Eliciting Zero-Shot Named Entity Recognition in Large Language Models via Instruction Semantic Elaboration

Kaiyin Zhou[1,2], **Zhiyi Song**[1,2], **Xiangling Fu**[1,2]*, **Chenwei Yan**[3], **Xinxin You**[4], **Shaohui Liu**[1,2], **Ji Wu**[4,5,6], **Xien Liu**[4]*
[1]School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing, China
[2]Key Laboratory of Trustworthy Distributed Computing and Service (BUPT), Ministry of Education, Beijing, China.
[3]School of Artificial Intelligence and Data Science, University of International Business and Economics, Beijing, China
[4]Department of Electronic Engineering, Tsinghua University, Beijing, China
[5]College of AI, Tsinghua University, Beijing, China
[6]Beijing National Research Center for Information Science and Technology, Beijing, China

Abstract

Large language models (LLMs) have achieved notable success in information extraction, yet their zero-shot named entity recognition (NER) performance remains unsatisfactory. We argue that this limitation stems not from intrinsic defects within the models, but rather from the failure of existing instruction paradigms to effectively elicit their latent capabilities. Drawing inspiration from cognitive science, we propose an instruction optimization framework termed Instruction Semantic Elaboration (ISE), which effectively elicits models' latent NER capabilities by mimicking the semantic elaboration process—specifically by contrasting related concepts, decomposing constituent elements, and illustrating with concrete examples. We evaluate ISE under a zero-shot setting across four domain-specific datasets, where it achieves an 11.12% F1 improvement over initial instructions and outperforms strong baselines by 6.92%. Furthermore, the framework generalizes well across model architectures and parameter scales, while remaining robust to the quality of initial instructions. This study offers a novel perspective on activating NER capabilities in LLMs, facilitating their efficient and stable deployment in real-world scenarios. Code is available at https://github.com/THUMLP/ISE.

1 Introduction

Large Language Models (LLMs) have recently catalyzed a paradigm shift in unstructured data processing, emerging as a pivotal driver for Information Extraction (IE) [1–3]. Despite their remarkable generative capabilities, empirical applications reveal that LLMs exhibit significant instruction sensitivity [4–6], a phenomenon also evident in Named Entity Recognition (NER) tasks [7]. As illustrated in Figure 1, the direct usage of generic label descriptions often results in extraction failures, whereas optimized descriptions successfully steer the model toward precise recognition. This observation suggests that the challenge lies not in a deficiency of LLMs' NER capabilities, but in the failure to effectively activate their intrinsic potential via instructions, which compromises the stability and robustness of extraction performance and hinders real-world deployment [8, 9]. Consequently, it is imperative to devise effective methodologies to fully activate the NER capabilities of LLMs.

*Corresponding authors: Xiangling Fu and Xien Liu.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

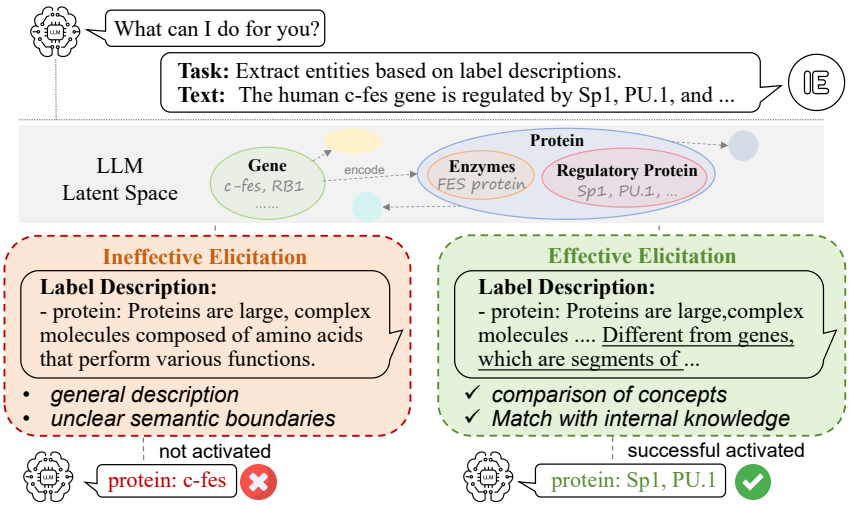


Figure 1: Illustration of instruction impact in activating LLMs' latent NER capabilities. In the latent space, related concepts cluster together. Left: Generic instruction with ambiguous boundaries fail to elicit NER capabilities. Right: Detailed instruction with clear boundaries elicit NER capabilities.

Existing research primarily addresses this challenge through adaptive fine-tuning. Methods such as UIE [10] and InstructUIE [11] employ natural language instructions, whereas KnowCoder [12] and GoLLIE [13] introduce code-style instructions for supervised fine-tuning. Although these approaches effectively elicit NER capabilities, their heavy reliance on extensive annotated data and computational resources impedes rapid adaptation to new tasks, particularly in zero-shot scenarios. Consequently, instruction engineering has emerged as a resource-efficient alternative, aiming to unleash model potential by optimizing the instructions themselves [14–17]. Early attempts relied on manual error analysis and heuristic rules [18], a process that is not only labor-intensive but also struggles to comprehensively cover diverse error patterns. While recent automated optimization techniques—such as task decomposition (e.g., Decomposed-QA [19], DEF [20]) and iterative self-correction (e.g., Self-improving [21], EvoPrompt [22])—have made significant strides, they predominantly operate within single-perspective optimization frameworks. As a result, current methods exhibit significant limitations in scalability and fall short of fully activating the model's intrinsic NER capabilities.

To bridge this gap, we propose **Instruction Semantic Elaboration (ISE)**, an annotation-free instruction optimization framework for zero-shot NER (Figure 2). Our framework is grounded in cognitive theories of complex concept acquisition. Humans rarely grasp the nuances and boundaries of complex concepts solely through simple definitions. Conversely, efficient learning necessitates a process termed "semantic elaboration"—wherein learners achieve a comprehensive understanding by establishing deep connections between new concepts and existing knowledge networks through contrasting with related concepts, decomposing concepts into constituent elements, and illustrating them with concrete examples [23, 24]. Building upon this insight, ISE simulates human learning processes from three distinct perspectives, fully associating the label definitions in instructions with the knowledge within LLMs, thereby comprehensively activating the NER capabilities of LLMs.

However, achieving semantic elaboration from different perspectives presents a critical bottleneck: due to the absence of annotated data, there is insufficient information to guide the precise construction of semantic elaboration instructions for each specific perspective. To overcome this information scarcity, we propose a co-evolutionary algorithm, which leverages mutual consensus among different semantic views to drive iterative instruction refinement. Furthermore, to mitigate the risk of semantic collapse [25–27]—where unconstrained evolution leads to instruction degradation or drift—we introduce a Relaxed Monotonicity Constraint (RMC) mechanism, ensuring the optimization process remains stable and consistently enhances the model's intrinsic capabilities.

Finally, our main contributions are:

1. We propose the ISE framework, a novel zero-shot instruction optimization paradigm inspired by cognitive science. It fully elicits LLMs' latent NER capabilities by simulating human learning processes through multi-perspective semantic elaboration (contrast, decomposition, and instantiation).

2. We devise a Co-evolutionary Algorithm to address the lack of annotated information during the instruction elaboration process, which leverages consensus across diverse views for iterative elaboration and incorporates a RMC mechanism to prevent semantic collapse.

3. We demonstrate performance improvements, with results across four domain-specific datasets showing a 11.12% gain in F1 score over initial instructions and a 6.92% lead over strong baselines, while maintaining high generalization across model architectures and robustness to instruction quality.

2 Related Work

LLMs for Named Entity Recognition. Early studies primarily focused on collecting large-scale datasets, standardizing these datasets into specific formats, and activating IE capabilities of LLMs through supervised fine-tuning. For example, UIE [10] and InstructUIE [11] used a natural language generation format, while KnowCoder [12] and GoLLIE [13] adopted a code format. While effective, these approaches rely heavily on large amounts of labeled data. Furthermore, training on general data fails to fully activate model's NER capabilities in specific domains [28, 29]. As a result, recent studies have shifted towards zero-shot NER. This approach avoids updating model parameters and instead relies directly on the LLMs' inherent semantic understanding [30, 19, 31]. However, experiments show that generic zero-shot instructions often struggle to effectively trigger the specific extraction skills of LLMs, highlighting the need to optimize how we approach zero-shot extraction [32, 33].

Optimization Strategies for Zero-Shot NER. Existing research uses various paradigms to improve the zero-shot performance of LLMs. Since manual instruction optimization [18, 34] is hard to scale, automated instruction generation has become a mainstream solution. For example, Self-improving [21] uses self-consistency mechanisms to select and update instruction examples, while EvoPrompt introduces an evolutionary framework to optimize instructions by analyzing label similarity [22]. Some studies reduce difficulty through task decomposition strategies like Decomposed-QA [19, 20], which breaks down complex NER goals into easier question-answering tasks. Additionally, retrieval augmentation is used to fill gaps in instruction understanding or internal knowledge, achieving better performance [35–37]. Recently, work has begun exploring multi-agent collaboration for zero-shot NER, using agent frameworks and teamwork mechanisms to enhance extraction capabilities [38–40]. Despite these improvements, existing methods still suffer from limited scalability or only optimize from a single perspective, failing to fully unlock the zero-shot NER potential of LLMs.

3 Methodology

We propose the ISE framework, which simulates three semantic elaboration processes—contrast, decomposition, and instantiation—to effectively activate the NER capabilities of the model. As shown in Figure 2, the framework first employs a co-evolutionary algorithm to iteratively optimize instructions, and then conducts inference to accomplish the final entity extraction. Before detailing these mechanisms, we first mathematically formulate our overall optimization objective.

3.1 Problem Formulation

Let $\mathcal{D} = \{x_i\}_{i=1}^{N}$ denote an unlabeled corpus used for instruction optimization, where each x_i is an input text. Let $\mathcal{Y}$ denote the predefined entity type set. For each input $x_i \in \mathcal{D}$, there exists an unknown true entity set $\mathcal{E}_i^*$, where each entity is represented as a tuple $e = (m, y)$. Here, m denotes a mention span in x_i, and $y \in \mathcal{Y}$ denotes its entity type. The goal is to optimize an instruction without accessing any gold annotations and then use it to predict entity sets for unseen inputs.

In this work, we formulate annotation-free zero-shot NER as an instruction optimization problem. Instead of updating LLM parameters, we seek an optimal instruction $\mathcal{P}^*$ that elicits the model to produce entity predictions close to latent true annotations. Given a candidate instruction $\mathcal{P}$, the LLM defines a conditional distribution $P_\theta(\mathcal{E} \mid x_i, \mathcal{P})$ over possible entity sets. Ideally, the optimal instruction should maximize the likelihood of latent true entity annotations over the unlabeled corpus:

$$\mathcal{P}^* = \underset{\mathcal{P}}{\operatorname{argmax}} \sum_{i=1}^{N} \mathbb{E}_{\mathcal{E} \sim q_i(\mathcal{E})} \left[\log P_\theta(\mathcal{E} \mid x_i, \mathcal{P}) \right], \tag{1}$$

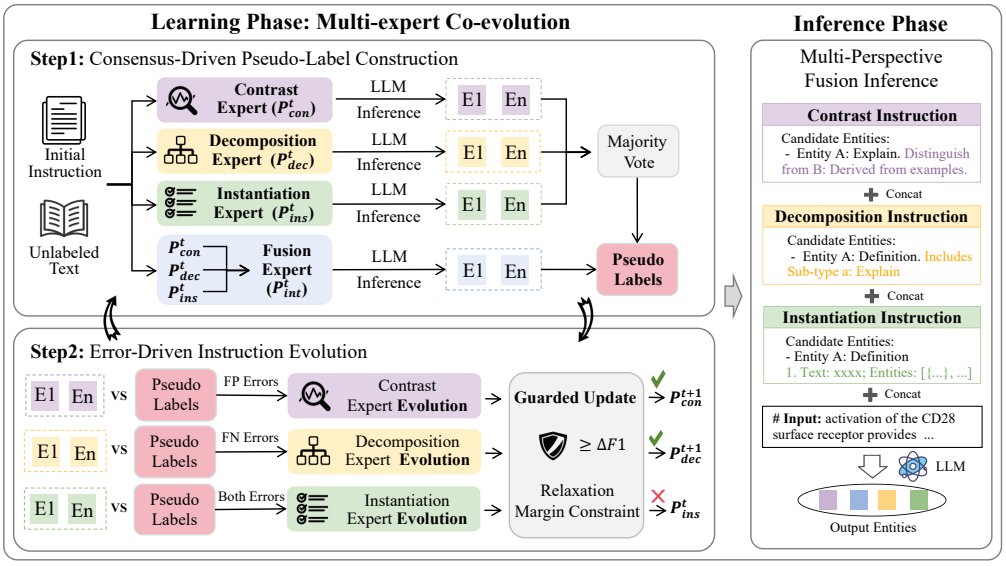


Figure 2: Overview of the ISE framework. Left: Learning phase based on a co-evolutionary algorithm. Right: Inference phase powered by the multi-expert instructions generated post-evolution.

where $q_i(\mathcal{E})$ denotes the latent annotation distribution that assigns high probability to the unknown true entity set $\mathcal{E}_i^*$. Since $q_i(\mathcal{E})$ is unavailable without gold annotations and the instruction space is discrete, ISE approximates this objective by iteratively constructing pseudo-label anchors and refining instructions through multi-perspective semantic elaboration experts.

3.2 Semantic Elaboration Experts

To implement multi-perspective semantic elaboration, we decompose the instruction $\mathcal{P}$ into three complementary refinement operations: contrast, decomposition, and exemplification. Let $\mathcal{K} = \{con, dec, ins\}$ denote the index set of the three semantic elaboration experts. Each operation is managed by a specialized expert $\mathcal{M}_k$ with instruction $\mathcal{P}_k$, where $k \in \mathcal{K}$:

1. Contrastive Elaboration Expert ($\mathcal{M}_{con}$): Refines $\mathcal{P}_{con}$ to distinguish the target concept y from semantically similar or easily confusable concepts, addressing boundary ambiguity.

2. Decompositional Elaboration Expert ($\mathcal{M}_{dec}$): Refines $\mathcal{P}_{dec}$ by decomposing a high-level tag y into fine-grained subtypes or attributes, improving semantic coverage.

3. Exemplificational Elaboration Expert ($\mathcal{M}_{ins}$): Refines $\mathcal{P}_{ins}$ by grounding label definitions with reliable contextual examples selected from pseudo-labeled data.

3.3 Co-evolutionary Algorithm

Due to the multi-perspective simulation, the global optimization of $\mathcal{P}$ is decomposed into the joint evolution of three expert-specific instruction sets $\{\mathcal{P}_{con}, \mathcal{P}_{dec}, \mathcal{P}_{ins}\}$. To achieve this without external supervision, we employ a co-evolutionary mechanism inspired by the EM paradigm [41, 42]. We initialize $\mathcal{P}^{(0)}$ using generic label definitions derived from task guidelines. At each iteration t, given the current instructions $\mathcal{P}^{(t)}$, the framework alternates between estimating the latent entities via hard pseudo-labeling (E-step) and updating the instructions (M-step) to mutually supervise and enhance one another. The procedure is summarized in Appendix Algorithm 1.

3.3.1 E-Step: Consensus-Driven Pseudo-Label Construction

In this phase, we aim to infer the ground-truth entities to serve as high-quality pseudo-labels for the subsequent iteration. To capture individual expert perspectives while resolving semantic ambiguities, we perform independent extractions followed by an integrated inference.

First, at each iteration t, the three experts independently parse the input text x using their current respective instructions. This yields three distinct sets of predicted entities:

$$\hat{\mathcal{E}}_k^{(t)} = \text{LLM}\left(x, \mathcal{P}_k^{(t)}\right), \quad k \in \mathcal{K}. \tag{2}$$

Simultaneously, to synergize the multi-perspective knowledge, we integrate contrastive boundaries $\mathcal{P}_{con}^{(t)}$, fine-grained subtypes $\mathcal{P}_{dec}^{(t)}$, and concrete exemplars $\mathcal{P}_{ins}^{(t)}$ into a unified instruction $\mathcal{P}_{int}^{(t)}$ (see Appendix 16 for details). We introduce a Fusion Expert ($\mathcal{M}_{fuse}$) that executes this unified instruction. The LLM, acting as this Fusion Expert, then utilizes this instruction, together with the three experts' predictions $\{\hat{\mathcal{E}}_k^{(t)}\}_{k\in\mathcal{K}}$ on x, to generate a candidate merged set $\tilde{\mathcal{E}}_{int}^{(t)}$ in a single inference pass.

$$\mathcal{P}_{int}^{(t)} = \text{Integrate}\left(\mathcal{P}_{con}^{(t)}, \mathcal{P}_{dec}^{(t)}, \mathcal{P}_{ins}^{(t)}\right), \quad \tilde{\mathcal{E}}_{int}^{(t)} = \mathcal{M}_{fuse}\left(x, \mathcal{P}_{int}^{(t)}, \{\hat{\mathcal{E}}_k^{(t)}\}_{k\in\mathcal{K}}\right) \tag{3}$$

To prevent semantic drift, the final pseudo-labels $\mathcal{S}_{pseudo}^{(t)}$ are constructed via a routine consensus heuristic. Let $\mathcal{C}^{(t)}$ denote the set of high-consensus entities obtained by majority voting over the three experts ($\hat{\mathcal{E}}_k^{(t)}$) together with the integrated prediction $\tilde{\mathcal{E}}_{int}^{(t)}$. The update rule is formulated as:

$$\mathcal{S}_{pseudo}^{(t)} = \mathcal{C}^{(t)} \cup \left(\tilde{\mathcal{E}}_{int}^{(t)} \cap \mathcal{S}_{pseudo}^{(t-1)}\right) \tag{4}$$

This formulation anchors the pseudo-labels on the current consensus among individual experts, while using the integrated prediction as a stability filter to retain historically validated hard entities. This ensures a stable and noise-filtered anchor for the subsequent M-step.

3.3.2 M-Step: Error-Driven Instruction Evolution

In this phase, the instruction set $\mathcal{P}^{(t)} = \{\mathcal{P}_{con}^{(t)}, \mathcal{P}_{dec}^{(t)}, \mathcal{P}_{ins}^{(t)}\}$ is updated to maximize the likelihood of the pseudo-labels $\mathcal{S}_{pseudo}^{(t)}$. Rather than using gradient-based optimization, a rewrite expert performs text-based rewriting by holistically analyzing error patterns (e.g., confusion cases or missing subtypes) to generate an improved instruction in a single inference pass.

The error patterns, crucial for this process, are derived as follows:

- **False Positives ($\mathcal{E}_{FP}^k$):** Entities predicted by the expert k but absent in the pseudo-labels:

$$\mathcal{E}_{FP}^k = \hat{\mathcal{E}}_k^{(t)} \setminus \mathcal{S}_{\text{pseudo}}^{(t)}. \tag{5}$$

- **False Negatives ($\mathcal{E}_{FN}^k$):** Entities present in the pseudo-labels but not predicted by expert k:

$$\mathcal{E}_{FN}^k = \mathcal{S}_{\text{pseudo}}^{(t)} \setminus \hat{\mathcal{E}}_k^{(t)}. \tag{6}$$

Using these error sets, we adaptively refine the instructions associated with each expert as follows:

Contrast Evolution: To sharpen class boundaries, the optimizer utilizes false positive samples ($\mathcal{E}_{FP}^{con}$) to identify easily confused concepts. It then summarizes the types of easily confused semantics and methods for semantic differentiation, integrating them into the rewritten instruction $\mathcal{P}_{con}^{(t+1)}$.

$$\mathcal{P}_{con}^{(t+1)} \leftarrow \text{Rewrite}(\mathcal{P}_{con}^{(t)}, \mathcal{E}_{FP}^{con}) \tag{7}$$

Decomposition Evolution: To enhance semantic completeness, the optimizer examines false negative samples ($\mathcal{E}_{FN}^{dec}$) to infer missing fine-grained subtypes. It regenerates $\mathcal{P}_{dec}^{(t+1)}$ by explicitly embedding these fine-grained concepts and their descriptions into the semantic scope of the instruction.

$$\mathcal{P}_{dec}^{(t+1)} \leftarrow \text{Rewrite}(\mathcal{P}_{dec}^{(t)}, \mathcal{E}_{FN}^{dec}) \tag{8}$$

Instantiation Evolution: To improve contextual relevance and specificity, the optimizer evaluates the hybrid error set $\mathcal{E}_{err} = \mathcal{E}_{FN}^{ins} \cup \mathcal{E}_{FP}^{ins}$ to identify complex and representative cases. It directly regenerates the instruction $\mathcal{P}_{ins}^{(t+1)}$ by replacing less effective demonstration samples with these high-value instances in a single generative step.

$$\mathcal{P}_{ins}^{(t+1)} \leftarrow \text{Rewrite}(\mathcal{P}_{ins}^{(t)}, \mathcal{E}_{err}) \tag{9}$$

All instructions for rewriting experts are shown in the Appendix Table 13, 14, 15.

3.3.3 Guarded Update and Convergence

We implement RMC mechanism to ensure stable instruction evolution. Let $F_1(\mathcal{P}, \mathcal{S})$ denote the F1 score using instruction $\mathcal{P}$ evaluated against the pseudo-label set $\mathcal{S}$. A candidate instruction $\tilde{\mathcal{P}}_k$ generated in the M-step is accepted as $\mathcal{P}_k^{(t+1)}$ only if it meets a relaxation margin δ:

$$F_1(\tilde{\mathcal{P}}_k, \mathcal{S}_{pseudo}^{(t)}) \geq F_1(\mathcal{P}_k^{(t)}, \mathcal{S}_{pseudo}^{(t)}) - \delta \tag{10}$$

If this condition fails, the previous instruction $\mathcal{P}_k^{(t)}$ is retained.

The evolution terminates when all experts converge. Specifically, an expert halts its updates if the change in its F_1 score on $\mathcal{S}_{pseudo}^{(t)}$ remains below a threshold ϵ for n consecutive iterations.

3.4 Inference via Multi-Perspective Semantic Fusion

After the co-evolutionary algorithm converges, we obtain the optimized instructions $\mathcal{P}^* = \{\mathcal{P}_{con}^*, \mathcal{P}_{dec}^*, \mathcal{P}_{ins}^*\}$. Unlike the E-step, which relies on the three expert-perspective instructions and the predictions of all three experts to generate an integrated result, the inference stage simply merges these three perspective instructions into a single unified instruction $\mathcal{P}_{int}^*$, eliminating the need for each expert to perform inference separately. For unseen texts, the final extraction is efficiently performed via a single inference pass by the Inference Expert ($\mathcal{M}_{inf}$):

$$\hat{\mathcal{E}} = \mathcal{M}_{inf}\left(x, \mathcal{P}_{int}^*\right) \tag{11}$$

By unifying the optimized boundaries, subtypes, and demonstrations into a single prompt, we ensure consistent extraction performance while minimizing computational overhead.

4 Experiments

4.1 Setup

Dataset: We evaluate our framework on four benchmarks: **GENIA** [43], **AI** [44], **Restaurant**, and **Movie** [45]. Label descriptions are derived from official annotation guidelines or, when unavailable, summarized by an LLM using 100 labeled samples. To manage computational overhead, we perform experiments on a fixed set of 200 instances randomly sampled from the test set of each dataset, following similar practices in prior work [19, 21]. Notably, following the zero-shot transductive setting [46, 21], all experiments utilize only the unlabeled texts of these 200 instances for learning. We conduct our evaluation across three representative Large Language Models as backbones: Qwen2.5 [47], Gemma3 [48], and GPT-3.5-turbo.

Baselines: We compare ISE with representative zero-shot NER baselines in four groups. (1) **Non-LLM methods**: **UIE-base-en** [10] treats information extraction as a unified text-to-structure generation task and supports transfer to unseen schemas. **GLiNER-large** [49] is a lightweight BERT-like model that matches entity type embeddings with span representations to identify arbitrary entity types. (2) **Prompt-based methods**: **Vanilla** prompts LLMs to extract target entities directly from the input text without label descriptions, while **Vanilla**$_{desc}$ additionally provides label descriptions. **Decomposed-QA** [19] formulates NER as label-wise question answering, and **SALLM** [19] further incorporates syntactic augmentation and voting. (3) **Retrieval-enhanced methods**: **KGB-Retrieval** retrieves entity-level matches from a dictionary constructed using the training set and uses them as prior context, while **SILLM** [21] retrieves reliable self-labeled unlabeled instances as demonstrations. (4) **Self-improving methods**: **EvoPrompt** [22] iteratively refines type definitions and pseudo-annotated samples based on label confusions.

Metrics: We employ the standard exact match criterion for evaluation: a predicted entity is considered correct only if both its predicted text span and entity type align perfectly with the ground truth. To ensure a nuanced assessment across all entity categories regardless of class distribution, we report the average Macro Precision, Recall, and F1-score over three independent runs.

Experiment details: During the iterations, all expert temperatures are set to 0.0, except for the three semantic elaboration experts, which are initialized to 0.6 in the first round. The evolution loop runs up to 10 iterations with a convergence threshold $\epsilon = 0.04$ and a tolerance count $n = 3$. The guarded update margin is $\delta = 0.4$. Refer to Appendix E for the parameter sensitivity analysis.

4.2 Main Results

As shown in Table 1, ISE achieves the highest average F1 under all three backbones, reaching 63.40%, 60.18%, and 65.63% with Qwen2.5-32B, Gemma3-27B, and GPT-3.5-turbo, respectively. Averaged across datasets and backbones, ISE improves over Vanilla$_{desc}$ by 11.12 F1 points and surpasses the strongest baseline, EvoPrompt, by 6.92 F1 points. These gains indicate that multi-perspective semantic elaboration provides more effective task guidance than using label descriptions or optimized prompts alone. Compared with non-LLM baselines, ISE achieves a higher overall F1 than UIE-base-en and GLiNER-large, whose average scores are 42.25% and 44.13%, respectively.

In addition, ISE also demonstrates strong robustness across different domains and backbone models. Across three LLM backbones, ISE consistently outperforms LLM-based methods on domain-specific datasets such as GENIA and AI, and ambiguous real-world datasets such as Restaurant and Movie. For example, under Qwen2.5-32B, ISE improves the best baseline F1 from 55.70% to 66.07% on GENIA and from 53.39% to 55.82% on Movie. Compared with EvoPrompt, the strongest baseline on average, ISE improves the backbone-level average F1 by 6.23, 5.85, and 8.67 points under Qwen2.5-32B, Gemma3-27B, and GPT-3.5-turbo, respectively. These consistent gains suggest that the optimized semantic instructions generalize well across entity distributions, annotation characteristics, and LLM architectures in practice.

Table 1: Main F1. Best and second-best results are marked in **bold** and underlined within each block.

Methods	GENIA	AI	Res.	Movie
Non-LLM Baselines				
UIE-base-en	25.24	33.64	76.22	33.91
GLiNER-large	39.72	62.23	41.16	33.42
Backbone: Qwen2.5-32B				
Vanilla	35.40	54.44	45.10	46.94
Vanilla$_{desc}$	42.33	63.03	56.16	50.76
Decomposed-QA	44.88	46.39	43.12	27.89
SALLM	42.63	47.89	44.43	35.13
KGB-Retrieval	50.61	57.10	62.60	53.39
SILLM	50.83	60.12	51.45	48.90
EvoPrompt	55.70	65.27	58.68	49.02
ISE(our)	**66.07**	**69.08**	**62.62**	**55.82**
Backbone: Gemma3-27B				
Vanilla	42.72	50.48	44.25	41.95
Vanilla$_{desc}$	37.39	55.63	54.88	50.36
Decomposed-QA	44.54	52.79	48.13	44.03
SALLM	46.75	52.28	47.76	41.70
KGB-Retrieval	45.84	63.68	53.87	52.47
SILLM	38.39	62.21	52.78	46.96
EvoPrompt	54.83	63.34	54.89	44.27
ISE(our)	**57.78**	**66.27**	**61.17**	**55.51**
Backbone: GPT-3.5-turbo				
Vanilla	41.74	48.41	42.78	43.89
Vanilla$_{desc}$	40.79	60.75	61.21	50.11
Decomposed-QA	46.56	54.32	51.73	40.91
SALLM	51.15	52.59	49.43	49.08
KGB-Retrieval	51.22	63.21	55.60	55.79
SILLM	42.95	63.80	60.62	50.93
EvoPrompt	55.39	69.74	56.46	46.24
ISE(our)	**61.40**	**73.94**	**69.20**	**57.98**

4.3 Ablation Study

Table 2 verifies the distinct roles of each elaboration expert. Semantic Contrast functions as the primary boundary guardian; its exclusion results in the most significant Precision drop ($\Delta = -1.38\%$) among all elaboration experts, confirming its necessity in suppressing false positives by explicitly distinguishing confounding concepts. Semantic Decomposition ensures semantic comprehensiveness; its absence triggers the steepest Recall decline among the experts ($\Delta = -4.85\%$), leading to the most severe overall degradation ($\Delta = -3.15\%$ F1). Meanwhile, Semantic Instantiation bridges abstract definitions with specific contexts, evidenced by a decline in recall ($\Delta = -2.14\%$) and overall F1-score ($\Delta = -1.00\%$) upon its removal, despite a slight apparent increase in precision ($\Delta = +1.28\%$).

Table 2: Ablation results for ISE with Qwen2.5-32B, averaged across benchmarks. Parentheses show percentage-point changes from the full ISE.

Method	**Precision**	**Recall**	**F1-Score**
EvoPrompt	62.63	56.49	57.17
ISE	**63.88**	**65.17**	**63.40**
Semantic Elaboration Experts			
w/o Contrast	62.50 (-1.38)	61.61 (-3.56)	60.70 (-2.70)
w/o Decomp.	63.00 (-0.88)	60.32 (-4.85)	60.25 (-3.15)
w/o Instant.	65.16 (+1.28)	63.03 (-2.14)	62.40 (-1.00)
Expert Instruction Fusion			
w/o Fusion	67.23 (+3.35)	59.64 (-5.53)	61.58 (-1.82)

The Fusion Expert proves indispensable for harmonizing conflicting signals. Replacing it with simple majority voting (≥ 2 experts) causes a severe Recall drop ($\Delta = -5.53\%$) and overall F1 decline ($\Delta = -1.82\%$), despite a superficial Precision gain ($\Delta = +3.35\%$). This highlights its vital role in

balancing the expansive nature of instantiation and decomposition with the restrictive precision of contrast, ensuring structurally robust instructions. Detailed results are provided in Appendix Table 11.

4.4 Evolutionary Dynamics and Scaling Analysis

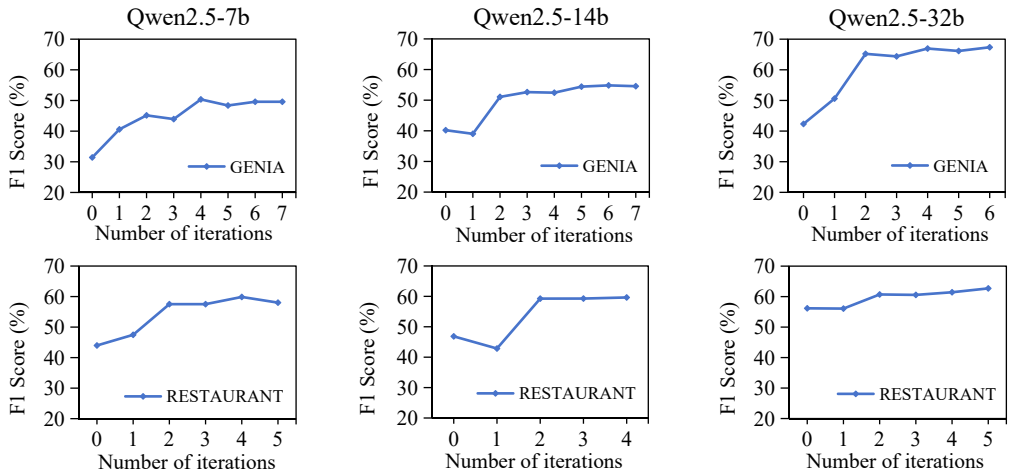


Figure 3: Evolutionary dynamics across model scales. F1-score trajectories of Qwen2.5 (7B, 14B, 32B) show rapid adaptation and stable convergence on GENIA and RESTAURANT datasets.

To examine optimization stability and model-scale effects, Figure 3 reports the F1 trajectories of Qwen2.5 models on GENIA and RESTAURANT, with additional results in Appendix Figure 9. Specifically, ISE achieves most gains within the first 3–5 iterations (e.g., the 32B model achieves a 22.88-point F1 increase across the first two steps on GENIA) and then gradually stabilizes, indicating stable convergence and supporting our hypothesis that LLMs possess latent NER capabilities that can be elicited through semantic elaboration. Moreover, larger models obtain stronger performance, especially on the domain-specific GENIA dataset, where the 7B model saturates at an F1-score of approximately 50%, whereas the 32B model peaks near 67%. In addition, the gains are domain-sensitive: across all model scales, improvements on GENIA consistently exceed those on RESTAURANT, with the 32B model achieving a total gain of +25.03% on GENIA compared to +6.53% on RESTAURANT. Overall, these results suggest that ISE consistently benefits different model scales, while the improvement magnitude depends on model capacity and dataset characteristics.

4.5 Pseudo-Label Quality and Error Dynamics

To further examine the optimization process, we track task performance, pseudo-label F1, and pseudo-label error rate during instruction refinement, as shown in Figure 4. The pseudo-label metrics are computed only for post-hoc analysis using gold annotations and are not used during optimization. On GENIA, extraction F1 rises from 50.46% to 65.59%, while pseudo-label F1 improves from 52.95% to about 63.00%, and the error rate quickly drops before stabilizing around 33–34%. On Restaurant, extraction F1 improves more moderately from 58.77% to 61.43%, with pseudo-label F1 increasing from 59.02% to 62.58% and the error rate fluctuating within a narrow range. These results show that ISE can benefit from sufficiently stable, though imperfect, pseudo-label anchors. More importantly, pseudo-label errors do not exhibit progressive or uncontrolled accumulation across iterations, suggesting that the iterative process avoids substantial error amplification during instruction refinement.

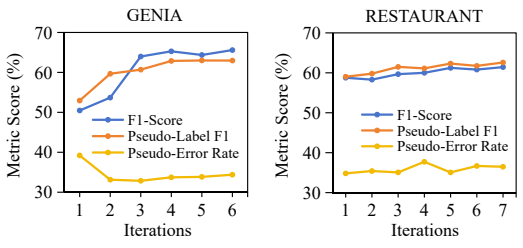


Figure 4: Evolution of extraction F1, pseudo-label F1, and pseudo-label error rate during ISE optimization on GENIA and Restaurant.

4.6 Relaxation Margin Impact Analysis

We analyze the impact of the relaxation margin δ on framework performance to evaluate the effectiveness of the RMC. As shown in Figure 5, this controlled mechanism is necessary: enforcing strict monotonicity (i.e., $\delta = 0$) causes optimization to get stuck in local optima (e.g., achieving an F1-score of only 61.31% on the GENIA dataset). Conversely, removing the RMC may lead to semantic drift, resulting in performance degradation. A moderately relaxed boundary ($\delta \in [0.2, 0.4]$) achieves the best trade-off, improving the F1-score by +3.93% compared to the unconstrained setting ($\delta = 1$) on GENIA. This confirms the RMC as a crucial stabilizer: it allows strategic exploration—by accepting short-term fluctuations to uncover better semantic structures—while effectively preventing semantic collapse. Comparable trends are observed on the remaining datasets, as reported in Appendix Figure 10.

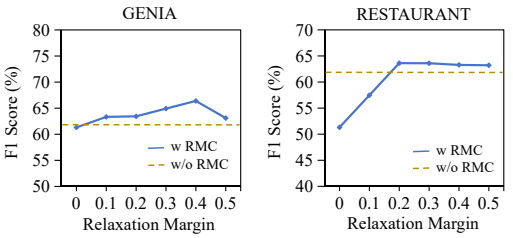


Figure 5: Effect of the relaxation margin δ. Dashed lines denote the unconstrained setting; solid lines denote different δ values.

4.7 Initial Instruction Sensitivity Analysis

To evaluate robustness, we test ISE under three initialization settings: label-only (Init A), LLM-generated descriptions (Init B), and guideline-based descriptions used in main experiments (Init C). As shown in Figure 6, despite significant initial performance gaps—e.g., a 17.01% disparity on GENIA between the weakest (Init A: 35.40%) and strongest (Init B: 52.41%) setups—ISE demonstrates strong semantic convergence. Post-evolution, performance aligns across all settings, confirming the effectiveness of our co-evolution in mitigating initialization differences and consistently attaining a stable optimum. Detailed results on additional datasets are reported in Appendix Table 12.

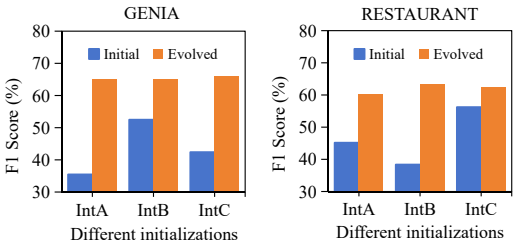


Figure 6: Performance under different initialization methods: Init A (label-only), Init B (LLM-generated), and Init C (guideline-based).

4.8 Iterative Data Efficiency Analysis

To assess the data efficiency of ISE, we analyze its performance under varying levels of unlabeled corpus usage, ranging from 0% to 50%, and use the full dataset at 100% as a reference. The results are shown in Figure 7, with complete experimental results provided in Appendix Figure 11. ISE exhibits favorable data efficiency: using only 30% of the data improves the Restaurant F1-score from 56.16% to 63.93%. Moreover, performance stabilizes relatively early. On GENIA, using 20% of the data achieves an F1-score of 64.92%, which is comparable to the 66.07% obtained with the full dataset. These results indicate that semantic error patterns within the corpus are largely repetitive, enabling ISE to capture core misalignments and refine instructions with lower computational overhead.

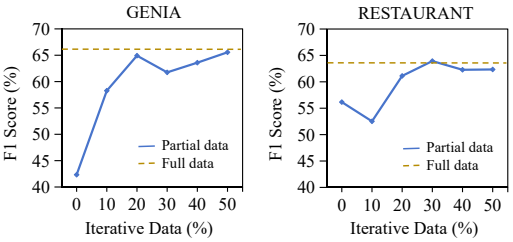


Figure 7: Impact of data scale. Solid lines show ISE with varying unlabeled data proportions; dashed lines denote full-data ISE using 100% data.

5 Qualitative Analysis

Table 3 presents two GENIA cases to illustrate the extraction changes after instruction evolution. In Case S1, the initial output incorrectly assigns the nucleic-acid phrase “antisense oligonucleotide” to the protein category and misses it as DNA; after evolution, the false protein prediction is removed

Table 3: Qualitative analysis on the GENIA dataset. Red denotes extraction errors or missed entities, while green denotes corrected or recalled entities.

Case / Gold Annotation	Initial Output	Evolved Output
Case S1: Several small molecules, an antisense oligonucleotide, ... inhibit NF-kappa b activation. ***Gold:*** DNA: "antisense oligonucleotide"; Protein: "NF-kappa b".	**Protein:** "NF-kappa b", "antisense oligonucleotide" **DNA:** N/A	**Protein:** "NF-kappa b"; **DNA:** "antisense oligonucleotide"
	Error: The DNA-related oligonucleotide is falsely labeled as protein.	***Correction:*** Clear DNA–protein boundaries remove the false protein prediction and recover the DNA entity.
Case S2: Involvement of the N-terminal region of the human mineralocorticoid receptor hormone-binding domain ... by a new monoclonal antibody. ***Gold:*** Protein: "N-terminal region", "human mineralocorticoid receptor hormone-binding domain", "new monoclonal antibody".	**Protein:** "human mineralocorticoid receptor hormone-binding domain"	**Protein:** "N-terminal region", "human ... receptor hormone-binding domain", "new monoclonal antibody"
	Error: The model detects the central domain but misses related protein components.	***Correction:*** Richer protein semantics cover components and antibodies, recovering all gold protein mentions.

and the correct DNA entity is recovered. In Case S2, the initial output captures only the central receptor-domain span but misses related protein mentions, including "N-terminal region" and "new monoclonal antibody". The evolved output recalls all gold protein spans. These results indicate that ISE improves both boundary discrimination and semantic coverage. As detailed in Appendix G, such improvements are supported by expert-specific elaborations: contrastive cues reduce confusion between proteins and nucleic-acid entities, decomposition broadens the coverage of protein subtypes and components, and instantiation examples guide span-boundary decisions.

6 Conclusion

This paper proposes ISE, a novel framework that effectively unleashes the latent zero-shot NER capabilities of LLMs. By simulating human cognitive learning processes—specifically through semantic contrast, decomposition, and instantiation—coupled with a constrained co-evolutionary algorithm, ISE achieves state-of-the-art performance across diverse domain-specific benchmarks. Our comprehensive analysis yields vital insights: capability elicitation exhibits distinct scaling behaviors, requires minimal iterative data, and necessitates mechanisms like our Relaxed Monotonicity Constraint to prevent semantic drift. Overall, ISE demonstrates that systematically optimizing instruction semantics serves as a highly efficient, robust, and model-agnostic paradigm, paving a new path for deploying reliable LLM-based information extraction systems in real-world scenarios.

Despite these promising results, ISE has several limitations that point to future research directions. First, as an instruction-driven approach, ISE remains bounded by the intrinsic parametric knowledge of LLMs, suggesting that highly specialized domains may benefit from integration with external knowledge retrieval. Second, the multi-perspective co-evolutionary process increases token consumption and inference latency, as detailed in Appendix F. Finally, the Relaxed Monotonicity Constraint serves as a weak constraint mechanism: although it can mitigate extreme anomalous semantic drift to some extent, it cannot fundamentally eliminate the inherent drift issue in pseudo-label-based iterative algorithms, especially when the base model exhibits strong systematic biases in niche domains.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 72274022), the Hainan Province Health Science and Technology Innovation Joint Project (Grant No. WSJK2024MS233), and the Beijing Natural Science Foundation (Grant No. L2603023). The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

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

[2] Imed Keraghel, Stanislas Morbieu, and Mohamed Nadif. A survey on recent advances in named entity recognition. *arXiv preprint arXiv:2401.10825*, 2024.

[3] Zikang Zhang, Wangjie You, Tianci Wu, Xinrui Wang, Juntao Li, and Min Zhang. A survey of generative information extraction. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 4840–4870, 2025.

[4] Alberto Mario Ceballos-Arroyo, Monica Munnangi, Jiuding Sun, Karen Zhang, Jered Mcinerney, Byron C Wallace, and Silvio Amir. Open (clinical) llms are sensitive to instruction phrasings. In *Proceedings of the 23rd Workshop on Biomedical Natural Language Processing*, pages 50–71, 2024.

[5] Sheng Lu, Hendrik Schuff, and Iryna Gurevych. How are prompts different in terms of sensitivity? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5833–5856, 2024.

[6] Federico Errica, Davide Sanvito, Giuseppe Siracusano, and Roberto Bifulco. What did i do wrong? quantifying llms' sensitivity and consistency to prompt engineering. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1543–1558, 2025.

[7] Yubo Chen, Baoli Zhang, Sirui Li, Zhuoran Jin, Zhengyuan Cai, Yingzheng Wang, Delai Qiu, ShengPing Liu, and Jun Zhao. Prompt robust large language model for chinese medical named entity recognition. *Information Processing & Management*, 62(5):104189, 2025.

[8] Chen Liao, Gang Cheng, Shilei Huang, and Lin Yao. Llm-based automating product information retrieval for industry analysis: A real-world application. In *International Conference on Cognitive Computing*, pages 116–128. Springer, 2024.

[9] Yu-An Liu, Ruqing Zhang, Jiafeng Guo, and Maarten de Rijke. Robust information retrieval. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, pages 1008–1011, 2025.

[10] Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. Unified structure generation for universal information extraction. *arXiv preprint arXiv:2203.12277*, 2022.

[11] Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, et al. Instructuie: Multi-task instruction tuning for unified information extraction. *arXiv preprint arXiv:2304.08085*, 2023.

[12] Zixuan Li, Yutao Zeng, Yuxin Zuo, Weicheng Ren, Wenxuan Liu, Miao Su, Yucan Guo, Yantao Liu, Lixiang Lixiang, Zhilei Hu, et al. Knowcoder: Coding structured knowledge into llms for universal information extraction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8758–8779, 2024.

[13] Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. Gollie: Annotation guidelines improve zero-shot information-extraction. *arXiv preprint arXiv:2310.03668*, 2023.

[14] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM computing surveys*, 55(9):1–35, 2023.

[15] Dhananjay Ashok and Zachary C Lipton. Promptner: Prompting for named entity recognition. *arXiv preprint arXiv:2305.15444*, 2023.

[16] Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*, 2024.

[17] Fina Polat, Ilaria Tiddi, and Paul Groth. Testing prompt engineering methods for knowledge extraction from text. *Semantic Web*, 16(2):SW–243719, 2025.

[18] Yan Hu, Qingyu Chen, Jingcheng Du, Xueqing Peng, Vipina Kuttichi Keloth, Xu Zuo, Yujia Zhou, Zehan Li, Xiaoqian Jiang, Zhiyong Lu, et al. Improving large language models for clinical named entity recognition via prompt engineering. *Journal of the American Medical Informatics Association*, 31(9):1812–1820, 2024.

[19] Tingyu Xie, Qi Li, Jian Zhang, Yan Zhang, Zuozhu Liu, and Hongwei Wang. Empirical study of zero-shot ner with chatgpt. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7935–7956, 2023.

[20] Reza Averly and Xia Ning. Entity decomposition with filtering: A zero-shot clinical named entity recognition framework. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2935–2951, 2025.

[21] Tingyu Xie, Qi Li, Yan Zhang, Zuozhu Liu, and Hongwei Wang. Self-improving for zero-shot named entity recognition with large language models. In *Proceedings of the 2024 conference of the North American chapter of the association for computational linguistics: human language technologies (volume 2: short papers)*, pages 583–593, 2024.

[22] Zeliang Tong, Zhuojun Ding, and Wei Wei. Evoprompt: Evolving prompts for enhanced zero-shot named entity recognition with large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 5136–5153, 2025.

[23] Claire E Weinstein. Elaboration skills as a learning strategy. In *Learning strategies*, pages 31–55. Elsevier, 1978.

[24] Joel R Levin. Elaboration-based learning strategies: Powerful theory= powerful application. *Contemporary Educational Psychology*, 13(3):191–205, 1988.

[25] David Herel and Tomas Mikolov. Collapse of self-trained language models. *arXiv preprint arXiv:2404.02305*, 2024.

[26] Zhengwei Tao, Ting-En Lin, Xiancai Chen, Hangyu Li, Yuchuan Wu, Yongbin Li, Zhi Jin, Fei Huang, Dacheng Tao, and Jingren Zhou. A survey on self-evolution of large language models. *arXiv preprint arXiv:2404.14387*, 2024.

[27] Yixing Chen, Yiding Wang, Siqi Zhu, Haofei Yu, Tao Feng, Muhan Zhang, Mostofa Patwary, and Jiaxuan You. Multi-agent evolve: Llm self-improve through co-evolution. *arXiv preprint arXiv:2510.23595*, 2025.

[28] Hongyu Lin, Yaojie Lu, Jialong Tang, Xianpei Han, Le Sun, Zhicheng Wei, and Nicholas Jing Yuan. A rigorous study on named entity recognition: Can fine-tuning pretrained model lead to the promised land? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7291–7300, 2020.

[29] Yichu Zhou and Vivek Srikumar. A closer look at how fine-tuning changes bert. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 1046–1061, 2022.

[30] Farhad Pourpanah, Moloud Abdar, Yuxuan Luo, Xinlei Zhou, Ran Wang, Chee Peng Lim, Xi-Zhao Wang, and QM Jonathan Wu. A review of generalized zero-shot learning methods. *IEEE transactions on pattern analysis and machine intelligence*, 45(4):4051–4070, 2022.

[31] Mingyuan Qin, Lei Feng, Jing Lu, Ziyan Sun, Zhengyu Yu, and Lianyi Han. Zerotunebio ner: A three-stage framework for zero-shot and zero-tuning biomedical entity extraction using large language models and prompt engineering. *Computer Methods and Programs in Biomedicine*, page 109070, 2025.

[32] John Dagdelen, Alexander Dunn, Sanghoon Lee, Nicholas Walker, Andrew S Rosen, Gerbrand Ceder, Kristin A Persson, and Anubhav Jain. Structured information extraction from scientific text with large language models. *Nature communications*, 15(1):1418, 2024.

[33] Aishik Nagar, Viktor Schlegel, Thanh-Tung Nguyen, Hao Li, Yuping Wu, Kuluhan Binici, and Stefan Winkler. Llms are not zero-shot reasoners for biomedical information extraction. In *The Sixth Workshop on Insights from Negative Results in NLP*, pages 106–120, 2025.

[34] Meifang Chen, Suzhi Zhang, Luyang Liu, Gaozhan Li, and Xuezheng Shi. Improving named entity recognition performance for large language models using a prompt framework. In *2025 IEEE 12th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, volume 12, pages 828–832. IEEE, 2025.

[35] Ruichu Cai, Junhao Lu, Zhongjie Chen, Boyan Xu, and Zhifeng Hao. Handling missing entities in zero-shot named entity recognition: Integrated recall and retrieval augmentation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10790–10802, 2025.

[36] Haijiang Li, Cangqi Zhou, Jing Zhang, and Dianming Hu. Open-world knowledge augmentation for zero-shot information extraction in llms. In *International Conference on Intelligent Computing*, pages 304–315. Springer, 2025.

[37] Ivan Lopez, Akshay Swaminathan, Karthik Vedula, Sanjana Narayanan, Fateme Nateghi Haredasht, Stephen P Ma, April S Liang, Steven Tate, Manoj Maddali, Robert Joseph Gallo, et al. Clinical entity augmented retrieval for clinical information extraction. *npj Digital Medicine*, 8(1):45, 2025.

[38] Meng Lu, Yuzhang Xie, Zhenyu Bi, Shuxiang Cao, and Xuan Wang. Crossagentie: Cross-type and cross-task multi-agent llm collaboration for zero-shot information extraction. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 13953–13977, 2025.

[39] Zihan Wang, Ziqi Zhao, Yougang Lyu, Zhumin Chen, Maarten de Rijke, and Zhaochun Ren. A cooperative multi-agent framework for zero-shot named entity recognition. In *Proceedings of the ACM on Web Conference 2025*, pages 4183–4195, 2025.

[40] Yu Tian, Minghao Hu, Jianyong Duan, Wenpeng Hu, Zhunchen Luo, and Wei Luo. Knowledge-optimized multi-agent dynamics framework for zero-shot relation triplet extraction. In *2025 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2025.

[41] Todd K Moon. The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6):47–60, 1996.

[42] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

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

[44] Zihan Liu, Yan Xu, Tiezheng Yu, Wenliang Dai, Ziwei Ji, Samuel Cahyawijaya, Andrea Madotto, and Pascale Fung. Crossner: Evaluating cross-domain named entity recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 13452–13460, 2021.

[45] Asahi Ushio and Jose Camacho-Collados. T-ner: An all-round python library for transformer-based named entity recognition. In *Proceedings of the 16th conference of the European Chapter of the Association for Computational Linguistics: System demonstrations*, pages 53–62, 2021.

[46] Zhicai Wang, Yanbin Hao, Tingting Mu, Ouxiang Li, Shuo Wang, and Xiangnan He. Bi-directional distribution alignment for transductive zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19893–19902, 2023.

[47] Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Yunyang Wan, Yuqi Liu, Zeyu Cui, Zhenru Zhang, Zihan Qiu, Shanghaoran Quan, and Zekun Wang. Qwen2.5 technical report. *ArXiv*, abs/2412.15115, 2024.

[48] Gemma Team Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ram'e, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean-Bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gael Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Róbert Istvan Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, Andr'as Gyorgy, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Boxi Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, Cj Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Pluci'nska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, J. Michael Wieting, Jonathan Lai, Jordi Orbay, Joe Fernandez, Joshua Newlan, Junsong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson,

Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Ping mei Xu, Piotr Stańczyk, Pouya Dehghani Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Ardeshir Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vladimir Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab S. Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeffrey Dean, Demis Hassabis, Koray Kavukcuoglu, Clément Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and L'eonard Hussenot. Gemma 3 technical report. *ArXiv*, abs/2503.19786, 2025.

[49] Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. Gliner: Generalist model for named entity recognition using bidirectional transformer. In *Proceedings of the 2024 conference of the north american chapter of the association for computational linguistics: Human language technologies (volume 1: Long papers)*, pages 5364–5376, 2024.

A Algorithmic Implementation Details

To provide a clearer understanding of the execution flow of the ISE framework, we detail the step-by-step procedure of the co-evolutionary mechanism in Algorithm 1. The algorithm outlines an EM-style optimization loop consisting of the E-step (Consensus-Driven Pseudo-Label Construction) and the M-step (Error-Driven Instruction Evolution). Specifically, it alternates between constructing stable pseudo-labels through multi-expert consensus and fusion, and rewriting perspective-specific instructions based on expert-specific error patterns. This process continues until all experts satisfy the convergence criteria under the Relaxed Monotonicity Constraint based guarded update mechanism.

Algorithm 1 Co-evolutionary Algorithm for Instruction Optimization

Input: Unlabeled Corpus $\mathcal{D}$, Initial Instructions $\mathcal{P}^{(0)} = \{\mathcal{P}_{con}^{(0)}, \mathcal{P}_{dec}^{(0)}, \mathcal{P}_{ins}^{(0)}\}$, Margin δ, Convergence Threshold ϵ, Patience n
Output: Optimized Instruction Set $\mathcal{P}^*$
Initialize: Experts $\mathcal{K} = \{con, dec, ins\}$, Active Experts $\mathcal{A} \leftarrow \mathcal{K}$, Iteration $t \leftarrow 0$, $\mathcal{S}_{pseudo}^{(-1)} \leftarrow \emptyset$
while $\mathcal{A} \neq \emptyset$ **do**
  // E-Step: Consensus-Driven Pseudo-Label Construction
  for each expert $k \in \mathcal{K}$ **in parallel do**
    $\hat{\mathcal{E}}_k^{(t)} \leftarrow \text{LLM}(\mathcal{D}, \mathcal{P}_k^{(t)})$ // Independent extraction Eq. (2)
  end for
  $\mathcal{P}_{int}^{(t)} \leftarrow \text{Integrate}(\mathcal{P}_{con}^{(t)}, \mathcal{P}_{dec}^{(t)}, \mathcal{P}_{ins}^{(t)})$
  $\tilde{\mathcal{E}}_{int}^{(t)} \leftarrow \mathcal{M}_{fuse}(\mathcal{D}, \mathcal{P}_{int}^{(t)}, \{\hat{\mathcal{E}}_k^{(t)}\}_{k\in\mathcal{K}})$ // Integrated inference Eq. (3)
  $\mathcal{C}^{(t)} \leftarrow \text{MajorityConsensus}\Big(\{\hat{\mathcal{E}}_k^{(t)}\}_{k\in\mathcal{K}} \cup \{\tilde{\mathcal{E}}_{int}^{(t)}\}\Big)$
  $\mathcal{S}_{pseudo}^{(t)} \leftarrow \mathcal{C}^{(t)} \cup \Big(\tilde{\mathcal{E}}_{int}^{(t)} \cap \mathcal{S}_{pseudo}^{(t-1)}\Big)$ // Pseudo-label construction Eq. (4)
  // M-Step: Error-Driven Instruction Evolution
  for each expert $k \in \mathcal{K} \setminus \mathcal{A}$ **do**
    $\mathcal{P}_k^{(t+1)} \leftarrow \mathcal{P}_k^{(t)}$
  end for
  for each expert $k \in \mathcal{A}$ **in parallel do**
    $\mathcal{E}_{FP}^k \leftarrow \hat{\mathcal{E}}_k^{(t)} \setminus \mathcal{S}_{pseudo}^{(t)}$, $\mathcal{E}_{FN}^k \leftarrow \mathcal{S}_{pseudo}^{(t)} \setminus \hat{\mathcal{E}}_k^{(t)}$
    Select $\mathcal{E}_\Delta^k$ as $\mathcal{E}_{FP}^k$ for $\mathcal{M}_{con}$, $\mathcal{E}_{FN}^k$ for $\mathcal{M}_{dec}$, and $\mathcal{E}_{FP}^k \cup \mathcal{E}_{FN}^k$ for $\mathcal{M}_{ins}$
    $\tilde{\mathcal{P}}_k \leftarrow \text{Rewrite}(\mathcal{P}_k^{(t)}, \mathcal{E}_\Delta^k)$ // Expert-specific rewrite Eq. (7)-(9)
    Guarded Update via Relaxed Monotonicity Constraint (RMC):
    $s_{old} \leftarrow F_1(\mathcal{P}_k^{(t)}, \mathcal{S}_{pseudo}^{(t)})$; $s_{new} \leftarrow F_1(\tilde{\mathcal{P}}_k, \mathcal{S}_{pseudo}^{(t)})$
    if $s_{new} \geq s_{old} - \delta$ **then**
      $\mathcal{P}_k^{(t+1)} \leftarrow \tilde{\mathcal{P}}_k$; $s_{acc} \leftarrow s_{new}$ // Accept candidate instruction
    else
      $\mathcal{P}_k^{(t+1)} \leftarrow \mathcal{P}_k^{(t)}$; $s_{acc} \leftarrow s_{old}$ // Reject candidate and retain previous instruction
    end if
    Expert-Level Convergence Check:
    if $|s_{acc} - s_{old}| < \epsilon$ for n consecutive steps **then**
      $\mathcal{A} \leftarrow \mathcal{A} \setminus \{k\}$ // Expert k halts updates
    end if
  end for
  $t \leftarrow t + 1$
end while
return $\mathcal{P}^* = \{\mathcal{P}_{con}^{(t)}, \mathcal{P}_{dec}^{(t)}, \mathcal{P}_{ins}^{(t)}\}$

B Evidence for Latent Knowledge Activation

To provide behavioral evidence for the latent knowledge activation hypothesis, we conduct a post-hoc mention-level probing experiment with Qwen2.5-32B. This analysis is diagnostic and is not involved in instruction optimization or inference; gold annotations are used only after prediction to identify

Vanilla errors and ISE newly corrected entities. We operationalize latent NER knowledge as the model's ability to recognize the semantic status or entity type of a candidate mention under a probing formulation, even when the mention is not correctly extracted under the Vanilla NER instruction. Specifically, we collect Vanilla false negatives and false positives from the 200-instance evaluation subset, provide each candidate mention with its context, and ask the model to select an entity type or choose "None" if the mention should not be extracted. Since the candidate span is provided, this probe evaluates mention-level semantic recognition rather than full NER extraction.

We then examine entities newly extracted correctly by ISE compared with Vanilla and compute the proportion that also receive positive probing evidence, termed the activation rate:

$$\text{Activation Rate} = \frac{\#\{\text{ISE newly correct entities with positive probing evidence}\}}{\#\{\text{ISE newly correct entities}\}}. \tag{12}$$

As shown in Table 4, 70.6% of Vanilla error candidates on GENIA and 54.9% on Restaurant can be correctly resolved under this simplified mention-level probe, indicating that a substantial portion of Vanilla errors correspond to knowledge already accessible to the model. More importantly, 97.9% and 87.5% of ISE's newly correct entities are supported by positive probing evidence. Under our operational definition, these results provide behavioral evidence for latent knowledge activation: ISE mainly recovers entities whose semantic status the model can already recognize when properly queried, suggesting that its gains arise largely from better eliciting internal knowledge rather than from external annotated supervision during optimization or inference.

Table 4: Post-hoc probing results with Qwen2.5-32B. Probing Acc. is computed over Vanilla errors; Activation Rate is the proportion of ISE's newly correct entities with positive probing evidence.

Dataset	Vanilla Errors	Probing Acc.	ISE New Correct	Activation Rate
GENIA	528	70.6% (373 / 528)	190	97.9% (186 / 190)
Restaurant	304	54.9% (167 / 304)	56	87.5% (49 / 56)

C Comparison with Few-shot LoRA Fine-tuning

To further examine the relationship between annotation-free instruction optimization and parameter-efficient supervised adaptation, we compare ISE with LoRA fine-tuning under a small number of gold annotations. Specifically, we fine-tune Qwen2.5-32B with LoRA using 10, 20, 40, and 60 labeled samples, and evaluate on the same benchmark setting as the main experiments.

Table 5: Comparison between few-shot LoRA fine-tuning and annotation-free zero-shot methods using Qwen2.5-32B. Results are macro-F1 scores averaged over three runs.

Method	GENIA	AI	Restaurant	Movie	Avg.
LLM-LoRA-10	51.69 ± 0.99	63.42 ± 0.18	62.65 ± 1.01	55.33 ± 0.68	58.27 ± 0.72
LLM-LoRA-20	51.01 ± 1.33	67.08 ± 0.27	63.39 ± 0.24	57.75 ± 0.09	59.81 ± 0.48
LLM-LoRA-40	51.78 ± 0.59	70.34 ± 0.90	66.92 ± 0.57	53.29 ± 0.70	60.58 ± 0.69
LLM-LoRA-60	58.46 ± 0.70	$\mathbf{78.47} \pm 0.89$	$\mathbf{72.05} \pm 0.42$	$\mathbf{72.71} \pm 0.32$	$\mathbf{70.42} \pm 0.58$
ISE (Ours)	$\mathbf{66.07} \pm 1.23$	69.08 ± 1.76	62.62 ± 1.14	55.82 ± 1.31	63.40 ± 1.36

As shown in Table 5, LoRA becomes effective when gold annotations are available, achieving the best performance with 60 labeled samples. However, ISE remains competitive in the low-resource regime despite using no gold labels or parameter updates. In particular, ISE outperforms LoRA with 10, 20, and 40 labeled samples on average, and still surpasses LoRA-60 on GENIA. These results suggest that ISE is not intended to replace supervised fine-tuning when labeled data and training resources are available; rather, it provides an inference-only alternative for annotation-free scenarios.

D Held-out Inductive Evaluation

Since ISE optimizes instructions on the unlabeled texts of the 200 evaluation instances in the transductive zero-shot setting, the instruction may adapt to sample-specific vocabulary. To assess inductive generalization, we run the full instruction evolution process only on these 200 unlabeled instances, freeze the final integrated instruction after convergence, and directly evaluate it on the remaining disjoint test instances, which are never used for prompt evolution, pseudo-label construction, prompt selection, fusion, or intermediate feedback. For optimization-based baselines, we apply the same protocol by evaluating their frozen optimized prompts on the held-out split without re-optimization.

As shown in Table 6, ISE shows a moderate F1 drop on GENIA (-4.25), suggesting that a small unlabeled optimization subset may not fully cover long-tail biomedical terminology and diverse entity contexts. On Restaurant, however, ISE improves by 1.11 F1 on the held-out split, indicating stronger transfer in a more regular general-domain setting. Overall, ISE still outperforms the compared baselines on both held-out datasets, demonstrating that the optimized instruction preserves meaningful inductive generalization rather than merely overfitting to the transductive subset.

Table 6: Held-out inductive evaluation on Qwen2.5-32B. The instruction is optimized only on the original 200 unlabeled instances and then frozen for evaluation on the remaining disjoint test instances. Δ denotes the F1 difference between held-out and transductive evaluation.

Dataset	Method	Transductive			Held-out Inductive			Δ F1
		P	R	F1	P	R	F1	
GENIA	Vanilla$_{desc}$	52.66	40.07	42.33	51.37	42.50	41.32	-1.01
	EvoPrompt	61.00	53.34	55.70	57.36	55.84	51.10	-4.60
	ISE	**66.98**	**67.26**	**66.07**	**63.31**	**64.09**	**61.82**	-4.25
Restaurant	Vanilla$_{desc}$	**68.90**	48.17	56.16	66.45	47.00	54.48	-1.68
	EvoPrompt	61.91	57.94	58.68	64.60	58.89	60.95	+2.27
	ISE	63.93	**62.09**	**62.62**	**67.70**	**62.08**	**63.73**	+1.11

E Parameter sensitivity analysis

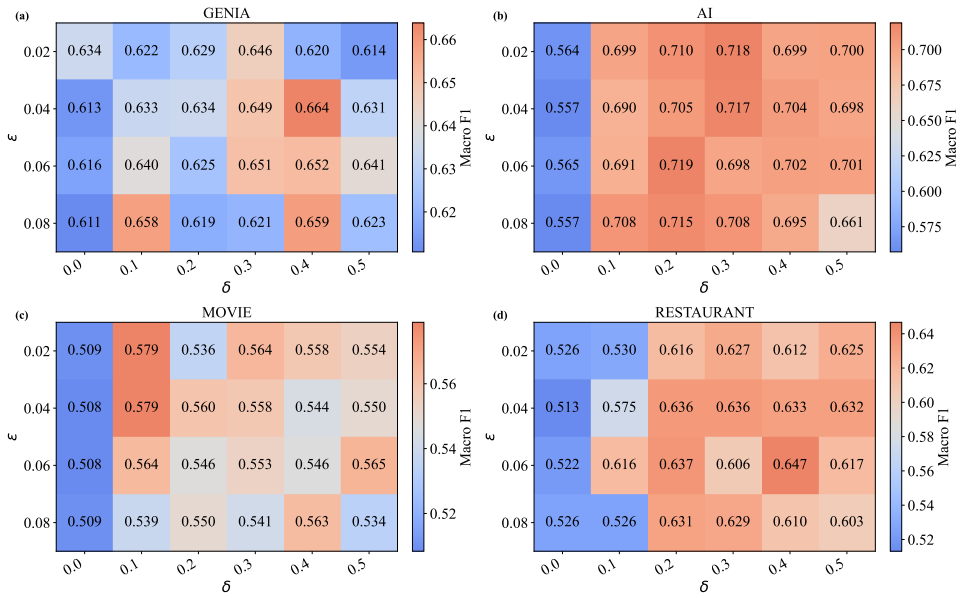


Figure 8: Parameter sensitivity analysis of the guarded update relaxation margin (δ) and convergence threshold (ϵ) across four datasets. The color intensity represents the Macro F1 score.

Appendix Figure 8 presents the sensitivity analysis of the guarded update relaxation margin δ and the convergence threshold ϵ on four datasets, measured by Macro F1. The results show that performance is more sensitive to δ, confirming the necessity of the RMC. When δ is too small (e.g., 0.0 or 0.1), the update process becomes overly conservative, restricting instruction exploration and causing the optimization to fall into local optima. In contrast, an excessively large δ weakens the semantic constraint and leads to performance degradation, indicating a higher risk of semantic drift.

Overall, a moderate δ range of 0.2–0.4 achieves the best and most stable performance by balancing evolutionary flexibility and semantic stability. Meanwhile, ϵ has a relatively mild effect: under suitable δ values, Macro F1 remains largely stable when ϵ varies from 0.02 to 0.08.

F Token Consumption and Runtime Analysis

To quantify the computational overhead of ISE, we report token consumption and wall-clock latency on GENIA and Restaurant using Qwen2.5-32B. All efficiency experiments are conducted on a local serving environment with two NVIDIA A800 GPUs. We deploy the model with vLLM and enable prefix caching, suitable for ISE because the optimized instruction prefix is shared across instances during inference. As shown in Table 7, the optimization stage is an offline one-time cost. GENIA, as a more complex biomedical-domain dataset, requires more tokens and time than Restaurant. Overall, the full optimization process takes 11.4 minutes on GENIA and 7.2 minutes on Restaurant. The E-step consumes more input tokens because it performs multi-perspective extraction and consensus construction, whereas the M-step takes slightly longer due to error-driven instruction rewriting.

Table 7: Token consumption and runtime of the offline optimization phase. Experiments are conducted with Qwen2.5-32B served by vLLM on two A800 GPUs with prefix caching enabled.

Phase	Metric	GENIA	Restaurant
E-Step	Tokens (Input / Output)	3.6M / 67K	2.3M / 49K
	Time	4.7 mins	2.9 mins
M-Step	Tokens (Input / Output)	3.0M / 0.2M	2.0M / 0.1M
	Time	6.7 mins	4.3 mins
Total Offline Cost	Total Time	11.4 mins	7.2 mins

For deployment, ISE does not require multi-agent voting or repeated inference. After optimization, the three semantic perspectives are merged into a single instruction, so each test instance requires one API call. As shown in Table 8, compared with the basic LLM baseline, ISE uses longer prompts and incurs higher per-instance latency. However, with vLLM serving and prefix caching, the repeated instruction prefix can be reused across instances, keeping the latency moderate: 5.2 seconds on GENIA and 3.6 seconds on Restaurant. Meanwhile, ISE yields substantial F1 improvements over the basic LLM baseline. These results indicate that ISE trades a manageable increase in prompt length for significantly stronger extraction accuracy without increasing the number of online API calls.

Table 8: Online inference efficiency com parison. Latency is measured under vLLM serving on two A800 GPUs with prefix caching enabled.

Dataset	Method	API Calls	Prompt Tokens	Latency	F1
GENIA	Basic LLM	1	333	1.8 sec	42.33
	ISE	1	2.9K	5.2 sec	66.07
Restaurant	Basic LLM	1	285	1.3 sec	56.16
	ISE	1	1.9K	3.6 sec	62.62

G Detailed Qualitative Analysis

Table 9 gives the prompt-level details behind the qualitative cases in the main paper. The initial instruction contains a broad protein definition, which leads to two errors: type confusion in Case S1 and incomplete recall in Case S2. For Case S1, the Semantic Contrast Expert adds cues that separate

Table 9: Qualitative analysis of the instruction before and after optimization on the GENIA dataset. Blue text denotes expert-specific instruction components introduced by the optimized prompt. Red indicates extraction errors or missed entities, while green indicates corrected or recalled entities.

Panel A. Input Samples to be Processed

Case S1. Several small molecules, an antisense oligonucleotide, and gene therapeutic experts that inhibit NF-kappa b activation have been described.
Gold Annotation (S1): DNA: {"antisense oligonucleotide"}; protein: {"NF-kappa b"}
Case S2. Involvement of the N-terminal region of the human mineralocorticoid receptor hormone-binding domain in agonist and antagonist binding as revealed by a new monoclonal antibody.
Gold Annotation (S2): protein: {"N-terminal region", "human mineralocorticoid receptor hormone-binding domain", "new monoclonal antibody"}

Panel B. Initial Instruction and Outputs

(1) Initialization Instruction (Protein as an Example):
protein: Proteins are large, complex molecules composed of amino acids that perform various functions in organisms, such as structural support, catalysis, and signaling.

Output on S1: protein: "NF-kappa b", "antisense oligonucleotide"; DNA: N/A

Output on S2: protein: "human mineralocorticoid receptor hormone-binding domain" (Missed: "N-terminal region", "new monoclonal antibody")

Panel C. Optimized Instruction and Outputs

(2) Evolved Instruction (Protein as an Example):
1. Expert A (Semantic Contrast): Specializes in distinguishing between easily confused entity types...
- Definitions: ...large, complex molecules... **[Contrast Added]:** Distinguish from DNA: Not involved in coding or gene regulation. ## Entity Example: "Globin gene switching." → Span: "Globin gene" is identified as dna.
2. Expert B (Semantic Decomposition): Specializes in decomposing entity definitions into fine-grained ...
- Definitions: ...large, complex molecules... **[Decomposition Added]:** Includes generalized aspects like receptors, antibodies, transcription factors, heterodimers, and domains. ## Entity Example: "Involvement of the N-terminal region of the human..." → Span: "N-terminal region" is identified as protein.
3. Expert C (Exemplar Inference): Specializes in inferring entity boundaries and labels from annotation ...
Definitions: ...large, complex molecules... **[Instantiation Added]:** Uses representative annotation examples to guide span detection and type assignment. ## Annotation Example: "The T cell tyrosine kinase ZAP-70..." → Span: "kinase ZAP-70" is identified as protein.

Output on S1: protein: "NF-kappa b"; DNA: "antisense oligonucleotide"

Output on S2: protein: "N-terminal region", "human mineralocorticoid receptor hormone-binding domain", "new monoclonal antibody"

proteins from nucleic-acid entities, allowing the model to remove the protein prediction "antisense oligonucleotide" and recover it as DNA. For Case S2, the Semantic Decomposition Expert expands the protein category to include components such as regions, domains, receptors, and antibodies, while the Semantic Instantiation Expert provides annotation-style examples for boundary decisions. These examples demonstrate how ISE converts generic label descriptions into operational instructions that improve boundary discrimination, semantic coverage, and span-level extraction accuracy.

H Complete Main Experimental Results

Appendix Table 10 reports macro Precision, Recall, and F1 with standard deviations for each dataset and backbone. Overall, ISE achieves the best average F1 across LLM backbones and domains, consistent with the main results. The Precision–Recall breakdown shows that ISE improves zero-shot NER not by biasing toward overly conservative or aggressive extraction, but by maintaining a stronger balance: competing methods may achieve high Precision with low Recall or improve Recall at the cost of more false positives. These results suggest that multi-perspective Semantic Elaboration enhances semantic coverage while sharpening decision boundaries with reliable label-grounded examples.

Table 10: Complete main experimental results with macro Precision, Recall, and F1. Standard deviations are reported only for F1 scores. Best and second-best scores for each metric within each LLM backbone are marked in **bold** and underlined, respectively.

Methods	GENIA			AI			Restaurant			Movie		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Non-LLM Baselines												
UIE-base-en	25.37	27.35	$25.24_{\pm 0.26}$	37.28	51.35	$33.64_{\pm 0.54}$	71.32	82.42	$76.22_{\pm 1.32}$	31.59	41.23	$33.91_{\pm 0.47}$
GLiNER-large	39.25	61.69	$39.72_{\pm 0.66}$	57.38	71.88	$62.23_{\pm 1.19}$	46.04	41.77	$41.16_{\pm 0.98}$	31.48	40.40	$33.42_{\pm 0.44}$
Backbone: Qwen2.5-32B												
Vanilla	55.12	28.38	$35.40_{\pm 1.07}$	59.78	55.36	$54.44_{\pm 0.96}$	63.39	35.64	$45.10_{\pm 0.31}$	48.52	50.40	$46.94_{\pm 1.04}$
Vanilla$_{desc}$	52.66	40.07	$42.33_{\pm 0.56}$	70.25	63.72	$63.03_{\pm 0.41}$	**68.90**	48.17	$56.16_{\pm 0.76}$	53.69	49.54	$50.76_{\pm 0.86}$
Decomposed-QA	60.00	40.79	$44.88_{\pm 0.81}$	53.42	44.65	$46.39_{\pm 0.01}$	55.26	36.45	$43.12_{\pm 0.15}$	**60.40**	21.93	$27.89_{\pm 0.07}$
SALLM	48.64	41.42	$42.63_{\pm 1.18}$	62.81	46.83	$47.89_{\pm 0.72}$	60.49	38.62	$44.43_{\pm 0.39}$	56.97	29.37	$35.13_{\pm 0.16}$
KGB-Retrieval	55.01	47.30	$50.61_{\pm 0.10}$	69.39	52.66	$57.10_{\pm 0.16}$	59.31	**67.53**	$62.60_{\pm 0.09}$	51.61	57.38	$53.39_{\pm 0.10}$
SILLM	64.66	41.87	$50.83_{\pm 0.77}$	67.30	59.10	$60.12_{\pm 0.49}$	60.75	44.62	$51.45_{\pm 0.90}$	51.38	52.03	$48.90_{\pm 0.86}$
EvoPrompt	61.00	53.34	$55.70_{\pm 0.98}$	**76.32**	64.38	$65.27_{\pm 0.54}$	61.91	57.94	$58.68_{\pm 1.59}$	51.30	50.28	$49.02_{\pm 0.36}$
ISE	**66.98**	**67.26**	$\mathbf{66.07}_{\pm 1.23}$	70.05	**71.03**	$\mathbf{69.08}_{\pm 1.76}$	63.93	62.09	$\mathbf{62.62}_{\pm 1.14}$	54.55	**60.31**	$\mathbf{55.82}_{\pm 1.31}$
Backbone: Gemma3-27B												
Vanilla	41.57	52.61	$42.72_{\pm 0.44}$	49.69	55.11	$50.48_{\pm 0.48}$	49.56	41.83	$44.25_{\pm 0.60}$	41.20	47.39	$41.95_{\pm 0.17}$
Vanilla$_{desc}$	39.94	39.68	$37.39_{\pm 0.63}$	61.13	58.13	$55.63_{\pm 0.56}$	59.92	53.17	$54.88_{\pm 0.54}$	47.15	57.08	$50.36_{\pm 0.89}$
Decomposed-QA	45.41	47.03	$44.54_{\pm 0.63}$	50.01	59.52	$52.79_{\pm 0.19}$	51.49	48.69	$48.13_{\pm 0.20}$	42.95	50.54	$44.03_{\pm 0.14}$
SALLM	45.19	50.36	$46.75_{\pm 0.92}$	48.80	59.61	$52.28_{\pm 0.61}$	50.07	48.50	$47.76_{\pm 0.35}$	40.62	47.93	$41.70_{\pm 0.52}$
KGB-Retrieval	47.13	45.41	$45.84_{\pm 0.33}$	65.54	68.50	$63.68_{\pm 0.05}$	47.32	**65.79**	$53.87_{\pm 0.22}$	51.93	56.22	$52.47_{\pm 0.38}$
SILLM	51.34	34.22	$38.39_{\pm 0.69}$	66.63	62.80	$62.21_{\pm 0.46}$	**60.29**	49.57	$52.78_{\pm 0.79}$	48.48	50.64	$46.96_{\pm 0.75}$
EvoPrompt	**58.15**	56.77	$54.83_{\pm 0.76}$	**75.99**	60.66	$63.34_{\pm 0.82}$	52.46	61.87	$54.89_{\pm 1.58}$	44.31	48.09	$44.27_{\pm 1.45}$
ISE	56.66	**63.05**	$\mathbf{57.78}_{\pm 0.91}$	65.60	**71.72**	$\mathbf{66.27}_{\pm 1.75}$	59.25	63.75	$\mathbf{61.17}_{\pm 1.03}$	**53.13**	**61.65**	$\mathbf{55.51}_{\pm 1.49}$
Backbone: GPT-3.5-turbo												
Vanilla	44.84	43.24	$41.74_{\pm 0.80}$	58.04	48.10	$48.41_{\pm 3.58}$	65.15	33.47	$42.78_{\pm 0.26}$	47.83	44.81	$43.89_{\pm 1.02}$
Vanilla$_{desc}$	46.18	41.01	$40.79_{\pm 0.55}$	69.70	56.95	$60.75_{\pm 0.08}$	67.42	57.14	$61.21_{\pm 0.46}$	48.77	53.22	$50.11_{\pm 0.77}$
Decomposed-QA	53.07	48.46	$46.56_{\pm 0.74}$	69.70	56.06	$54.32_{\pm 0.04}$	60.68	49.37	$51.73_{\pm 0.18}$	39.74	43.70	$40.91_{\pm 0.10}$
SALLM	49.71	54.78	$51.15_{\pm 1.02}$	54.83	56.20	$52.59_{\pm 0.68}$	49.91	53.18	$49.43_{\pm 0.45}$	47.32	53.05	$49.08_{\pm 0.57}$
KGB-Retrieval	53.94	49.40	$51.22_{\pm 0.32}$	72.86	59.19	$63.21_{\pm 4.91}$	49.91	64.46	$55.60_{\pm 0.94}$	51.50	56.68	$55.79_{\pm 0.50}$
SILLM	40.72	48.11	$42.95_{\pm 0.72}$	70.05	53.67	$63.80_{\pm 0.46}$	61.11	58.73	$60.62_{\pm 0.84}$	53.17	49.01	$50.93_{\pm 0.80}$
EvoPrompt	57.81	55.25	$55.39_{\pm 0.88}$	**75.78**	68.64	$69.74_{\pm 0.66}$	55.35	66.60	$56.46_{\pm 1.14}$	49.50	47.58	$46.24_{\pm 1.05}$
ISE	**58.53**	**68.78**	$\mathbf{61.40}_{\pm 1.65}$	75.30	**73.50**	$\mathbf{73.94}_{\pm 2.55}$	**67.68**	**71.44**	$\mathbf{69.20}_{\pm 0.79}$	**55.59**	**65.49**	$\mathbf{57.98}_{\pm 1.57}$

I Detailed Ablation Results

In this section, we present the granular performance breakdown of the ablation experiments across the GENIA, AI, Restaurant, and Movie datasets (Appendix Table 11). These domain-specific metrics naturally exhibit minor contextual variations but closely corroborate our aggregated findings. Specifically, the removal of **Semantic Decomposition** generally leads to notable Recall degradation across the datasets, while omitting **Semantic Contrast** hurts Precision in most domains. Furthermore, the detailed results clearly illuminate the balancing mechanism of the **Fusion Expert**. Excluding Fusion consistently triggers a stark trade-off across all domains: a noticeable increase in Precision at the cost of a severe drop in Recall. Although this strict filtering behavior happens to yield a slightly higher F1-score in the Restaurant dataset, it disrupts the balance between comprehensiveness and accuracy. Ultimately, the Fusion expert remains essential for harmonizing these conflicting signals, preventing excessive recall loss, and ensuring stable performance across diverse domains.

Table 11: Detailed ablation results by dataset. We report the Precision (P), Recall (R), and F1-score for the ISE framework and its ablated variants on the GENIA, AI, Restaurant, and Movie benchmarks. The best results for each metric are marked in **bold**, and the second-best are underlined.

Methods	GENIA			AI			Restaurant			Movie		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
ISE	66.98	**67.26**	**66.07**	70.05	**71.03**	69.08	63.93	**62.09**	62.62	54.55	**60.31**	55.82
w/o Contrast	62.83	62.57	61.13	67.61	64.32	64.05	65.54	60.12	62.24	54.01	59.44	55.37
w/o Decomposition	64.03	59.49	60.14	68.17	64.93	65.22	63.86	57.43	59.78	55.94	59.42	**55.86**
w/o Instantiation	66.06	63.09	63.35	71.76	69.59	**69.37**	66.27	61.30	62.54	56.54	58.14	54.32
w/o Fusion	**70.12**	59.70	61.54	**73.14**	64.37	65.91	**69.07**	61.09	**64.52**	**56.57**	53.38	54.33

J Evolutionary Dynamics across Model Scales

To further verify the robustness of the ISE co-evolutionary mechanism, Appendix Figure 9 reports the complete iteration trajectories on GENIA, AI, Movie, and Restaurant using Qwen2.5 models at three scales: 7B, 14B, and 32B. The results show a clear scale-dependent performance ceiling: all models benefit from instruction elaboration, while larger models consistently achieve higher final performance. For example, on the AI dataset, the converged scores increase from 49.16% for 7B to 64.30% for 14B and 71.44% for 32B. Meanwhile, the optimization process converges rapidly across datasets and model sizes, with most gains appearing within the first three iterations; on GENIA, the 32B model improves from 50.61% at iteration 2 to 65.21% at iteration 3.

The trajectories also reveal clear domain sensitivity. Specialized and knowledge-intensive domains, such as GENIA and AI, benefit more from model scaling than general domains. The performance gap between 7B and 32B is relatively small on Movie and Restaurant, averaging 5.14%, but much larger on GENIA and AI, averaging 20.14%. In particular, on GENIA, the 32B model exhibits strong elicitation elasticity, improving from 42.33% to 67.36%, which suggests that larger models can better exploit decomposed concepts and contrastive constraints.

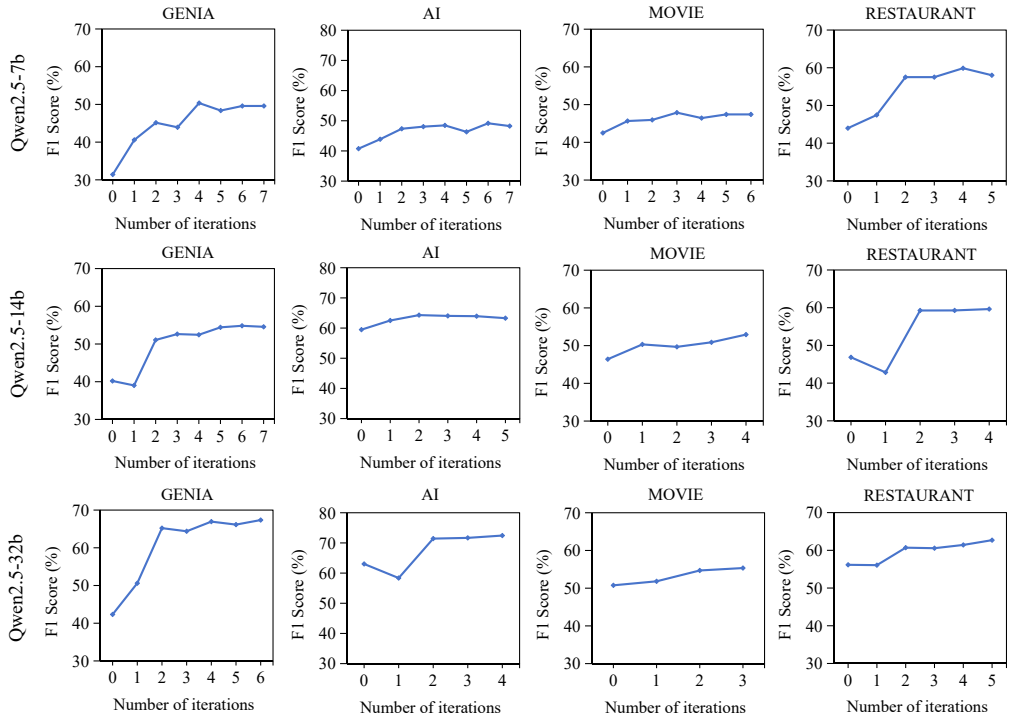


Figure 9: Evolutionary Dynamics Across Model Scales. Iteration trajectories of Qwen2.5 models (7B, 14B, 32B) demonstrate the robustness of the ISE framework across varying model capacities.

K Complete Results of Instruction Initialization Sensitivity

Appendix Table 12 details the performance of the three initialization strategies across all four datasets. The results highlight two critical strengths of the ISE framework: consistent enhancement and strong convergence. Regardless of the starting strategy, ISE steadily elevates F1-scores across all domains. Most notably, the iterative evolution drastically mitigates initial performance disparities. For instance, in the GENIA and Restaurant datasets, initial performance gaps of up to 17 points across strategies are narrowed to a mere 1–3 points post-evolution. This underscores that the framework effectively reduces dependency on meticulously crafted initial prompts, robustly converging to high-quality instructions even from suboptimal starting states.

Table 12: Complete Impact of Instruction Initialization on ISE Performance. We report F1-scores before (Initial) and after (Evolved) the ISE iterative process across three different initialization strategies for all four datasets. The results show consistent convergence across varied domains.

Strategy	GENIA		AI		Restaurant		Movie	
	Initial	Evolved	Initial	Evolved	Initial	Evolved	Initial	Evolved
No Description (Init A)	35.40	65.02	54.44	66.93	45.10	60.39	46.94	50.98
LLM-Generated (Init B)	52.41	64.94	55.08	68.28	38.31	63.44	42.85	50.50
Guideline-based (Init C)	42.33	66.07	63.03	69.08	56.16	62.62	50.76	55.82

L Full Analysis of Relaxed Monotonicity Constraint

In the main text, we evaluated the impact of the RMC on the GENIA and Restaurant datasets. We present the comprehensive results across all four benchmark datasets (GENIA, AI, Restaurant, and Movie) in Appendix Figure 10. The extended results corroborate the robustness of setting δ within the range of $[0.2, 0.4]$. However, further analysis reveals that this threshold range does not universally guarantee maximum performance gains; for instance, it proves to be a suboptimal setting on the Movie dataset (where $\delta = 0.1$ is optimal). Moreover, the benefits derived from the relaxed constraint exhibit notable dataset variance. Specifically, the performance improvements on the AI and Restaurant datasets are substantially more pronounced than those on the GENIA and Movie datasets (yielding an average maximum F1-score improvement of 14.12% versus 6.05%).

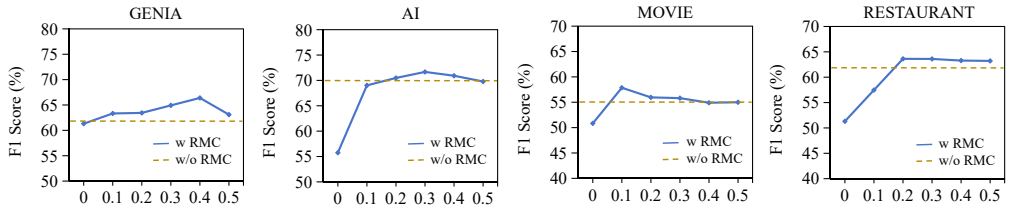


Figure 10: Impact of Relaxed Monotonicity Constraint. Performance (F1-score) variations across different thresholds (δ) versus unconstrained evolution.

M Detailed Analysis of Iterative Data Scale

Appendix Figure 11 presents the complete performance trajectories across all four benchmark datasets (GENIA, AI, Movie, and Restaurant) with respect to the unlabeled data volume used during evolution.

Consistent with the findings in Section 4.8, all datasets exhibit a "rapid rise and early saturation" trend. The detailed results further confirm the framework's data efficiency. Specifically, on the GENIA dataset, using just 10% of the data yields an F1-score of 58.27%, demonstrating a substantial absolute improvement of 15.94% over the zero-usage baseline (42.33%). On the AI dataset, performance stabilizes at roughly 30% data usage, achieving an F1-score of 70.55%, which is highly comparable to the full dataset performance (69.08%). Similarly, the Movie dataset shows remarkably early saturation, where merely 10% data usage achieves an F1-score of 55.53%, performing nearly on par with the full dataset (55.82%). These comprehensive results validate that the ISE framework does not rely on massive-scale unlabeled corpora to function; instead, it efficiently extracts high-value semantic signals from limited samples to drive instruction evolution.

N Broader Impact

This work advances the democratization of specialized information extraction by enabling domain experts to elicit high-performance NER capabilities from LLMs without manual labeling, which is particularly valuable for clinical research and data governance in low-resource scenarios. While ISE significantly reduces human annotation costs and demonstrates high data efficiency, its iterative co-evolutionary mechanism poses a risk of reinforcing inherent foundation model biases through pseudo-labeling. Additionally, the multi-expert optimization process increases computational energy

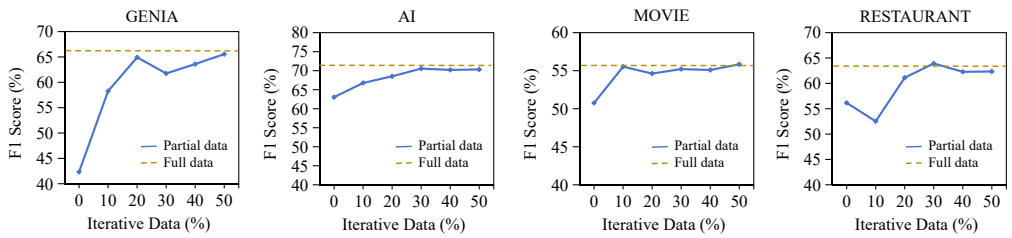


Figure 11: Impact of unlabeled data scale on zero-shot performance across four benchmark datasets. Dashed lines indicate the baseline performance using 100% of the data.

consumption compared to vanilla prompting. We recommend that practitioners implement human-in-the-loop audits to verify the fairness of evolved instructions and consider the trade-off between extraction performance and the environmental footprint of intensive token usage.

O Instruction for ISE

The instructions used in this study are presented as follows. The initial instruction for system initialization is provided in Appendix Table 12. Next, we describe the instructions used for expert evolution during the M-step, including the Contrast Expert evolution instruction (Appendix Table 13), the Decomposition Expert evolution instruction (Appendix Table 14), and the Instantiation Expert evolution instruction (Appendix Table 15). Finally, the Fusion Expert instructions, which aggregate the entity semantic cognition from multiple semantic elaboration experts, are shown in Appendix Tables 16 and 17, corresponding to the training and inference/testing stages, respectively. Note that the parts enclosed in "<>" in each instruction are placeholders that should be filled during use.

Initial Instruction for All Experts

You are an excellent linguist. Given the following entity types, please complete the entity extraction task and output the results as required:

Entities List:
<*Entity and Definitions*>

Notes:
Output entity extraction results according to the following format:
"'json[{"type": "...", "entity": "..."}, {"type": "...", "entity": "..."}, ...]"'
- If there are no valid entities, output an empty list "'json[]"'.

Target Text:
<*Target Text*>

Figure 12: Initial Instruction for All Experts.

Instruction for Semantic Contrast Expert

You are an expert in entity definition refinement. Your task is to incrementally optimize entity definitions based on confusion cases and provide illustrative annotation examples.

Input Data:
Current Entity Definitions:
<*Entity and Definitions*>

Confusion Examples (complete entity annotations with error information):
<*Confusion Examples*>

Task Requirements:
1. Analyze the confusion examples. Each example shows the COMPLETE entity list for a text, along with error information indicating which entities were confused.
2. Refine the definitions INCREMENTALLY and SAFELY:
- CONCISENESS: Keep definitions extremely concise. Do not create bloated definitions with endless appended clauses.
- ANTI-OVERFITTING: Do not overfit to the specific confusion examples. The revised definition must remain universally applicable to standard cases.
- ELEGANT MERGING: If adding distinguishing features, merge them elegantly into the core definition. Only use "Distinguish from:" if absolutely necessary.
- NO NEW CONFLICTS: Ensure that your adjustments to separate two entities do not accidentally overlap with the definition of a THIRD entity.
3. Select and update Annotation Examples based on the Confusion Examples:
- Select examples that best illustrate the corrected boundaries.
- STABILITY: Preserve existing basic/classic examples that illustrate standard cases. Only replace an example if the new one provides a significantly higher diversity gain for distinguishing boundaries.
- All newly added or updated examples must use the exact Context and Span from the Confusion Examples. Entities for each context must be complete. For each span, keep only one of the most correct span label, and it must come from the true labels in the given examples.
- Maintain a highly curated set of examples. The 'Context' text MUST BE UNIQUE.
4. Output ONLY the refined definitions and the selected examples in the specified format below. DO NOT include analysis steps or explanations.

Critical Output Rules:
- MUST include all entity labels in the definitions section: <*Target Entity List*>
- If "Confusion Examples" is empty or "None", return the current definitions and existing examples unchanged.
- Output format MUST consist of exactly two sections: "## Entity Definitions" and "## Entity Examples".

Output Format:
Entity Definitions
1. [Entity Name]: [Core Definition integrated with distinguishing boundaries if necessary]. [Optional: Distinguish from [Confusing Entity]: [Brief feature].]
2. [Entity Name]: [Core Definition integrated with distinguishing boundaries if necessary]. [Optional: Distinguish from [Confusing Entity]: [Brief feature].]

Entity Examples
- 1. "[Context Text]" -> Span: "[Target Span]" is identified as [Entity Name]; Span: "[Target Span]" is identified as [Entity Name]; ...

Figure 13: Instruction for Semantic Contrast Expert.

Instruction for Semantic Decomposition Expert

You are an expert in entity semantic decomposition. Your task is to incrementally expand entity definitions by identifying fine-grained sub-semantic categories from recall error cases.

Input Data:
Current Entity Definitions:
<*Entity and Definitions*>

Recall Error Examples (complete entity annotations with error information):
<*Recall Error Examples*>

Task Requirements:
1. Analyze the recall error examples. Each example shows the COMPLETE entity list for a text, along with error information indicating which entities were missed.
2. Refine the definitions INCREMENTALLY and SAFELY:
- CONFLICT AVOIDANCE (CRITICAL): Before adding a sub-semantic category, verify that it does not logically overlap with the definitions of OTHER existing entities. Do not introduce ambiguity.
- CONCISENESS: Merge sub-semantics elegantly. Avoid long, verbose lists.
- ANTI-OVERFITTING: Ensure the abstracted dimensions represent a true class of entities, not just a quirk of the provided examples.
3. Select and update Annotation Examples based on the Recall Error Examples:
- Select examples that best illustrate the newly added sub-semantic categories.
- STABILITY: Preserve core example contexts when useful; if a preserved context appears in Recall Error Examples, update its spans/labels to the complete true annotations there. Replace a context only to better cover new sub-semantics or clear diversity gain.
- All newly added or updated examples must use the exact Context and Span from the Recall Error Examples. Entities for each context must be complete. For each span, keep only one of the most correct span label, and it must come from the true labels in the given examples.
- Maintain a highly curated set of examples. The ‘Context‘ text MUST BE UNIQUE.
4. Output ONLY the refined definitions and the selected examples in the specified format below. DO NOT include analysis steps or explanations.

Critical Output Rules:
- MUST include all entity labels in the definitions section: <*Target Entity List*>
- If “Recall Error Examples” is empty or “None”, return the current definitions and existing examples unchanged.
- Output format MUST consist of exactly two sections: “## Entity Definitions” and “## Entity Examples”.

Output Format:
Entity Definitions
1. [Entity Name]: [Core Definition]. [If sub-semantics found: Includes generalized aspects like [Category 1] and [Category 2].]
2. [Entity Name]: [Core Definition]. [If sub-semantics found: Includes generalized aspects like [Category 1] and [Category 2].]

Entity Examples
- 1. “[Context Text]” -> Span: “[Target Span]” is identified as [Entity Name]; Span: “[Target Span]” is identified as [Entity Name]; ...

Figure 14: Instruction for Semantic Decomposition Expert.

Instruction for Semantic Instantiation Expert

You are an expert in few-shot prompt optimization. Your task is to incrementally refine entity extraction instructions by selecting and updating high-quality annotation examples.

Input Data:
Current Entity Definitions:
<*Entity and Definitions*>

Candidate Examples:
<*Candidate Examples*>

Task Requirements:
1. Verify the Candidate Examples based strictly on the Current Entity Definitions. An example is valid ONLY IF annotated labels strictly match the entity definitions and unambiguous contextual evidence exists.
2. Refine the Example Set INCREMENTALLY and SAFELY:
- ANCHOR STABILITY (CRITICAL): Keep fundamental, standard examples intact. Do NOT drastically replace the entire example set.
- DIVERSITY GAIN: Only add or substitute an existing example if the new candidate introduces a distinct, high-value edge case (e.g., implicit expressions, previously confused boundaries) without duplicating existing structural triggers.
- NO REDUNDANCY: Keep only the single most representative example for a specific contextual scenario.
- Maintain a highly curated set of examples. The ‘Context‘ text MUST BE UNIQUE.
3. Output ONLY the unified instruction in the specified format below. DO NOT include analysis steps, intermediate verification steps, or explanations.

Critical Output Rules:
- MUST include all entity labels in the definitions section: <*Target Entity List*>
- The output MUST consist of exactly two sections: “## Entity Definitions” and “## Annotation Examples”.
- Do NOT put examples inside the Entity Definitions section. List all verified examples together at the end.

Output Format:
Entity Definitions
1. [Entity Name]: [Definition]
2. [Entity Name]: [Definition]
...

Annotation Examples
- 1. “[Context Text]” -> Answer: [{"type": "xx", "entity": "x"}, {"type": "xx", "entity": "x"}, ...]

Figure 15: Instruction for Semantic Instantiation Expert.

Instruction for Multi-expert Fusion For Train

Your goal is to complete information extraction tasks based on multi-expert definitions and their predictions.

Experts Definitions And Their Predictions:
1. Expert A (Comparative Analysis): Specializes in distinguishing between easily confused entity types. Prioritize Expert A's type assignment when two experts disagree on the label for the same span.
- Definitions: *<Definitions Expert A>*
- Predictions: *<Predictions Expert A>*
2. Expert B (Semantic Decomposition): Specializes in recalling entities that are easily missed due to broad or implicit semantics. Prioritize Expert B's recall when deciding whether a singleton prediction is a genuine entity.
- Definitions: *<Definitions Expert B>*
- Predictions: *<Predictions Expert B>*
3. Expert C (Exemplar Inference): Specializes in context-based identification using annotated examples. Use Expert C as a tiebreaker when A and B conflict, or to verify ambiguous spans.
- Definitions: *<Definitions Expert C>*
- Predictions: *<Predictions Expert C>*

Target Text:
<Target Text>

Final Output Format:
Output only the JSON list. Do not include explanation.
“‘json[{"type": "xx", "entity": "x"}, ...]“‘
- If there are no valid entities, output: “‘json[]“‘.

Figure 16: Instruction for Multi-expert Fusion.

Instruction for Multi-expert Fusion For Inference

Your goal is to complete information extraction tasks based on multi-expert perspectives.

The Three Experts:
1. Expert A (Comparative Analysis): Specializes in distinguishing between easily confused entity types. Prioritize Expert A's type assignment when two experts disagree on the label for the same span.
- Definitions: *<Definitions Expert A>*
2. Expert B (Semantic Decomposition): Specializes in recalling entities that are easily missed due to broad or implicit semantics. Prioritize Expert B's recall when deciding whether a singleton prediction is a genuine entity.
- Definitions: *<Definitions Expert B>*
3. Expert C (Exemplar Inference): Specializes in context-based identification using annotated examples. Use Expert C as a tiebreaker when A and B conflict, or to verify ambiguous spans.
- Definitions: *<Definitions Expert C>*

Target Text:
<Target Text>

Final Output Format:
Output only the JSON list. Do not include explanation.
“‘json[{"type": "xx", "entity": "x"}, ...]“‘
- If there are no valid entities, output: “‘json[]“‘.

Figure 17: Instruction for Multi-expert Fusion.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the proposed ISE framework, the co-evolutionary algorithm, the RMC mechanism, and explicitly list the three main contributions which align with the evaluated scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: The paper includes a dedicated "Limitations" section discussing constraints regarding intrinsic parametric knowledge, increased token consumption and inference latency, and the marginal risk of error propagation due to base model biases.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: This paper proposes an empirical instruction optimization framework and does not introduce formal theoretical theorems or mathematical proofs.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

Justification: The paper provides the complete pseudocode for the co-evolutionary algorithm (Algorithm 1), details on the dataset splits, main models, and their parameter counts (Section 4.1), and includes the exact prompt templates used for all experts in Appendix O.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The appendix provides the pseudocode of the proposed algorithm. In addition, Section 4.1 details the data and sampling methods used to aid reproducibility. All experiments in this project are based on open-source datasets. The code for the main and analytical experiments is currently being organized and will be released on GitHub subsequently.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.

- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the datasets and evaluation splits, zero-shot/transductive and held-out protocols, model backbones, baselines, metrics, iteration settings, key hyperparameters, and clarifies that ISE uses prompt-based co-evolutionary optimization rather than gradient-based training.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The main experiments were all conducted multiple times, and the mean and variance are reported. Detailed variance information is provided in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: The paper reports the compute setup and runtime details, including use of local vLLM serving with two NVIDIA A800 GPUs, prefix caching, token consumption, offline optimization time, online latency, and API-call costs for reproducing the experiments.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The research conforms to the NeurIPS Code of Ethics, focusing on standard NLP instruction optimization methodologies without violating ethical guidelines.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We provide a concise "Broader Impact" Appendix section addressing the framework's benefits for low-resource domain extraction and clinical data governance, alongside potential risks related to bias amplification and computational costs.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer:[N/A]

    Justification: The paper introduces a prompting and instruction evolution framework for existing LLMs and does not release high-risk foundational models or sensitive scraped datasets that require explicit safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All used assets, including the datasets (GENIA, AI, Restaurant, and Movie) and the Large Language Models (Qwen2.5, Gemma3, and GPT-3.5-Turbo), are properly credited via citations. We have verified that the datasets are open-access. Regarding the models: Qwen2.5 and Gemma3 are utilized under their respective open-weight license agreements (e.g., Apache 2.0 or specific Model License Agreement); GPT-3.5-Turbo is accessed via OpenAI's API in strict accordance with their Terms of Service.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The research relies on existing benchmark datasets and pre-trained LLMs, and does not introduce new datasets or foundational models as standalone assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The research relies on automated instruction evolution with LLMs and does not involve crowdsourcing or human subject experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Not applicable, as the research does not involve human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: The core methodology fundamentally relies on LLMs for the co-evolutionary instruction optimization process, which is explicitly detailed in the abstract, methodology, and algorithmic implementation sections.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.