GEAR-UIE: Graph-Embedded Augmented Retrieval for Unified Information Extraction

Anonymous ACL submission

Abstract

RAG-based Unified Information Extraction (UIE) typically retrieves exemplars by semantic similarity, which can be syntactically misaligned and hurt in-context learning in complex or cross-domain inputs. To overcome this, we propose GEAR-UIE, a novel framework that introduces syntactic structure as a first-class retrieval signal for UIE. Our approach uniquely constructs a hybrid syntactic graph by integrating dependency and constituency parses, encodes it via Graphormer to obtain structure-aware embeddings, and retrieves semantically and structurally aligned candidates. These are further refined by a lightweight, LLM-preference-aligned reranker. Crucially, the entire pipeline operates without task-specific fine-tuning, enabling a frozen LLM to generate accurate, schema-constrained outputs. Extensive evaluation on 11 Chinese and English benchmarks across three core UIE tasks demonstrates that GEAR-UIE consistently outperforms strong baselines, achieving particularly substantial gains on structure-sensitive extraction tasks. Our code is anonymously available at `https://anonymous.4open.science/r/GEAR-UIE`.

1 Introduction

Information Extraction (IE), a fundamental task in natural language processing, aims to automatically identify and structure key information from unstructured text. Core IE subtasks include Named Entity Recognition (NER), Relation Extraction (RE), and Event Extraction (EE) (Lu et al., 2022; Xu et al., 2024; Li et al., 2023; Xu et al., 2024). The rise of pre-trained and large language models (PLMs/LLMs) has recently established Unified Information Extraction (UIE) as a powerful paradigm (Lou et al., 2023; Fang et al., 2025; Guo et al., 2024b; Wang et al., 2024; Gui et al., 2024; Liao et al., 2025). By reformulating diverse extraction tasks into a unified text-generation framework through prompting or instruction tuning, UIE demonstrates impressive capabilities in cross-task transfer and cross-domain generalization.

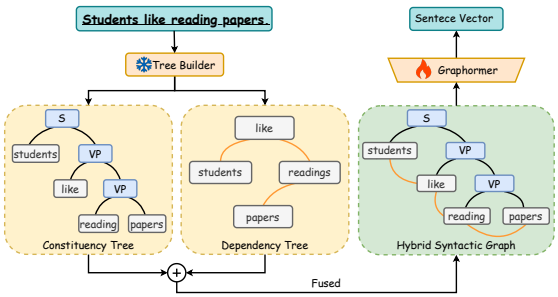

Figure 1: An illustrative example of constructing two types of syntactic trees and a syntactic graph.

Despite these advances, deploying UIE in open and dynamic real-world scenarios remains challenging. **First**, conventional full-parameter fine-tuning methods are heavily dependent on large-scale, high-quality annotated data (Gui et al., 2024; Li et al., 2025b,c), leading to substantial training and maintenance costs. Such approaches also bind models to specific backbone architectures, hindering rapid adaptation to evolving foundation models like the GPT series. **Second**, while Retrieval-Augmented Generation (RAG) offers a promising direction to mitigate few-shot learning and knowledge obsolescence (Liao et al., 2025), current RAG-based IE systems typically retrieve exemplars based solely on semantic similarity, neglecting the explicit syntactic structure within sentences. For structure-sensitive tasks like RE and EE, this often leads to syntactically misaligned examples that weaken in-context learning and may induce hallucinations in the generated outputs.

To overcome these limitations, we introduce GEAR-UIE (Graph-Embedded Augmented Retrieval for Unified Information Extraction), a novel structure-aware RAG framework designed to enhance retrieval quality through explicit syntactic grounding. Our core innovation is the integra-

tion of rich, hybrid syntactic graphs into the retrieval pipeline, ensuring both semantic relevance and structural congruence between retrieved exemplars and the input query. GEAR-UIE follows a three-stage pipeline: (1) structure-aware retrieval, (2) preference-aligned re-ranking, and (3) schema-constrained generation.

In the retrieval stage, we parse an input sentence into a unified Hybrid Syntactic Graph that combines dependency relations, constituency structures, word order, and part-of-speech tags. As shown in Figure 1, for the sentence "*Students like reading papers*", we first derive separate dependency and constituency trees, then merge them into a single graph that captures both word-level dependencies and phrase-level constituents. A dedicated graph encoder (e.g., Graphormer) then encodes this graph into structure-aware embeddings, enabling retrieval of exemplars that are semantically relevant and syntactically isomorphic to the query.

Subsequently, a lightweight, probability-based reranker—trained to align with the target LLM's generative preferences—filters the candidate exemplars. The top-ranked ones, which satisfy both semantic-structural alignment and preference compatibility, are then formatted into a prompt alongside explicit JSON schema constraints. This enables a frozen off-the-shelf LLM to generate accurate and schema-compliant extractions in a zero-shot manner, entirely without task-specific fine-tuning. Our main contributions are as follows:

- **A Novel Structure-Aware RAG Framework for UIE**: We propose GEAR-UIE, the first framework to systematically integrate syntactic graph representations into the retrieval process for IE. This approach directly mitigates the structural mismatch problem inherent in semantic-only RAG, providing a more reliable basis for in-context learning.
- **A Training-Free, Unified Paradigm**: GEAR-UIE supports NER, RE, and EE within a single generative setup without any domain-specific fine-tuning. It demonstrates strong zero-shot and few-shot adaptability across both Chinese and English, offering a practical and scalable deployment paradigm.
- **Empirical Superiority**: Extensive evaluations on 11 Chinese and English benchmarks under zero/few-shot settings show that GEAR-UIE significantly outperforms strong baselines. The marked improvements, particularly on complex RE and EE tasks, validate that our structure-semantic retrieval and constrained generation strategy effectively reduces format errors and hallucinations.

2 Related Work

2.1 Unified Information Extraction

Early IE systems mainly relied on task-specific pipelines or joint modeling frameworks, where EE, NER, and RE were handled separately or loosely coupled, often suffering from error propagation and limited cross-task interaction, which restricted scalability and generalization (Sun et al., 2021).

UIE was subsequently proposed to address these issues by solving multiple IE tasks within a single framework via shared representations or unified output formats (Lu et al., 2022; Vithanage et al., 2025; Rombach and Fettke, 2024; Zhang et al., 2025b). With the advent of LLMs, generative UIE methods have further gained popularity by formulating IE as a sequence-to-sequence problem through prompt engineering or instruction tuning, demonstrating strong zero-shot and cross-domain capabilities (Jin et al., 2025; Srivastava et al., 2025; Rombach and Fettke, 2024; Mishra et al., 2024).

More recently, retrieval-augmented generation (RAG) has been incorporated into UIE to enhance inference via in-context example retrieval (Liao et al., 2025). However, existing RAG-based UIE approaches typically retrieve exemplars based primarily on semantic similarity, implicitly assuming that semantic proximity implies structural alignment (Guo et al., 2024a; Liao et al., 2025). This assumption often fails in structurally complex or cross-domain scenarios, motivating our work to explicitly integrate syntactic signals into the retrieval process for more reliable, structure-aware UIE.

2.2 Syntactic Graph Representations in NLP

Syntactic structure is essential for modeling hierarchical relations and long-range dependencies in natural language. Dependency and constituency parses are widely used to construct structured sentence representations, enabling explicit encoding of grammatical roles and topological relations that are difficult for purely sequential models to capture (Manning et al., 2014). Representing sentences as syntactic graphs has therefore been shown to consistently benefit a wide range of NLP tasks.

In IE, syntactic graph representations are particularly effective for modeling complex relations

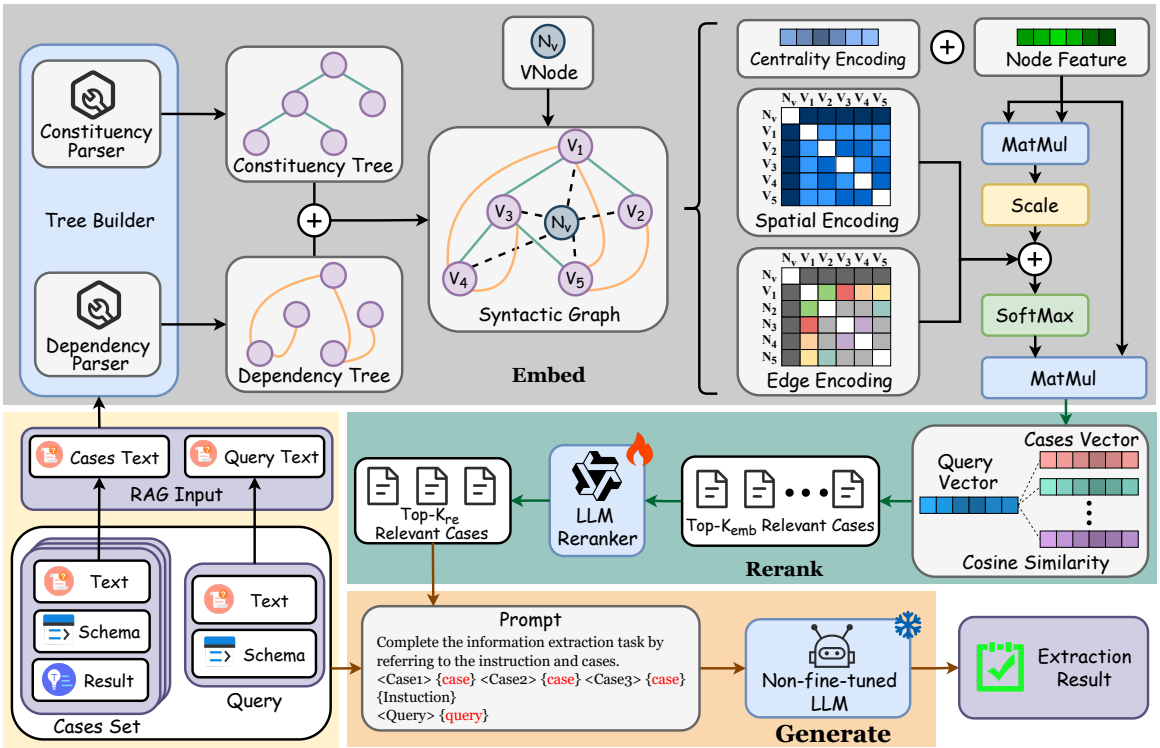

Figure 2: The architecture of GEAR-UIE.

and event argument structures. Graph-based models propagate contextual information along dependency edges to capture higher-order structural dependencies, leading to notable improvements in RE and EE tasks, especially for long-tail and structurally complex instances (Zhang et al., 2018, 2024; He et al., 2018). These findings highlight the complementary role of explicit syntactic bias beyond surface-level semantics.

However, most prior work integrates syntactic structures only into internal encoding or inference modules, while the use of syntactic information to guide example retrieval in retrieval-augmented frameworks remains largely unexplored.

2.3 Graphormer for Structural Encoding

Graphormer is a graph representation learning architecture that integrates Transformer self-attention with explicit structural inductive biases, enabling direct modeling of graph topology while preserving global dependency modeling (Ying et al., 2021). By incorporating structural encodings such as spatial and centrality information into the attention mechanism, Graphormer effectively overcomes the limitations of message-passing GNNs in capturing long-range dependencies and global structure.

Graphormer and its variants have achieved strong performance in graph-centric domains such as molecular property prediction and code representation learning (Dwivedi and Bresson, 2020; Wang et al., 2023). In NLP, recent studies explore Graphormer-style models for structured text encoding by incorporating dependency trees, abstract syntax trees, or heterogeneous linguistic graphs, demonstrating consistent gains over sequential encoders in tasks requiring structural reasoning (Li et al., 2025a; Zhang et al., 2022; Li and Ji, 2019; Jin et al., 2020).

Nevertheless, most retrieval-augmented IE systems still rely on semantic retrieval, overlooking structural misalignment between retrieved examples and the target. Our work employs Graphormer to learn structure-aware representations from hybrid syntactic graphs, enabling retrieval that explicitly accounts for structural compatibility.

3 Methodology

We propose GEAR-UIE, a retrieval-augmented framework for Unified Information Extraction that explicitly incorporates syntactic structure into the example retrieval process. As illustrated in Figure 2, GEAR-UIE operates through three core stages: 1) constructing hybrid syntactic graphs and

learning structure-aware embeddings for retrieval, 2) performing a two-stage retrieval and reranking process to select optimal examples, and 3) generating schema-constrained outputs using a frozen large language model (LLM). By aligning retrieved examples with the target input in terms of structure, semantics, and output schema, GEAR-UIE enables accurate and compliant extractions without task-specific fine-tuning.

3.1 Syntactic Graph-Based Text Embedding

Current semantic retrieval methods often overlook the structural information critical for IE tasks, leading to suboptimal example retrieval. To address this, we design a structure-aware embedding module based on hybrid syntactic graphs, which jointly encodes semantic and structural information (see the Embed module in Figure 2).

3.1.1 Syntactic Graph Construction

Given an input sentence S, we first parse it using Stanza to obtain both its dependency tree and constituency tree. Dependency syntax captures word-level head-dependent relations, while constituency syntax models phrase-level hierarchical structures; the two are highly complementary. We integrate these parses to construct a hybrid syntactic graph $G = (V, E)$, where the node set V includes word nodes and their part-of-speech tags, and the edge set E consists of heterogeneous dependency and constituency relations.

To enhance global information modeling, we introduce a virtual node (VNode) that is bidirectionally connected to all real nodes, acting as a global aggregator. This design allows the model to capture both global dependencies and fine-grained local syntax. Since the VNode's connections do not represent genuine syntactic relations, we assign it a special distance value d_{vnode}, which is larger than the maximum valid syntactic distance in the graph, to prevent the model from misinterpreting these virtual links as real structural paths.

3.1.2 Graphormer-based Embedding

Before inputting the graph to Graphormer, we encode its structural information through centrality, spatial, and edge encodings, which are explicitly injected into the self-attention mechanism to incorporate structural inductive biases.

- **Centrality Encoding.** We first measure node importance via its degree, integrating this into the initial node representation:

$$\mathbf{h}_i^{(0)} = \mathbf{x}_i + \mathbf{z}^+_{\deg^+(v_i)} + \mathbf{z}^-_{\deg^-(v_i)} \quad (1)$$

  where $\mathbf{x}_i$ is the original feature (word and POS embeddings), $\mathbf{z}^+$ and $\mathbf{z}^-$ are learnable vectors associated with the out-degree and in-degree of node v_i, respectively, capturing its structural importance in terms of outgoing and incoming connections in a directed graph.

- **Spatial Encoding.** To model pairwise structural relationships, we compute the shortest path distance $b_\phi(v_i, v_j)$ between any two nodes v_i and v_j, mapping each distance to a learnable embedding used as an attention bias.

- **Edge Encoding.** For node pair (v_i, v_j), we aggregate features along one shortest path between them to obtain bias $c_{i,j}$:

$$c_{i,j} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}_{e_n} (\mathbf{w}_n^E)^\top \quad (2)$$

  where $\mathbf{x}_{e_n}$ is the feature embedding of the n-th edge on the path, and $\mathbf{w}_n^E$ is a learnable weight vector associated with this path, shared across all node pairs.

After obtaining all node features and structural encodings, the graph is then processed by a Graphormer encoder. Within each layer l, the multi-head self-attention score between nodes v_i and v_j integrates semantic and structural information as:

$$A_{i,j} = \frac{(\mathbf{h}_i^{(l)} \mathbf{W}_Q)(\mathbf{h}_j^{(l)} \mathbf{W}_K)^\top}{\sqrt{d}} + b_\phi(v_i, v_j) + c_{i,j} \quad (3)$$

where $\mathbf{W}_Q$ and $\mathbf{W}_K$ are projection matrices, and d is the hidden dimension. The attention weights are normalized via softmax:

$$\alpha_{i,j} = \text{softmax}_j(A_{i,j}) = \frac{e^{A_{i,j}}}{\sum_{k=1}^{n} e^{A_{i,k}}} \quad (4)$$

The output representation for v_i is computed as:

$$\mathbf{o}_i = \sum_{j=1}^{n} \alpha_{i,j} \cdot (\mathbf{h}_j^{(l)} \mathbf{W}_V) \quad (5)$$

We then apply residual connection followed by LayerNorm, a position-wise FFN, and another residual connection with LayerNorm to obtain the updated node representation $\mathbf{h}^{(l+1)}$. After stacking L layers, the final VNode representation $\mathbf{h}_{\text{VNode}}^{(L)}$ is used as the sentence embedding.

To obtain retrieval-friendly embeddings, we train the Graphormer encoder using a contrastive learning objective. For each sentence graph, two augmented views are generated by applying independent dropout masks. Let $\mathbf{N}_{\mathbf{v}_i}$ and $\mathbf{N}_{\mathbf{v}_i}^{+}$ be the ℓ_2-normalized VNode embeddings from the two views of the same sentence. The model is optimized using an InfoNCE loss with in-batch negatives:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{\mathrm{sim}(\mathbf{N}_{\mathbf{v}_i}, \mathbf{N}_{\mathbf{v}_i}^{+})/\tau}}{\sum_{j=1}^{N} e^{\mathrm{sim}(\mathbf{N}_{\mathbf{v}_i}, \mathbf{N}_{\mathbf{v}_j}^{+})/\tau}} \quad (6)$$

where τ is a temperature hyperparameter and $\mathrm{sim}(\cdot,\cdot)$ denotes cosine similarity. This objective encourages embeddings from the same sentence to be close while pushing apart those from different sentences, yielding structure-aware sentence vectors suitable for efficient nearest-neighbor retrieval.

3.2 Two-Stage Retrieval and Rerank

To balance efficiency and accuracy, GEAR-UIE employs a two-stage strategy for example selection (see the Rerank module in Figure 2).

3.2.1 Structured Vector Retrieval

In the first stage, the system performs fast retrieval in vector space using the structure-aware embeddings. Candidates are ranked according to cosine similarity scores, and the top-K_{emb} (e.g., 50) examples are selected as the initial candidate set, efficiently filtering out irrelevant samples.

3.2.2 PAR Training Dataset Construction

To align the reranker's behavior with the preferences of the downstream LLM, we automatically construct a Preference-Aligned Reranker (PAR) training dataset without manual annotation. As shown in Figure 3, for each training instance, an untrained reranker retrieves 50 candidate examples. We then construct 51 prompts: 50 prompts each containing one candidate example, and one prompt with no example. These prompts—along with the input text and schema—are processed by multiple state-of-the-art LLMs (e.g., GPT-5.2, Gemini 3, Claude 4.5) to generate predictions. Using multiple models captures a more comprehensive and robust spectrum of prediction preferences. By comparing the quality of predictions with and without retrieved examples, we automatically construct positive-negative sample pairs for contrastive learning. Further details are provided in Appendix B.

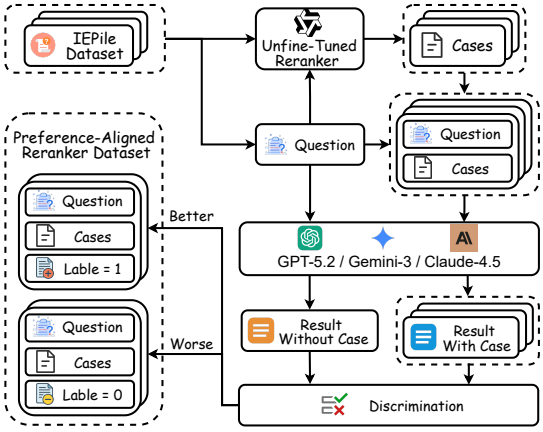

Figure 3: PAR training dataset composition.

3.2.3 LLM-based Reranker Inference

During inference, Qwen3-Reranker-0.6B (Zhang et al., 2025a), trained on PAR, scores each retrieved candidate. Given the query q and a candidate c_i, it outputs a relevance score:

$$S_{\text{rerank}} = P(\text{Relevant} \mid \text{Input}(q, c_i)) \quad (7)$$

This defines the reranker score as the probability that c_i is relevant to q, given $\text{Input}(q, c_i)$. Candidates are ranked by these scores, and the top-K_{re} examples are selected for the final generation stage.

3.3 Information Extraction

The final generation stage constructs a prompt that integrates the task instruction, a predefined output schema, the top-K (e.g.,3) retrieved examples (each with its text and structured output), and the query sentence (see Figure 2, Generate block). This structured prompt enables the frozen LLM to perform in-context learning, inferring transferable extraction patterns and generating outputs that strictly follow the schema in machine-parsable JSON format. By combining structure-aware retrieval with schema-guided generation, GEAR-UIE achieves robust and accurate extraction, particularly in cross-domain and low-resource scenarios, all without task-specific fine-tuning.

4 Experiments

4.1 Datasets & Metrics

We evaluate GEAR-UIE on 11 benchmark datasets spanning three core IE tasks: (1) Event Extraction (EE): CCF Law, FewFC, RAMS, and WikiEvents; (2) Named Entity Recognition (NER): Boson, Weibo, and CrossNER; and (3) Relation Extraction

	EE			NER			RE		
Model	CCF Law	FewFC	Avg	Weibo	Boson	Avg	COAE2016	SKE2020	Avg
GLM-4	39.13	41.81	40.47	20.06	29.79	24.93	26.89	42.09	34.49
Gemma3	38.99	43.24	41.12	21.95	32.43	27.19	25.10	41.72	33.41
Qwen3	43.31	47.01	45.16	23.14	33.59	28.36	26.24	45.02	35.63
Deepseek-V3.2	51.31	56.24	53.78	29.81	40.04	34.92	35.07	50.44	42.76
IEPile-LLama2	59.90	70.10	65.00	34.97	54.45	44.71	46.70	72.18	58.45
ChunkUIE	61.41	79.75	70.58	35.11	59.00	47.06	48.20	70.91	59.56
OneKE	62.19	80.11	71.15	35.06	72.61	53.84	49.83	72.61	61.22
MR-UIE	66.09	83.80	74.95	36.44	54.90	45.67	48.71	72.08	60.40
GEAR-GLM-4	69.34	74.36	71.85	40.60	65.76	53.18	68.85	71.97	70.41
GEAR-Gemma3	68.74	77.32	73.03	46.46	68.73	57.60	67.32	68.12	67.72
GEAR-Qwen3	72.52	78.24	75.47	50.45	69.74	60.10	68.62	80.34	74.48
GEAR-DeepSeek-V3.2	**73.99**	**84.29**	**79.14**	**51.66**	**73.03**	**62.34**	**72.56**	**85.19**	**78.88**

Table 1: Performance comparison in Chinese IE Tasks. Best results are in bold and the second best are underlined.

	EE			NER	RE		
Model	RAMS	WikiEvents	Avg	CrossNER	Wiki-ZSL	FewRel	Avg
GLM-4	12.65	5.98	9.82	37.24	23.90	19.75	21.83
Gemma3	15.53	7.39	11.46	39.26	21.67	21.04	21.36
Qwen3	16.10	8.76	12.43	40.08	26.03	23.87	24.95
DeepSeek-V3.2	19.26	10.98	15.12	46.37	29.31	27.98	28.65
IEPile-LLama2	23.62	13.93	18.78	56.50	36.18	37.14	36.66
ChunkUIE	19.71	8.67	14.19	58.13	32.23	35.76	33.99
OneKE	22.58	12.43	17.51	60.91	42.18	39.19	40.69
RUIE	26.06	**40.64**	33.35	65.41	53.16	49.93	51.55
MR-UIE	24.48	14.20	19.34	62.21	40.47	37.64	39.06
GEAR-GLM-4	29.34	21.04	25.19	62.11	41.59	44.35	42.97
GEAR-Gemma3	36.82	23.61	30.22	63.23	47.15	48.30	47.73
GEAR-Qwen3	35.08	27.43	31.26	65.84	48.24	50.33	49.29
GEAR-DeepSeek-V3.2	**40.78**	32.87	**36.82**	**68.06**	**55.13**	**53.85**	**54.49**

Table 2: Performance comparison in English IE Tasks. Best results are in bold and the second best are underlined.

(RE): COAE2016, SKE2020, and FewRel. Dataset details are provided in Appendix A. We employ the span-based Micro-F1 score as the primary evaluation metric. Based on hyper-parameter analysis, we set the number of in-context examples to 3, which yields the best performance. All reported results follow this configuration.

4.2 Baselines

We compare against a diverse set of baselines: (1) representative open-source LLMs (GLM-4, Gemma-3, Qwen3, DeepSeek-V3.2); (2) Specific fine-tuned UIE models (IEPile-LLama2, OneKE, ChunkUIE); and (3) retrieval-augmented or reasoning-enhanced UIE framework (RUIE, MR-UIE). This selection ensures coverage across model scale, training methodology, and retrieval strategy. All models are evaluated under a consistent few-shot prompting setup with the same metrics.

- **GLM-4-9B** (GLM et al., 2024)[1]: From Zhipu AI, supporting efficient inference.
- **Gemman3.1-12B** (Team et al., 2025)[2]: Google's LLM excels in reasoning, coding, and multilingual tasks with near-large-model performance at medium scale.
- **Qwen3-14B** (Yang et al., 2025)[3]: Alibaba's LLM featuring RL-optimized performance in STEM and coding domains.
- **DeepSeek-V3.2-Exp** (DeepSeek-AI et al., 2025) [4]: Built on DeepSeek-V3.1, using DeepSeek Sparse Attention for efficient long-context processing with strong performance.
- **IEPile-LLama2** (Gui et al., 2024)[5]: A 13B LLM fine-tuned on the IEPile dataset.
- **OneKE** (Luo et al., 2024)[6]: A bilingual IE framework with cross-domain generalization.
- **ChunkUIE** (Li et al., 2025b)[7]: Uses chunked instruction processing and hard negative sampling to tackle bilingual IE ambiguity.

[1]https://github.com/zai-org/GLM-4.
[2]https://deepmind.google/models/gemma.
[3]https://github.com/QwenLM/Qwen3.
[4]https://github.com/deepseek-ai/DeepSeek-V3.2-Exp.
[5]https://github.com/zjunlp/IEPile.
[6]https://github.com/zjunlp/OneKE.
[7]https://github.com/ChunkUIE/chunkuie.

Task	Lang.	w/o Both	w/o E	w/o R	FULL
EE	ZH	45.16	54.08	61.43	**75.47**
	EN	12.43	19.98	24.37	**31.26**
NER	ZH	28.36	42.97	46.45	**60.10**
	EN	40.08	49.29	51.37	**65.84**
RE	ZH	35.63	51.44	62.90	**74.48**
	EN	24.95	28.06	42.05	**49.29**

Table 3: Module ablation study results.

- **RUIE** (Liao et al., 2025)[8]: English-only UIE framework utilizes BM25 sparse retrieval.
- **MR-UIE** (Li et al., 2025c)[9]:UIE framework using multi-perspective reasoning and RL to enhance LLM extraction accuracy.

4.3 Main Results

We conducted two-sided paired t-tests over results from five random seeds for each task. Our best model achieves higher average F1 scores than the strongest baseline, with statistically significant improvements across all tasks ($p < 0.05$).

4.3.1 Chinese Dataset Performance

Table 1 shows the results on Chinese benchmarks. GEAR models consistently outperform all baselines across EE, NER, and RE, with performance gains scaling with the underlying LLM's capability.

For EE, GEAR-DeepSeek-V3.2 achieves an average F1 of 79.14, exceeding the best baseline (MR-UIE) by 4.2 points, showing the benefit of explicit structural modeling for complex event semantics. On NER, GEAR-DeepSeek-V3.2 reaches an average F1 of 62.34, outperforming ChunkUIE by over 15 points, indicating improved entity boundary detection and type discrimination. The largest gains are observed in RE, where GEAR-DeepSeek-V3.2 attains an average F1 of 78.88—nearly 20 points higher than OneKE and MR-UIE—highlighting the effectiveness of structure-aware modeling for capturing complex inter-entity relations.

4.3.2 English Dataset Performance

Table 2 presents results on English benchmarks. GEAR-UIE maintains strong performance, confirming its cross-lingual generalization capability. For EE, GEAR-DeepSeek-V3.2 achieves an F1 score of 36.82, substantially outperforming OneKE (17.51), indicating that our structure-enhanced framework effectively handles long-distance dependencies between event triggers and arguments.

[8]https://github.com/OStars/RUIE.
[9]https://github.com/shiquan-wang/MR-UIE.

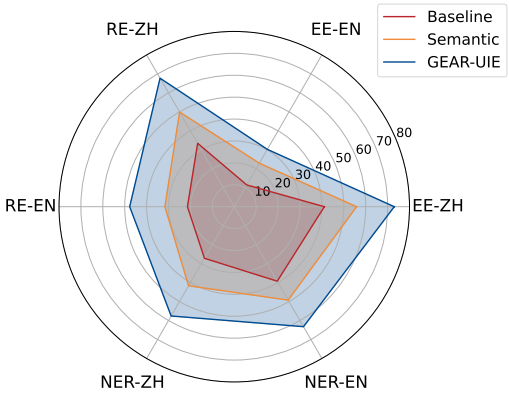

Figure 4: Comparison of semantic vs. structure-semantic retrieval methods.

On RE, GEAR-DeepSeek-V3.2 reaches 54.49 F1, clearly surpassing existing baselines, suggesting that the incorporation of global nodes and structural encodings enhances modeling of cross-sentence and implicit relations.

Overall, these results demonstrate the robustness and effectiveness of GEAR-UIE across bilingual benchmarks, supporting its applicability to UIE.

4.4 Ablation Study

We perform ablation studies on module, semantic-structure, alignment, and hybrid graph; details are provided in later sections and the appendix C.

4.4.1 Ablation Study on Module

Table 3 reports ablation results using Qwen3-14B as the frozen extractor. Removing both retrieval and reranking modules (w/o Both) leads to consistently poor performance, confirming the necessity of retrieval augmentation. Using only structure-aware embeddings (w/o R) yields substantially larger gains than using only the reranker (w/o E), especially on structure-sensitive EE and RE, underscoring the critical role of syntactic structure in retrieval quality. The reranker further refines candidate selection. Combining both modules, GEAR-UIE achieves the best results across all tasks and languages, demonstrating that structure-aware embedding and preference-aligned reranking are complementary and jointly essential.

4.4.2 Semantic vs. Structure-Semantic

Figure 4 compares retrieval strategies with Gemma-3-12B, including: **Baseline** (no retrieval), **Semantic** retrieval (standard Transformer embeddings), and our **GEAR-UIE** with syntactic graph-based embeddings. Results show that structure-semantic

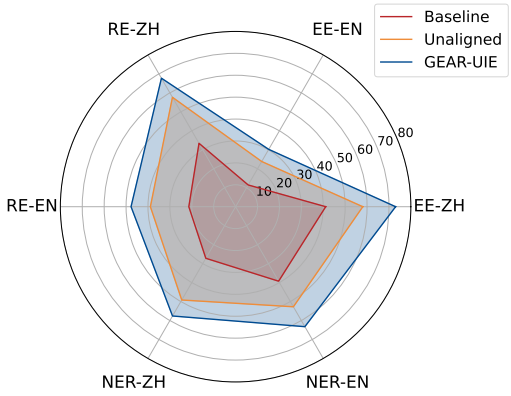

Figure 5: Ablation study of alignment preferences.

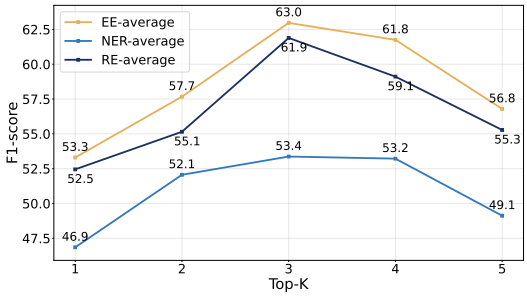

Figure 6: Hyper-parameter analysis.

Task	Time	Performance
Semantic Embed	+3.58%	+21.09%
Structure-Semantic Embed	+7.89%	+54.41%
GEAR-UIE	+11.65%	+71.38%

Table 4: Time cost versus performance gain.

retrieval consistently outperforms semantic-only retrieval, achieving substantially higher F1 scores. Gains are most pronounced on EE and RE tasks, where syntactic structure governs argument roles and relation directions. For example, EE improves by over 17 F1 points in Chinese and 7 points in English, with RE showing similar gains. These results indicate that semantic-only retrieval fails to ensure structural alignment even with strong LLMs. In contrast, GEAR-UIE's structure-semantic retrieval enforces structural compatibility between retrieved examples and target instances, which is crucial for reliable unified information extraction.

4.4.3 Ablation on Alignment Preferences

Figure 5 evaluates the impact of preference alignment using Gemma-3-12B. **Baseline** denotes no retrieval, **Unaligned** uses retrieval without preference-aligned reranking, and **GEAR-UIE** incorporates the preference-aligned reranker. GEAR-UIE achieves an average F1 of 56.59, yielding a gain of 9.87 over Unaligned, with particularly large improvements on EE-ZH, RE-ZH, and NER-EN where format drift and boundary inconsistency are more prevalent. These results indicate that alignment preferences effectively stabilize structured outputs, reduce noise, and improve cross-task and cross-lingual extraction consistency.

4.5 Hyper-parameter Analysis on Top-K

Figure 6 examines the effect of retrieved examples K (1–5). Performance consistently improves up to $K = 3$, with EE showing the largest gain (F1 from 53.3 to 63.0), indicating that structure-sensitive tasks benefit most from additional high-quality context. Beyond $K = 3$, performance declines across tasks, especially for EE, likely due to overly long prompts reducing the model's focus. Consequently, $K = 3$ is adopted as the default setting, providing an optimal balance between contextual richness and prompt conciseness.

4.6 Time Costs and Performance Gains

Table 4 reports the average inference time overhead and performance gains over a direct extraction baseline, evaluated on EE, NER, and RE in both Chinese and English using DeepSeek-V3.2-EXP as the frozen extractor. The semantic-only embedding baseline yields a limited performance improvement of 21.09% with a +3.58% time cost. In contrast, the structure–semantic embedding module achieves a substantially larger performance gain of 54.41% at a moderate additional computational cost (+7.89%), highlighting the critical role of structural information. The full GEAR-UIE framework further boosts performance to 71.38% with only an 11.65% increase in inference time, demonstrating an effective balance between efficiency and extraction accuracy.

5 Conclusion

This paper proposes GEAR-UIE, a structure-aware retrieval-augmented framework that uses syntactic structure as a primary retrieval signal for UIE. Diverging from prior semantic-only RAG methods, GEAR-UIE employs hybrid syntactic graphs and Graphormer embeddings to perform structure-aware example retrieval. Combined with a preference-aligned reranker and structured prompting, it aligns retrieved examples with target inputs structurally, semantically, and schematically, enabling effective UIE with frozen LLMs. Experiments across tasks, languages, and domains confirm its robustness and generality.

6 Limitations

Although GEAR-UIE demonstrates strong performance on unified information extraction tasks, several aspects remain open for further exploration. First, the current framework primarily models syntactic structures at the sentence level and does not explicitly incorporate discourse-level or cross-sentence structural information, which may limit its applicability to long documents or more complex contextual settings. Second, while the structure-aware retrieval and reranking pipeline improves extraction stability, it also introduces additional system complexity; future work may explore more lightweight or end-to-end alternatives. Finally, our experiments focus mainly on Chinese and English unified information extraction tasks, and the generalization of the proposed approach to a broader range of languages and application scenarios warrants further investigation.

References

DeepSeek-AI, Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenhao Xu, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, and 79 others. 2025. Deepseek-v3.2: Pushing the frontier of open large language models. *arXiv preprint*, arXiv:2512.02556.

Vijay Prakash Dwivedi and Xavier Bresson. 2020. A generalization of transformer networks to graphs. *CoRR*, abs/2012.09699.

Yushen Fang, Jianjun Li, Mingqian Ding, Chang Liu, Xinchi Zou, and Wenqi Yang. 2025. Scir: A self-correcting iterative refinement framework for enhanced information extraction based on schema. *arXiv preprint arXiv:2512.12337*.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, and 37 others. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *Preprint*, arXiv:2406.12793.

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, ACL 2024 - Short Papers, Bangkok, Thailand, August 11-16, 2024*, pages 127–146.

Yucan Guo, Zixuan Li, Xiaolong Jin, Yantao Liu, Yutao Zeng, Wenxuan Liu, Xiang Li, Pan Yang, Long Bai, Jiafeng Guo, and Xueqi Cheng. 2024a. Retrieval-augmented code generation for universal information extraction. In *Natural Language Processing and Chinese Computing - 13th National CCF Conference, NLPCC 2024, Hangzhou, China, November 1-3, 2024, Proceedings, Part II*, volume 15360 of *Lecture Notes in Computer Science*, pages 30–42. Springer.

Zixuan Guo, Yan Zhang, Xiaohui Wang, and et al. 2024b. Code4uie: Information extraction with code generation and refinement. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4803–4809.

Zhengqiu He, Wenliang Chen, Zhenghua Li, Meishan Zhang, Wei Zhang, and Min Zhang. 2018. See: Syntax-aware entity embedding for neural relation extraction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Lifeng Jin, Linfeng Song, Yue Zhang, Kun Xu, Wei-Yun Ma, and Dong Yu. 2020. Relation extraction exploiting full dependency forests. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8034–8041. AAAI Press.

Yu Jin, Jie Liu, and Shaowei Chen. 2025. Multi-lora continual learning based instruction tuning framework for universal information extraction. *Knowl. Based Syst.*, 308:112750.

Changjian Li, Yang Song, and Aiping Li. 2025a. GRAG-ZRE: graph retrieval-augmented generation for zero-shot relation extraction in domain-sensitive scenarios. In *Advanced Intelligent Computing Technology and Applications - 21st International Conference, ICIC 2025, Ningbo, China, July 26-29, 2025, Proceedings, Part XX*, volume 15861 of *Lecture Notes in Computer Science*, pages 273–285. Springer.

Diya Li and Heng Ji. 2019. Syntax-aware multi-task graph convolutional networks for biomedical relation extraction. In *Proceedings of the Tenth International Workshop on Health Text Mining and Information Analysis (LOUHI 2019)*, pages 28–33.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2023. A survey on deep learning for named entity recognition : Extended abstract. In *39th IEEE International Conference on Data Engineering, ICDE 2023, Anaheim, CA, USA, April 3-7, 2023*, pages 3817–3818. IEEE.

Wei Li, Yingzhen Liu, Yinling Yang, Ting Zhang, and Wei Men. 2025b. Chunkuie: Chunked instruction-based unified information extraction. *PLOS ONE*, 20(6):e0326764.

Zhongqiu Li, Shiquan Wang, Ruiyu Fang, Mengjiao Bao, Zhenhe Wu, Shuangyong Song, Yongxiang Li, and Zhongjiang He. 2025c. Mr-uie: Multi-perspective reasoning with reinforcement learning for universal information extraction. *arXiv preprint*, arXiv:2509.09082.

Xincheng Liao, Junwen Duan, Yixi Huang, and Jianxin Wang. 2025. RUIE: retrieval-based unified information extraction using large language model. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pages 9640–9655.

Zihan Liu, Yan Xu, Tiezheng Yu, Wenliang Dai, Ziwei Ji, Samuel Cahyawijaya, Andrea Madotto, and Pascale Fung. 2021. Crossner: Evaluating cross-domain named entity recognition. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13452–13460.

Jie Lou, Yaojie Lu, Dai Dai, Wei Jia, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2023. Universal information extraction as unified semantic matching. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 13318–13326. AAAI Press.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 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), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 5755–5772. Association for Computational Linguistics.

Yujie Luo, Xiangyuan Ru, Kangwei Liu, Lin Yuan, Mengshu Sun, Ningyu Zhang, Lei Liang, Zhiqiang Zhang, Jun Zhou, Lanning Wei, Da Zheng, Haofen Wang, and Huajun Chen. 2024. Oneke: A dockerized schema-guided LLM agent-based knowledge extraction system. *CoRR*, abs/2412.20005.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, System Demonstrations*, pages 55–60. The Association for Computer Linguistics.

Lokesh Mishra, Sohayl Dhibi, Yusik Kim, Cesar Berrospi Ramis, Shubham Gupta, Michele Dolfi, and Peter W. J. Staar. 2024. Statements: Universal information extraction from tables with large language models for ESG kpis. *CoRR*, abs/2406.19102.

Alexander Rombach and Peter Fettke. 2024. Deep learning based key information extraction from business documents: Systematic literature review. *CoRR*, abs/2408.06345.

Jiawei Sheng, Shu Guo, Bowen Yu, Qian Li, Yiming Hei, Lihong Wang, Tingwen Liu, and Hongbo Xu. 2021. Casee: A joint learning framework with cascade decoding for overlapping event extraction. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 164–174.

Saurabh Srivastava, Sweta Pati, and Ziyu Yao. 2025. Instruction-tuning llms for event extraction with annotation guidelines. In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 13055–13071. Association for Computational Linguistics.

Kai Sun, Richong Zhang, Samuel Mensah, Yongyi Mao, and Xudong Liu. 2021. Progressive multi-task learning with controlled information flow for joint entity and relation extraction. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13851–13859. AAAI Press.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 188 others. 2025. Gemma 3 technical report. *arXiv preprint*, arXiv:2503.19786.

Dinithi S. Vithanage, Chao Deng, Lei Wang, Mengyang Yin, Mohammad Alkhalaf, Zhenyu Zhang, Yunshu Zhu, and Ping Yu. 2025. Adapting generative large language models for information extraction from unstructured electronic health records in residential aged care: A comparative analysis of training approaches. *J. Heal. Informatics Res.*, 9(2):191–219.

Shuhai Wang, Xin Liu, Xiao Pan, Hanjie Xu, and Mingrui Liu. 2023. Heterogeneous graph transformer for meta-structure learning with application in text classification. *ACM Trans. Web*, 17(3):21:1–21:27.

Yucheng Wang, Bowen Yu, Yilin Liu, and Shudong Lu. 2024. TRUE-UIE: two universal relations unify information extraction tasks. In *Proceedings of the 2024 Conference of the North American Chapter of*

the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024, pages 1863–1876. Association for Computational Linguistics.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, Yang Wang, and Enhong Chen. 2024. Large language models for generative information extraction: a survey. *Frontiers Comput. Sci.*, 18(6):186357.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. 2021. Do transformers really perform badly for graph representation? In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 28877–28888.

Shuaicheng Zhang, Qiang Ning, and Lifu Huang. 2022. Extracting temporal event relation with syntax-guided graph transformer. In *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 379–390. Association for Computational Linguistics.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025a. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

Yini Zhang, Yuxuan Zhang, Zijing Wang, Huanchun Peng, Yongsheng Yang, and Yuanxiang Li. 2024. Multi-information interaction graph neural network for joint entity and relation extraction. *Expert Systems with Applications*, 235:121211.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2205–2215. Association for Computational Linguistics.

Zikang Zhang, Wangjie You, Tianci Wu, Xinrui Wang, Juntao Li, and Min Zhang. 2025b. A survey of generative information extraction. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pages 4840–4870. Association for Computational Linguistics.

A Composition and Structure of Test Data

Table 5 shows the statistics of the test datasets used in our experiments. The introduction to the datasets are as follows:

- **COAE2016**: A Chinese relation extraction dataset constructed from large-scale web texts, focusing on extracting factual relations expressed in diverse and informal online contexts.
- **SKE2020**: A Chinese commercial-domain relation extraction dataset designed to evaluate models' ability to identify structured relations from business and industrial texts.
- **Wiki-ZSL**: An English relation extraction benchmark derived from Wikipedia, emphasizing zero-shot learning scenarios where models must generalize to unseen relations.
- **FewRel** (Han et al., 2018): A widely used English few-shot relation extraction dataset that evaluates a model's capability to learn new relations from a limited number of labeled examples.
- **Boson**: A Chinese named entity recognition dataset in the financial domain, targeting fine-grained entity types commonly appearing in economic and business texts.
- **Weibo**: A Chinese social media named entity recognition dataset built from microblog posts, characterized by informal language and noisy user-generated content.
- **CrossNER** (Liu et al., 2021): An English cross-domain named entity recognition benchmark that assesses model robustness and generalization across multiple unseen domains.
- **CCF Law**: A Chinese legal event extraction dataset focusing on identifying event triggers and arguments from formal law-related documents.
- **FewFC** (Sheng et al., 2021): A Chinese few-shot event extraction dataset in the judicial domain, designed to test event detection and classification under limited supervision.
- **RAMS**: An English news-based event extraction dataset that emphasizes complex event structures with multiple arguments and roles.

Dataset	Lang.	Task	Domain
COAE2016	ZH	RE	Web Text
SKE2020	ZH	RE	Commercial
Wiki-ZSL	EN	RE	Encyclopedia
FewRel	EN	RE	Experiment
Boson	ZH	NER	Financial
Weibo	ZH	NER	Social Media
CrossNER	EN	NER	Experiment
CCF Law	ZH	EE	Legal
FewFC	ZH	EE	Judicial
RAMS	EN	EE	News
WikiEvents	EN	EE	Encyclopedia

Table 5: Dataset details.

```
{
  "instruction":
  {
    "instruction": <Extraction Instruction>,
    "text": <Extraction Text>,
    "schema": <Target Schema>
  }
  "label": <Gold Answer>
}
```

Figure 7: Test Dataset Format.

- **WikiEvents**: An English event extraction dataset constructed from Wikipedia articles, covering a wide range of event types in an encyclopedic setting.

Figure 7 illustrates the structure of the test data, which consists of two main parts: Input and Label. The Input section contains the IE instruction, the text to be processed, and the extraction schema.

B Reranker Dataset and Prompt Examples

As shown in Figure 8, each training instance is a JSON object with two fields: a **question** string and a **candidates** list. The **question** contains the query (i.e., the input sentence with its IE instruction and schema, formatted as a single string). Each element in **candidates** is a candidate example paired with the query, including (i) a **text** field that stores the candidate content in the same unified format, and (ii) a binary relevance label **label** indicating whether the candidate is relevant to the query (`1` for relevant and `0` for irrelevant). During training, the reranker consumes the concatenated query–candidate pair to predict $P(\text{Relevant} \mid \text{Input}(q, c))$. Figure 9 shows the exact prompt templates used to serialize the query and candidates for the reranker.

```
{
    "question": <Question Sentence + Schema>
    "candidates":
    [{
        "text": <Relevant Sentence + Schema>,
        "label": 1
    },
    {
        "text": <Irrelevant Sentence + Schema>,
        "label": 0
    }]
}
```

Figure 8: Train Dataset Format

```
Judge whether the Document meets the requirements
based on the Query and the Instruct provided. Note that
the answer can only be "yes" or "no".
<Instruct>: Given a search query, retrieve passages that
            are most relevant to the query.

<Query>:  Task: {Task Name}
          Schema: {Input Schem}
          Input: {Input Text}

<Document>:  Task: {Task Name}
             Schema: {Cases Schem}
             Input: {Cases Text}
```

Figure 9: Examples of re-ranking prompts.

Task	Lang.	w/o Both	w/o D	w/o C	FULL
EE	ZH	47.67	63.42	67.17	**71.85**
	EN	14.23	19.89	21.05	**25.19**
NER	ZH	33.89	42.73	46.31	**53.18**
	EN	42.41	53.96	58.12	**62.11**
RE	ZH	41.94	57.68	62.55	**70.41**
	EN	23.55	33.23	36.80	**42.97**

Table 6: Ablation Study on the Impact of Different Syntactic Graphs in Retrieval

Task	Dataset	Schema	Size	Label
NER	ACE2004	7	6202	Train
	clue	9	9674	Train
	MSRA	3	14041	Train
	resume_ner	16	3799	Train
	WEIBONER	3	1208	Eval
	boson	6	191	Eval
RE	ADE corpus	1	3417	Train
	CMeIE	52	14339	Train
	DuIE2.0	47	171126	Train
	SKE2020	48	3601	Eval
	COAE2016	8	971	Eval
EE	ACE2005	33	3342	Train
	DuEE-fin	13	7015	Train
	DuEE1.0	64	11908	Train
	FewFC	5	2879	Eval
	ccf_law	9	971	Eval

Table 7: The Chinese datasets employed in this work. "Train" denotes training-only datasets and "Eval" denotes testing-only datasets.

C Impact of Syntactic Graph Types on Retrieval Performance

Table 6 reports the impact of different syntactic graph types on retrieval performance across EE, NER, and RE in both Chinese and English. Removing all syntactic structures (w/o Both) leads to a substantial performance degradation on all tasks, indicating that purely semantic reranking is insufficient to capture fine-grained structural alignment between queries and candidates. Introducing a single syntactic representation yields consistent improvements, yet the gains remain limited. Specifically, dependency-only modeling (w/o C) generally outperforms constituency-only modeling (w/o D), suggesting that head–dependent relations are more effective in enhancing retrieval recall by explicitly encoding predicate–argument interactions and long-range dependencies. However, relying solely on dependency syntax still fails to fully capture span-level and hierarchical phrase information, which is crucial for precise structural matching. In contrast, the full model that integrates both dependency and constituency graphs consistently achieves the best performance across all tasks and languages. This demonstrates that the two syntactic formalisms provide complementary structural signals: dependency syntax facilitates semantic connectivity and recall, while constituency syntax contributes hierarchical and boundary-aware constraints. Their combination enables a more comprehensive structural alignment, leading to more robust and accurate retrieval results than using either syntactic representation alone.

D Basic (IEPile) Dataset Composition

IEPile assembled a diverse collection of bilingual (Chinese–English) IE datasets covering three fundamental tasks to enhance cross-domain robustness and real-world usability. These include NER for identifying and categorizing entity spans, RE for capturing semantic relations between entities, and EE for constructing structured event information. During corpus preparation, we adhered to consistent procedures such as data cleaning, unified formatting, and standardized annotation guidelines, thereby integrating datasets from different sources into a single benchmark framework suitable for fair training and evaluation of models. Details of the training and test datasets are reported in Table 7 and Table 8.

Task	Dataset	Schema	Size	Label
NER	ACE2004	7	6202	Train
	ACE2005	7	7299	Train
	Broad Tweet	3	5334	Train
	CoNLL2003	4	14041	Train
	multiNERD	16	1341444	Train
	Ontonotes	18	59924	Train
	Polyglot-NER	3	393982	Train
	tweetNER7	7	7111	Train
	Wikiann	3	20000	Train
	wikineural	3	92729	Train
	anatEM	1	5861	Train
	Bc2gm	1	12500	Train
	Bc4chemd	1	30682	Train
	Bc5cd	2	4560	Train
	FabNER	12	9435	Train
	FindVehicle	21	21565	Train
	GENIA	5	15023	Train
	HarveyNER	4	3967	Train
	Ncbi-disease	1	5432	Train
	MIT Movie Review	12	2442	Train
	MIT Restaurant Review	8	1520	Train
	CrossNER AI	14	431	Eval
	CrossNER Literature	12	416	Eval
	CrossNER Music	13	465	Eval
	CrossNER Politics	9	650	Eval
	CrossNER Science	17	543	Eval
RE	ADE corpus	1	3417	Train
	CoNLL2004	5	922	Train
	GIDS	4	8526	Train
	Kbp37	18	15917	Train
	NYT	24	56196	Train
	NYT11 HRL	12	62648	Train
	SciERC	7	1366	Train
	SemEval RE	10	6507	Train
	FewRel	83	17291	Eval
	Wiki	100	23113	Eval
EE	ACE2005	33	3342	Train
	CASIE	5	3751	Train
	GENIA	5	15023	Train
	PHEE	2	2898	Train
	CrudeOilNews	18	356	Eval
	RAMS	106	887	Eval
	WikiEvents	31	249	Eval

Table 8: The table lists all English datasets used in our method, where "label" indicates the purpose of the dataset: "train" denotes datasets used solely for training, and "eval" denotes datasets reserved exclusively for testing.