Academic Journal of Applied Sciences
ISSN: 3105-0549 (Print), ISSN: 3105-0557 (Online) | Vol. 3, No. 3, 2026

Event and Relation Extraction for Low-Production and Inefficient Oil Wells Based on LLMs

Xuesong Chen, Xiao Lian *

School of Electrical and Information Engineering, Northeast Petroleum University, Daqing 163318, China

* **Corresponding author**: (Email: 1793399061@qq.com)

Abstract: To address the challenges of terminology comprehension and insufficient seman-tic relationship modeling in knowledge extraction for petroleum engineering us-ing large LLMs, this paper proposes an Instruction-Heterogeneous Tuning-based Event and Relation Extraction framework (IHT-ERE). The framework employs a triple-heterogeneous instruction strategy, which constructs diversified instruc-tions for each sample and dynamically samples them during training. It integrates LoRA for parameter-efficient fine-tuning, enabling effective adaptation to the oil-field domain with minimal parameter adjustments. During inference, self-consistency decoding and rule-based verification are introduced to enhance output stability. Experimental comparisons based on six mainstream open-source models demonstrate that Mistral-7B fine-tuned with IHT-ERE achieves the best perfor-mance, with a ROUGE-L score of 73.11% and a BERTScore F1 of 92.45%, sig-nificantly outperforming other models under the same framework. Ablation stud-ies confirm that the instruction heterogeneity strategy contributes a core perfor-mance improvement of 3.31 percentage points. This study validates the effective-ness of the IHT-ERE framework combined with LoRA fine-tuning for knowledge extraction in petroleum engineering.

Keywords: Event Extraction, Large Language Models, Prompt Learning.

1. Introduction

The increase of low-production and inefficient wells in the mid-to-late stages of oilfield development necessitates accurate identification of their problems, causes, and measures to formulate stabilization strategies. Automatically extracting event knowledge from professional texts and constructing a computable system is crucial for intelligent management.

Early approaches to event and relation extraction relied on rule-based pattern matching. For instance, the FASTUS system [1] employed a cascaded finite-state transducer for domain-specific event extraction but required substantial manual feature engineering and exhibited limited generalization. To reduce template dependency, machine learning methods were introduced. Ahn [2] utilized Support Vector Machines to improve event classification and argument identification, while Ji and Grishman [3] applied a Maximum Entropy model for joint event and argument classification. However, these traditional methods struggled with capturing deep semantic and non-linear relationships [4]. The advent of deep learning brought improvements; Chen et al. [5] used Convolutional Neural Networks for event trigger detection, effectively capturing local context. Pre-trained language models further advanced the field, as shown by Choi et al. [6], who achieved significant gains by fine-tuning BERT for event extraction, though such methods remain heavily reliant on annotated data and lack interpretability [7].

Recently, large language models (LLMs) have propelled generative information extraction, reframing tasks as conditional text generation to better handle structural and semantic complexity [8]. General-domain research has explored various prompt-based methods: KnowPrompt [9] integrates knowledge graphs to optimize relation extraction; UIE [10] is a unified text-to-structure framework for multi-task adaptive modeling; Mirror [11] facilitates cross-task knowledge sharing; on-demand extraction [12] produces tabular outputs based on user instructions; and Instruct-UIE [13] employs instruction fine-tuning to guide LLMs. Instruction fine-tuning enhances model adaptability by explicitly defining tasks, while parameter-efficient techniques like Low-Rank Adaptation (LoRA) [14] enable domain adaptation with minimal trainable parameters by approximating weight updates via low-rank decomposition. However, existing studies focus predominantly on general or standardized texts. Their applicability to oilfield remains unverified, particularly for low-production well analysis, which demands robust domain understanding and structured generation.

This paper proposes a event and relation extraction method. It features a multi-perspective heterogeneous instruction mechanism that constructs complementary instruction sets and dynamically samples them to improve LLM generalization and robustness in professional texts. A LoRA-based parameter-efficient fine-tuning strategy injects low-rank adapters into key Transformer layers for effective domain knowledge transfer with minimal parameters. A reasoning enhancement module integrates self-consistency decoding and domain rule verification, leveraging multi-path sampling and structured validation to boost the accuracy and reliability of extracted events and relations.

2. Methodology Framework

This paper proposes an instruction-heterogeneous tuning-based event and relation extraction framework (IHT-ERE). (see Fig. 1).

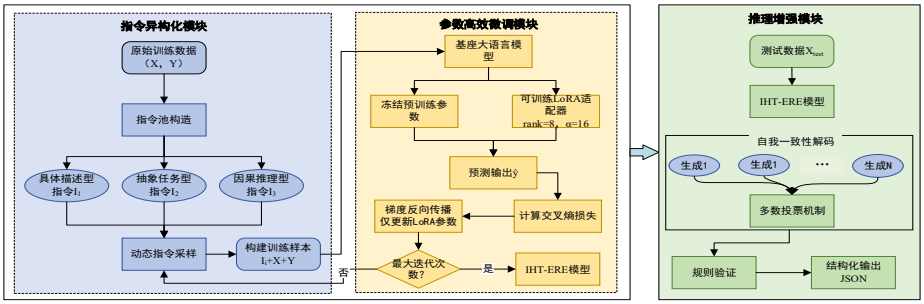


Figure 2. IHT-ERE Framework Diagram.

2.1. Instruction Heterogeneity Module

Although instruction learning enables large models to perform tasks in a zero-shot manner, their performance heavily relies on the precision and robustness of instruction formulation. In complex tasks such as information extraction for low-production and inefficient oil wells, using only a single instruction template can lead the model to overfit to specific expressions. To address this issue, this paper proposes an instruction heterogeneity strategy, constructing a set of three instructions, denoted as $I_i=\{I_1, I_2, I_3\}$. During training, an instruction is dynamically and randomly selected for each sample from this set, extending the original sample space from $\{(X, Y)\}$ to $\{(I_i, X, Y)|i=1, 2, 3\}$.

To ensure alignment between instructions and domain-specific tasks, each instruction follows the paradigm of "Role Definition – Task Description – Constraints – Output Example." Role definition activates the model's domain knowledge; task description is based on predefined event and relation types; constraints embed domain-specific rules; and output examples use structured JSON to clarify field requirements. Based on this paradigm, this paper designs three types of functionally complementary heterogeneous instructions, as detailed in Table 1.

During the training phase, one instruction is randomly and uniformly sampled from the instruction set I_i for each sample in every iteration I_i^j, combining it with the text X to form a training sample (I_i^j, X, Y). This process encourages the model to learn task invariance, thereby significantly enhancing its instruction generalization capability.

Table 1. Overall Evaluation.

Instruction Type	Instruction Prompt
Descriptive Typeitle	Extract all events and relationships related to low-production and inefficient oil wells. Events cover geological features, development dynamics, technical measures, monitoring activities, and engineering effects, with attributes for well type, hierarchical well number, and time. Relationships are limited to inclusion, causality, sequence, condition, and coordination. Output a single standardized JSON object.
Task-Oriented Type	As a reservoir analysis expert, identify the governance framework for low-production wells by sequentially: determining task type, extracting events from five predefined categories, associating events with well data, and inferring logical relationships. Use structured well numbers and time format, outputting standardized JSON with NULL for missing values. Ensure event descriptions match the original text.
Relationship Reasoning Type	Process text [X] via the logic chain first principle: identify the causal chain "problem → measure → effect"; extract event details from five predefined categories with formatted well numbers; mark missing data as NULL. Output causal relationships in the relations field, using only predefined types and original text descriptions.

2.2. Parameter-Efficient Fine-Tuning

Given the vast parameter scale of large language models, full-parameter fine-tuning demands substantial computational resources. Under resource constraints, this study adopts the LoRA method for efficient fine-tuning. Research shows that LoRA achieves performance comparable to full fine-tuning while significantly reducing the number of trainable parameters and GPU memory usage.For a pre-trained weight matrix $\boldsymbol{W}_0 \in \mathbb{R}^{d\times k}$, where d denotes the input feature dimension and k denotes the output feature dimension, its update ΔW is approximated via low-rank decomposition, expressed mathematically as:

$$\boldsymbol{W_0} + \boldsymbol{\Delta W} = \boldsymbol{W_0} + \boldsymbol{BA} \tag{1}$$

Where $A\in\mathbb{R}^{d\times k}$ is an input projection matrix mapping from the low-rank space to the output dimension, and $B\in\mathbb{R}^{d\times k}$ is a low-rank matrix mapping from the input dimension to the low-rank space. The rank $r << \min(d, k)$ is significantly smaller than the minimum of the input and output feature dimensions. During fine-tuning, the original pre-trained parameters $\boldsymbol{W}_0$ remain frozen, and only the low-rank matrices B and A are updated. The forward propagation process can be represented as:

$$h = \boldsymbol{W}_0 x + \boldsymbol{\Delta W} x = \boldsymbol{W_0} x + \boldsymbol{BA} x \tag{2}$$

Where X is the input feature vector and h is the output feature vector after linear layer computation. Here, $\boldsymbol{\Delta W} = \boldsymbol{BA}$ represents the weight update. During initialization, A follows a normal distribution to mitigate gradient vanishing, while B is initialized as a zero matrix to ensure $\boldsymbol{\Delta W} = 0$ at the start of training, thereby preserving pre-trained knowledge. LoRA is typically applied to multi-head self-attention layers and multi-layer perceptron layers. To address the semantic complexity of low-production oil well texts, LoRA adapters are injected into all key linear projection layers of the Transformer.

Given the context composed of instruction Ii and text X, the model's optimization objective is to minimize the cross-entropy loss between the predicted sequence and the ground truth sequence:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T} y_t^{(i)} \log P(\hat{y}_t^{(i)} \left| x^{(i)}; \theta_{LoRA} \right.) \tag{3}$$

Where T is the target sequence length, N is the batch size, $y_t^{(i)}$ denotes the ground truth label for the i-th sample at position t, $\hat{y}_t^{(i)}$ represents the model's prediction for the i-th sample at position t, and $P(\hat{y}_t^{(i)}|x^{(i)};\theta_{LoRA})$ is the probability of predicting $\hat{y}_t^{(i)}$ given input $x^{(i)}$ and LoRA parameters θ_{LoRA}. Here, θ_{LoRA} denotes the parameter set of all LoRA adapters. Through iterative training, the model gradually learns to map heterogeneous instructions to standardized structured outputs, ultimately achieving precise domain-specific information extraction.

2.3. Reasoning Enhancement

To address these challenges, this module incorporates self-consistency decoding and domain rule verification to enhance the quality of extraction results. For each test sample X_{test}, it is combined with a descriptive instruction I_1 to form a complete input prompt. The trained model is then used to perform multiple parallel sampling generations. Let Y denote the i-th candidate output generated by the model, and N=5 independent sampling processes are conducted to obtain a set of candidate outputs. Subsequently, an event-level voting mechanism is applied: events from each candidate output are parsed, and their triple features $Y_1', Y_2', \ldots, Y_5'$, are extracted. The frequency of each triple feature across all candidate outputs is counted. Events with a frequency greater than or equal to 3 are retained to form the final event set. The same strategy is applied to filter relationship triples. This collective decision-making approach effectively identifies the most reliable results.

The output results are further parsed and validated against domain-specific rules. First, the generated string is parsed into a JSON object. If parsing fails, an error is recorded, and an empty result is returned. Regular expressions are used to validate the format of the "well_id" field, with non-compliant values replaced by "NULL". Similarly, the "time" field is validated against the date format, with invalid entries replaced by "NULL". Finally, the "relations" list is traversed to ensure that the "from event" and "to event" values exist in the "event" field of the "events" list; otherwise, such invalid relationships are filtered out.

3. Experimental Design and Analysis

3.1. Experimental Setup and Evaluation Metrics

Due to the lack of specialized event and relation datasets in the low-production and inefficient oil well domain, this study independently constructed a domain-specific instruction dataset. The data sources include 891 production practice documents from typical oilfields provided by a company, along with 235 theoretical documents selected from platforms such as CNKI and Wanfang. After preprocessing, 4,296 sentence-level valid samples were obtained and divided into training and test sets in an 8:2 ratio. The experiments were conducted on an Ubuntu system equipped with NVIDIA vGPUs and Intel Xeon processors, using Python 3.10, PyTorch 2.1.2, and CUDA 11.8. As this study models the extraction task as a text generation task, ROUGE [15] and BERTScore [16] were adopted to evaluate the consistency between generated results and annotated content.

3.2. Experimental Results

Considering data sensitivity and differences in parameters, architecture, and knowledge adaptation among models, this paper selects six representative models—Qwen2.5-7B-Instruct, ChatGLM3-9B, Meta-Llama-3-8B-Instruct, Mistral-7B-Instruct, DeepSeek-LLM-7B-Chat, and Qwen2.5-1.5B-Instruct—for comparative experiments. The results are presented in Table 2.

Table 2. Overall Evaluation %.

Model	R-1	R-2	R-L	BS-F1
Qwen2.5-1.5B-Instruct	66.91	58.38	66.00	90.77
ChatGLM3-9B	38.78	32.59	38.40	73.49
DeepSeek-LLM-7B-Chat	66.02	57.72	65.63	90.19
Qwen2.5-7B-Instruct	71.04	63.02	70.58	91.60
Meta-Llama-3-8B-Instruct	72.39	64.09	71.82	92.34
Mistral-7B-Instruct	73.79	67.33	73.11	92.45

Overall evaluation indicates that Mistral-7B-Instruct achieves optimal and balanced performance across all metrics. Meta-Llama-3-8B-Instruct shows comparable performance but lags by 3.24 percentage points in ROUGE-2, reflecting the limitations of general architectures in specific tasks. Qwen2.5-7B-Instruct delivers balanced results; however, due to insufficient coordination between parsing and generation in its decoder-only architecture, it slightly underperforms compared to Mistral. DeepSeek-LLM-7B-Chat exhibits strong reasoning capabilities but is highly sensitive to prompts, while Qwen2.5-1.5B, constrained by its limited parameter scale, struggles to support complex relationship inference. ChatGLM3-9B performs significantly worse, with frequent structural errors in its outputs, highlighting its notable deficiencies in handling highly specialized and structured industrial information extraction tasks.

Table 3. Results of ablation experiments unit %.

Model	R-1	R-2	R-L	BS-F1
IHT-ERE	73.79	67.33	73.11	92.45
w/o Instruction Heterogeneity Module	70.52	63.84	69.80	90.57
w/o Parameter-Efficient Fine-Tuning	30.14	19.66	29.49	75.34
w/o Reasoning Enhancement	72.23	65.62	72.19	91.64

To validate the effectiveness of each module in the IHT-ERE framework, an ablation study was conducted based on Mistral-7B-Instruct, with the results shown in Table 8. Removing the LoRA-based parameter-efficient fine-tuning module resulted in a significant performance decline, confirming the necessity of domain adaptation. The removal of the dynamic instruction construction module led to decreases of 1.67 percentage points in ROUGE-L and 2.22 percentage points in BERTScore F1, demonstrating that the instruction heterogeneity strategy effectively enhances generalization capability. The removal of the reasoning enhancement module only caused a slight performance drop, indicating that the adequately fine-tuned model already possesses stable generation capabilities.

4. Conclusions and Future Works

The IHT-ERE framework proposed in this paper integrates instruction heterogeneity strategies with parameter-efficient fine-tuning, providing an effective solution for applying

LLMs in the specialized field of oilfield engineering. Experimental results demonstrate that Mistral-7B-Instruct achieves optimal performance on the low-production and inefficient oil well dataset. Compared to ChatGLM3-9B, it improves ROUGE-L by 34.71 percentage points and BERTScore F1 by 18.96 percentage points. It also maintains a 1.29 percentage point advantage over the next best model of similar scale, highlighting the significant impact of model architecture and pre-training quality on performance. Future research will explore automatic instruction generation and multimodal information extraction to build a more comprehensive oilfield knowledge system.

References

[1] Hobbs, J. R., Appelt, D., Bear, J., & Israel, D. (1997). FASTUS: A cascaded finite-state transducer for extracting information from natural-language text. MIT Press.

[2] Ahn, D. (2006). The stages of event extraction. In Proceedings of the Workshop on Annotating and Reasoning about Time and Events (pp. 1–8). Association for Computational Linguistics.

[3] Ji, H., & Grishman, R. (2008). Refining event extraction through cross-document inference. In Proceedings of ACL-08: HLT (pp. 254–262). Association for Computational Linguistics.

[4] Nguyen, T. H., Cho, K., & Grishman, R. (2016). Joint event extraction via recurrent neural networks. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 300–309). Association for Computational Linguistics.

[5] Chen, Y., Xu, L., Liu, K., Zeng, D., & Zhao, J. (2015). Event extraction via dynamic multi-pooling convolutional neural networks. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (pp. 167–176). Association for Computational Linguistics.

[6] Choi, H., Kim, J., Joe, S., & Lee, J. (2021). Evaluation of BERT and ALBERT sentence embedding performance on downstream NLP tasks. In 2020 25th International Conference on Pattern Recognition (ICPR) (pp. 5482–5487). IEEE.

[7] Zhu, M., Mao, Y., Cheng, Y., & Wu, J. (2023). Event extraction method based on dual attention mechanism. Journal of Software, 34(7), 3226–3240. (in Chinese)

[8] Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Zhang, Y., Peng, Y., Li, X., Chen, Y., Zhang, H., & Wen, J. R. (2023, November 24). A survey of large language models. arXiv. https://arxiv.org/pdf/2303.18223.pdf

[9] Chen, X., Zhang, N., Xie, X., Deng, S., Tan, C., Huang, F., & Si, L. (2022). KnowPrompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In Proceedings of the ACM Web Conference 2022 (pp. 2778–2788). Association for Computing Machinery.

[10] Lu, Y., Liu, Q., Dai, D., Xiao, X., Lin, H., Han, X., Sun, L., & Wu, H. (2022). Unified structure generation for universal information extraction. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 5755–5772). Association for Computational Linguistics.

[11] Zhu, T., Ren, J., Yu, Z., Feng, Y., Zhang, L., Wang, Y., & Wang, R. (2023). MIRROR: A universal framework for various information extraction tasks. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (pp. 8861–8876). Association for Computational Linguistics.

[12] Jiao, Y., Zhong, M., Li, S., Zhu, H., Wang, P., Zhang, R., Zhang, T., & Liu, Y. (2023). Instruct and extract: Instruction tuning for on-demand information extraction. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (pp. 10030–10051). Association for Computational Linguistics.

[13] Wang, X., Zhou, W., Zu, C., Xia, R., Chen, J., & Zhang, R. (2023, April 17). InstructUIE: Multi-task instruction tuning for unified information extraction. arXiv. https://arxiv.org/pdf/2304.08085.pdf

[14] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Chen, W., & Chen, W. (2021, October 16). LoRA: Low-rank adaptation of large language models. arXiv. https://arxiv.org/pdf/2106.09685.pdf

[15] Lin, C. Y. (2004). ROUGE: A package for automatic evaluation of summaries. In Proceedings of the Workshop on Text Summarization Branches Out (pp. 74–81). Association for Computational Linguistics.

[16] Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artetxe, M. (2020, February 24). BERTScore: Evaluating text generation with BERT. arXiv. https://arxiv.org/pdf/2002.08835.pdf