Capability Decomposition for Unified Information Extraction via Hierarchical Mixture-of-Experts

Jing Zhou[1], Peng Wang[1,2]*, Wenjun Ke[1,2], Jiajun Liu[1], Yao He[2]

[1]School of Computer Science and Engineering, Southeast University
[2]Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications (Southeast University), Ministry of Education, China
[3]Institute of Collaborate Innovation, University of Macau

{zhoujing0201, pwang, kewenjun, jiajliu}@seu.edu.cn, mc46477@um.edu.mo

Abstract

Unified Information Extraction (UIE) aims to handle heterogeneous IE tasks within a single framework, but existing methods often suffer from inconsistent schema representation, implicitly intermediate reasoning and full-parameter adaptation, which limit generalization, interpretability and parameter efficiency. To address these issues, we propose UC-UIE (Universal Capabilities-based Unified Information Extractor), a unified framework based on Large Language Model (LLM), which introduces a unified frame-and-slots schema for IE tasks and explicitly decomposes IE reasoning into three universal capabilities: judging, locating, and associating. Furthermore, UC-UIE adopts a Low-Rank Adaptation (LoRA) based hierarchical Mixture-of-Experts (MoE) adapter to fine-tune LLMs for IE tasks, which explicitly models these three capabilities in a task-driven way while ensuring parameter efficiency. With only 1.24% trainable parameters, UC-UIE outperforms full-parameter tuning methods, showing excellent parameter efficiency. Zero-shot evaluation reveals its strong generalization ability to unseen domains and schemas, benefiting from unified schema representation and explicit capability decomposition. Further experiments validate that the hierarchical MoE adapter learns capability specialization and composition, which enhances both UIE performance and interpretability.

1 Introduction

Information Extraction (IE) is a fundamental task in Natural Language Processing (NLP), aiming to extract structured knowledge such as entities, relations and events from unstructured or semi-structured text (Cowie and Lehnert, 1996; Grishman, 2015; Yang et al., 2022; Xu et al., 2024; Zhang et al., 2025b). IE tasks, such as Named Entity Recognition (NER) (Yadav and Bethard, 2018; Hu et al., 2024), Relation Extraction (RE) (Zhao et al., 2024; Nasar et al., 2021), Event Extraction (EE) (Xiang and Wang, 2019; Li et al., 2022) and Aspect-Based Sentiment Analysis (ABSA) (Zhang et al., 2022; Hua et al., 2024), vary widely in their structured extraction objectives (Lu et al., 2022), named schemas (Hogan et al., 2021). As shown in Figure 1, NER extracts *typed entities*, RE identifies *two typed entities together with their relation*, EE constructs event structures consisting of *triggers and multiple argument roles*, and ABSA detects *aspect–opinion–sentiment* tuples. Consequently, heterogeneous IE tasks are traditionally treated as distinct problems with task-tailored models and optimization objectives, leading to redundant designs, limited knowledge sharing and poor cross-task generalization (Paolini et al., 2021; Lu et al., 2022; Xu et al., 2024; Zhang et al., 2025b).

Recently, Unified IE (UIE) emerges to integrate heterogeneous IE tasks within a single framework (Paolini et al., 2021; Lu et al., 2022; Xu et al., 2024; Zhang et al., 2025b). Pretrained Language Models (PLMs), especially Large Language Models (LLMs) make UIE feasible by providing a shared semantic space, reasoning ability and low-resource generalization. However, existing PLM-based UIE methods still face three key limitations. First, despite sharing a unified model architecture, most UIE methods suffer from inconsistent schema and output representations across heterogeneous IE tasks, requiring separate templates or decoders for diverse output formats (Paolini et al., 2021; Lu et al., 2022; Wang et al., 2023), which limits generalization to unseen IE tasks. Second, although some studies incorporate reasoning into the extraction process (Lou et al., 2023; Ping et al., 2023; Zhu et al., 2023), most UIE frameworks formulate IE as a direct sequence generation problem (Lu et al., 2022; Wang et al., 2023; Li et al., 2024b; Liao et al., 2025), implicitly modeling all reasoning behaviors within shared parameters, resulting

*Corresponding author

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 12304–12318
July 2-7, 2026

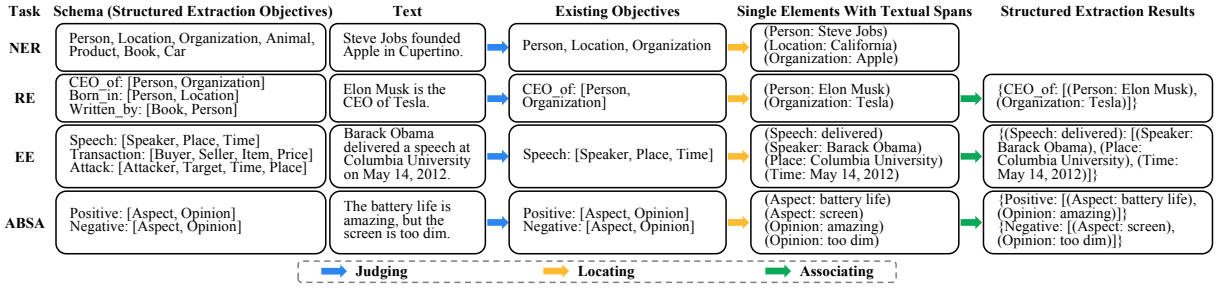


Figure 1: Universal capabilities across IE tasks.

in limited interpretability. Finally, most approaches rely on full-parameter tuning to adapt PLMs to UIE task (Lu et al., 2022; Wang et al., 2023; Lou et al., 2023), which remains feasible for million-parameter PLMs, but impractical when scaling to billion-parameter LLMs due to high training cost.

In response, we propose UC-UIE (Universal Capabilities-based Unified Information Extractor), an LLM-based framework that addresses the above challenges. First, to resolve schema inconsistency, we introduce a unified frame-and-slots representation, grounded in frame semantics (Fillmore, 1976; Petruck, 2022) and slot-filling (Zhang et al., 2017). Each structured extraction objective is abstracted as a semantic frame (e.g., a relation) with typed slots (e.g., relation subject and object), yielding format-consistent outputs and better generalization to unseen schemas. Second, inspired by the frame-based meaning construction (Das et al., 2014; Rai et al., 2025), where linguistic comprehension involves *frame evocation*, *role filling* and *role binding*, we observe that IE tasks share common reasoning behaviors under the unified frame–and-slots schema, despite differing in their structured objectives. We therefore decompose UIE into three task-agnostic universal reasoning capabilities as shown in Figure 1 to overcome the implicit modeling of intermediate reasoning: **judging** the presence of frames (e.g., relations), **locating** their textual spans (e.g., relation subject and object mentions), and **associating** related elements into coherent structures (e.g., linking subject and object via a relation), therefore disentangling intermediate reasoning and enhancing interpretability. Finally, to avoid full-parameter tuning when adapting LLMs to IE tasks, we design a Low-Rank Adaptation (LoRA) (Hu et al., 2022) based hierarchical Mixture-of-Experts (MoE) adapter for Parameter-Efficient Fine-Tuning (PEFT), where experts are specialized for the three capabilities and adaptively activated via a task-driven router.

The main contributions in this paper are summarized as follows:

- A frame-and-slots schema is introduced to represent heterogeneous IE tasks in a consistent way, improving IE schema generalization.
- UIE reasoning is explicitly decomposed into three universal capabilities, including judging, locating and associating, enhancing model interpretability. A LoRA-based hierarchical MoE adapter is designed to model the three capabilities while enabling parameter-efficient adaptation of LLMs to UIE.
- Experiments on 4 IE tasks across 34 IE benchmarks demonstrate the effectiveness of UC-UIE. It outperforms full-parameter tuning methods with only 1.24% trainable parameters, showing excellent parameter efficiency. Zero-shot UC-UIE reaches over half the performance of supervised methods, revealing strong generalization to unseen domains and schemas. Further experiments validate the effectiveness of capability decomposition and hierarchical MoE design.

2 Related work

Unified Information Extraction. Recent advances in UIE can be categorized into two paradigms: linking-based and generation-based methods. Linking-based UIE methods rely on encoder-only PLMs to jointly encode task schema and text into a shared semantic space, and unify heterogeneous IE tasks by structured linking among textual tokens, spans and schema elements, such as USM (Lou et al., 2023), UniEX (Ping et al., 2023), RexUIE (Liu et al., 2023a), Mirror (Zhu et al., 2023) and TRUE-UIE (Wang et al., 2024). However, their schema-aware linking mechanisms are tightly coupled with predefined schemas, and thus difficult to handle unseen schemas. Moreover, relying on full-parameter tuning of encoder-only PLMs limits scalability to larger and more expressive LLMs. Generation-based methods pro-

vide more flexible frameworks through encoder-decoder PLMs or decoder-only LLMs, which formulate IE as a sequence-to-sequence generation task (Sutskever et al., 2014). They differ mainly in output serialization strategies: TANL (Paolini et al., 2021) and ADELIE (Qi et al., 2024) generate natural-language descriptions, UIE (Lu et al., 2022), LasUIE (Fei et al., 2022), InstructUIE (Wang et al., 2023), YAYI-UIE (Xiao et al., 2023)) RUIE (Liao et al., 2025) and LLM-UIE (Zhang et al., 2025a) linearize structured outputs using special symbols, while recent works express extraction results as code-style snippets, such as CodeIE (Li et al., 2023), GoLLIE (Sainz et al., 2024), KnowCoder (Li et al., 2024b), KnowCoder-X (Zuo et al., 2025b)) and GoLLIE-TF (Chen et al., 2025). These methods share PLMs parameters across IE tasks and ignore intermediate reasoning, which limits interpretability, and many of them further rely on full-parameter pretraining.

Mixture-of-Experts. MoE, consisting of multiple independent experts and a router, is proposed to capture different aspects of data (Jacobs et al., 1991; Jordan and Jacobs, 1994) and has proven effective for scaling model capacity with controlled computation (Shazeer et al., 2017; Lepikhin et al., 2021; Dai et al., 2024). Recent works further leverage LoRA-based MoE to achieve LLMs PEFT (Liu et al., 2023b; Dou et al., 2024; Li et al., 2024a; Gao et al., 2024). Beyond scaling and PEFT, MoE has shown strong benefits for multi-task learning, such as multilingual machine translation (NLLB-Team, 2024; Kudugunta et al., 2021), multi-task medical applications (Liu et al., 2023b; Zhu et al., 2024). Tuning to IE, RTE-GMoE (Wulamu et al., 2025) designs a graph-based MoE for Relation Triplet Extraction (RTE), and Tea-MOELoRA (Tang et al., 2025) introduces LoRA experts to learn the cross-task and cross-era (historical and modern Chinese) knowledge. Analogously, we design a hierarchical MoE adapter that explicitly specialize experts for different UIE reasoning capabilities while maintaining parameter efficiency through LoRAs.

3 Methodology

3.1 Problem Formulation

UIE aims to handle multiple IE tasks within a unified framework. We consider 4 representative tasks, namely, NER, RE, EE and ABSA. In this paper, we adopt an LLM as the backbone and formulate UIE as a sequence generation problem. Given a text $X = \{x_1, x_2, \ldots, x_{|X|}\}$ and a task instruction that encodes the task schema $I = \{i_1, i_2, \ldots, i_{|I|}\}$, the model generates a linearized structured sequence $Y = \{y_1, y_2, \ldots, y_{|Y|}\}$ as extraction results. The combined input $Z = [I; X]$ is fed into the model, and training is performed using autoregressive language modeling objective (next token prediction loss) (Du et al., 2022):

$$\mathcal{L}_{task} = -\sum_{(X,Y)\in\mathcal{D}_{train}} \sum_{i}^{|Y|} log P_\Theta(y_i|Z, y_{<i}) \quad (1)$$

where $\mathcal{D}_{train}$ denotes the training dataset and Θ denotes the trainable parameters.

3.2 Unified Schema and Output for IE Tasks

Unified Schema. Extraction objectives in heterogeneous IE tasks (e.g., entities in NER, relation triples in RE and events in EE) can be viewed as semantic information units. Inspired by frame semantics (Fillmore, 1976; Petruck, 2022) and slot-filling (Zhang et al., 2017), we reinterpret each information unit as a **frame**, consisting of a semantic type and a set of typed roles. Formally, a frame is composed of: (1) a **FrameType (FT)** denoting the semantic category (e.g., event type, relation type, sentiment polarity), and its textual realizations **FrameFillers (FFs)**, grounded in the input text; (2) **SlotTypes (STs)** representing roles or attributes within the frame (e.g., relation subjects and objects, event arguments, sentiment aspects and opinions), and their textual realizations **SlotFillers (SFs)**. Accordingly, the unified frame-and-slots schema is represented as a predefined set of frames: $\{FT$: $[ST_1, ST_2, \cdots]\}$. For example, in the EE task shown in Figure 1, the event type *Speech* can be represented in the schema as: {*Speech*: [*Speaker*, *Place*, *Time*]}.

Unified Output. Given the frame-and-slots schema, UC-UIE extracts FFs and SFs to instantiate frames, producing structured information units that are linearized into the unified format: $\{(FT$: $FF)$: $[(ST_1$: $SF_1)$,$(ST_2$: $SF_2)$,$\cdots]\}$. For tasks not needing to specify particular frame elements (e.g., RE does not need FT spans), the corresponding entries are set to *Null*, allowing the representation to flexibly cover different IE tasks. For example, the event frame in Figure 1 is instantiated as {(*Speech*: *delivered*): [(*Speaker*: *Barack Obama*), (*Place*: *Columbia University*), (*Time*: *May 14, 2012*)]}.

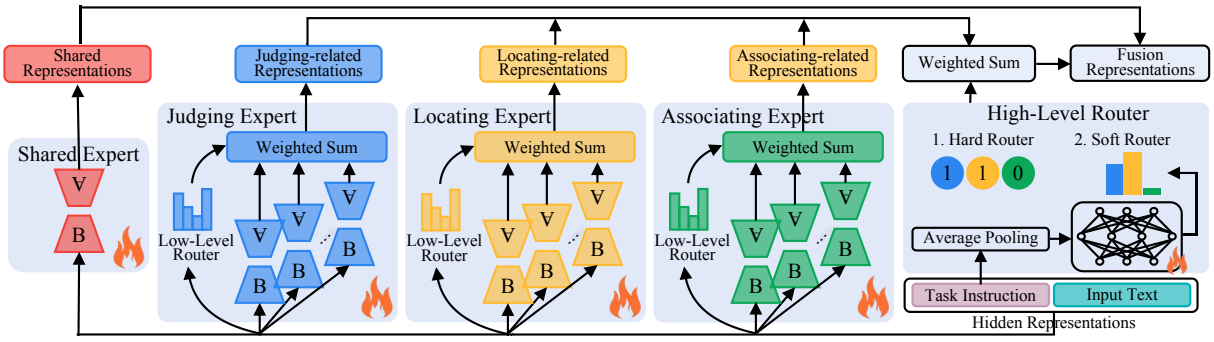


Figure 2: The architecture of hierarchical MoE adapter. The pretrained weights of backbone LLM are frozen. The task instruction and input text of hierarchical MoE adapter refer to the hidden representations.

3.3 Capability Decomposition

Inspired by the frame-based meaning construction (Das et al., 2014; Rai et al., 2025) in linguistics and cognitive semantics (Fillmore, 1976), where comprehension involves: (1) evoking appropriate frames, (2) grounding their roles in text, and (3) binding these roles into coherent semantic structures, we observe that IE tasks share common reasoning behaviors under the unified frame–and-slots schema, despite differing in their structured objectives. We therefore decompose UIE into three task-agnostic universal capabilities: (1) **judging** decides which frames and slot types should be instantiated (e.g., entity categories, relation types, event types and sentiment polarities), (2) **locating** identifies textual spans that fill these frames and slots (e.g., entity mentions, event triggers and arguments, aspects and opinions). (3) **associating** organizes extracted elements into coherent structures by linking them within frames (e.g., subject–object pairing in RE and arguments binding in EE). Judging and locating are commonly required across IE tasks, while associating is additionally needed when relational or compositional structures must be formed.

3.4 Hierarchical MoE Adapter

We design a hierarchical MoE as an adapter to explicitly learn the dedicated three capability while parameter-efficiently fine-tuning the backbone LLM, as illustrated in Figure 2. The adapter is inserted into a selected layer of the backbone whose pretrained weight is denoted as $\boldsymbol{W}_0 \in \mathbb{R}^{d\times k}$ (d is the hidden size and k is the output size) and remains frozen during fine-tuning, while only the adapter parameters are updated. Given a task instruction I and an input text X, their token embeddings are concatenated as $\boldsymbol{Z} = [\boldsymbol{I}; \boldsymbol{X}]$. Feeding $\boldsymbol{Z}$ into the backbone LLM yields hidden representations $\boldsymbol{H} \in \mathbb{R}^{(|I|+|X|)\times d}$ at the target layer, which serve as the input to the hierarchical MoE adapter.

High-level MoE. Hierarchical MoE adapter learns universal capabilities using dedicated experts, E^J (judging), E^L (locating) and E^A (associating), all processing $\boldsymbol{H}$ and producing capability-specific outputs $E^i(\boldsymbol{H}) \in \mathbb{R}^{(|I|+|X|)\times k}$ $(i \in \{J, L, A\})$. To capture task-agnostic knowledge and reduce redundant learning across capability experts (Dai et al., 2024), we additionally include a shared expert E^S implemented as a LoRA (Hu et al., 2022):

$$E^S(\boldsymbol{H}) = \frac{\alpha}{r} \cdot \boldsymbol{H} \cdot \boldsymbol{B}^S \cdot \boldsymbol{A}^S \tag{2}$$

where $\boldsymbol{B}^S \in \mathbb{R}^{d\times r}$, $\boldsymbol{A}^S \in \mathbb{R}^{r\times k}$ are trainable LoRA parameters of E^S, $r \ll min(d, k)$ is the rank, α is a scaling factor.

Router of MoE is responsible for coordinating the expert contributions. In our design, the shared expert is always activated, whereas capability experts selectively activated according to task requirements, motivating a **task-driven routing mechanism**. We implement two alternatives. (1) **Hard routing** heuristically maps each task to required capabilities and produces binary routing labels $[g_J, g_L, g_A] \in \{0, 1\}^3$ (e.g., judging and locating experts are activated in NER, all three capability experts are activated in RE and EE). (2) **Soft routing** learns a trainable router $G(\cdot)$ that predicts expert contributions based on the task instruction. For robustness, we construct task instruction pools based on few manually-crafted instructions by performing offline instruction paraphrasing via LLMs, and randomly sample an instruction I for each instance. The router is formulated as:

$$G(\boldsymbol{h}_{inst}) = \sigma(\boldsymbol{W}_G \cdot \boldsymbol{h}_{inst} + \boldsymbol{b}_G) \tag{3}$$

yielding $[g'_J, g'_L, g'_A] \in \mathbb{R}^3$ as the soft contribution weights. $\boldsymbol{h}_{inst}$ is obtained by average pooling over task instruction tokens. $\sigma(\cdot)$ is the sigmoid function, $\boldsymbol{W}_G \in \mathbb{R}^{3\times d}$, $\boldsymbol{b}_G \in \mathbb{R}^3$ are trainable router parameters. To ensure meaningful capability activation, the soft router is supervised using the hard-routing labels $[g_J, g_L, g_A]$ via a Binary Cross-Entropy (BCE) loss (Goodman et al., 1991):

$$\mathcal{L}_{bce} = -\sum_{(X,Y)\in\mathcal{D}_{train}} \sum_{i\in\{J,L,A\}} [g_i log(g'_i) + (1-g_i) log(1-g'_i)] \quad (4)$$

Low-level MoEs. Each capability expert is further designed as a token-level MoE to capture fine-grained intra-capability patterns. Taking the judging expert E^J as example, it consists of Q LoRA-based sub-experts $\{E^J_q\}_{q=1}^Q$ and a router G^J. Recent studies show that dense activation is particularly effective for LoRA-based MoE fine-tuning (Dou et al., 2024; Liu et al., 2024b; Cai et al., 2025). Consequently, all sub-experts are activated for each token. Compared to sparse activation (Lepikhin et al., 2021; Fedus et al., 2022; Jiang et al., 2024), dense activation prevents load imbalances and mitigates optimization challenges introduced by discrete operations (Dou et al., 2024; Mu and Lin, 2025; Cai et al., 2025).

Each sub-expert E^J_q and the router G^J operates on all token hidden representation $\boldsymbol{h}_t \in \mathbb{R}^d$ in $\boldsymbol{H}$. The sub-expert E^J_q is implemented as a LoRA:

$$E^J_q(\boldsymbol{h}_t) = \frac{\alpha}{r} \cdot \boldsymbol{h}_t \cdot \boldsymbol{B}^J_q \cdot \boldsymbol{A}^J_q \quad (5)$$

where $\boldsymbol{B}^J_q \in \mathbb{R}^{d\times r}$, $\boldsymbol{A}^J_q \in \mathbb{R}^{r\times k}$ are trainable LoRA parameters.

The router G^J is formulated as:

$$G^J(\boldsymbol{h}_t) = softmax(\boldsymbol{W}^J_G \cdot \boldsymbol{h}_t + \boldsymbol{b}^J_G) \quad (6)$$

yielding $[g^J_1, \ldots, g^J_Q] \in \mathbb{R}^Q$ as the sub-expert weights, where $\boldsymbol{W}^J_G \in \mathbb{R}^{Q\times d}$, $\boldsymbol{b}^J_G \in \mathbb{R}^Q$ are trainable router parameters. Then the capability-specific representation produced by E^J is the weighted sum over all sub-experts:

$$E^J(\boldsymbol{h}_t) = \sum_{q=1}^{Q} g^J_q \cdot E^J_q(\boldsymbol{h}_t) \quad (7)$$

Consistent with MOLE (Wu et al., 2024), ModuleFormer (Shen et al., 2023) and Mod-Squad (Chen et al., 2023), we observe low-level routing weights tend to collapse to a few dominant sub-experts during fine-tuning. To encourage balanced expert contributions, we introduce an entropy-based loss. Formally, for a set of tokens in a batch, $\mathcal{X}$, the entropy-based loss is formulated as:

$$\begin{aligned}\mathcal{L}_{entropy} &= -\sum_{i\in\{J,L,A\}} \mathcal{H}(E^i) \\ &= \sum_{i\in\{J,L,A\}} \sum_{q=1}^{Q} p(E^i_q) \log p(E^i_q)\end{aligned} \quad (8)$$

where $\mathcal{H}(E^i)$ is the entropy of distribution of the sub-experts in capability expert E^i. $p(E^i_q) = \sum_t g^i_q p(t)$, $p(t)$ is the probability of token t inside the batch. Following ModuleFormer, we assume that t is uniform over $\mathcal{X}$, therefore $p(t) = 1/|\mathcal{X}|$, $|\mathcal{X}|$ is the number of tokens.

Finally, the hierarchical MoE adapter output is:

$$\Delta\boldsymbol{O} = E^S(\boldsymbol{H}) + \sum_{i\in\{J,L,A\}} g'_i \cdot E^i(\boldsymbol{H}) \quad (9)$$

which is added to the output of fine-tuning layer in the backbone.

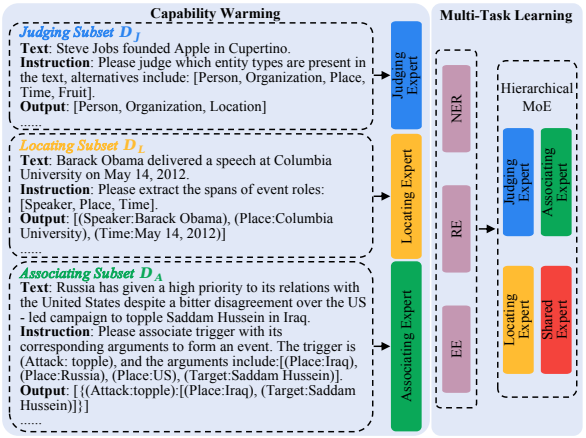


Figure 3: Two-stage fine-tuning of hierarchical MoE.

UC-UIE Optimization. UC-UIE is optimized with the task loss (Eq. (1)) as the primary optimizing objective, and two auxiliary losses, the BCE loss (Eq. (4)) and the entropy-based loss (Eq. (8)), for guiding the routing behaviors. Then the final objective is:

$$\mathcal{L}_{total} = \mathcal{L}_{task} + \gamma\mathcal{L}_{bce} + \beta\mathcal{L}_{entropy} \quad (10)$$

where γ and β control the strengths of the two auxiliary terms.

Training the hierarchical MoE adapter faces challenges in maintaining specialization of high-level experts and cooperation among low-level sub-experts. We adopt a two-stage fine-tuning strategy (Figure 3). In stage 1 (capability warming), three capability-specific subsets $\mathcal{D}_J$, $\mathcal{D}_L$ and $\mathcal{D}_A$ are constructed, and only the corresponding capability expert and the shared expert are activated for each subset. In stage 2 (multi-task learning), the model is fine-tuned on the full multi-task and multi-domain datasets to learn adaptive capability composition.

4 Experiments

4.1 Experimental Setup

Datasets and Metrics. We evaluate performance of UC-UIE across 21 datasets for NER, 6 datasets for RE, 3 datasets for EE and 4 datasets for ABSA. 30 of the total 34 datasets come from IE INSTRUCTIONS (Wang et al., 2023), and we further supplement SemEval 14res (Pontiki et al., 2014), SemEval 14lap (Pontiki et al., 2014), SemEval 15res (Pontiki et al., 2015) and SemEval 16res (Pontiki et al.). Details of the datasets are presented in Appendix A. Following previous methods (Lu et al., 2022; Li et al., 2024b), span-based offset Micro-F1 is used as the evaluation metric. For more reliable and stable evaluation, we run 5 trials for all experiments and report the average results.

Data Preparation. To make UC-UIE more robust to different task instructions, instruction paraphrasing is conducted by DeepSeek-V3 (Liu et al., 2024a) based on few manually-crafted instructions. The detailed prompt and examples of instruction paraphrasing are shown in Figure 4.

Baselines. We compare UC-UIE with 13 baselines under supervised, low-resource and zero-shot settings, covering linking-based methods including USM (Lou et al., 2023), UniEX (larger) (Ping et al., 2023), RexUIE (Liu et al., 2023a), Mirror (Zhu et al., 2023), and TRUE-UIE (Wang et al., 2024), as well as generation-based methods including UIE (larger for supervised and zero-shot settings, base for low-resource setting) (Lu et al., 2022), LasUIE (large) (Fei et al., 2022), InstructUIE (Jiao et al., 2023), YAYI-UIE (Xiao et al., 2023), GoLLIE (7B) (Sainz et al., 2024), KnowCoder (Li et al., 2024b), RUIE (Liao et al., 2025) and KnowCoder-X (Zuo et al., 2025b), and report the results under comparable settings.

Prompt

You are a helpful AI assistant. Please rewrite the following task instruction into another version that is functionally equivalent but uses different wording and structure. Avoid changing the core meaning.

Examples (Input → Paraphrased Output)

NER:
Input: Please list all entity words in the text that fit the types.
Output: Identify every named entity in the sentence and specify its corresponding types.

RE:
Input: Given a sentence, detect the subject and object entities that are connected via a relation from the provided list.
Output: Please identify the subject and object pairs in the sentence that are involved in any of the specified relation types.

EE:
Input: Extract the trigger word and arguments for each event mentioned in the sentence.
Output: For every event described in the sentence, find the trigger and the associated argument roles.

Figure 4: Prompt and examples of instruction paraphrasing.

Implementation Details. UC-UIE adopts LLaMA-2-7B (Touvron et al., 2023) as the backbone LLM following KnowCoder, and fine-tunes the linear layers in the Feed-Forward Network (FFN) of each decoder layer (mlp.gate_proj, mlp.down_proj and mlp.up_proj) with our hierarchical MoE adapter. All experiments are conducted on a machine equipped with 3 NVIDIA RTX 3090Ti GPUs, utilizing the PyTorch framework (Paszke et al., 2019). For all settings, UC-UIE maintains the same configuration. Each capability expert is composed of 4 sub-experts. All LoRAs are used with the rank r of 4 and the scaling factor α of 8. UC-UIE contains approximately 84M trainable parameters, representing merely 1.24% of the entire model parameters. The hyperparameters γ, β for controlling auxiliary constraints strength are both set to 0.1. We train UC-UIE for 3 epochs with a learning rate of 3×10^{-4} and a batch size of 16. Given the total of 112,229 training instances, this results in 21,042 optimization steps.

4.2 Main Results

Supervised Evaluation. Results on 21 datasets across NER, RE, ED and EAE tasks under super-

	Task/Dataset	Linking-based methods					Generation-based methods									Ours	
		USM$^{\diamondsuit}_{\triangle}$	UniEX$^{\diamondsuit}$	RexUIE$^{\diamondsuit}$	Mirror$^{\diamondsuit}$	TRUE-UIE$^{\diamondsuit}_{\triangle}$	UIE$^{\diamondsuit}$	LasUIE$^{\diamondsuit}_{\triangle}$	InstructUIE$_{\triangle}$	YAYI-UIE	GoLLIE$_{\triangle}$	KnowCoder$^{\diamondsuit}_{\triangle}$	RUIE	KnowCoder-X	LLM-UIE	UC-UIE^{h}	UC-UIE
NER	ACE2004	87.34	87.12	87.25	87.16	**89.91**	86.89	86.80	-	-	-	86.20	56.53	-	88.31	88.71	89.64
	ACE2005	-	87.02	87.23	85.34	-	85.78	86.00	86.66	81.78	88.10	86.10	55.86	87.49	87.17	89.45	**89.91**
	Broad Twitter	-	75.21*	-	76.07*	-	80.94*	76.89*	83.14	83.52		-	69.25	82.36	83.59	83.20	**83.77**
	CoNLL2003	92.97	92.65	93.67	92.73	94.13	92.99	93.20	92.94	**96.77**	92.80	95.10	78.34	94.69	93.75	95.49	95.96
	OntoNotes	-	86.03*	-	85.52*	-	88.57*	88.91*	90.19	87.04	-	-	62.94	87.91	-	90.14	**91.34**
	bc5cdr	-	83.92*	-	84.16*	-	87.70*	83.54*	89.59	83.67	87.50	89.30	74.49	88.46	89.57	90.27	**91.11**
	ncbi-disease	-	84.60*	-	85.39*	-	87.05*	85.97*	90.23	87.29	85.40	83.80	58.81	85.49	89.79	90.11	**91.07**
	MultiNERD	-	95.22*	-	91.14*	-	94.89*	93.77*	92.32	88.42	-	**96.10**	88.79	95.94	-	94.76	94.98
	bc2gm	-	81.06*	-	82.58*	-	83.23*	83.02*	85.16	82.05	-	82.00	49.78	84.49	**88.65**	84.87	86.08
	FabNER	-	69.24*	-	68.11*	-	78.47*	67.35*	76.20	72.63	-	82.90	38.51	83.19	82.60	83.41	**85.73**
	WikiANN	-	84.14*	-	85.57*	-	86.05*	85.82*	85.13	72.63	-	**87.00**	68.81	84.69	-	86.52	86.79
	FindVehicle	-	91.56*	-	90.42*	-	96.90*	93.85*	89.47	98.47	-	99.40	92.26	**99.47**	-	98.92	99.18
	GENIA_NER	-	75.34*	-	75.65*	-	76.55*	74.06*	74.71	75.21	-	76.70	59.85	78.97	79.24	77.14	**79.42**
	HarveyNER	-	72.88*	-	73.12*	-	73.42*	82.86*	88.79	69.57	-	-	37.72	73.91	**89.89**	88.35	88.61
RE	ADE corpus	-	-	-	-	-	80.51*	81.04*	82.31	84.14	-	84.30	71.24	84.45	86.51	84.03	**84.57**
	CoNLL2004	77.12	73.40	78.39	75.22	78.94	75.00	75.30	78.48	79.73	-	73.30	54.61	73.14	75.56	79.56	**79.95**
	NYT	-	-	94.55	93.85	94.83	93.05*	94.2	90.47	89.97	-	93.70	72.30	96.08	92.99	95.71	**96.28**
	SemEvalRE	-	-	-	-	-	-	-	73.23	61.02	-	66.30	36.77	64.79	-	73.12	**74.19**
ED	ACE2005	72.31	74.08	75.17	74.44	76.42	73.36	73.86*	77.13	65.00	72.20	74.20	54.41	73.57	77.13	77.42	**77.58**
	CASIE	71.56	71.46	73.01	71.81	**73.02**	69.33	70.20*	67.80	63.00	-	-	40.46	63.91	68.27	71.91	**73.02**
	PHEE	-	-	-	-	-	69.42*	69.04*	70.14	63.00	-	-	47.16	67.03	71.24	70.71	**71.77**
EAE	ACE2005	53.57	53.92	59.15	55.88	56.81	54.79	53.43*	72.94	62.71	66.00	70.30	44.29	69.95	73.64	73.26	**74.14**
	CASIE	63.00	62.91	63.87	61.27	63.90	61.30	60.77*	63.53	64.23	-	-	43.76	64.96	**66.57**	64.61	65.27
	PHEE	-	-	-	-	-	61.23*	60.87*	62.91	77.19	-	-	65.13	76.24	77.85	76.54	**77.87**
	Average	73.98	78.88	79.14	79.77	78.50	78.52	77.73	81.02	77.78	82.00	83.61	59.25	80.92	82.23	83.55	**84.51**

Table 1: Results (%) under supervised setting. $\diamondsuit$ indicates the model pretrained on large-scale extraction-related corpus. $\triangle$ indicates the model employing multi-task learning for IE tasks. * indicates the results obtained from our replication experiments, while USM, RexUIE and TRUE-UIE do not release trained models or reproducible code.

vised setting are shown in Table 1. We observe that: (1) Overall, UC-UIE and its hard-routing variant UC-UIE^{h} achieve leading results on 18/21 datasets, particularly on structurally complex tasks such as RE and EE. UC-UIE obtains the best average F1 and outperforms the second-best baseline by 0.9%. These gains indicate that unified schema, explicit capability decomposition and hierarchical MoE adapter jointly enhance the UIE ability of UC-UIE. (2) Billion-parameter LLM-based methods generally outperform million-parameter PLM-based ones, including linking-based methods, which depend on full-parameter tuning and hinder scalability to LLMs for further IE improvements. Compared with (Q)LoRA-tuned (Hu et al., 2022; Dettmers et al., 2023) GoLLIE, KnowCoder and KnowCoder-X, hierarchical MoE adapter of UC-UIE remains highly parameter-efficient, using only $\approx$1.24% trainable parameters ($\approx 14\times$ standard LoRA) yet achieves superior performance. Although UC-UIE^{h} slightly lags KnowCoder, the latter benefits from full-parameter pretraining on large-scale extraction-related corpus. (3) LLM-based UIE methods often benefit from code-style schema, as seen in GoLLIE, KnowCoder and KnowCoder-X. In contrast, UC-UIE achieves superior results without this, owing to unified frame-and-slots schema and universal capabilities, which together yield more reliable structured IE outputs.

Low-resource Evaluation. To evaluate the generalization ability of UC-UIE, we fine-tune models using only 1%, 5% and 10% of training data on CoNLL2003 (NER), CoNLL2004 (RE) and ACE2005 (ED, EAE). We compare UC-UIE against UIE (base), LasUIE and KnowCoder as these models have reported results under comparable settings. Results in Table 2 show that UC-UIE consistently achieves the best averaged performance across all ratios and particularly excels on complex structured tasks such as RE and EAE. In ratio 1%, it outperforms baselines by 3.8%$\sim$15.2%, demonstrating strong generalization. Unlike UIE, LasUIE and KnowCoder which benefit from large-scale extraction-oriented pretraining with about 220M, 770M and 7B updating parameters respectively, UC-UIE updates only about 84M parameters via the hierarchical MoE adapter, highlighting its parameter efficiency. Furthermore, UC-UIE consistently outperforms UC-UIE^{h}, with a larger margin as data becomes scarcer, indicating that soft routing enable more flexible capability composition, which is beneficial for low-resource learning.

Zero-shot Evaluation. To further investigate the generalization ability of UC-UIE, we conduct zero-shot experiments on 9 datasets for unseen domains (NER, RE), and a dataset for unseen task(ABSA) and report the results in Table 3. Across 7 unseen-domain NER datasets, UC-UIE achieves the best

Ratio	Task	UIE	LasUIE	KnowCoder	UC-UIE^h	UC-UIE
1%	NER	**82.8**	82.1	79.2	80.1	80.7
	RE	30.8	32.0	43.3	42.8	**50.6**
	ED	41.5	30.5*	50.3	50.7	**53.6**
	EAE	12.8	28.2*	38.5	38.8	**43.8**
	Average	42.0	43.4	52.8	53.6	**57.2**
5%	NER	88.3	88.1*	**90.6**	89.7	90.3
	RE	51.7	50.6*	51.1	53.6	**57.0**
	ED	55.7	36.8*	59.0	59.4	**61.9**
	EAE	30.4	35.0*	48.3	51.1	**55.6**
	Average	56.5	52.6	62.3	63.5	**66.2**
10%	NER	89.6	91.6	**92.2**	91.3	91.9
	RE	59.2	60.8	53.6	59.7	**62.5**
	ED	60.3	42.4*	62.2	63.1	**64.6**
	EAE	36.3	40.7*	55.1	56.4	**58.2**
	Average	61.4	58.9	65.8	67.6	**69.3**

Table 2: Results (%) under low-resource setting.

NER							
Method	AI	Literature	Music	Politics	Science	Movie	Restaurant
USM	34.91	65.69	60.07	56.65	55.26	42.11	26.01
Mirror	45.23	46.32	58.61	67.30	54.84	39.20	16.32
InstructUIE	49.00	47.21	53.16	48.15	49.30	63.00	20.99
GoLLIE	58.79*	61.07*	67.54*	56.98*	54.32*	61.77*	43.81*
YAYI-UIE	52.40	45.99	51.20	51.82	50.53	-	-
KnowCoder	60.30	61.10	70.00	**72.20**	59.10	50.00	48.20
LLM-UIE	54.34	49.01	55.79	45.38	55.89	**66.39**	47.9
UC-UIE	**62.41**	**66.82**	**70.11**	71.53	**60.27**	63.81	**49.36**
w/o Para.	62.03	66.15	69.27	70.88	59.73	62.86	48.45

RE			ABSA				
Method	FewRel	Wiki-ZSL	Method	14res	14lap	15res	16res
InstructUIE	39.55	35.20	USM (1-shot)	-	-	-	30.81
			TRUE-UIE (1-shot)	-	-	-	32.03
YAYI-UIE	36.09	**41.07**	UniEX (sp)	**74.77**	**65.23**	**68.58**	**76.02**
			UIE (sp)	74.52	63.88	67.15	75.07
UC-UIE	**39.96**	39.04	UC-UIE	40.66	36.28	37.46	42.43
w/o Para.	39.20	38.49	*w/o* Para.	37.22	34.93	36.08	41.94

Table 3: Results (%) under zero-shot setting on unseen domains (NER, RE) and unseen task (ABSA). *w/o* Para. indicates UC-UIE without instruction paraphrasing. sp indicates the model evaluated under supervised setting.

performance on 5/7 datasets, exceeding LLM-based baselines such as GoLLIE, YAYI-UIE and KnowCoder, showing clear cross-domain generalization. For RE, UC-UIE attains competitive performance compared to InstructUIE and YAYI-UIE. For unseen task ABSA, UC-UIE performs competitively without any task-specific tuning, outperforming the 1-shot USM and TRUE-UIE by 11.62% and 10.4% on 16res and reaching over half the supervised performance of UniEX and UIE.

4.3 Ablation Study

Ablations under supervised setting are summarized in Table 4. Overall, both model structure and optimization strategies contribute notably to UC-UIE.

Model Structure. Replacing soft router with hard router (UC-UIE^h) results in a clear performance drop from 84.03% to 83.30%, showing that

Method	NER	RE	ED	EAE	Average
UC-UIE (*w* Soft Router)	**89.54**	**83.75**	**74.12**	**72.43**	**84.51**
Model Structure					
UC-UIE^h (*w* Hard Router)	88.67	83.11	72.35	71.47	83.55
w/o Shared Expert	88.02	81.47	72.52	66.79	82.34
Backbone+LoRA	77.54	71.06	67.97	60.18	73.09
Backbone+LoRA-Based MoE	81.52	76.93	69.24	62.38	76.83
Model Optimization					
w/o Capability Warming	89.10	82.00	73.30	66.80	83.22
w/o $\mathcal{L}_{entropy}$	87.43	82.20	72.83	68.51	82.37
w/o $\mathcal{L}_{bce}$	87.95	81.04	73.00	68.29	82.47

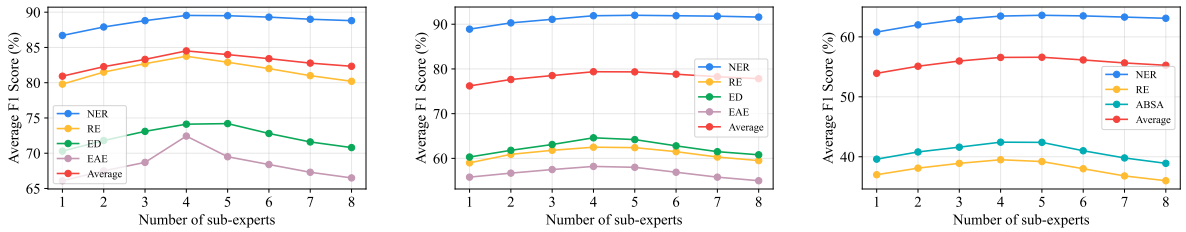
Table 4: Ablation results (%) under supervised setting. *w* and *w/o* indicate the UC-UIE variants with and without corresponding component, respectively.

adaptive capability composition is more beneficial than fixed activation. Removing the shared expert decreases performance to 82.34% (↓2.17%), it is because explicitly modeling task-agnostic knowledge helps avoid redundant learning across capability experts. When replacing the hierarchical MoE with a single LoRA adapter (Backbone+LoRA), performance drops sharply to 73.09% (↓11.42%), and LoRA-based MoE improves but still lags behind (76.83%), demonstrating effectiveness of the the hierarchical MoE adapter.

Model Optimization. Removing the capability warming leads to a moderate decline of 1.29%, validating its effectiveness in capability specialization. Eliminating the auxiliary routing losses also harms performance: removing the entropy-based loss reduces the average F1 by 2.14%, while discarding the BCE-based loss reduces it by 2.04%, showing both effectively regularize routing behavior and strengthen capability modeling. Finally, paraphrasing task instructions consistently improves robustness in zero-shot evaluation (*w/o* Para. in Table 3), benefiting both unseen domains and unseen tasks.

4.4 In-depth Analysis

Analysis on Number of Sub-experts. We study the effect of sub-expert numbers (1∼8) in low-level MoEs under supervised, low-resource (10%) and zero-shot settings, the results are shown in Figure 5. Performance generally rises when increasing sub-experts from 1 to 4, but saturates or slightly declines beyond 4, suggesting that moderate parameter size offers the best intra-capability diversity. The degradation becomes more obvious in low-resource and zero-shot scenarios, where limited supervision is insufficient to train and coordinate too many fine-grained sub-experts, thus harming gen-



(a) Supervised setting (b) Low-resource (10%) setting (c) Zero-shot setting

Figure 5: Effect of number of sub-experts. The red line illustrates the average F1 score (%) over all evaluated tasks.

Task	J	L	A	J+L	J+A	L+A	J+L+A
Judging	**92.38**	74.92	73.34	85.47	83.76	72.85	83.01
Locating	75.88	**91.84**	74.62	83.45	73.85	81.74	80.91
Associating	58.96	60.41	**82.92**	57.01	75.47	73.63	72.92

Table 5: Results (%) on capability-oriented tasks.

Task	J	L	A	J+L	J+A	L+A	J+L+A	*J+L+A*
NER	79.16	78.03	74.62	88.65	77.92	76.47	84.30	**89.43**
RE	67.14	68.28	70.15	73.83	75.94	75.21	82.61	**83.25**
EAE	55.93	56.19	57.84	59.92	62.41	61.86	71.70	**72.43**

Table 6: Results (%) on representative IE tasks.

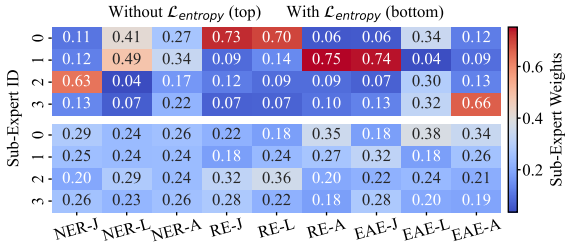


Figure 6: Visualization of average routing weights of low-level sub-experts. Task-capability labels denote the specific capability experts under specific tasks.

eralization. Task-wise analysis further shows that complex structured tasks (RE, ED, EAE, ABSA) benefit more from moderate refinement than span-centric tasks such as NER. Note that this trend is not caused by load imbalance, since all sub-experts are densely activated and explicitly regularized.

Analysis of Capability Specialization and Composition. To verify whether the hierarchical MoE learns capability specialization and composition, we evaluate capability-oriented subtasks (judging, locating, associating) and representative IE tasks (NER, RE, EAE) under different expert activation configurations. As shown in Table 5, each subtask achieves its best performance when only its designated expert is activated, while removing the required expert leads to drops of 17.99%∼25.91%, confirming clear functional specialization rather than interchangeable behaviors.

For IE tasks (Table 6), performance aligns with expected capability demands: on NER, J+L performs best, while performance consistently rises on RE and EAE as more required experts are added, and drops by 15.46% and 15.77% when associating is absent. This demonstrates effective capability composition. Moreover, learned soft routing (*J+L+A*) achieves improvements of 0.64%∼5.13% compared to the binary activation (i.e., hard routing) under the same J+L+A setting, indicating that adaptive activation further refines expert collaboration beyond fixed activation.

Analysis on Entropy-based Loss. We examine the effect of the entropy-based loss $\mathcal{L}_{entropy}$ by visualizing the average routing weights of low-level sub-experts across NER, RE and EE. For each task, we randomly sample 1,000 tokens and compute average routing weights per sub-expert. As shown in Figure 6, removing $\mathcal{L}_{entropy}$ leads to highly skewed routing, where a few sub-experts dominate with some weights exceeding 0.7. While $\mathcal{L}_{entropy}$ yields more balanced weights, promoting better collaboration among sub-experts and enabling more effective fine-grained capability modeling.

5 Conclusion

We proposed UC-UIE, a unified LLM-based framework for heterogeneous IE tasks. It introduces a frame-and-slots schema, explicitly decomposes UIE into judging, locating and associating capabilities, and employs a hierarchical MoE adapter for capability specialization and parameter-efficient tuning. Experiments across supervised, low-resource and zero-shot settings show that UC-UIE achieves competitive or superior performance, with strong generalization to unseen domains and schemas.

Limitations

Although UC-UIE achieves strong performance, this work still leaves several directions unexplored. First, UC-UIE is evaluated on 4 representative IE tasks while extending it to a broader spectrum of IE scenarios (e.g., document-level IE (Xue et al., 2024; Zuo et al., 2025a), multilingual IE (Xiao et al., 2023; Zuo et al., 2025b; Chen et al., 2025)) or even other NLP tasks (e.g., text classification) remains an interesting direction. Second, UC-UIE adopts a frame-and-slots schema, which is well-suited for schema-predefined tasks, applying it to schema-free IE such as open IE (Zhou et al., 2022) and on-demand IE (Jiao et al., 2023) requires further investigation. Third, while UC-UIE is already parameter-efficient, exploring more lightweight capability modeling (e.g., smaller adapters, distilled capability experts) may further reduce computational overhead and broaden applicability in resource-constrained scenarios.

Acknowledgments

This work was supported by National Science Foundation of China (Grant Nos.62376057).

References

Weilin Cai, Juyong Jiang, Fan Wang, Jing Tang, Sunghun Kim, and Jiayi Huang. 2025. A survey on mixture of experts in large language models. *IEEE Trans Knowl Data Eng*, 37(07):3896–3915.

Chih-Yao Chen and Cheng-Te Li. 2021. Zs-bert: Towards zero-shot relation extraction with attribute representation learning. In *the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 3470–3479.

Pei Chen, Haotian Xu, Cheng Zhang, and Ruihong Huang. 2022. Crossroads, buildings and neighborhoods: A dataset for fine-grained location recognition. In *the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 3329–3339.

Yang Chen, Vedaant Shah, and Alan Ritter. 2025. Translation and fusion improves cross-lingual information extraction. In *the Annual Meeting of the Association for Computational Linguistics*, pages 7744–7764.

Zitian Chen, Yikang Shen, Mingyu Ding, Zhenfang Chen, Hengshuang Zhao, Erik G Learned-Miller, and Chuang Gan. 2023. Mod-squad: Designing mixtures of experts as modular multi-task learners. In *the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11828–11837.

Jim Cowie and Wendy Lehnert. 1996. Information extraction. *Commun ACM*, 39(1):80–91.

Damai Dai, Chengqi Deng, Chenggang Zhao, R. X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y. K. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. 2024. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *ArXiv:2401.06066*.

Dipanjan Das, Desai Chen, André FT Martins, Nathan Schneider, and Noah A Smith. 2014. Frame-semantic parsing. *Computational Linguistics*, 40(1):9–56.

Leon Derczynski, Kalina Bontcheva, and Ian Roberts. 2016. Broad twitter corpus: A diverse named entity recognition resource. In *the International Conference on Computational Linguistics*, pages 1169–1179.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36:10088–10115.

Rezarta Islamaj Dogan, Robert Leaman, and Zhiyong Lu. 2014. Ncbi disease corpus: a resource for disease name recognition and concept normalization. *J Biomed Inform*, 47:1–10.

Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Wei Shen, Limao Xiong, Yuhao Zhou, Xiao Wang, Zhiheng Xi, Xiaoran Fan, Shiliang Pu, Jiang Zhu, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024. Loramoe: Alleviating world knowledge forgetting in large language models via moe-style plugin. In *the Annual Meeting of the Association for Computational Linguistics*, pages 1932–1945.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. Glm: General language model pretraining with autoregressive blank infilling. In *the Annual Meeting of the Association for Computational Linguistics*, pages 320–335.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *J Mach Learn Res*, 23(120):1–39.

Hao Fei, Shengqiong Wu, Jingye Li, Bobo Li, Fei Li, Libo Qin, Meishan Zhang, Min Zhang, and Tat-Seng Chua. 2022. Lasuie: Unifying information extraction with latent adaptive structure-aware generative language model. In *Advances in Neural Information Processing Systems*, volume 35, pages 15460–15475.

Charles J Fillmore. 1976. Frame semantics and the nature of language. *Ann N Y Acad Sci*, 280(1):20–32.

Chongyang Gao, Kezhen Chen, Jinmeng Rao, Baochen Sun, Ruibo Liu, Daiyi Peng, Yawen Zhang, Xiaoyuan Guo, Jie Yang, and VS Subrahmanian. 2024. Higher layers need more lora experts. *ArXiv:2402.08562*.

R Goodman, JW Miller, and P Smyth. 1991. Objective functions for neural network classifier design. In *IEEE International Symposium on Information Theory*, pages 87–87. IEEE.

Ralph Grishman. 2015. Information extraction. *IEEE Intell Syst*, 30(5):8–15.

Runwei Guan. 2022. Findvehicle and vehiclefinder: A ner dataset for a text-image cross-modal vehicle retrieval system.

Harsha Gurulingappa, Abdul Mateen Rajput, Angus Roberts, Juliane Fluck, Martin Hofmann-Apitius, and Luca Toldo. 2012. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *J Biomed Inform*, 45(5):885–892.

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 *The Conference on Empirical Methods in Natural Language Processing*.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Sabbir M. Rashid Axel Polleres, Anisa Rula, Lukas Schmelzeisen, Juan F. Sequeda, Steffen Staab, and Antoine Zimmermann. 2021. Knowledge graphs. *ACM Comput Surv*, 54(4):1–37.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *the International Conference on Learning Representations*.

Zhentao Hu, Wei Hou, and Xianxing Liu. 2024. Deep learning for named entity recognition: a survey. *Neural Comput Appl*, 36(16):8995–9022.

Yan Cathy Hua, Paul Denny, Jörg Wicker, and Katerina Taskova. 2024. A systematic review of aspect-based sentiment analysis: domains, methods, and trends. *Artif Intell Rev*, 57(11):296.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. Adaptive mixtures of local experts. *Neural Comput*, 3(1):79–87.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, and 7 others. 2024. Mixtral of experts.

Yizhu Jiao, Ming Zhong, Sha Li, Ruining Zhao, Siru Ouyang, Heng Ji, and Jiawei Han. 2023. Instruct and extract: Instruction tuning for on-demand information extraction. In *the Conference on Empirical Methods in Natural Language Processing*, pages 10030–10051.

Michael I Jordan and Robert A Jacobs. 1994. Hierarchical mixtures of experts and the em algorithm. *Neural Comput*, 6(2):181–214.

J-D Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2003. Genia corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl_1):i180–i182.

Veysel Kocaman and David Talby. 2021. Biomedical named entity recognition at scale. In *the International Conference on Pattern Recognition*, pages 635–646. Springer.

Sneha Kudugunta, Yanping Huang, Ankur Bapna, Maxim Krikun, Dmitry Lepikhin, Minh-Thang Luong, and Orhan Firat. 2021. Beyond distillation: Task-level mixture-of-experts for efficient inference. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 3577–3599.

Aman Kumar and Binil Starly. 2022. "fabner": information extraction from manufacturing process science domain literature using named entity recognition. *J Intell Manuf*, 33(8):2393–2407.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2021. Gshard: Scaling giant models with conditional computation and automatic sharding. In *the International Conference on Learning Representations*.

Dengchun Li, Yingzi Ma, Naizheng Wang, Zhengmao Ye, Zhiyuan Cheng, Yinghao Tang, Yan Zhang, Lei Duan, Jie Zuo, Cal Yang, and Mingjie Tang. 2024a. Mixlora: Enhancing large language models fine-tuning with lora-based mixture of experts. *ArXiv:2404.15159*.

Jiao Li, Yueping Sun, Robin J Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wiegers, and Zhiyong Lu. 2016. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database*, 2016.

Peng Li, Tianxiang Sun, Qiong Tang, Hang Yan, Yuanbin Wu, Xuan-Jing Huang, and Xipeng Qiu. 2023. Codeie: Large code generation models are better few-shot information extractors. In *the Annual Meeting of the Association for Computational Linguistics*, pages 15339–15353.

Qian Li, Jianxin Li, Jiawei Sheng, Shiyao Cui, Jia Wu, Yiming Hei, Hao Peng, Shu Guo, Lihong Wang, Amin Beheshti, and Philip S. Yu. 2022. A survey on deep learning event extraction: Approaches and applications. *IEEE Trans Neural Netw Learn Syst*, 35(5):6301–6321.

Zixuan Li, Yutao Zeng, Yuxin Zuo, Weicheng Ren, Wenxuan Liu, Miao Su, Yucan Guo, Yantao Liu, Xiang Li, Zhilei Hu, Long Bai, Wei Li, Yidan Liu, Pan Yang, Xiaolong Jin, Jiafeng Guo, and Xueqi Cheng. 2024b. Knowcoder: Coding structured knowledge into llms for universal information extraction. In *the Annual Meeting of the Association for Computational Linguistics*, pages 8758–8779.

Xincheng Liao, Junwen Duan, Yixi Huang, and Jianxin Wang. 2025. Ruie: Retrieval-based unified information extraction using large language model. In *the International Conference on Computational Linguistics*, pages 9640–9655.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024a. Deepseek-v3 technical report. *ArXiv:2412.19437*.

Chengyuan Liu, Fubang Zhao, Yangyang Kang, Jingyuan Zhang, Xiang Zhou, Changlong Sun, Kun Kuang, and Fei Wu. 2023a. Rexuie: A recursive method with explicit schema instructor for universal information extraction. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 15342–15359.

Qidong Liu, Xian Wu, Xiangyu Zhao, Yuanshao Zhu, Derong Xu, Feng Tian, and Yefeng Zheng. 2023b. Moelora: An moe-based parameter efficient fine-tuning method for multi-task medical applications. *CoRR*.

Qidong Liu, Xian Wu, Xiangyu Zhao, Yuanshao Zhu, Derong Xu, Feng Tian, and Yefeng Zheng. 2024b. When moe meets llms: Parameter efficient fine-tuning for multi-task medical applications. In *SIGIR*.

Yijin Liu, Fandong Meng, Jinchao Zhang, Jinan Xu, Yufeng Chen, and Jie Zhou. 2019. Gcdt: A global context enhanced deep transition architecture for sequence labeling. In *the Annual Meeting of the Association for Computational Linguistics*, pages 2431–2441.

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 *the Association for the Advancement of Artificial Intelligence*.

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 *the Association for the Advancement of Artificial Intelligence*, volume 37, pages 13318–13326.

Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021. Text2event: Controllable sequence-to-structure generation for end-to-end event extraction. In *the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, pages 2795–2806.

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 *the Annual Meeting of the Association for Computational Linguistics*, pages 5755–5772.

Alexis Mitchell, Stephanie Strassel, Shudong Huang, and Ramez Zakhary. 2005. Ace 2004 multilingual training corpus. *Linguistic Data Consortuim*, 1:1–1.

Siyuan Mu and Sen Lin. 2025. A comprehensive survey of mixture-of-experts: Algorithms, theory, and applications. *ArXiv:2503.07137*.

Zara Nasar, Syed Waqar Jaffry, and Muhammad Kamran Malik. 2021. Named entity recognition and relation extraction: State-of-the-art. *ACM Comput Surv*, 54(1):1–39.

NLLB-Team. 2024. Scaling neural machine translation to 200 languages. *Nature*, 630(8018):841–846.

Eduard Hovy Mitchell Marcus Martha Palmer and Lance Ramshaw Ralph Weischedel. 2006. Ontonotes: The 90% solution. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 57–60.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *the Annual Meeting of the Association for Computational Linguistics*, pages 1946–1958.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *the International Conference on Learning Representations*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and 1 others. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32.

Miriam RL Petruck. 2022. *Frame semantics*. John Benjamins Publishing Company.

Yang Ping, JunYu Lu, Ruyi Gan, Junjie Wang, Yuxiang Zhang, Pingjian Zhang, and Jiaxing Zhang. 2023. Uniex: An effective and efficient framework for unified information extraction via a span-extractive perspective. In *the Annual Meeting of the Association for Computational Linguistics*, pages 16424–16440.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Nuria Bel,

Salud María Jiménez-Zafra, and Gülşen Eryiğit. Semeval-2016 task 5: Aspect based sentiment analysis.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. Semeval-2015 task 12: Aspect based sentiment analysis. In *the International Workshop on Semantic Evaluation*, page 486–495.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *the International Workshop on Semantic Evaluation*, page 27–35.

Yunjia Qi, Hao Peng, Xiaozhi Wang, Bin Xu, Lei Hou, and Juanzi Li. 2024. Adelie: Aligning large language models on information extraction. In *the Conference on Empirical Methods in Natural Language Processing*, pages 7371–7387.

Shahid Iqbal Rai, Danilo Croce, and Roberto Basili. 2025. Injecting frame semantics into large language models via prompt-based fine-tuning. In *the Joint Conference on Lexical and Computational Semantics*, pages 31–47.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 148–163. Springer.

Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *the Conference on Computational Natural Language Learning at HLT-NAACL*, pages 1–8.

Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. 2024. Gollie: Annotation guidelines improve zero-shot information-extraction. In *the International Conference on Learning Representations*.

Erik Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *the conference on Natural language learning at HLT-NAACL*, pages 142–147.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *the International Conference on Learning Representations*.

Yikang Shen, Zheyu Zhang, Tianyou Cao, Shawn Tan, Zhenfang Chen, and Chuang Gan. 2023. Moduleformer: Modularity emerges from mixture-of-experts. *ArXiv:2306.04640*.

Zhaoyue Sun, Jiazheng Li, Gabriele Pergola, Byron C Wallace, Bino John, Nigel Greene, Joseph Kim, and Yulan He. 2022. Phee: A dataset for pharmacovigilance event extraction from text. In *the Conference on Empirical Methods in Natural Language Processing*, pages 5571–5587.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*.

Xuemei Tang, Chengxi Yan, Jinghang Gu, and Chu-Ren Huang. 2025. Joint information extraction across classical and modern chinese with tea-moelora. *ArXiv:2509.01158*.

Simone Tedeschi and Roberto Navigli. 2022. Multinerd: A multilingual, multi-genre and fine-grained dataset for named entity recognition (and disambiguation). In *Findings of the Association for Computational Linguistics: NAACL*, pages 801–812.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv:2307.09288*.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. *(No Title)*.

Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, Jihua Kang, Jingsheng Yang, Siyuan Li, and Chunsai Du. 2023. Instructuie: Multi-task instruction tuning for unified information extraction. *ArXiv:2304.08085*.

Yucheng Wang, Bowen Yu, Yilin Liu, and Shudong Lu. 2024. True-uie: Two universal relations unify information extraction tasks. In *the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 1863–1876.

Xun Wu, Shaohan Huang, and Furu Wei. 2024. Mixture of lora experts. In *the International Conference on Learning Representations*.

Aziguli Wulamu, Kaiyuan Gong, Lyu Zhengyu, Yu Han, Zhihong Zhu, and Bowen Xing. 2025. Rte-gmoe: A model-agnostic approach for relation triplet extraction via graph-based mixture-of-expert mutual learning. In *the Conference on Empirical Methods in Natural Language Processing*, pages 7488–7499.

Wei Xiang and Bang Wang. 2019. A survey of event extraction from text. *IEEE Access*, 7:173111–173137.

Xinglin Xiao, Yijie Wang, Nan Xu, Yuqi Wang, Hanxuan Yang, Minzheng Wang, Yin Luo, Lei Wang, Wenji Mao, and Daniel Zeng. 2023. Yayi-uie: A

chat-enhanced instruction tuning framework for universal information extraction. *ArXiv:2312.15548*.

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. *Front Comput Sci*, 18(6):186357.

Lilong Xue, Dan Zhang, Yuxiao Dong, and Jie Tang. 2024. Autore: Document-level relation extraction with large language models. In *the Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 211–220.

Vikas Yadav and Steven Bethard. 2018. A survey on recent advances in named entity recognition from deep learning models. In *the International Conference on Learning Representations*, pages 2145–2158.

Yang Yang, Zhilei Wu, Yuexiang Yang, Shuangshuang Lian, Fengjie Guo, and Zhiwei Wang. 2022. A survey of information extraction based on deep learning. *Appl Sci*, 12(19):9691.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2022. A survey on aspect-based sentiment analysis: Tasks, methods, and challenges. *IEEE Trans Knowl Data Eng*, 35(11):11019–11038.

Xieyun Zhang, Shimin Cai, Xiaorong Shen, Han Yang, Wenhao Hu, and Yanru Zhang. 2025a. Efficient unified information extraction model based on large language models. *Appl Soft Comput*, page 113302.

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

Zikang Zhang, Wangjie You, Tianci Wu, Xinrui Wang, Juntao Li, and Min Zhang. 2025b. A survey of generative information extraction. In *the International Conference on Computational Linguistics*, pages 4840–4870.

Xiaoyan Zhao, Yang Deng, Min Yang, Lingzhi Wang, Rui Zhang, Hong Cheng, Wai Lam, Ying Shen, and Ruifeng Xu. 2024. A comprehensive survey on relation extraction: Recent advances and new frontiers. *ACM Comput Surv*, 56(11):1–39.

Shaowen Zhou, Bowen Yu, Aixin Sun, Cheng Long, Jingyang Li, Haiyang Yu, Jian Sun, and Yongbin Li. 2022. A survey on neural open information extraction: Current status and future directions. In *the International Joint Conference on Artificial Intelligence*, pages 5694–5701.

Tong Zhu, Junfei Ren, Zijian Yu, Mengsong Wu, Guoliang Zhang, Xiaoye Qu, Wenliang Chen, Zhefeng Wang, Baoxing Huai, and Min Zhang. 2023. Mirror: A universal framework for various information extraction tasks. In *the Conference on Empirical Methods in Natural Language Processing*, pages 8861–8876.

Xun Zhu, Ying Hu, Fanbin Mo, Miao Li, and Ji Wu. 2024. Uni-med: a unified medical generalist foundation model for multi-task learning via connector-moe. *Advances in Neural Information Processing Systems*, 37:81225–81256.

Yue Zuo, Yuxiao Fei, Wanting Ning, Jiayi Huang, Yubo Feng, and Lishuang Li. 2025a. Rule-guided extraction: A hierarchical rule optimization framework for document-level event argument extraction. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 21155–21171.

Yuxin Zuo, Wenxuan Jiang, Wenxuan Liu, Zixuan Li, Long Bai, Hanbin Wang, Yutao Zeng, Xiaolong Jin, Jiafeng Guo, and Xueqi Cheng. 2025b. Knowcoder-x: Boosting multilingual information extraction via code. In *Findings of the Association for Computational Linguistics: ACL*, pages 14486–14509.

A Dataset Statistics

To evaluate the effectiveness of UC-UIE, extensive experiments are undertaken across 34 datasets, 30 of which come from IE INSTRUCTIONS (Wang et al., 2023) for NER, RE and EE tasks. Additionally, we supplement SemEval 14res (Pontiki et al., 2014), SemEval 14lap (Pontiki et al., 2014), SemEval 15res (Pontiki et al., 2015) and SemEval 16res (Pontiki et al.) for ABSA task, which are processed following the method in InstructUIE (Wang et al., 2023) to align with the unified format. Also following InstructUIE, we limit the training samples per dataset to at most 10,000 for dataset balance. Details of the datasets are shown in Table 7.

Task	Dataset	Domain	Train	Test	Setting	#Label	#Train	#Validation	#Test
NER	ACE2004 (Mitchell et al., 2005)	News	✓	✓	sp	7	6,202	745	812
	ACE2005 (Walker et al., 2006)	News	✓	✓	sp	7	7,299	971	1,060
	Broad Twitter (Derczynski et al., 2016)	Social	✓	✓	sp	3	5,334	2,000	2,001
	CoNLL2003 (Sang and De Meulder, 2003)	News	✓	✓	sp, lr	4	14,041	3,250	3,453
	OntoNotes (Palmer and Weischedel, 2006)	Multi-domain	✓	✓	sp	18	107,032	14,110	10,838
	bc5cdr (Li et al., 2016)	Biomedical	✓	✓	sp	2	4,560	4,581	4,797
	ncbi-disease (Dogan et al., 2014)	Biomedical	✓	✓	sp	1	5,432	923	940
	MultiNERD (Tedeschi and Navigli, 2022)	Wikipedia	✓	✓	sp	16	134,144	10,000	10,000
	bc2gm (Kocaman and Talby, 2021)	Biomedical	✓	✓	sp	1	12,500	2,500	5,000
	FabNER (Kumar and Starly, 2022)	Science	✓	✓	sp	12	9,435	2,182	2,064
	WikiANN (Pan et al., 2017)	Wikipedia	✓	✓	sp	3	20,000	10,000	10,000
	FindVehicle (Guan, 2022)	Traffic	✓	✓	sp	21	21,565	20,777	20,777
	GENIA_NER (Kim et al., 2003)	Biomedical	✓	✓	sp	5	15,023	1,669	1,854
	HarveyNER (Chen et al., 2022)	Social	✓	✓	sp	4	3,967	1,301	1,303
	CrossNER-AI (Liu et al., 2021)	Science		✓	zs	14	-	-	431
	CrossNER-Literature (Liu et al., 2021)	Literature		✓	zs	12	-	-	416
	CrossNER-Music (Liu et al., 2021)	Music		✓	zs	13	-	-	465
	CrossNER-Politics (Liu et al., 2021)	Politics		✓	zs	9	-	-	650
	CrossNER-Science (Liu et al., 2021)	Science		✓	zs	17	-	-	543
	MIT Movie (Liu et al., 2019)	Reviews		✓	zs	12	-	-	2,442
	MIT Restaurant (Liu et al., 2019)	Reviews		✓	zs	8	-	-	1,520
RE	ADE corpus (Gurulingappa et al., 2012)	Biomedical	✓	✓	sp	1	3,417	427	428
	CoNLL2004 (Roth and Yih, 2004)	News	✓	✓	sp, lr	5	922	231	288
	NYT (Riedel et al., 2010)	News	✓	✓	sp	24	56,196	5,000	5,000
	SemEvalRE (Pontiki et al.)	General	✓	✓	sp	10	6,507	1,493	2,717
	FewRel (Han et al., 2018)	Wikipedia		✓	zs	25	-	-	17,291
	Wiki-ZSL (Chen and Li, 2021)	Wikipedia		✓	zs	25	-	-	23,113
EE	ACE2005 (Walker et al., 2006)	News	✓	✓	sp, lr	33	3342	327	293
	CASIE (Lu et al., 2021)	Cybercrime	✓	✓	sp	5	3751	788	1500
	PHEE (Sun et al., 2022)	General	✓	✓	sp	2	2,898	961	968
ABSA	SemEval 14res (Pontiki et al., 2014)	Reviews		✓	zs	3	-	-	492
	SemEval 14lap (Pontiki et al., 2014)	Reviews		✓	zs	3	-	-	328
	SemEval 15res (Pontiki et al., 2015)	Reviews		✓	zs	3	-	-	322
	SemEval 16res (Pontiki et al.)	Reviews		✓	zs	3	-	-	326

Table 7: Datasets used in experiments. sp, lr and zs denote the supervised, low-resource and zero-shot setting, respectively. #Label denotes the number of categories, #Train, #Validation and #Test denote the number of instances of the split datasets.