoints.

All reasoning processes are contained within dedicated <think for verification> </think for verification> tags to maintain analytical clarity. This structured way guarantees a systematic and focused analysis at every stage.

4.4 Policy Making Agent

This agent implements system control policies based on its reasoning results. It manages the workflow by coordinating key tasks: *process termination*, *schema updating*, and *triples updating*. These operations must run sequentially rather than in parallel because of the tight coupling between schema configuration and triples extraction. We set a maximum number of iterations to ensure efficient execution.

5 Experiments

5.1 Data Setting

We conducted experiments on two datasets. **CaRB** (Bhardwaj et al., 2019) is a crowdsourced OpenIE dataset consisting of 1,282 sentences. It has become widely used for evaluating information extraction capabilities. **CmiCommentIE** focuses on reviews from the manufacturing and 3C industries. We collected 10,000 reviews from Chinese e-commerce platforms Taobao and JD. Over 6,000 triples were retained after following the data processing described in Section 3.3. Sensitive information such as names, addresses, and contact details has been removed to protect privacy and ensure data security.

Models	Size (B)	GPU (GB)	Latency (S)	CPU only
GPT-4o	-	-	6	×
Qwen2.5-72B	72	140	9	×
Qwen2.5-32B	32	63.72	3	×
Ours	**0.5**	**1.23**	**0.8**	✓

Table 2: Resource requirements and response latency for different models.

5.2 Baseline

ChunkOIE (Dong et al., 2023) utilizes BERT as its base model, replacing tokens with sentence-level chunks to construct a dependency graph. **DetIELSOIE** (Vasilkovsky et al., 2022) employs an order-agnostic loss function based on bipartite matching to ensure unique predictions for sequence labeling. Meanwhile, **GPT-4o** (Achiam et al., 2024), **Qwen2.5-72B** (Yang et al., 2025), and **Qwen2.5-32B** (Yang et al., 2025) all implement a few-shot approach for information extraction.

5.3 Results

Performance Comparison. The results are presented in Table 1 with precision(P), recall(R) and F1 score(F1). Results are reported as averages over 5-fold cross-validation. We can see that our method significantly improves the performance of information extraction, particularly in recall.

Our method shows distinct advantages over Qwen2.5-32B, with an average improvement of 5% in recall and 11.6% in F1. It also performs nearly on par with the large-scale 72B model. However,

despite these gains, GPT-4o still outperforms our method. Nevertheless, the slight improvement in recall is a promising result. We hypothesize that the agent framework translates the model's self-reasoning patterns into clear, actionable instructions, thereby enhancing its ability to identify valuable information. This finding holds significant potential in resource-limited industrial settings, where leveraging multiple weak agents may lead to enhanced information recognition capabilities.

We also compare our method with transformer-based models, achieving an average improvement of 5.8% in F1 score. Our method offers another key advantage: it excels in domain transfer, eliminating the need for specialized training in specific fields.

Resource Usage Comparison. We evaluate the resource requirements and response latency. As shown in Table 2, our agents show superior performance. Specifically, it requires just 1.23 GB of GPU memory, whereas the Qwen2.5-32B and Qwen2.5-72B models demand significantly more, at 63.72 GB and 140 GB, respectively. In terms of response latency, our agents respond in approximately 800ms without any performance optimizations, which means it is 3.7 times faster than Qwen2.5-32B, 11 times faster than Qwen2.5-72B, and 7.5 times faster than GPT-4o. Furthermore, Our framework can be deployed in a CPU-only environment, which is highly advantageous for industrial applications.

During evaluation we also notice a potential source of unfairness to the baseline. After fine tuning, SMOIE adheres to the prescribed output format much more reliably. Whenever the baseline deviates from that format and parsing fails, we must discard its prediction even if the extracted answer is correct, so its performance is undercounted.

5.4 Applications

SMOIE has been applied to RAG-based PC Troubleshooting. In the RAG-based setting for troubleshooting, domain-specific repair knowledge is retrieved to support the LLM. However, user queries often contain limited information, resulting in low similarity with relevant content and incomplete or inaccurate retrieval. To address this, SMOIE was introduced, yielding a 3% improvement in overall task performance.

SMOIE enhances the structure of sliced official repair manuals by incorporating entities alongside text embeddings as retrieval signals, leading to richer representations and more relevant knowledge retrieval. An ablation study on retrieval alone shows that SMOIE more effectively identifies key entities representing the main topic of a fragment, achieving a 4.7% gain in retrieval completeness over baseline methods.

We observe an interesting phenomenon: SMOIE tends to extract entities that are more useful for resolving the current query, which differs from the common assumption of exhaustive extraction.

6 Conclusion

In this paper, we explore an effective step-by-step deepening group decision-making approach to address the issues of insufficient knowledge and resource constraints in open information extraction. We introduce a small-scale 0.5B LLM and present an innovative multi-agent framework to support and enhance the model. Our proposed agent framework clarifies label information in instructions through schema mining, reasoning, and decision-making, thereby stimulating the reasoning abilities of small-scale models. Experimental results show the effectiveness of our approach in enhancing information extraction performance while significantly reducing dependence on computational resources, offering substantial benefits in resource-constrained scenarios.

7 Limitations

While our work offers valuable insights into the use of lightweight LLMs for information extraction in resource-limited scenarios, it does have some limitations. First, the reasoning process of the agent is still guided by traditional chain-of-thought techniques, which restricts the model to a predefined thought pattern rather than fostering self-validation. Second, cross-lingual performance is another limitation. While our training set includes data in Chinese, English, Korean, and Russian, this may hinder the framework's performance with other languages.

Future research could focus on integrating reinforcement learning to replace fixed supervised learning, enabling the model's reasoning process to evolve more independently and creatively, with supervision limited to the outcomes. Additionally, expanding the range of languages included in the model's fine-tuning process could improve its cross-lingual capabilities.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62376166).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Sangnie Bhardwaj, Samarth Aggarwal, and Mausam. 2019. CaRB: A crowdsourced benchmark for open IE. 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 6262–6267, Hong Kong, China. Association for Computational Linguistics.

Sergey Brin. 1998. Extracting patterns and relations from the world wide web. In *Selected Papers from the International Workshop on The World Wide Web and Databases*, WebDB '98, page 172–183, Berlin, Heidelberg. Springer-Verlag.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang,

Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. 2024. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Zepeng Ding, Wenhao Huang, Jiaqing Liang, Yanghua Xiao, and Deqing Yang. 2024. Improving recall of large language models: A model collaboration approach for relational triple extraction. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8890–8901, Torino, Italia. ELRA and ICCL.

Kuicai Dong, Aixin Sun, Jung-jae Kim, and Xiaoli Li. 2023. Open information extraction via chunks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15390–15404, Singapore. Association for Computational Linguistics.

Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S. Weld. 2008. Open information extraction from the web. *Commun. ACM*, 51(12):68–74.

Jun Gao, Huan Zhao, Changlong Yu, and Ruifeng Xu. 2023. Exploring the feasibility of chatgpt for event extraction. *Preprint*, arXiv:2303.03836.

Honghao Gui, Lin Yuan, Hongbin Ye, Ningyu Zhang, Mengshu Sun, Lei Liang, and Huajun Chen. 2024. Iepile: unearthing large scale schema-conditioned information extraction corpus. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 127–146.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685.

Katharina Jeblick, Balthasar Schachtner, Jakob Dexl, Andreas Mittermeier, Anna Theresa Stüber, Johanna Topalis, Tobias Weber, Philipp Wesp, Bastian Sabel, Jens Ricke, and Michael Ingrisch. 2022. Chatgpt makes medicine easy to swallow: An exploratory case study on simplified radiology reports. *Preprint*, arXiv:2212.14882.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. *Preprint*, arXiv:1412.6980.

Ruixiang Tang, Xiaotian Han, Xiaoqian Jiang, and Xia Hu. 2023. Does synthetic data generation of llms help clinical text mining? *Preprint*, arXiv:2303.04360.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Michael Vasilkovsky, Anton Alekseev, Valentin Malykh, Ilya Shenbin, Elena Tutubalina, Dmitriy Salikhov, Mikhail Stepnov, Andrey Chertok, and Sergey Nikolenko. 2022. Detie: Multilingual open information extraction inspired by object detection. *Preprint*, arXiv:2206.12514.

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.

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, Yong Jiang, and Wenjuan Han. 2024. Chatie: Zero-shot information extraction via chatting with chatgpt. *Preprint*, arXiv:2302.10205.

Zhiheng Yan, Chong Zhang, Jinlan Fu, Qi Zhang, and Zhongyu Wei. 2021. A partition filter network for joint entity and relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 185–197, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Fan Yin, Jesse Vig, Philippe Laban, Shafiq Joty, Caiming Xiong, and Chien-Sheng Jason Wu. 2023. Did you read the instructions? rethinking the effectiveness of task definitions in instruction learning. *Preprint*, arXiv:2306.01150.

Xinghua Zhang, Haiyang Yu, Cheng Fu, Fei Huang, and Yongbin Li. 2024. Iopo: Empowering llms with complex instruction following via input-output preference optimization. *arXiv preprint arXiv:2411.06208*.

Tianyang Zhao, Zhao Yan, Yunbo Cao, and Zhoujun Li. 2021. A unified multi-task learning framework for joint extraction of entities and relations. In *AAAI Conference on Artificial Intelligence*.

A Prompt for Correctness Validation

You are a fair judge, and please evaluate the quality of an AI assistant's responses to user query based on three key dimensions: relevance, fluency, and accuracy. You must provide detailed scoring and explanations for each dimension.

Requirements

1. For relevance evaluation, you must provide:
   - A relevance score (0-10)
   - Detailed explanations
2. For fluency evaluation, you must provide:
   - Analyze whether the prompt description is concise and correct
   - Analyze whether the prompt description is reader-friendly and user-friendly
   - Detailed explanations
3. For accuracy evaluation, you must provide:
   - Verify the completeness of the answer
   - Verify the accuracy of the answer
   - Detailed explanations

##Example##
#INPUT#

<Task Description>:
Relation extraction
<Constraints>:
NULL
<Schema>:
NULL
<Input>:
which also has offices in Putnam, Orange, Dutchess and Rockland Counties, as well as in the Bronx, Manhattan and Staten Island and in Bergen and Passaic Counties in New Jersey – already has some experience rejecting an insurer.
<Answer>:
"location contains": ["subject": "New Jersey", "object": "Orange"]

#OUTPUT#

<Relevance Score>:
10/10
<Explain>:
The response directly addresses the extraction task and provides clearly organized results for both requested categories. The format is clean and easy to understand.

<Fluency>:
Concise and reader-friendly
<Explain>:
The prompt describes the task simply and clearly, using appropriate words

<Accuracy Assessment>:
Incomplete and Correct
<Explain>:
- The relation "location contains" is not defined in the schema
- No false positives information

#INPUT#

<Task Description>:
Extract info from this text. En..Maybe, at the same time, extract the products. Do not extract others' information.
<Constraints>:
NULL
<Schema>: ["city", "product"]
<Input>:
This laptop is lightweight, perfect for work and travel.

<Response>:
"cities": [], "product": ["lightweight", "travel"]

#OUTPUT#

<Relevance Score>:
10/10
<Explain>:
The response does attempt to address the extraction task

<Fluency>:
Not concise
<Explain>:
The requirements in the task description are not concise and clear enough. It is recommended to use a more concise method, such as "Extract the cities and products from this text."

<Accuracy Assessment>:
Incomplete and Partially Incorrect
<Explain>:
- Cities: correct
- Products: travel is not a kind of product
- Multiple errors of both omission and commission

#Input#

<Instruction>: {structures}

B Prompt for Error Correction

You are a prompt correction expert. Please modify the given data according to the provided information <Prompt Structure, Relevance and Accuracy of Answer>

##Requirements##

1. Correct the prompt based on the information given, if the prompt does not need to be modified, please output the original prompt directly.

2. Make sure the modified prompt is correct and meets the purpose of the original prompt.

3. The information provided in the original prompt cannot be modified, but it can be organized in a more user-friendly way.

4. If the original prompt is in JSON format, you need to carefully analyze the JSON content to ensure the fields in JSON are parsed correctly.

5. Do not modify the fields in JSON, but you can optimize the values of these fields.

6. Remember to analyze the original prompt carefully to ensure that the modified prompt does not lose important information.

7. Just output the modified prompt without any explanation.

##Example##
#INPUT#

#Original Prompt
“instruction”:“Extract information based on following schema.
“schema”: [“city”, “product”],
“input”: “This laptop is lightweight, perfect for work and travel.”, “output”: “cities”: [], “product”: [“laptop”, “travel”]

#Prompt Structure

<Task Description>:
Extract information based on given schema
<Constraints>:
1. schema: ["city", "product"]
<Input>: This laptop is lightweight, perfect for work and travel.
<Response>:
{"cities": [], "paoduct": [‘laptop”, “travel”]}

#Relevance and Accuracy of Answer

<Relevance Score>:
10/10
<Explain>:
The response does attempt to address the extraction task

<Fluency>:
Concise and user-friendly
<Explain>:
The requirements in the task description are concise and clear.

<Accuracy Assessment>:
Incomplete and Partially Incorrect

<Explain>:
- Cities: correct, the given text have no cities.
- product: Only laptop need to be extracted.
- Multiple errors of both omission and commission

#OUTPUT#

“instruction”: “Extract information based on following schema.”
“schema”: [“city”, “name”]", “input”: “This laptop is lightweight, perfect for work and travel.”, “output”: “cities”: [], “product”: [“laptop”]

#Input#
Original Prompt prompt

#Original Prompt
{instruction}

#Relevance and Accuracy of Answer#

{relevance and accuracy}

#OUTPUT#