

Beyond Known Event Types: Open-Domain Event Detection via Ontology-Guided Grounding–Discovery Resolution

Yuhan Hu^{1*} Yupeng Qi^{1*} Yongxi Luo² Chen Su² Shan Huang² Ziyu Lyu^{1†}

¹School of Cyber Science and Technology, Shenzhen Campus of Sun Yat-sen University

²Didi Chuxing Co. Ltd.

{huyh76,qiyp7}@mail2.sysu.edu.cn lvzy7@mail.sysu.edu.cn

Abstract

Open-domain event detection aims to identify known event types while discovering previously unseen ones from unstructured documents. In practice, event detection systems often operate with an evolving event set rather than a complete type inventory. Under this setting, existing zero-shot event detection methods can fail in two ways when the event type is absent from the predefined set: the model may either miss the event or assign it to a plausible but wrong known type, hiding the need for discovery. To address this problem, we propose OGED, an Ontology-Guided open-domain Event Detection framework for grounding–discovery resolution, i.e., deciding whether an event mention should be assigned to an existing type or should induce a new type. OGED organizes the available event types into a hierarchical search structure, uses cluster-level scheduler and adaptive planning to keep known-type assignment and new-type discovery jointly active, and applies retrieval-gated verification together with conflict selection to resolve competing assignments before output. Experiments on a unified benchmark with 564 event types show that OGED outperforms strong baselines on both known-type detection and unknown-type discovery. Code is available at <https://github.com/chenzhouli/OGED>.

1 Introduction

Open-domain event detection is a core task in open-domain information extraction. It aims to extract event mentions, including both event triggers and event types, from unstructured documents, and supports a wide range of downstream applications including knowledge graph construction (Yao et al., 2023), information retrieval (Zhang et al., 2024), knowledge-intensive question answering (Liu et al., 2025), and domain-specific extraction over medical (Ma et al., 2023a) and scientific (Lu et al.,

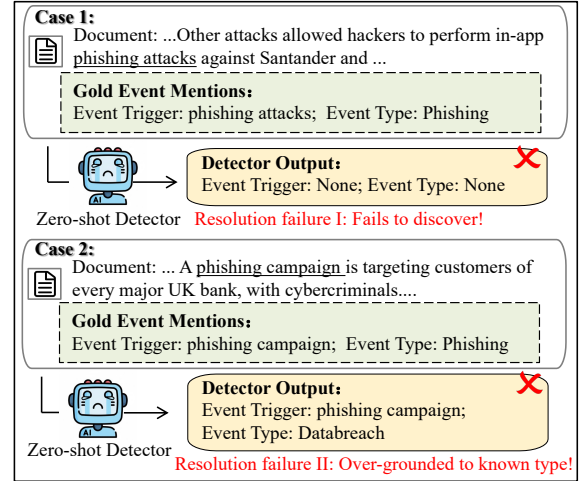


Figure 1: When the event type is absent from the predefined set, zero-shot detectors either miss the event or ground it to a plausible but wrong known type. Both reflect failures of grounding–discovery resolution, where the model must decide whether to assign the mention to an existing type or induce a new one.

2026) text. In practice, real systems usually operate with an evolving set of event types. This setting is common in dynamic domains such as cyber threat analysis, where resources such as MITRE ATT&CK (The MITRE Corporation, 2026) organize adversary behavior into tactics, techniques, and sub-techniques, yet incident reports still introduce emerging behaviors that are not covered by the current ontology of adversary behaviors.

Despite the need, most existing LLM-based event detection methods formulate the task as zero-shot extraction over a predefined set of event type definitions (Srivastava et al., 2025; Wang et al., 2023; Sainz et al., 2024). This formulation is limited when the event types are evolving. In such cases, the model may exhibit two failures. As shown in Figure 1-Case 1, the zero-shot detector fails to discover the new type “Phishing” from the trigger “phishing attacks”. More subtly, it may as-

*Equal contribution.

†Corresponding author.

sign the trigger to a plausible but wrong known type, as in Figure 1-Case 2, where “phishing campaign” is mapped to “Databreach” although the correct type is “Phishing”. Besides, existing LLM-based fully open-domain methods extract free-form event types (Lu et al., 2025), which are not aligned with predefined event types that downstream applications rely on, and thus serve offline label construction. The key challenge is therefore to use the predefined event types as useful prior knowledge without grounding every event mention into the existing type set. This motivates a unified framework in which known-type grounding and new-type discovery remain jointly active until the final output is resolved.

In light of these challenges, we study open-domain event detection with an incomplete type set and propose **OGED**, an **O**ntology-**G**uided open-domain **E**vent **D**etection framework for grounding–discovery resolution. Rather than treating known-type assignment and new-type discovery as separate stages, OGED maintains both as competing explanations and resolves them before final output. For efficiency, OGED organizes the predefined event type set into a hierarchical search structure built from event definitions, allowing the model to focus on a relevant semantic cluster instead of the entire type set at once.

OGED explicitly addresses the two failure modes. For missing new events, a *cluster-based semantic scheduler* prioritizes clusters that are relevant to the document but remain insufficiently resolved, and a *hybrid reasoning planner* decides whether the evidence in the selected cluster should favor known-type assignment or new-type discovery. For over-grounding, a *retrieval-gated verifier* judges whether a proposed new type is genuinely novel or should instead be mapped back to a predefined type. Finally, a *conflict selection* module resolves competing known-type and new-type assignments, allowing a newly induced type to replace a plausible but incorrect known-type label when supported by the document evidence.

In summary, we make three contributions. (1) We formulate a practical open-domain event detection setting in which the available event type set is incomplete and evolving, so the system must determine whether each event mention should be assigned to an existing type or should trigger the induction of a new one. (2) We propose OGED, an ontology-guided framework for grounding–discovery resolution. By structuring

the predefined event type set into a hierarchical ontology, OGED keeps known-type grounding and new-type discovery jointly active within a unified inference process. (3) We conduct experiments on a unified benchmark with 564 event types, showing that OGED improves both known-type detection and unknown-type discovery over zero-shot baselines.

2 Related Work

2.1 Event Detection

Early event detection research explored data-efficient learning strategies such as contrastive learning, prototype learning, and trigger-aware prompting (Zhang et al., 2022; Ma et al., 2023b; Yue et al., 2023). With LLMs, recent work has shifted towards zero-shot and instruction-based inference (Gao et al., 2023), improving extraction through multi-turn QA, retrieval, debate, code generation, or adaptation with definitions, guidelines, and synthetic supervision (Wei et al., 2023; Yao et al., 2023; Wang and Huang, 2024; Guo et al., 2026; Cai et al., 2024; Sainz et al., 2024; Fuente et al., 2025). Multi-agent and multi-stage methods such as SNaRe (Parekh et al., 2025a) and DiCoRe (Parekh et al., 2025b) further reduce errors in direct LLM inference. These methods improve detection under a predefined event inventory, while OGED targets inference-time grounding–discovery resolution under an incomplete and evolving type set.

For open-domain event extraction, prior work has explored event type discovery and ontology expansion from collections of event instances (Li et al., 2022; Cao et al., 2023), distant supervision with external resources (Araki and Mitamura, 2018), and scalable semantic evaluation as in SEOE (Lu et al., 2025). Existing type-discovery methods typically assume candidate or gold event mentions and focus on inducing types from a collection of event instances. In contrast, OGED operates directly on documents and jointly performs known-type grounding and new-type discovery under a partial seed ontology.

2.2 Schema-guided LLM reasoning

Recent work shows that explicit schemas or ontologies can guide LLM inference by constraining the search space and organizing intermediate reasoning. Examples include schema-activated in-context learning (Chen et al., 2025), ontology-guided re-

verse thinking for knowledge graph question answering (Liu et al., 2025), and structure-guided prompting for table question answering and task-oriented dialogue (Wang et al., 2025; Zhang et al., 2023). OGED adopts this general idea for open-domain event detection, and uses the event ontology to guide search to minimize the semantic gap.

3 Problem Definition

Event Detection. Given an input document $D = [w_1, \dots, w_L]$ as a token sequence of length L , trigger-based event detection aims to extract a set of event mentions $\mathcal{E} = \{(m_i, t_i)\}_{i=1}^N$. Each mention $m_i = (s_i, f_i)$ is a token span ($1 \leq s_i \leq f_i \leq L$) that denotes an event trigger where s_i and f_i denote the start and end indices of the event trigger, respectively, and t_i is the assigned event type. We focus on trigger identification and do not model event arguments in this work.

Types and Definitions. We assume access to a set of event types equipped with textual definitions:

$$\mathcal{O} = \{(t_i, d_i)\}_{i=1}^N,$$

where t_i is an event type name and d_i is its definition. An example of event definition is “The event refers to a ransomware attack targeting systems and data”.

Two Standard Paradigms. Existing LLM-based approaches can be broadly categorized by whether the output type space is constrained by a given event type set.

(i) *Zero-Shot Event Detection (ZSED)*. ZSED extracts mentions by mapping them to a provided type set:

$$\begin{aligned} \mathcal{E} &= \Phi_{\text{LLM}}(D, \mathcal{O}_{\text{in}}), \\ \text{s.t. } \forall (m_i, t_i) \in \mathcal{E}, (t_i, d_i) &\in \mathcal{O}_{\text{in}}. \end{aligned}$$

In this setting, out-of-set events are typically *missed* or *forced-mapped* to the closest in-set type, limiting adaptability in dynamic domains.

(ii) *Open-Domain Event Detection (ODED)*. ODED removes the predefined type constraint and allows the model to generate types from an open vocabulary:

$$(m_i, t_i, d_i) = \Phi_{\text{LLM}}(D), \quad t_i \in \mathcal{V}^*.$$

where $\mathcal{V}^*$ is the model vocabulary. In practice, LLMs often generate free-form type names and descriptions. While flexible, unconstrained type generation commonly yields non-canonical labels,

redundant categories, and weak semantic consistency.

Task Setting: Open-Domain Event Detection with Evolving Event Types. We define open-domain event detection with a seed type set as follows. The model is provided with a seed type-definition set $\mathcal{O}_{\text{seed}}$ as a semantic reference, rather than a closed label space. The model must output (i) event mentions grounded to existing types when appropriate, and (ii) a set of induced novel types with definitions:

$$(\mathcal{E}, \mathcal{O}_{\text{new}}) = \Phi_{\text{LLM}}(D, \mathcal{O}_{\text{seed}}),$$

where $\mathcal{O}_{\text{new}} = \{(t_j^{\text{new}}, d_j^{\text{new}})\}_{j=1}^M$. Accordingly, the final type-definition set is expanded as:

$$\mathcal{O}_{\text{final}} = \mathcal{O}_{\text{seed}} \cup \mathcal{O}_{\text{new}},$$

where t_{new} and d_{new} represent the new event type and its corresponding definition given by the LLM. For each mention $(m_i, t_i) \in \mathcal{E}$, if $(t_i, d_i) \in \mathcal{O}_{\text{seed}}$ the system performs closed-world grounding; otherwise, it generates a new type with its definition $(t_{\text{new}}, d_{\text{new}})$.

Rationale. Real-world applications require both semantic consistency and adaptability. A fixed type set provides stable semantics for downstream reasoning, yet cannot capture emerging events. On the other hand, unconstrained generation offers flexibility but risks contaminating the label space with noise. For example, in “cybersecurity” domains, a seed set may contain standard types such as “Phishing” and “DDoS”. When the document describes file encryption and ransom demands, a purely seed-bounded model may miss the critical “Ransomware” event, whereas a fully open-domain model may produce vague labels, e.g., “Hacking” or overly specific ones, e.g., “File Encryption”, causing category fragmentation. We argue that a practical framework should thus treat the seed set as a semantic anchor while allowing principled expansion when the text provides unsupported evidence.

4 Methodology

4.1 Overview

To enable efficient reasoning under a limited budget, we begin by organizing the seed event type set $\mathcal{O}_{\text{seed}}$ into a hierarchical tree structure $\mathcal{H}_{\text{seed}}$, as illustrated in Fig. 2, where internal nodes represent coarse-grained semantic domains and leaf

nodes correspond to fine-grained event types, each associated with a textual definition. This hierarchical tree supports guided exploration, in which the model, rather than matching every mention against all available types, focuses on a few relevant branches and, when necessary, adds new leaf types to the most compatible branch to obtain an expanded tree $\mathcal{H}_{\text{final}} (\mathcal{H}_{\text{final}} \supseteq \mathcal{H}_{\text{seed}})$.

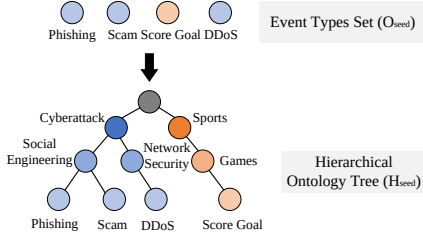


Figure 2: Hierarchical Ontology Tree Construction

The overall framework is shown in Fig. 3. OGED treats open-domain event detection as grounding–discovery resolution. Given a document and an incomplete seed ontology tree, the system must resolve whether each candidate event mention should be grounded to an existing leaf type or should induce a new leaf type. To operationalize this resolution, OGED proceeds in three stages. First, it localizes the grounding space by constructing a document-specific ontology subtree and prioritizing unresolved clusters. Second, it generates competing grounding and discovery candidates through an adaptive planner and cluster-conditioned detector. Third, it verifies proposed new types and resolves conflicts among competing event mentions before committing to the final output. These stages are implemented by five modules as follows.

4.2 Cluster-based Semantic Scheduler

To support grounding–discovery resolution under a limited inference budget, we first restrict reasoning to a compact ontology region. Starting from the initial ontology tree $\mathcal{H}_{\text{seed}}$, we retrieve the top- p relevant leaf types for the document and merge their ancestor paths into a document-specific dynamic search tree. The parent nodes of the leaves define a set of candidate semantic clusters C . As an example shown in Fig. 2, $C = \{\text{Social Engineering}, \text{Network Security}, \text{Games}\}$. Each cluster is associated with a local subtree $\mathcal{H}(c)$ that serves as an atomic search unit.

For each cluster c , we compute an initial rele-

vance score

$$P_0(c) = \text{Reranker}(D, c), \quad P_0(c) \in [0, 1],$$

which measures how well the document matches that semantic region. We then maintain a posterior score $V_k(c)$ at step k that summarizes how well the cluster has already explained the document through accepted candidates:

$$V_k(c) = \max \left(V_{k-1}(c), \max_{e \in \mathcal{E}_k^{\text{cand}}(c)} s(e) \right),$$

where $V_k(c) \in [0, 1]$, $s(e)$ is the candidate confidence defined in Sec. 4.5. Based on these two quantities, we define the semantic gap of cluster c as

$$\mathcal{G}_k(c) = P_0(c) \cdot (1 - V_{k-1}(c)).$$

A high semantic gap indicates that the cluster is relevant to the document but insufficiently explained by current predictions, suggesting that new-type discovery may be needed.

At each iteration, the scheduler selects the next cluster by balancing semantic gap and exploration:

$$c_k^* = \arg \max_{c \in C} \left[\mathcal{G}_k(c) + \lambda \sqrt{\frac{\ln(1+k)}{1+n(c)}} \right],$$

where $n(c)$ is the visit count of cluster c and λ controls the exploration strength. The semantic-gap term favors potential ontology regions, while the exploration term prevents the scheduler from early committing to a narrow set of regions and making the framework more robust to a noisy ontology tree.

4.3 Hybrid Reasoning Planner

Given the selected cluster c_k^* , the planner decides whether the current evidence should favor known-type grounding or new-type discovery. The detection strategy grounds triggers to existing event types within the current ontology, whereas the discovery strategy is invoked when the selected cluster remains unresolved. To achieve this goal, we adopt the cluster semantic gap as the strategy selection rule:

$$\pi_k = \begin{cases} \pi_{\text{dis}} & \text{if } n_{k-1}(c_k^*) \geq 1 \text{ and } \mathcal{G}_k(c_k^*) > \tau, \\ \pi_{\text{det}} & \text{otherwise,} \end{cases}$$

where τ controls when the framework moves from grounding to discovery. In the open-domain setting, each cluster can be visited at most twice: the first visit attempts known-type grounding, and the

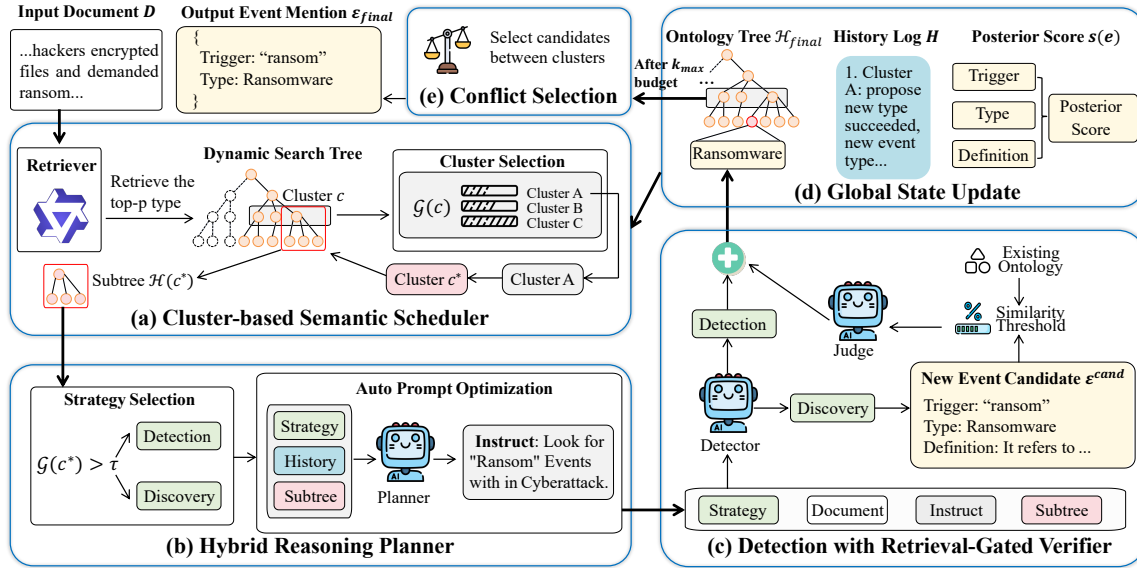


Figure 3: OGED performs ontology-guided grounding-discovery resolution in five phases. (a) A cluster-based scheduler prioritizes target clusters based on semantic gaps. (b) A planner selects between *Detection* and *Discovery* and generates context-aware instructions. (c) A detector extracts event candidates, while a retrieval-gated verifier judges new event type proposals. (d) A global state update strategy maintains the ontology tree, history log, and posterior score. (e) A conflict selection mechanism reviews aggregated candidates and selects the most plausible final mentions.

second is allowed only if the semantic gap remains high after grounding. Thus, discovery is focused on clusters that cannot be sufficiently explained by existing types, while potential conflicts between known-type and new-type candidates are deferred to the conflict selection module in Sec. 4.6.

To improve robustness across iterations, we use an LLM planner to generate step-specific guidance conditioned on the document D , the selected subtree $\mathcal{H}(c_k^*)$, and the history log H of previous decisions:

$$I_k = \text{LLM}_{\text{planner}}(\mathcal{P}(D, \mathcal{H}(c_k^*), H)).$$

Here $\mathcal{P}$ is a structured prompt template. Notably, our planner is inspired by recent advances in automatic prompt optimization (Wang et al., 2024). The resulting instruction I_k encourages the detector to focus on unresolved evidence in the selected cluster, thereby reducing repeated errors and making the transition between assignment and discovery more controlled.

4.4 Detection with Retrieval-Gated Verifier

After obtaining the selected strategy π_k and the generated instruction I_k , we perform generative event detection and discovery within the current cluster c_k^* . When the detector proposes a new event type t_{new} , we apply a retrieval-gated verifier to

check whether the proposal is supported by the document and sufficiently distinct from existing types, thereby maintaining controlled ontology expansion.

Generative Detection. The generative detector LLM first performs detection or discovery over the current semantic cluster subtree $\mathcal{H}(c_k^*)$. Guided by the strategy π_k and contextual instruction I_k , the detector analyzes the document D to identify event triggers and map them to the cluster-specific events $\mathcal{E}_k^{\text{cand}}$.

$$\mathcal{E}_k^{\text{cand}} = \text{LLM}_{\text{detector}}(D, \mathcal{H}(c_k^*), \pi_k, I_k),$$

where each candidate event $e \in \mathcal{E}_k^{\text{cand}}$ has a generated event type t_e and definition d_e . For the remaining candidate events whose types already exist in $\mathcal{O}_{\text{final}}$, i.e., $(t_e, d_e) \in \mathcal{O}_{\text{final}}$, we keep the original event definitions and event types.

Retrieval-Gated Verification. For the generated (t_e, d_e) , we embed definitions d_e into a shared space via an embedding model $\phi(\cdot)$ and compute the maximum vector similarity:

$$\text{sim}_{\text{max}}(d_e) = \max_{(t_i, d_i) \in \mathcal{O}_{\text{seed}}} \cos(\phi(d_e), \phi(d_i)).$$

If the $\text{sim}_{\text{max}}(d_e)$ is lower than a predefined hyperparameter threshold σ , i.e., $\text{sim}_{\text{max}}(d_e) < \sigma$, we accept the generated event type t_e as it is distinctly

far from existing types. Otherwise, we retrieve the top $p_v = 5$ nearest neighbors $\mathcal{N}_k(d_e)$:

$$\mathcal{N}_k(d_e) = \text{Top-}p_v \left(\cos(\phi(d_e), \phi(d_i)) \right)_{(d_i, d_i) \in \mathcal{O}_{\text{seed}}}$$

Then we use a judge LLM for verification:

$$\delta = \text{LLM}_{\text{judge}}(D, t_e, d_e, \mathcal{N}_k(d_e)),$$

where the action $\delta \in \{\text{MAPTOEXISTING}(t_{\text{map}}), \text{ACCEPTNEW}\}$. The judge LLM reads the document D as evidence and decides whether the proposed definition is genuinely new; if not, it returns an existing type from predefined event types for remapping. Finally, to update the existing ontology for the next iteration, we merge the current event set $\mathcal{O}_{\text{final}}$ with the newly induced candidate event set, and treat the merged set as the evolved ontology $\mathcal{H}_{\text{final}}$.

4.5 Global State Update

At the end of k -th iteration of event detection, we perform global framework state update, including the ontology, history, and posterior confidence update. For the ontology update, if a new event type o_{new} is accepted, it is attached to the current target cluster c_k^* as a new leaf node. Formally, we update the dynamic tree as:

$$\mathcal{H}(c_k^*) \leftarrow \mathcal{H}(c_k^*) \cup \{o_{\text{new}} \xrightarrow{\text{parent}} c_k^*\},$$

and synchronize the update to the global ontology tree $\mathcal{H}_{\text{final}}$.

For the history update, we append structured history text logs to H . For the posterior score update, each event candidate $e \in \mathcal{E}_k^{\text{cand}}$ receives a posterior score:

$$s(e) = 0.5S_{\text{pos}}(e) + 0.3S_{\text{sem}}(e) + 0.2P_0(c_k^*).$$

Here, $S_{\text{pos}}(e)$ measures trigger-span novelty by penalizing overlap with previously extracted spans:

$$S_{\text{pos}}(e) = 1.0 - 0.3 \cdot |\{e' \in \mathcal{E}_k : m_e \cap m_{e'} \neq \emptyset\}|,$$

clipped to $[0, 1]$. $S_{\text{sem}}(e) = \text{Reranker}(m_e, d_{t_e})$ measures the relevance between the trigger span m_e and the definition d_{t_e} of its assigned event type, and $P_0(c_k^*)$ is the prior relevance of the selected cluster. After scoring candidates, we update the visit count $n(c_k^*)$ for subsequent scheduling.

4.6 Conflict Selection

Different clusters may propose competing candidates for the same or overlapping trigger spans because they focus on different semantic regions. These candidates may correspond to either existing event types or newly induced types, making conflict selection the final step of grounding–discovery resolution. We first group candidates whose trigger spans overlap, i.e., $\text{IoU}(m_i, m_j) \geq 0.2$, into conflict groups.

For each conflict group G , an arbitration LLM uses the full document, trigger spans, candidate event types, and posterior scores as evidence to select a compact candidate subset:

$$\hat{G} = \text{LLM}_{\text{arb}}(D, G),$$

where $G = \{(m, t, s(e))\}$ denotes the competing candidates and $\hat{G}$ is the selected subset for the overlapping span region. In practice, the arbitration model keeps a single event mention unless the overlapping span expresses multiple distinct events. We retain candidates in $\hat{G}$ and discard the remaining candidates in G . The final output $\mathcal{E}_{\text{final}}$ contains event mentions assigned to either existing event types or newly induced types.

5 Experiment

5.1 Experiment Settings

We use *Qwen3-30B-A3B-Instruct-2507* (hereafter referred to as *Qwen3-30B-A3B*) as the main LLM backbone and *Qwen3-8B* for model-size generalization (Qwen Team, 2025). We use *Qwen3-Reranker-8B* as retriever and reranker, and *Qwen3-Embedding-4B* as the embedding model for encoding event definitions (Zhang et al., 2025). For each document, we retrieve the top $p = 15$ relevant event types to build the dynamic search tree. All reported results are averaged over three runs. Full hyperparameter settings and sensitivity analyses are provided in Appendix A.

5.1.1 Datasets.

We evaluate OGED on five datasets from the SEOE benchmark (Lu et al., 2025): MAVEN (Wang et al., 2020), CASIE (Satyapanich et al., 2020), Few-Event (Deng et al., 2020), MLEE (Pyysalo et al., 2012), and WikiEvents (Li et al., 2021). SEOE unifies these datasets and adds latent-event annotations, so evaluation covers both original labels and supplemented cross-dataset events. In total, the benchmark contains 564 event types. Detailed

Table 1: Main results on the open-domain event detection benchmark. We report F1 scores (%) for Trigger Identification (TI), Trigger Classification (TC), and Event Identification (EI). All methods use the same Qwen3-30B-A3B backbone and the same top- p retrieval module for fair comparison.

Setting	Method (p=15)	MAVEN			CASIE			FEWEVENT			MLEE			WIKIEVENTS			AVERAGE		
		TI	TC	EI	TI	TC	EI	TI	TC	EI	TI	TC	EI	TI	TC	EI	TI	TC	EI
Unknown	Baselines	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MD-Open	<u>0.53</u>	<u>0.53</u>	<u>7.11</u>	0	0	0	0	0	0	<u>0.33</u>	<u>0.33</u>	<u>0.33</u>	0	0	0	<u>0.17</u>	<u>0.17</u>	<u>1.49</u>
	OGED	14.90	14.90	26.98	13.90	13.90	45.99	13.27	13.27	31.22	22.88	22.88	50.45	2.59	2.59	10.95	13.51	13.51	33.12
Known	GEE	48.94	21.30	27.22	20.41	7.55	19.59	38.6	13.55	19.74	<u>30.36</u>	17.31	19.96	52.09	6.83	11.38	38.08	13.31	19.58
	DiCoRe	9.93	4.07	5.17	4.15	0.10	0.30	43.84	3.04	4.27	14.16	10.11	15.09	10.21	1.94	3.67	16.46	3.85	5.70
	MS	29.64	14.85	18.20	11.85	7.60	12.16	27.08	16.78	23.53	14.70	15.82	20.51	11.39	6.94	10.59	18.93	12.40	17.00
	ChatIE	49.98	20.84	25.07	31.10	15.12	28.07	<u>47.17</u>	25.48	34.85	33.57	25.18	38.81	35.28	<u>13.17</u>	<u>17.82</u>	<u>39.42</u>	<u>19.96</u>	28.92
	MD	<u>52.32</u>	<u>23.93</u>	<u>28.47</u>	<u>30.88</u>	<u>14.35</u>	<u>23.71</u>	44.70	20.45	26.12	19.99	25.07	27.83	27.58	10.99	15.60	35.09	18.96	24.35
	MD-Open	44.25	15.59	18.99	24.86	14.26	19.74	43.22	20.01	23.53	18.70	<u>25.38</u>	28.13	26.36	9.20	12.94	31.48	16.89	20.67
	OGED	55.94	28.88	32.57	30.27	13.65	22.55	52.50	<u>20.50</u>	<u>26.67</u>	23.93	25.50	<u>34.03</u>	<u>40.05</u>	15.35	20.01	40.54	20.78	<u>27.17</u>

dataset statistics are given in Table 2. The construction of ontology tree is illustrated in Appendix B.

Table 2: Detailed statistics of the SEOE benchmark. The benchmark covers a total of 564 unique event types across multiple domains.

Dataset	Domain	Size	# Event Types			Mentions
			Orig.	Supp.	Total	
MAVEN	General	1163	168	384	394	5.42
CASIE	Cybersecurity	33	5	92	92	13.94
FewEvent	General	981	100	379	381	2.82
MLEE	Biomedical	84	29	68	69	31.11
WikiEvents	Wikipedia	188	50	286	292	18.94

5.1.2 Evaluation Metrics.

Following established protocols (Parekh et al., 2025a,b), we report F1 scores for three standard metrics: (1) **Trigger Identification (TI)**, measuring the correct detection of event trigger spans; (2) **Event Identification (EI)**, evaluating the correct classification of event types; and (3) **Trigger Classification (TC)**, requiring both accurate identification of the trigger span and its corresponding event type. Detailed matching criteria for known and new events are provided in Appendix C.

5.1.3 Baselines for Comparison.

We compare OGED against five state-of-the-art zero-shot event detection methods: GuidelineEE (GEE) (Srivastava et al., 2025), Multi-event Staged (MS) (Parekh et al., 2025a), DiCoRe (Parekh et al., 2025b), ChatIE (Wei et al., 2023), and Multi-event Direct (MD) (Gao et al., 2023). To experiment whether a simple prompt method can support our task, we additionally introduce MD-Open, which augments MD with an instruction to generate new event types when no predefined type matches the trigger. To ensure a fair comparison, all baselines use the same top- p relevant event type definitions as context for inference. Detailed descriptions of each baseline are provided in Appendix D.

5.2 Performance in Open-domain Setting

Event Masking Strategy. To simulate an evolving environment, we keep 30% of the event types as unknown types and construct the seed ontology from the remaining 70% of event types. Results are summarized in Table 1.

Unknown Event Discovery. Closed-set baselines obtain zero F1 on unknown events. MD-Open, which adds open-generation to a single-pass detector, still yields only marginal recovery. OGED improves by +13.34 Unknown TI and +31.63 Unknown EI on average compared to MD-Open. The larger gain in Unknown EI indicates that OGED is effective in discovering new events and proposing semantically appropriate new types. The lower unknown TI on WIKIEVENTS is mainly due to the high semantic overlap between its masked types and retained MAVEN types, reflecting benchmark characteristics rather than a framework limitation. However, successful type discovery does not necessarily imply accurate event-mention extraction, where Unknown TI/TC scores remain lower than Unknown EI, showing that precise trigger localization is still challenging. A key reason is that unknown types have no type-specific trigger-span supervision at discovery time, while trigger boundaries are sensitive to dataset-specific annotation conventions.

Known Event Detection. OGED also remains competitive on known events. Compared with MD-Open, it improves average Known TI, TC and EI by +9.06, +3.89 and +6.50 F1, respectively, showing that unconstrained open generation can weaken known-event detection. Compared with the strongest closed-set baseline ChatIE, OGED improves average Known TI and TC by +1.12 and +0.82 F1, while lowering by 1.75 F1 on EI. Overall, OGED achieves the best average Known TI and TC while retaining substantial capability for

Table 3: Performance comparison on zero-shot event detection with the **Qwen3-30B-A3B** backbone. We report F1 scores (%) for Trigger Identification (TI), Trigger Classification (TC), and Event Identification (EI) across five datasets. All methods use the same backbone and the same top- p retrieval module for fair comparison. The AVERAGE columns report macro-averages over the five datasets.

Method ($p = 15$)	MAVEN			CASIE			FEWEVENT			MLEE			WIKIEVENTS			AVERAGE		
	TI	TC	EI	TI	TC	EI	TI	TC	EI	TI	TC	EI	TI	TC	EI	TI	TC	EI
GEE	<u>55.50</u>	<u>22.40</u>	<u>28.73</u>	26.33	7.55	23.47	45.64	15.72	22.07	<u>33.50</u>	12.90	23.85	21.06	5.59	<u>15.53</u>	36.41	12.83	22.73
DiCoRe	8.50	3.12	3.83	30.66	5.35	12.14	7.36	2.43	3.59	32.10	8.78	15.03	22.23	2.93	1.99	20.17	4.52	7.32
MS	35.27	16.88	20.11	15.82	10.10	16.16	34.42	19.20	26.55	13.33	18.78	23.14	7.07	3.77	5.38	21.18	13.75	18.27
ChatIE	51.77	19.82	23.81	<u>34.59</u>	16.85	29.93	52.43	24.39	33.18	33.35	19.61	<u>31.53</u>	<u>35.88</u>	<u>10.61</u>	13.72	<u>41.60</u>	18.26	<u>26.43</u>
MD	47.95	21.30	25.38	32.79	<u>12.47</u>	20.33	<u>56.14</u>	26.10	34.78	30.13	<u>22.61</u>	28.82	27.24	10.14	13.89	38.85	<u>18.52</u>	24.64
MD-Open	37.97	16.51	20.18	25.26	12.33	18.95	49.66	<u>24.64</u>	<u>34.51</u>	18.67	21.15	24.01	25.72	9.24	13.41	31.46	16.77	22.21
OGED	63.50	25.42	30.15	38.15	12.02	<u>25.20</u>	58.97	21.41	27.25	40.93	23.53	32.39	47.97	12.07	21.02	49.90	18.89	27.20

unknown-event discovery.

5.3 Performance in Zero-shot Setting

We evaluate OGED in the standard zero-shot setting, where all event types are available. The discovery mode is disabled. Table 3 shows that OGED achieves the best average results, outperforming ChatIE by +8.30 TI and +0.77 EI. Its gains are most evident on broad-inventory datasets such as MAVEN and WIKIEVENTS, where cluster-wise navigation better supports trigger coverage than flat retrieved-type prompting.

The main exceptions are CASIE and FEWEVENT. ChatIE obtains the best TC/EI on CASIE, where multi-turn type filtering is effective over a compact retrieved type set. MD obtains the best TC/EI on FEWEVENT, which has the lowest event density at 2.82 mentions per document, as shown in Table 2. In such sparse documents, a strong single-pass detector can already cover most candidates, leaving less room for iterative cluster-wise search. Overall, OGED is most useful when the type inventory is broad or the document provides event evidence for structured navigation.

5.4 Ablation Study

We conduct ablation studies to evaluate the contribution of the *Scheduler*, *Planner Instructions*, *Conflict Selection*, and *Retrieval-Gated Judge* under the open-domain setting, as shown in Table 4. For efficiency, we use a random 200-sample subset of MAVEN and full CASIE, which allows us to examine both a broad-domain dataset and a specialized-domain dataset. The corresponding zero-shot ablation results are reported in Appendix G.

Removing the Scheduler or Planner Instructions reduces unknown-event performance, confirming their role in cluster prioritization and discovery guidance. Conflict Selection mainly protects known-event performance by resolving over-

Table 4: Ablation study of key OGED components under the open-domain setting. We report F1 scores (%) for TI, TC, and EI on both Known and Unknown events. “w/o” denotes removing the corresponding module.

Dataset	Model Variant	Known Events			Unknown Events		
		TI	TC	EI	TI	TC	EI
MAVEN	OGED	57.3	26.51	30.54	10.09	10.09	24.28
	w/o Scheduler	57.27	26.38	31.08	9.35	9.35	21.53
	w/o Planner Instructions	54.14	26.22	31.34	8.37	8.37	19.69
	w/o Conflict Selection	45.98	21.77	26.44	11.50	11.50	23.55
	w/o Retrieval-Gated Judge	57.08	25.42	30.03	12.69	12.69	27.94
	Baseline (MD)	52.32	23.93	28.47	0	0	0
CASIE	OGED	35.86	13.09	26.44	7.02	7.02	31.58
	w/o Scheduler	33.73	16.33	25.97	0.00	0.00	16.33
	w/o Planner Instructions	36.03	14.04	25.17	4.00	4.00	28.00
	w/o Conflict Selection	31.21	16.00	24.52	3.45	3.45	31.30
	w/o Retrieval-Gated Judge	31.75	13.65	25.91	11.76	11.76	39.22
	Baseline (MD)	30.88	14.35	23.71	0	0	0

lapping candidates. The Retrieval-Gated Judge trades recall for precision. Removing it accepts more generated types but risks weaker known-type grounding. Together, these components balance discovery and grounding.

5.5 Computational Cost–Performance Analysis

Table 5 compares OGED with ChatIE with increasing ontology incompleteness. Overall F1 is computed over the union of known-event and unknown-event predictions, so it reflects the joint ability to preserve known-type detection and discover new event types. The TC improvement is +5.55 F1 (from 18.45 to 24.00), +7.04 F1 (from 18.76 to 25.80), and +19.49 F1 (from 13.81 to 33.30) under 30%, 50%, and 70% masking, respectively. This shows that OGED becomes more beneficial as the initial ontology becomes increasingly incomplete.

OGED introduces additional LLM calls compared with ChatIE, but these calls support ontology-guided grounding–discovery resolution rather than generic repeated extraction. Thus, OGED is intended for settings where ontology incompleteness is substantial and accurate recovery of both known

Table 5: Inference cost and overall performance on MAVEN under frequency-aware ontology masking. We mask 30%, 50%, and 70% of leaf event types and compare OGED with ChatIE. Overall F1 is the combination metric of known and unknown events, using a semantic threshold of 0.2 for unknown-type matching. “Calls/Doc” denotes average LLM calls per document; “+” values beside OGED scores denote absolute F1 gains over ChatIE.

Mask	Method	Calls/Doc	Overall F1		
			TI	TC	EI
30%	ChatIE	4.76	44.25	18.45	22.20
	OGED	13.2	50.10 (+5.85)	24.00 (+5.55)	29.50 (+7.30)
50%	ChatIE	4.37	31.61	18.76	23.62
	OGED	15.0	35.70 (+4.09)	25.80 (+7.04)	40.40 (+16.78)
70%	ChatIE	3.79	22.08	13.81	17.10
	OGED	16.3	37.60 (+15.52)	33.30 (+19.49)	47.40 (+30.30)

Table 6: LLM-call breakdown of OGED on MAVEN under increasing ontology incompleteness. Rows correspond to LLM roles used by OGED: *Detector* for event detection within selected clusters, *Planner* for strategy and instruction generation, *Judge* for retrieval-gated verification, and *Conflict Selection* for final arbitration. Each cell reports Calls/Doc (Percent), where Calls/Doc is the average number of LLM calls per document and Percent is its share of total OGED calls under the same mask ratio.

LLM Role	30% Mask Calls/Doc (Percent)	50% Mask Calls/Doc (Percent)	70% Mask Calls/Doc (Percent)
Detector	9.02 (68.4%)	8.68 (57.7%)	8.39 (51.6%)
Planner	2.70 (20.5%)	2.87 (19.1%)	3.49 (21.5%)
Judge	0.54 (4.1%)	2.77 (18.4%)	3.71 (22.8%)
Conflict Selector	0.93 (7.0%)	0.71 (4.7%)	0.67 (4.1%)
Total	13.19	15.03	16.26

and unknown events outweighs minimizing single-document latency.

Table 6 breaks down the average LLM calls per document across LLM roles. As the masking ratio increases from 30% to 70%, the total calls increase moderately from 13.19 to 16.26, with most additional calls concentrated in discovery-related roles. The detector remains the largest component, but its share drops from 68.4% to 51.6%, while the judge increases from 0.54 to 3.71 calls per document. This reflects the greater need to verify new-type proposals as more leaf types are removed. Planner calls also increase slightly, whereas conflict selection remains relatively stable. Overall, OGED spends additional computation on verification and planning instead of increasing calls uniformly across modules. We further analyze the impact of the iteration budget k_{max} in Appendix H.

Case 1

Document:

“...Other attacks allowed hackers to perform in-app **phishing attacks**_[Attack:Phishing*] against Santander and Allied Irish bank users, allowing attackers to take over part of the screen while the app was running...”

*Note: *Purple* indicates unknown (masked) types.

Prediction Comparison:

Trigger	Gold Type	Baseline (ChatIE)	Ours
phishing attacks	Atk:Phishing	✗ None	✓ NEW: In-App Phishing

Case 2

Document:

“... The attackers favour using URL shortening services to disguise the true links they are sending in spear phishing emails. ... Another method favoured by the attackers was **the setting up**_[Attack:Phishing*] of fake news sites...”

*Note: *Purple* indicates unknown (masked) types.

Prediction Comparison:

Trigger	Gold Type	Baseline (ChatIE)	Ours
the setting up	Atk:Phishing	✗ Using	✓ NEW: Fake News Site Setup

Figure 4: Case studies on CASIE. Both triggers correspond to unseen event types. OGED creates new labels (*In-App Phishing*, *Fake News Site Setup*), while the baseline either fails to detect the event or misroutes it to a known type (*Using*).

5.6 Case Study

Fig. 4 illustrates two failure modes of flat zero-shot detection under incomplete event type sets in CASIE. In Case 1, ChatIE misses the unseen trigger *phishing attacks*, while OGED identifies it as a new event type and induces the label *In-App Phishing*. In Case 2, ChatIE detects the trigger span but routes it to a plausible known type, *Using*, whereas OGED resolves it as a discovery case and induces *Fake News Site Setup*. These examples show that our method can help address these two failure modes.

6 Conclusion

In this paper, we study open-domain event detection under an evolving event set, where the system must both ground mentions to known event types and create new ones when the current ontology is insufficient. We propose OGED, an ontology-guided framework for grounding–discovery resolution that keeps known-type assignment and new-type discovery jointly active during inference through cluster-level scheduling, adaptive planning, retrieval-gated verification, and conflict selection. Experiments on SEOE benchmark show that OGED improves both known-type detection and unknown-type discovery over strong baselines.

Limitations

While OGED achieves better performance on the SEOE benchmark, several limitations remain. First, our experiments evaluate OGED at the document level, where the framework decides whether each mention should be assigned to an existing type or should trigger a new one. We do not yet study long-horizon ontology evolution as newly induced types accumulate across many documents. Future work should develop mechanisms for controlled ontology growth, including redundancy control, near-duplicate type merging, and structure maintenance during continual detection and discovery. Second, most OGED modules run with temperature 0.0, while only the planner instruction generator uses temperature 0.3 to encourage exploration of possible new event types. In practice, this stochasticity is confined to the discovery-hint stage, while the remaining modules are executed deterministically.

Ethical Considerations

We acknowledge the use of AI assistants for figure-generation code, literature search support, and writing polishing. All references, results, claims, and final wording were independently checked and finalized by the authors.

Acknowledgments

This work is sponsored by CCF-DiDi GAIA Collaborative Research Funds.

References

- Jun Araki and Teruko Mitamura. 2018. Open-domain event detection using distant supervision. In *Proceedings of the 27th international conference on computational linguistics*, pages 878–891.
- Zefan Cai, Po-Nien Kung, Ashima Suvarna, Mingyu Ma, Hritik Bansal, Baobao Chang, P Jeffrey Brantingham, Wei Wang, and Nanyun Peng. 2024. Improving event definition following for zero-shot event detection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2842–2863.
- Pengfei Cao, Yupu Hao, Yubo Chen, Kang Liu, Jiexin Xu, Huaijun Li, Xiaojian Jiang, and Jun Zhao. 2023. Event ontology completion with hierarchical structure evolution networks. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 306–320.
- Pan Chen, Shaohong Chen, Mark Wang, Shi Xuan Leong, Priscilla Fung, Varinia Bernales, and Alan Aspuru-Guzik. 2025. Schema for in-context learning. *arXiv preprint arXiv:2510.13905*.
- Shumin Deng, Ningyu Zhang, Jiaojian Kang, Yichi Zhang, Wei Zhang, and Huajun Chen. 2020. Meta-learning with dynamic-memory-based prototypical network for few-shot event detection. In *Proceedings of the 13th international conference on web search and data mining*, pages 151–159.
- Neil De La Fuente, Oscar Sainz, Iker García-Ferrero, and Eneko Agirre. 2025. [GUIDEX: Guided synthetic data generation for zero-shot information extraction](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 24248–24262, Vienna, Austria. Association for Computational Linguistics.
- Jun Gao, Huan Zhao, Changlong Yu, and Ruifeng Xu. 2023. Exploring the feasibility of chatgpt for event extraction. *arXiv preprint arXiv:2303.03836*.
- Quanjiang Guo, Sijie Wang, Jinchuan Zhang, Ben Zhang, Zhao Kang, Ling Tian, and Ke Yan. 2026. Extracting events like code: A multi-agent programming framework for zero-shot event extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 30880–30887.
- Qi Li, Heng Ji, and Liang Huang. 2013. [Joint event extraction via structured prediction with global features](#). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 73–82.
- Sha Li, Heng Ji, and Jiawei Han. 2021. [Document-level event argument extraction by conditional generation](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 894–908, Online. Association for Computational Linguistics.
- Sha Li, Heng Ji, and Jiawei Han. 2022. Open relation and event type discovery with type abstraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6864–6877.
- Runxuan Liu, Bei Luo, Jiaqi Li, Baoxin Wang, Ming Liu, Dayong Wu, Shijin Wang, and Bing Qin. 2025. [Ontology-guided reverse thinking makes large language models stronger on knowledge graph question answering](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15269–15284, Vienna, Austria. Association for Computational Linguistics.
- Yi-Fan Lu, Xian-Ling Mao, Tian Lan, Tong Zhang, Yu-Shi Zhu, and Heyan Huang. 2025. [SEOE: A scalable and reliable semantic evaluation framework for open domain event detection](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7201–7218, Vienna, Austria. Association for Computational Linguistics.

- Yi-Fan Lu, Xian-Ling Mao, Bo Wang, Xiao Liu, and Heyan Huang. 2026. [EXCEEDS: Extracting complex events via nugget-based grid modeling in scientific domain](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5997–6022, San Diego, California, United States. Association for Computational Linguistics.
- Mingyu Derek Ma, Alexander Taylor, Wei Wang, and Nanyun Peng. 2023a. Dice: Data-efficient clinical event extraction with generative models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15898–15917.
- Yubo Ma, Zehao Wang, Yixin Cao, and Aixin Sun. 2023b. [Few-shot event detection: An empirical study and a unified view](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11211–11236, Toronto, Canada. Association for Computational Linguistics.
- J. MacQueen. 1967. Some methods for classification and analysis of multivariate observations. In *Proc. of 5th Berkeley Symposium on Math. Stat. and Prob.*, pages 281–297.
- Tanmay Parekh, Yuxuan Dong, Lucas Bandarkar, Artin Kim, I-Hung Hsu, Kai-Wei Chang, and Nanyun Peng. 2025a. [SNaRe: Domain-aware data generation for low-resource event detection](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20583–20604, Suzhou, China. Association for Computational Linguistics.
- Tanmay Parekh, Kartik Mehta, Ninareh Mehrabi, Kai-Wei Chang, and Nanyun Peng. 2025b. [DiCoRe: Enhancing zero-shot event detection via divergent-convergent LLM reasoning](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20560–20582, Suzhou, China. Association for Computational Linguistics.
- Sampo Pyysalo, Tomoko Ohta, Makoto Miwa, Han-Cheol Cho, Jun’ichi Tsujii, and Sophia Ananiadou. 2012. Event extraction across multiple levels of biological organization. *Bioinformatics*, 28(18):i575–i581.
- Qwen Team. 2025. [Qwen3 technical report](#). Preprint, arXiv:2505.09388.
- Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. 2024. [GoLLIE: Annotation guidelines improve zero-shot information-extraction](#). In *The Twelfth International Conference on Learning Representations*.
- Taneeya Satyapanich, Francis Ferraro, and Tim Finin. 2020. Casie: Extracting cybersecurity event information from text. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8749–8757.
- Saurabh Srivastava, Sweta Pati, and Ziyu Yao. 2025. [Instruction-tuning LLMs for event extraction with annotation guidelines](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 13055–13071, Vienna, Austria. Association for Computational Linguistics.
- The MITRE Corporation. 2026. [MITRE ATT&CK](#). Accessed: 2026-05-21.
- Sijia Wang and Lifu Huang. 2024. [Debate as optimization: Adaptive conformal prediction and diverse retrieval for event extraction](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16422–16435, Miami, Florida, USA. Association for Computational Linguistics.
- Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, Jihua Kang, Jingsheng Yang, Siyuan Li, and Chunsai Du. 2023. Instructuie: Multi-task instruction tuning for unified information extraction. *arXiv preprint arXiv:2304.08085*.
- Xiaozhi Wang, Ziqi Wang, Xu Han, Wangyi Jiang, Rong Han, Zhiyuan Liu, Juanzi Li, Peng Li, Yankai Lin, and Jie Zhou. 2020. [MAVEN: A Massive General Domain Event Detection Dataset](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1652–1671, Online. Association for Computational Linguistics.
- Xinyuan Wang, Chenxi Li, Zhen Wang, Fan Bai, Haotian Luo, Jiayou Zhang, Nebojsa Jojic, Eric Xing, and Zhiting Hu. 2024. [Promptagent: Strategic planning with language models enables expert-level prompt optimization](#). In *The Twelfth International Conference on Learning Representations*.
- Xixi Wang, Miguel Costa, Jordanka Kovaceva, Shuai Wang, and Francisco C. Pereira. 2025. [Plugging schema graph into multi-table QA: A human-guided framework for reducing LLM reliance](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 5829–5842, Suzhou, China. Association for Computational Linguistics.
- Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, Yong Jiang, and Wenjuan Han. 2023. Chatie: Zero-shot information extraction via chatting with chatgpt. *arXiv preprint arXiv:2302.10205*.
- Bishan Yang and Tom Mitchell. 2016. [Joint extraction of events and entities within a document context](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 289–299.
- Yunzhi Yao, Shengyu Mao, Ningyu Zhang, Xiang Chen, Shumin Deng, Xi Chen, and Huajun Chen. 2023.

Schema-aware reference as prompt improves data-efficient knowledge graph construction. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 911–921.

Zhenrui Yue, Huimin Zeng, Mengfei Lan, Heng Ji, and Dong Wang. 2023. [Zero- and few-shot event detection via prompt-based meta learning](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7928–7943, Toronto, Canada. Association for Computational Linguistics.

Senhui Zhang, Tao Ji, Wendi Ji, and Xiaoling Wang. 2022. Zero-shot event detection based on ordered contrastive learning and prompt-based prediction. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2572–2580.

Xiaoying Zhang, Baolin Peng, Kun Li, Jingyan Zhou, and Helen Meng. 2023. [SGP-TOD: Building task bots effortlessly via schema-guided LLM prompting](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13348–13369, Singapore. Association for Computational Linguistics.

Yanan Zhang, Xiaoling Bai, and Tianhua Zhou. 2024. Event-enhanced retrieval in real-time search. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 6594–6606.

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

A Implementation Details

We list the implementation details used in our experiments. For the scheduler, we set the exploration coefficient to $\lambda = 1.0$. For the planner threshold τ , we use a domain-level rule rather than per-dataset tuning. Specialized domains use a more permissive threshold ($\tau = 10^{-4}$), while broad-domain datasets use a stricter threshold ($\tau = 0.4$). The domain category is determined from the dataset description only, without using test labels. The implications of this threshold are analyzed in Appendix I.1. For retrieval-gated verification, we set the similarity gate to $\sigma = 0.3$. For posterior confidence scoring, we use a fixed linear combination of trigger-span novelty, reranker relevance, and cluster prior, as described in Sec. 4.5. The combination weights are selected in a small pilot analysis on MAVEN based on the performance of *Qwen3-Reranker-8B*, and then kept unchanged for all datasets, models, and masking settings. Open-source models are run locally using up to four NVIDIA H20 GPUs.

B Ontology Tree Construction

To support efficient navigation and expansion, we convert the flat SEOE event type inventory into a hierarchical ontology tree. Because SEOE preserves source-specific labels, the inventory contains near-duplicate labels from different datasets. Two domain experts normalize these cases by merging only semantically equivalent variants caused by punctuation, suffixes, or source-ontology naming conventions, while keeping distinct event semantics separate. This reduces the original 564 fine-grained labels to 463 normalized event types, with 101 labels merged into 36 groups and 427 labels unchanged. Normalized inventory is used consistently for ontology construction, masking, prediction normalization, and evaluation.

The ontology tree is then constructed through a semi-automatic process that integrates semantic clustering with expert verification. We first encode event type definitions using *Qwen3-Embedding-4B* and apply two-stage K-Means (MacQueen, 1967): a fixed $k = 14$ for high-level clusters and adaptive $k = 2-8$ within each coarse cluster for family-level grouping. This initial clustering produces 14 coarse categories and 95 family clusters.

Two domain experts review the clusters and make three types of refinements: (1) splitting an over-aggregated generic-event cluster, mainly from

MAVEN, into semantically distinct sub-categories; (2) merging a Contact cluster split across two coarse groups into a unified Communication category; (3) moving singleton family clusters, such as *Attack:Ransom*, into appropriate parent groups. These refinements reduce the family count from 95 to 61, resulting in a reference ontology tree with a root node, 14 coarse categories, 61 family categories and fine-grained leaf event types.

C Evaluation Metric Details

To rigorously assess the alignment between predicted and gold results, we apply different matching criteria for known and new events. For grounding predicted triggers in the input text, we use a span-overlap criterion, considering a predicted trigger correct if its Intersection over Union (IoU) with the gold span exceeds $\tau_{\text{IoU}} = 0.2$. Evaluation is performed at the mention level, where repeated occurrences of the same trigger string in a document are not collapsed, and each occurrence contributes separately to micro-averaged TP, FP, and FN counts, following standard event extraction evaluation based on trigger offsets and event types (Li et al., 2013; Yang and Mitchell, 2016).

For new event types generated in the open-world setting, we assess semantic alignment by computing embedding similarity, regarding a generated new event type as correct if its cosine similarity with the gold latent event type exceeds a semantic threshold τ_{sim} . In practice, we set $\tau_{\text{sim}} = 0.2$ because generated event types are often expressed as semantically rich descriptions rather than short canonical labels; their embedding similarities to gold latent types are therefore more dispersed, even when the underlying event semantics are aligned. We use the same thresholds for all methods and all masking settings.

D Baseline Methods

We provide detailed descriptions of the six baselines used in our experiments.

GuidelineEE (GEE) (Srivastava et al., 2025) formulates event detection as a code generation task by mapping event schemas to Python classes.

Multi-event Staged (MS) (Parekh et al., 2025a) adopts a two-stage pipeline, first identifying potential event types and then extracting their corresponding triggers.

DiCoRe (Parekh et al., 2025b) is a baseline that employs a divergent-convergent reasoning frame-

Table 7: Performance comparison on zero-shot event detection with the **Qwen3-8B** backbone. We report micro F1 scores (%) for Trigger Identification (TI), Trigger Classification (TC), and Event Identification (EI) across five datasets.

Method (p=15)	MAVEN			CASIE			FEWEVENT			MLEE			WIKIEVENTS			AVERAGE		
	TI	TC	EI	TI	TC	EI	TI	TC	EI	TI	TC	EI	TI	TC	EI	TI	TC	EI
GEE	48.18	16.83	19.59	14.29	6.80	13.95	62.80	18.92	23.72	11.38	10.46	13.15	11.29	2.25	3.91	29.59	11.05	14.86
DiCoRe	6.33	2.23	2.65	7.10	0.37	1.12	4.96	1.79	2.90	8.05	6.92	10.18	39.75	0.87	2.24	13.24	2.44	3.82
MS	36.31	16.30	19.28	13.12	5.62	8.30	36.75	19.43	27.27	12.95	16.48	21.28	7.79	3.14	5.85	21.38	12.19	16.40
ChatIE	<u>56.17</u>	<u>21.20</u>	<u>24.25</u>	52.54	9.49	20.79	47.77	21.95	27.32	44.34	16.22	<u>24.88</u>	49.59	5.93	8.42	50.08	14.96	21.13
MD	40.73	18.53	21.17	22.46	13.05	<u>23.98</u>	52.70	27.71	35.09	17.18	<u>19.07</u>	22.89	25.98	<u>7.70</u>	11.99	31.81	<u>17.21</u>	23.02
MD-Open	37.92	16.23	20.10	23.56	<u>12.62</u>	26.09	49.66	<u>24.64</u>	<u>34.51</u>	17.12	18.17	22.76	21.47	6.78	<u>12.76</u>	29.95	<u>15.69</u>	<u>23.24</u>
OGED	62.44	23.16	27.32	<u>37.53</u>	11.53	20.13	<u>57.58</u>	20.75	26.46	<u>37.35</u>	20.01	25.61	<u>42.63</u>	10.86	19.11	<u>47.51</u>	17.26	23.73

work, using a dreamer LLM and a Grounder LLM to fully exploit the intrinsic knowledge and reasoning capabilities of LLMs.

ChatIE (Wei et al., 2023) decomposes the detection process into a multi-turn conversation with the LLM, sequentially identifying event types and relevant triggers.

Multi-event Direct (MD) (Gao et al., 2023) prompts the LLM to extract all events and triggers in a single inference pass. In practice, we employ the exact same prompt as our proposed method to ensure a fair comparison.

MD-Open extends MD with an additional instruction that asks the LLM to also generate novel event types when no predefined type matches the trigger, while otherwise reusing MD’s single-pass prompt.

To further compare OGED with training-based zero-shot information extraction methods, we evaluate GoLLIE-7B (Sainz et al., 2024) on the MAVEN and CASIE open-domain setting and compare it with OGED on Qwen3-8B. This comparison is intended to test whether a guideline-tuned extractor can handle the open-domain setting.

Table 8: Open-domain comparison between OGED with Qwen3-8B and GoLLIE-7B on MAVEN and CASIE. We report F1 (%) for known, unknown, and overall evaluation.

Dataset	Setting	GoLLIE-7B			OGED		
		TI	TC	EI	TI	TC	EI
MAVEN	Known	35.59	12.63	17.70	54.12	22.34	27.53
	Unknown	0.00	0.00	0.00	9.07	9.07	19.20
	Overall	30.17	10.70	15.00	47.51	20.40	26.31
CASIE	Known	76.53	0.73	2.47	33.92	11.47	21.95
	Unknown	0.00	0.00	0.00	7.27	7.27	21.82
	Overall	76.04	0.73	2.45	32.21	11.20	21.94

E Generalization to Smaller Backbones

Table 7 evaluates OGED with the smaller *Qwen3-8B* backbone for model-scale generalization. OGED achieves the best average TC and EI scores,

Table 9: Ablation study of key OGED components under the zero-shot event detection setting. We report F1 scores (%) for TI, TC, and EI. “w/o” denotes removing the corresponding module.

Dataset	Model Variant	TI	TC	EI
MAVEN	OGED	68.76	35.98	38.97
	w/o Scheduler	68.29	35.81	39.13
	w/o Planner Instructions	66.25	34.7	37.76
	w/o Conflict Selection	49.85	27.12	31.14
	Baseline (MD)	52.61	22.3	25.29
CASIE	OGED	38.15	12.02	25.2
	w/o Scheduler	38.27	11.44	21.24
	w/o Planner Instructions	36.76	9.19	21.08
	w/o Conflict Selection	32.17	11.13	22.61
	Baseline (MD)	32.79	12.47	20.33

with 17.26 TC and 23.73 EI, but ranks second in average TI behind ChatIE. This suggests that structured grounding and conflict resolution remain useful after candidate triggers are found, while trigger recall is more sensitive to the smaller model’s ability to follow cluster-conditioned extraction instructions.

F Comparison with Fine-tuned Method

As shown in Table 8, OGED shows stronger type grounding and unknown-event discovery than GoLLIE-7B. On MAVEN, OGED improves known-event TI, TC, and EI by +18.53, +9.71 and +9.83 F1 points, respectively, and increases overall EI from 15.00% to 26.31%. GoLLIE-7B obtains zero F1 on unknown events, whereas OGED achieves non-zero unknown-event discovery. On CASIE, GoLLIE-7B achieves high TI but very low TC/EI, while OGED raises overall EI from 2.45% to 21.94% and reaches 21.82% EI on unknown events. This suggests that GoLLIE-7B can detect spans in some cases, but fails at open-domain type grounding and discovery.

Table 9 presents the ablation results under the zero-shot event detection setting, complementing

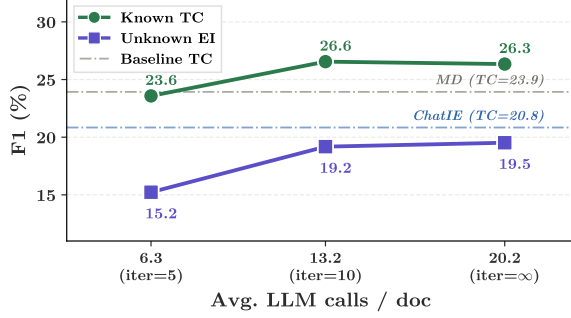


Figure 5: Budget-performance trade-off of OGED on MAVEN with 30% incompleteness. Known TC and Unknown EI are reported at varying iteration budgets. Dashed lines indicate baseline Known TC for reference. At iter=10 (13 calls/doc), OGED surpasses all baselines on Known TC while maintaining stable unknown event discovery.

the open-domain ablation in the main paper.

G Zero-shot Ablation Study

In the zero-shot setting, Conflict Selection is the most important component. Removing it reduces MAVEN TI by 18.91 F1 and CASIE TI by 5.98, showing that cluster-wise detection requires global arbitration to remove overlapping or redundant candidates. Planner Instructions also improve extraction quality, with their removal lowering TC by 1.28 F1 on MAVEN and 2.83 F1 on CASIE. The Scheduler is less influential in the zero-shot setting than in the open-domain setting, since its main role is to prioritize relevant but unresolved ontology regions for potential discovery. When the task is limited to known-type detection, this discovery-oriented scheduling signal becomes less central.

H Impact of the iteration budget

We examine the impact of the iteration budget $k_{\max}$ on cost and performance. Unless otherwise specified, experiments and call-breakdown analyzes use $k_{\max} = 10$. As shown in Fig. 5, OGED achieves most of its gains within the first 10 iterations. Increasing the budget from 5 to 10 improves both known-type detection and unknown-type discovery, while further increasing the budget yields only marginal additional gains.

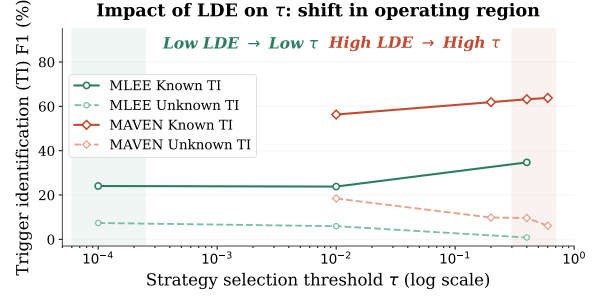


Figure 6: Impact of discovery threshold τ on TI performance. Optimal τ correlates with data entropy. Specialized domains (low entropy) favor lower thresholds for discovery, while diverse domains (high entropy) require higher thresholds for stability.

I Parameter Analysis

I.1 Implication of Strategy Selection Threshold τ

In our framework, the threshold τ governs the trade-off between detection and discovery. We argue that its optimal value correlates with the **Label Distribution Entropy (LDE)**, denoted as $\mathcal{H}_{\text{dist}}$, which quantifies the semantic dispersion of event types:

$$\mathcal{H}_{\text{dist}}(\text{Dataset}) = - \sum_{c \in \mathcal{C}} p(c) \log p(c)$$

where $p(c)$ is the proportion of events falling under semantic cluster c . A high $\mathcal{H}_{\text{dist}}$ indicates a scattered semantic space, while a low $\mathcal{H}_{\text{dist}}$ implies concentration in specific subtrees.

Fig. 6 analyzes how the planner threshold τ relates to dataset-level semantic concentration. Lower-entropy datasets concentrate event mentions within fewer ontology regions, so a smaller threshold leaves more room for discovering missing types that are close to existing ones. This explains why MLEE ($\mathcal{H}_{\text{dist}} = 1.52$) benefits from $\tau = 10^{-4}$, while $\tau = 0.4$ suppresses many discovery cases. Higher-entropy datasets spread event mentions across more diverse ontology regions, so a larger threshold is needed to avoid opening discovery for weakly supported clusters. MAVEN follows this pattern, with $\mathcal{H}_{\text{dist}} = 2.92$ and stronger performance at $\tau = 0.4$. Thus, the analysis supports our domain-level choice of a smaller τ for specialized domains and a larger τ for broad-domain datasets. It is worth noting that LDE is used only as an explanatory diagnostic.

I.2 Impact of Retrieval Number p

We investigate the model’s sensitivity to the number of retrieved event types p . As shown in Fig. 7, OGED outperforms the baseline MD across all p values on MAVEN, which demonstrates the robustness of our framework. Regarding the impact of p , increasing p from 10 to 15 improves TI on MAVEN. However, a further increase in p to 20 introduces a slight performance decline. Consequently, we adopt $p = 15$ as the optimal setting to balance candidate coverage and noise control.

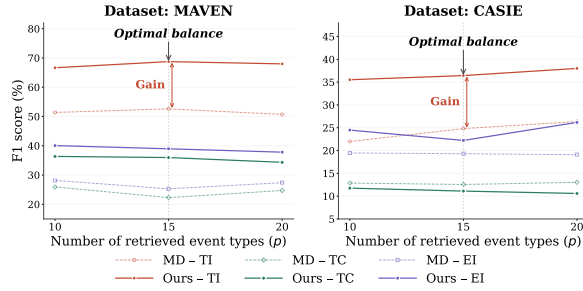


Figure 7: Sensitivity analysis of the number of retrieved event types (p) on the MAVEN and CASIE.

J Robustness Analysis

Table 10 evaluates robustness to additional ontology noise beyond the original ontology. The perturbation has two dimensions, where ρ controls the fraction of selected leaf nodes moved to incorrect clusters, while distance controls how semantically distant the target clusters are from the original ones. Thus, near perturbations move leaves to relatively similar clusters, whereas far perturbations move them to semantically distant clusters. We additionally report NameQ to measure the semantic quality of induced unknown type descriptions. Given the matched unknown true-positive pairs $\mathcal{M}_{TP}^{\text{unk}}$, we define

$$\text{NameQ} = \frac{1}{|\mathcal{M}_{TP}^{\text{unk}}|} \sum_{(\hat{t}, t) \in \mathcal{M}_{TP}^{\text{unk}}} \text{sim}(\hat{t}, t),$$

where $\text{sim}(\hat{t}, t)$ is the embedding cosine similarity between the predicted and gold type name-definition texts. If no unknown true-positive pair is matched, NameQ is set to 0. Light perturbations cause only moderate changes. Under heavy perturbation, OGED maintains stable known-event performance, and Unknown EI changes from 9.18% to 17.22% (+8.04 F1). However, NameQ drops from 0.440 to 0.371 (-0.069), suggesting that rewritten cluster names and definitions mainly affect the

Table 10: Robustness under additional ontology noise. The unperturbed setting uses the original constructed ontology without extra perturbation. Light perturbation moves selected leaf nodes to incorrect clusters, where ρ is the leaf-misplacement rate and distance indicates whether the target cluster is semantically near or far from the original one. Heavy perturbation further rewrites cluster names and definitions. Results are reported on the same randomly selected 200-document MAVEN subset. NameQ denotes the average embedding similarity between predicted and gold type names/definitions over matched unknown true-positive pairs.

Mode	Setting	Known F1			Unknown F1 / Quality			
		TI	TC	EI	TI	TC	EI	NameQ
Light	Base ontology	66.75	35.71	39.53	7.92	7.92	13.86	0.470
Light	$\rho = 0.20$, far	65.78	35.26	38.84	4.17	4.17	10.42	0.481
Light	$\rho = 0.20$, near	67.22	35.02	38.43	5.10	5.10	10.20	0.438
Light	$\rho = 0.40$, far	65.12	35.98	40.28	5.10	5.10	9.18	0.480
Heavy	Base ontology	66.07	37.43	40.23	5.10	5.10	9.18	0.440
Heavy	30-swap + rewrite	65.76	38.24	41.09	6.70	6.70	17.22	0.371

semantic alignment of induced type descriptions rather than event recovery itself.

A case in Fig. 8 illustrates why OGED can remain robust under heavy ontology noise. The perturbation moves the gold type `Social_event` into an incorrect financial cluster and rewrites the affected cluster names and definitions, which suppresses the cluster-level prior of F19. However, the leaf-level event definition of `Social_event` is still preserved. In the execution trace, F19 is selected at step 4 because its exploration bonus raises the scheduler score to 2.000 despite a near-zero prior. Its semantic gap remains below $\tau = 0.4$, so the planner chooses detection rather than discovery. Once F19 is visited, the detector recovers `Social_event` from the trigger “festival” using the preserved leaf-level semantics. This case shows that noisy cluster organization can delay a relevant type, but exploration can still expose it to the detector.

K Representative Prompt Templates

We provide representative prompt templates for the components that directly determine event outputs: the Detector, Retrieval-Gated Judge, Conflict Selection, and the MD-Open baseline. The Planner prompt is omitted for brevity because it only generates intermediate strategy instructions rather than final event predictions; its inputs and outputs are described in Sec. 4.3. The complete prompts are included in the code.

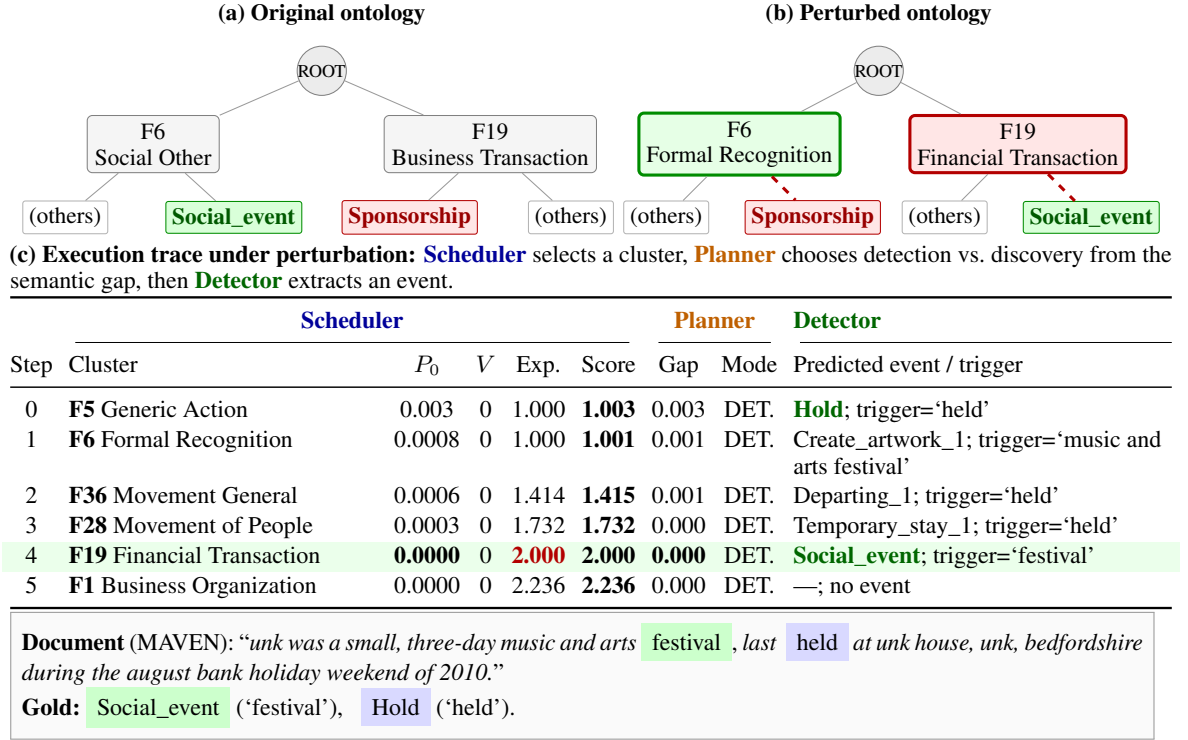


Figure 8: Case study under heavy ontology noise. (a) The original ontology places the gold leaf Social_event under cluster F6 and an unrelated leaf Sponsorship under cluster F19. (b) The perturbation swaps the two leaves and rewrites the affected cluster names and definitions. (c) The execution trace records scheduler, planner, and detector decisions. Here, P_0 is the cluster prior, V is the explained score, Exp. is the exploration term, Score is the scheduler score, Gap is the planner semantic gap, and Mode denotes the planner strategy, where DET. means detection. Despite the near-zero prior of the perturbed F19 cluster, exploration eventually selects it, and the detector recovers Social_event from “festival” using preserved leaf-level semantics.

K.1 Detector Prompts

The Detector runs with two prompt variants corresponding to the planner’s strategy. Under DETECTION in Fig. 9, the prompt presents enumerated candidate event types and the document; the planner-generated hint is injected when available. Under DISCOVERY in Fig. 10, the detector operates as an ontology designer, where it scans the entire document for action verbs, event-denoting nouns, and state-change indicators, filters them against the target domain, and proposes new event types with generalizable one-sentence definitions.

Detector — DETECTION Mode
System You are an expert Event Extraction Agent.
Event Type Candidates {event_types_section} # Document Text {document_text} # Task Instructions Goal: Extract events that match the Candidate definitions.

{hint_section}
Rules
1) Trigger Selection: Trigger words MUST come directly from the document text. Select the minimal span, prefer SHORT triggers of 1-3 words, and avoid long phrases. 2) Semantic Matching: Extract events that align with the Candidate definitions. 3) Context: Use document context to determine event semantics. 4) Format: Output valid JSON. If no event matches, return an empty list.
Output Format
<pre>{ "events": [{"event_type_name": ..., "trigger_text": ...}, ...] }</pre>

Figure 9: Detector prompt under the DETECTION mode. The placeholders are instantiated with candidate event types, the current document text, and the planner-generated hint.

```
Detector — DISCOVERY Mode

System
You are an expert event extraction specialist and ontology designer.

-----

# Task
Discover and define new event types by analyzing a document within the {cluster_name} domain.

# Document Text
{document_text}

# Domain Context
Target Domain: {cluster_name}
Domain Definition: {cluster_definition}

# Planner Hint
{hint_section}

Instructions

1)Comprehensive Scan: Systematically scan the ENTIRE document for action verbs, event-denoting nouns, and state-change indicators appearing verbatim.
2)Domain Filtering: Select triggers that semantically align with the {cluster_name} domain.
3)Novelty: Exclude already-extracted tokens and triggers already covered by existing event types.
4)Generalizability: Each proposed event type must generalize beyond this specific document.
5)Definition Generation: For each novel, domain-relevant trigger, produce a brief and generalizable event type definition.

Constraints
Extract ONLY triggers (prefer 1-3 words) that exist verbatim in the document text. No abstract conceptual terms not grounded in document lexicon. Exclude tokens already marked with asterisk prefix. Event definitions must be generalizable, not document-specific

Output Format
{
  "new_events":      [{"event_type_id":      "NEW:
<TypeName>",        "trigger_text":        ...},
  "new_type_definition":  ..., "reason":    ...},
  ...]
}
```

Figure 10: Detector prompt under the DISCOVERY mode. The detector identifies novel domain-relevant triggers and proposes a generalizable event type definition for each discovered event type.

K.2 Retrieval-Gated Judge Prompt

The retrieval-gated verifier is used for each proposed new event type that passes the similarity threshold. The Judge receives the proposal alongside its top-5 nearest existing types and decides whether to accept the new type or map the trigger to an existing one. Fig. 11 shows the prompt template.

```
Judge — RETRIEVAL-GATED VERIFIER

System
You are an expert in event ontology design and semantic analysis. Your task is to determine whether a proposed new event type can be adequately represented by any existing event type in the ontology, or if it requires a genuinely new category.

-----

# Task
Given a trigger word in a document, determine if any of the Existing Candidate Event Types can adequately express this event, or if we need to create the Proposed New Event Type.

# Document Context
{context}

# Proposed New Event Type
Name: {new_type_name}
Definition: {new_type_definition}

# Existing Candidate Event Types
{candidates}

Instructions

1)Decision Criteria:
-EXISTING TYPE CAN EXPRESS: Choose this if any candidate can adequately label this event instance.
*The trigger clearly fits an existing type’s definition.
*Using the existing type would not lose critical semantic information.
-NEW TYPE NEEDED: Choose this if none of the candidates adequately capture the event.
*The event represents a fundamentally different concept.
*Key semantic nuances would be lost by using existing types.
2)Output Requirements:
-If an existing type can express this event, set “is_same” to true, provide “matched_candidate_id” as a number from 1 to {N}, and explain why.
-If a new type is needed, set “is_same” to false and explain what is missing from the existing types.

Output Format
{
  "is_same": true or false,
  "matched_candidate_id": <number 1-{N} or null>,
  "reasoning": "Detailed explanation of the decision (2-4 sentences)"
}
```

Figure 11: Judge prompt used by the gatekeeper to determine whether a proposed new event type can be adequately represented by existing candidate event types or should be retained as a genuinely new type.

K.3 Conflict Selection Prompt

The Conflict Selection prompt is used after candidate generation to resolve overlapping trigger assignments proposed by different clusters. Given the full document and a conflict group, the prompt

asks the model to select a compact final subset of candidates, as shown in Fig. 12.

Conflict Selection

System

You are an expert event detection adjudicator. Given a document and candidate event types for the same trigger region, select the best matching event type(s) based on the trigger's meaning in context.

Document

{document_text}

Candidate Groups

Each group contains candidates whose trigger spans have $\text{IoU} \geq 0.2$, corresponding to the same trigger region.

Task

CRITICAL: You MUST provide a selection result for EVERY SINGLE group listed below.
For EACH of the {N_groups} groups, select 1-2 candidates:

- Select ONE by default, namely the single best event type for the trigger in context.
- Select TWO ONLY if the document context strongly supports that both event types are genuinely appropriate for the same trigger region, meaning two independently defensible interpretations or events, and you cannot reasonably disambiguate in favor of one.
- Do NOT select two just because both are somewhat plausible or similar.

IMPORTANT: Your response MUST include exactly {N_groups} entries in the "selected" array, one for each group_id: {expected_group_ids}.

Output Format

```
{
  "selected": [
    {example_selections},
    ...
  ],
  "reasoning": "2-3 sentences overall"
}
```

Return JSON only. No extra text.

Groups

{blocks}

Figure 12: Conflict selector prompt. Given groups of event candidates with overlapping trigger spans, the selector chooses the best matching candidate event type for each group, or retains two candidates only when the document context supports two independently defensible types.

K.4 MD-Open Baseline Prompt

MD-Open uses a single-pass prompt, as shown in Fig. 13, which asks the model to ground existing event types and generate a new event type when no existing type is appropriate. This baseline directly tests whether open-domain event detection can be solved by simply allowing free-form new labels in

a flat prompt.

MD-Open Detector

System

You are an expert Event Extraction Agent.

Event Type Candidates

{event_types_section}

Document Text

{visualized_text}

Task Instructions

Goal: Extract events that match the Candidate definitions.

Rules

- 1)**Trigger Selection:** Trigger words MUST come directly from the document text. Select the minimal span, prefer SHORT triggers of 1-3 words, and avoid long phrases.
- 2)**Semantic Matching:** Extract events that align with the Candidate definitions.
- 3)**Context:** Use document context to determine event semantics.
- 4)**Format:** Output valid JSON. If no event matches, return an empty list.
- 5)**Propose New Types:** If a clear event is present in the text but **none** of the Candidate definitions fit, you MAY propose a new event type instead of forcing a bad match.
 - Use "event_type_name" formatted as "NEW: <TypeName>".
 - Provide a brief, generalizable "new_type_definition", not one specific to this document.
 - The new type must be semantically distinct from every Candidate above, not a synonym or sub-case.
 - Prefer reusing a Candidate when there is any reasonable fit; only propose a new type when truly needed.

Output Format

Return a JSON object with an "events" array. Each element should follow: {

```
"events": [
  {
    "event_type_name": "EventTypeName OR NEW: NewTypeName",
    "trigger_text": "exact text from document",
    "new_type_definition": "Brief generalizable definition. Omit or set to empty string for existing types.",
    "reasoning": "Why this fits the definition, or why none of the candidates fit and a new type is warranted"
  }
]
```

Final Constraint

If no events match and no new type is warranted, return { "events": [] }.

Figure 13: Baseline detector prompt, which allows the detector to either assign an existing candidate event type or directly propose a new event type within the same generation step.