EXCEEDS: Extracting Complex Events via Nugget-based Grid Modeling in Scientific Domain

Yi-Fan Lu[1], **Xian-Ling Mao**[1*], **Bo Wang**[1], **Xiao Liu**[2], **Heyan Huang**[1]
[1]Beijing Institute of Technology, [2]Microsoft Research Asia
yifanlu@bit.edu.cn, maoxl@bit.edu.cn, bwang@bit.edu.cn,
xiaoliu2@microsoft.com, hhy63@bit.edu.cn

Abstract

It is crucial to understand a specific domain by events. Extensive event extraction research has been conducted in many domains such as news, finance, and biology. However, event extraction in scientific domain is still insufficiently supported by comprehensive datasets and tailored methods. Compared with other domains, scientific domain has two characteristics: (1) denser nuggets and events, and (2) more complex information forms. To solve the above problem, considering these two characteristics, we first construct SciEvents, a large-scale multi-event document-level dataset with a schema tailored for scientific domain. It consists of 2,508 documents and 24,381 events under multi-stage manual annotation and quality control. Then, we propose EXCEEDS, an end-to-end scientific event extraction framework by encoding dense nuggets into a grid matrix and simplifying complex event extraction as a nugget-based grid modeling task. Experiments on SciEvents demonstrate state-of-the-art performances of EXCEEDS. Both the SciEvents dataset and the EXCEEDS framework are released publicly to facilitate future research.[1]

1 Introduction

Event extraction (EE) is a fundamental information extraction task aiming to extract structural event knowledge from plain texts (Peng et al., 2023). It is typically decomposed into two pipeline subtasks: event detection (ED) and event argument extraction (EAE). Specifically, ED identifies a word span (hereafter referred to as a **nugget**) that most clearly refers to the occurrence of an event, *i.e.*, event trigger, and also detects the event type evoked by the event trigger (Pouran Ben Veyseh et al., 2022). Given an event trigger and its event type, EAE further identifies nuggets as event arguments and classifies their roles in the event.

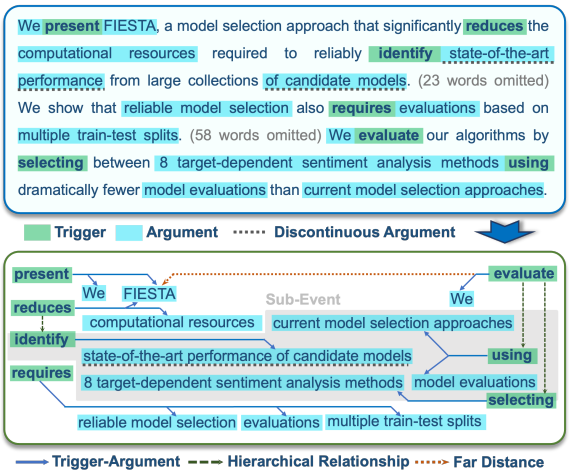


Figure 1: A real example from SciEvents. The upper panel displays a scientific paper abstract, and the lower panel shows the extracted events, highlighting the dense information and complex event structures characteristic of scientific texts.

EE provides an effective abstraction for representing domain knowledge and supports downstream tasks such as reasoning, summarization, and knowledge discovery. Consequently, extensive EE research has been conducted in various scenarios and domains such as internet news, radio conversations, internet blogs (Sundheim, 1992; Aguilar et al., 2014; Ebner et al., 2020), business (Yang et al., 2018; Liang et al., 2020), biology (Kim et al., 2011; Pyysalo et al., 2012), legislation (Shen et al., 2020; Yao et al., 2022), cybersecurity (Man Duc Trong et al., 2020; Satyapanich et al., 2020) and so on. Despite this progress, scientific literature has been growing rapidly in recent decades, with millions of new publications released every year. Such growth poses an urgent challenge for managing scientific domain knowledge, calling for effective EE solutions.

However, EE in scientific domain remains insufficiently characterized by existing datasets and methods. In particular, current resources and for-

*Corresponding Author.
[1]https://github.com/HammerScholar/EXCEEDS

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 5997–6022
July 2-7, 2026

mulations struggle to capture two salient characteristics of scientific texts. **First**, compared with other domains, scientific domain tends to contain **more complex information forms**. Although many EE methods are task-specialized and rely on domain-specific ontologies (Lu et al., 2022), these ontologies typically adopt flat tabular schema, which (1) neglects the hierarchical structure of events, (2) restricts the continuity of arguments, and (3) complicates the coreference problem, while these complex information forms are common in scientific literature. For example in Figure 1, (1) the trigger *evaluate* and the trigger *using* form a hierarchical relationship; (2) the trigger *identity* connects a discontinuous argument *state-of-the-art performance of candidate models*; (3) the trigger *evaluate* connects an argument *FIESTA* with a far distance from itself. These three examples demonstrate complex nuggets and events in scientific domain. **Second**, scientific texts, especially literature abstracts, tend to contain **denser nuggets and events** (with statistical evidence presented in Section 3.2). Unlike many existing datasets that focus on sentence-level extraction or single-event documents, scientific domain EE requires modeling dense, document-level multi-event interactions (see Figure 1 and Section 3.2). Together, these two characteristics motivate the need for dedicated EE resources and methods for scientific domain.

To explore the unique characteristics of scientific domain EE, we first introduce SciEvents, a large-scale document-level multi-event dataset tailored for scientific literature. SciEvents contains 2,508 manually annotated abstracts with 24,381 events, and is constructed with a refined schema designed to capture dense and structurally complex event patterns. Dataset statistics show that SciEvents exhibits both **denser nugget distributions** and **more complex event structures** than existing domain-specific EE datasets, reflecting the information-intensive nature of scientific texts.

Denser nuggets and more complex events pose two fundamental challenges to existing EE methods. On the one hand, the high density requires models to capture global information, rather than extracting events only at the sentence level or as isolated instances. On the other hand, the complexity of scientific events calls for models that can represent hierarchical relationships, handle discontinuous nuggets, and associate triggers with arguments across long textual distances. However, most existing EE approaches are developed under assumptions of non-hierarchical structures and locally bounded contexts, which limit their effectiveness in modeling the complex event patterns commonly observed in scientific texts.

To address these challenges, we further propose **EXCEEDS**, an end-to-end framework to **EX**tract **C**omplex **E**vents via nugg**E**t-based gri**D** modeling in **S**cientific domain. EXCEEDS represents pairwise token relations across the entire document in a word-word event grid, enabling unified modeling of dense multi-event contexts as well as complex nugget and event structures, including hierarchical relations, discontinuous arguments, and long-distance dependencies. This formulation allows EXCEEDS to effectively address the challenges posed by scientific texts under a single end-to-end framework.

We evaluate state-of-the-art and recent EE methods on SciEvents under an extended evaluation protocol that incorporates an event correlation metric for hierarchical EE. Experimental results show that EXCEEDS achieves consistently strong performance across tasks, while further analysis reveals that complex nugget structures, especially under dense scientific contexts, remain challenging for existing models.

In summary, our contributions are two-fold: (1) We introduce SciEvents, a large-scale document-level EE dataset for the scientific domain with a refined schema, providing a comprehensive benchmark for studying dense and complex scientific events. (2) We propose EXCEEDS, an end-to-end EE framework tailored to the challenges of density and structural complexity in scientific texts, which achieves state-of-the-art performance on SciEvents.

2 Related Works

Event Extraction Datasets EE datasets have been constructed across a wide range of domains. Early efforts mainly focus on the news domain, describing events in realistic scenarios (Walker et al., 2006; Aguilar et al., 2014; Song et al., 2015). General domain datasets are further built from diverse sources like Wikipedia (Wang et al., 2020; Li et al., 2021; Tong et al., 2022), Reddit (Ebner et al., 2020), Baidu news (Li et al., 2020c), FrameNet (Parekh et al., 2023) and multi-lingual candidate data (Pouran Ben Veyseh et al., 2022). In addition, domain-specific datasets have been developed for finance (Yang et al., 2018; Liu et al., 2019a), biomedicine and related fields (Kim et al., 2011;

Dataset	Domain	#Docs	#Tokens	#ETs	#Events	#ATs	#Arguments
ACE2005 (Doddington et al., 2004)	News	599	297,842	33	5,348	22	8,097
Genia2011 (Kim et al., 2011)	Biomedical	1,375	345,371	9	13,537	10	11,865
Genia2013 (Kim et al., 2013)	Biomedical	20	149,856	13	6,001	7	5,660
RAMS (Ebner et al., 2020)	News	9,124	1,218,622	139	9,124	65	21,237
CASIE (Satyapanich et al., 2020)	Cybersecurity	999	387,275	5	8,479	26	22,679
M2E2 (Li et al., 2020b)	Multimedia	6,013	169,990	8	1,105	15	1,659
WikiEvents (Li et al., 2021)	News	246	189,718	50	3,951	59	5,536
PHEE (Sun et al., 2022)	Drug Safety	4,827	106,447	2	5,019	16	25,760
Maccrobat-EE (Ma et al., 2023)	Clinical	200	107,130	14	13,128	22	8,599
SciEvents (Ours)	Science	2,508	439,890	10	24,381	20	56,411

Table 1: Basic statistics of widely-used domain-specific event datasets. This table only presents publicly available event datasets that include argument annotations. #ETs: number of event types. #ATs: number of argument types.

Sun et al., 2022; Ma et al., 2023), as well as other domains such as cybersecurity, law, and literature (Man Duc Trong et al., 2020; Shen et al., 2020; Sims et al., 2019). Despite these advances, event datasets for the scientific domain remain limited, and existing resources rarely analyze the characteristics of scientific texts. In this work, we systematically examine the characteristics of scientific abstracts and construct SciEvents, a document-level EE dataset tailored to the scientific domain.

Event Extraction Approaches EE has evolved from early sequence labeling methods to more advanced neural architectures. To jointly model heterogeneous elements in EE datasets (Peng et al., 2023), early work focuses on joint extraction frameworks that capture dependencies within and across events (Liu et al., 2018; Yang et al., 2019; Nguyen et al., 2021; Lin et al., 2020). Subsequent studies reformulate EE as machine reading comprehension, enabling more flexible trigger and argument extraction via question answering (Chen et al., 2020; Li et al., 2020a; Zhou et al., 2021; Wei et al., 2021). More recent approaches adopt sequence-to-structure generation with Transformer-based models, unifying ED and EAE within a single framework (Lu et al., 2021; Lou et al., 2023; Wang et al., 2023a; Liu et al., 2022; Yang et al., 2024). With the emergence of large language models (LLMs), EE has further benefited from strong generalization and zero-shot capabilities (Wei et al., 2023; Gao et al., 2023a; Wang et al., 2023b; Sainz et al., 2024; Gao et al., 2023b; Li et al., 2024). Despite these advances, many existing methods struggle to model structurally complex event mentions. Some work partially mitigates this problem by modeling token-level relations (Lou et al., 2023; Liu et al., 2023; Zhu et al., 2023), or by adopting a more universal information extraction paradigm (Lu et al., 2022; Li et al., 2024). However, these approaches either rely on span-boundary representations (Lou et al., 2023; Liu et al., 2023), require instruction-style inputs with schema conditioning (Zhu et al., 2023), or rely on multi-task and multi-dataset training (Lu et al., 2022; Li et al., 2024). In this paper, we propose EXCEEDS, an end-to-end pairwise token relation modeling framework over the entire document, using only raw text as input and targeting the EE-only, single-dataset setting.

3 The SciEvents Dataset

To support systematic research on scientific EE, we construct SciEvents, a large-scale document-level EE dataset tailored for scientific literature. In this section, we will introduce the dataset construction process in Section 3.1, and present a comprehensive statistical analysis in Section 3.2.

3.1 Dataset Construction Process

Schema Design Scientific abstracts are typically organized around four rhetorical components: *background*, *related work*, *methodology*, and *results*. Motivated by this regularity, we design an event schema comprising 10 event types that cover these components. For instance, abstracts often summarize prior approaches and highlight their limitations; we capture such information using the *RelatedWorkStep* and *RelatedWorkFault* event types, respectively.

For each event type, we define a set of argument types to encode the information that readers typically seek in scientific abstracts. To ensure both coverage and annotation feasibility, we develop the schema iteratively with domain experts. Specifically, two professors and three senior Ph.D. students in computer science annotate a set of seed documents and revise the schema over four rounds.

Dataset	Domain	Density: Every 100 Tokens Contains			Complexity: Complex Forms			
		#Events	#Args	#Nugget Tokens	#D(%)	#O(%)	#R(%)	#S(%)
ACE2005	News	1.80	2.72	4.62	–	13.88	–	–
Genia2011	Biomedical	3.92	3.44	9.31	–	36.62	–	–
Genia2013	Biomedical	4.00	3.78	8.58	–	34.00	–	–
RAMS	News	0.75	1.74	4.06	–	10.51	–	–
CASIE	Cybersecurity	2.19	5.86	17.80	–	1.40	–	–
M2E2	Multimedia	0.65	0.98	1.89	–	6.08	–	–
WikiEvents	News	2.08	2.92	5.69	–	8.19	–	–
PHEE	Drug Safety	4.72	24.20	53.43	–	64.03	–	–
Maccrobat-EE	Clinical	12.25	8.03	38.65	–	3.71	–	–
SciEvents (Ours)	Science	5.54	12.82	39.49	3.08	33.70	1.01	25.63

Table 2: Statistics of density and complexity of widely-used domain-specific event datasets. This table only presents publicly available event datasets that include argument annotations. #D: Discontinuous nugget. #O: Overlapping nugget. #R: Reverse-order nugget. #S: Sub-event.

The final schema and detailed definitions of all event types and argument types are provided in Appendix F. We collect papers from the recent 4 years (2019-2022) ACL main conference paper abstracts as candidate data.

Annotation and Quality Control During the pre-annotation stage, we train three supervisors and some candidates, resulting in seven qualified annotators for the official annotation. For reproducibility, detailed descriptions of the annotation protocol are provided in Appendix I.

Quality inspection is conducted by three supervisors and two well-performing annotators. The annotator and the inspector of a document are strictly separated. If a document contains more than two annotation conflicts (including missing annotations), it is returned to the original annotator together with detailed revision comments provided by a quality inspector. Otherwise, minor conflicts are corrected by the inspection team, and corresponding feedback is still provided to annotators to facilitate continuous improvement. The first-pass inspection acceptance rate is 73.05%. A fully annotated example document can be found in Appendix G.

3.2 Dataset Statistics Analysis

This section will provide a comprehensive statistical analysis of SciEvents, with particular emphasis on information density and complexity.

Basic Statistics Table 1 presents basic statistics of SciEvents and other widely-used domain-specific EE datasets covering diverse domains. Among these datasets, SciEvents is distinguished as a large-scale dataset specifically constructed for the scientific domain. In terms of annotation scale, SciEvents 24,381 event instances and 56,411 arguments, substantially exceeding most existing domain-specific event datasets. Notably, SciEvents achieves this scale with only 10 event types. This suggests that the large number of event instances in SciEvents primarily arises from frequent event occurrences within scientific documents, rather than an expanded or fine-grained schema, reflecting the information-intensive nature of scientific texts. Statistics can be found in Appendix H.

Information Density Statistics As shown in Table 2, SciEvents exhibits high information density under all token-normalized metrics, with 5.54 events, 12.82 arguments, and 39.49 nugget tokens per 100 tokens, indicating that a large proportion of tokens in scientific documents directly participate in event expressions.

Among other domain-specific datasets, similarly high density values are mainly observed in medical datasets such as PHEE and Maccrobat-EE, whose documents describe inherently information-intensive content (*e.g.*, drug safety reports and clinical records). By contrast, remaining datasets generally exhibit substantially lower densities. Overall, these results suggest that the elevated density of SciEvents reflects intrinsic properties of scientific texts under domain-specific settings.

Information Complexity Statistics The right part of Table 2 reports the proportions of complex event forms. Unlike most existing domain-specific datasets that mainly annotate contiguous nuggets, SciEvents explicitly covers diverse complex structures, including overlapping, discontinuous, reverse-order nuggets, and sub-events. Specif-

ically, SciEvents contains a substantial proportion of overlapping nuggets (33.70%) and sub-events (25.63%), together with non-negligible occurrences of discontinuous (3.08%) and reverse-order nuggets (1.01%). By contrast, other datasets only partially capture overlapping structures. These statistics reflect the structural complexity of scientific events and highlight the challenges they pose for EE models.

4 Event Extraction Problem Formulation

In domain-specific EE, a predefined schema is given as $S = \{T_E, T_A\}$, where T_E denotes an event type set and T_A denotes an argument type set. Each event type $t_e \in T_E$ is associated with a specific argument type set $T_A(t_e)$. Given a document D, EE aims to extract a set of events $E = \{e_1, e_2, \ldots, e_M\}$ in D, where each event $e = \{t_e, t, A\}$ consists of an event type $t_e \in T_E$, a trigger t and a set of arguments $A = \{a_1, a_2, \ldots, a_N\}$. Each argument $a = \{t_a, m\}$ consists of an argument type $t_a \in T_A(t_e)$ and a word span m. Both triggers and arguments are referred to as nuggets, whose word spans should be combinations of tokens in D. In SciEvents, we add an event correlation task to extract hierarchical event structures. Specifically, the trigger t_s of a sub-event $e_s = \{t_{se}, t_s, A_s\}$ will be regarded as an argument of a main-event $e_m = \{t_{me}, t_m, A_m\}$ with a certain argument type t_{sa}, *i.e.* $\{t_{sa}, t_s\} \in A_m$.

5 The EXCEEDS Method

To address the challenges of high density and complexity in scientific EE, we propose EXCEEDS, an end-to-end framework that simplifies EE into a nugget-based grid modeling task. Section 5.1 introduces the word-word event grid construction, Section 5.2 presents the overall framework, and Section 5.3 describes the training objectives and inference procedure.

5.1 Word-Word Event Grid

To effectively capture the complex structures of nuggets and events, we encode relations within a nugget and across different nuggets through a word-word grid. Formally, given a document $D = \{x_1, x_2, \ldots, x_l\}$, we construct an $l \times l$ grid G, where each cell $G[i, j]$ stores the relation type $r \in R$ between token x_i (row) and token x_j (column). Specifically, within a nugget, we use head-tail-link (HTL) to represent the successive order

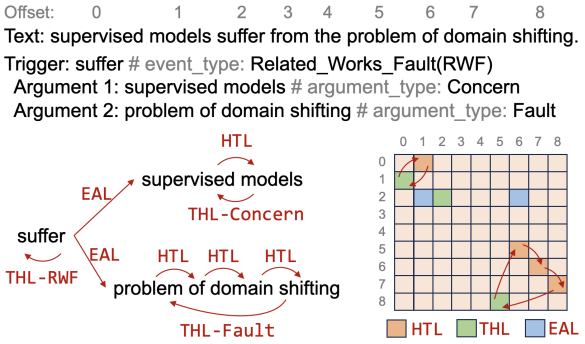


Figure 2: Illustration of the word-word event grid. HTL captures successive token order within a nugget; THL connects the last and first tokens to indicate nugget types; and EAL encodes relations across nuggets. The grid matrix (*right*) presents these relations.

between adjacent tokens and tail-head-link (THL) to connect the last token back to the first token, which conveys nugget type information. Across different nuggets, we define event-argument-link (EAL) to represent the relation between a trigger and its argument, or between an event trigger and a sub-event trigger. For example, Figure 2 shows how HTL, THL and EAL are instantiated in SciEvents, and how they are encoded into the corresponding cells of the grid, resulting in a unified representation for nuggets and events.

The benefits of this grid and relation design are threefold. First, it enables encoding of complex nugget structures within a document, including overlapping, discontinuous, and reverse-order nuggets. Second, it provides a unified formulation of event detection and event argument extraction in an end-to-end manner, allowing the framework to fully leverage contextual information without relying on separate pipeline modules. Third, it naturally captures hierarchical event relations by encoding relations between trigger pairs.

5.2 The Overall Framework

Figure 3 illustrates the overall architecture of our framework. Given a document, the model encodes contextual token representations and constructs a word-word grid to jointly model nugget structures and event relations in an end-to-end manner.

Contextual Token Encoding We encode the input document using a pretrained language model (Liu et al., 2019b) to obtain initial contextual representations, which are further refined by a bidirectional LSTM (Huang et al., 2015) to capture sequential dependencies. The resulting representa-

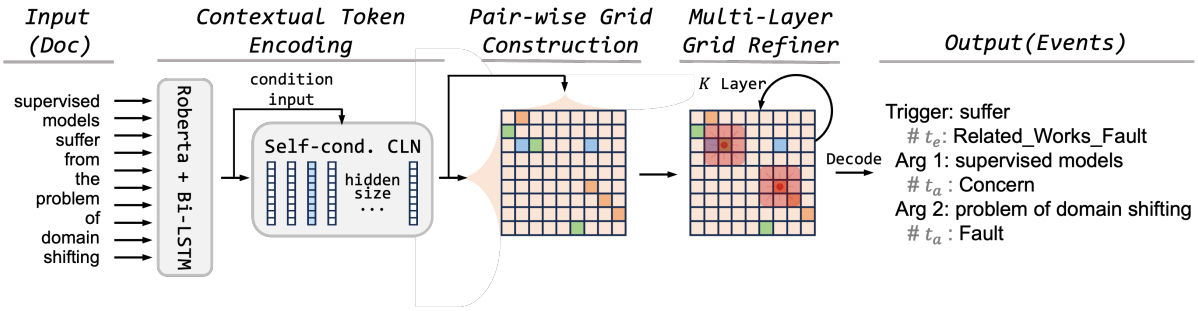


Figure 3: Overall architecture of EXCEEDS. The model encodes contextual token representations, constructs a word-word event grid to model pairwise relations, refines the grid, and decodes events from the refined grid.

tions are normalized using Conditional Layer Normalization (CLN) (Liu et al., 2021) to enhance stability and contextual adaptability. Specifically, given the LSTM outputs $\mathbf{L} \in \mathbb{R}^{l \times d}$, CLN performs normalization with dynamically generated affine parameters conditioned on the contextual representations themselves:

$$\mathbf{H} = \text{MLP}_\gamma(\mathbf{L}) \odot \frac{\mathbf{L} - \mu}{\sigma + \epsilon} + \text{MLP}_\beta(\mathbf{L}), \quad (1)$$

where μ and σ denote the mean and standard deviation computed along the feature dimension, ϵ is a smoothing parameter, and $\odot$ denotes element-wise multiplication. The output $\mathbf{H} \in \mathbb{R}^{l \times d}$ serves as the contextualized word representations for subsequent pair-wise grid construction.

Pair-wise Grid Construction Given the contextualized word representations $\mathbf{H}$, we construct a word-word grid $\mathbf{G} \in \mathbb{R}^{l \times l \times C_g}$, where each cell corresponds to relation of a token pair (x_i, x_j).

For each pair, we form a pair-wise representation by concatenating the token representations with a relative distance embedding:

$$\mathbf{z}_{i,j} = [\mathbf{h}_i \, ; \, \mathbf{h}_j \, ; \, \mathbf{d}_{i,j}], \quad (2)$$

which is projected into the grid feature space via a multilayer perceptron:

$$\mathbf{g}_{i,j} = \text{MLP}_{\text{pair}}(\mathbf{z}_{i,j}). \quad (3)$$

The resulting $\mathbf{g}_{i,j} \in \mathbb{R}^{C_g}$ constitutes the initial grid representation.

Grid Refiner The initial grid representations encode pair-wise token relations independently. To enable information propagation across related token pairs, we refine the grid with a stack of lightweight residual refinement blocks operating on the grid space.

Let $\mathbf{G}^{(0)} = \mathbf{G}$ and $\mathbf{G}^{(k)} \in \mathbb{R}^{l \times l \times C_g}$ denotes the grid features after the k-th refinement layer. Each block updates the grid by aggregating information from local neighborhoods and applying a residual transformation:

$$\mathbf{G}^{(k+1)} = \text{Norm}\Big(\mathbf{G}^{(k)} + \mathcal{F}\Big(\mathbf{G}^{(k)}\Big)\Big), \quad (4)$$

where $\mathcal{F}(\cdot)$ denotes a learnable local aggregation function on the grid, and is instantiated as stacked 2D convolutional refinement blocks in our implementation. After K refinement layers, we obtain the refined grid representations $\tilde{\mathbf{G}} = \mathbf{G}^{(K)}$.

5.3 Training and Inference

Loss Function Given $\tilde{\mathbf{G}} \in \mathbb{R}^{l \times l \times C_g}$, we project each grid cell to relation logits via a linear classifier, yielding $\mathbf{Y} \in \mathbb{R}^{l \times l \times |R|}$. Since multiple relation types may simultaneously hold for a token pair, we formulate grid prediction as a multi-label classification problem.

We adopt a multi-label categorical cross-entropy loss (Su et al., 2022), which jointly optimizes positive and negative labels without requiring a predefined number of active labels. Formally, for each grid cell (i, j), the loss is defined as

$$\mathcal{L}_{i,j} = \log(1 + \sum_{r \in \Omega^-} e^{y^r_{i,j}}) + \log(1 + \sum_{r \in \Omega^+} e^{-y^r_{i,j}}), \quad (5)$$

where Ω^+ and Ω^- denote the sets of positive and negative relation types for (x_i, x_j), respectively.

Inference Following the multi-label formulation, we obtain the predicted relation set for each grid cell by a zero-threshold decision:

$$\hat{\mathbf{M}}_{i,j,r} = \mathbb{I}\left[y^r_{i,j} > 0\right], \quad (6)$$

where $\hat{\mathbf{M}} \in \{0, 1\}^{l \times l \times |R|}$ is the binary word-word relation grid. We then decode $\hat{\mathbf{M}}$ into a set of events

Model		TI	TC	AI	AC	EC
Global	OneIE (Lin et al., 2020)	**75.72**$_{\pm 0.14}$	62.93$_{\pm 0.17}$	30.30$_{\pm 1.85}$	28.81$_{\pm 1.77}$	37.41$_{\pm 1.51}$
	ScentedEAE† (Yang et al., 2024)	73.27$_{\pm 0.52}$	63.03$_{\pm 0.20}$	36.70$_{\pm 2.99}$	35.74$_{\pm 2.41}$	37.88$_{\pm 2.10}$
Discriminative	EEQA (Du and Cardie, 2020)	74.85$_{\pm 0.78}$	62.15$_{\pm 0.73}$	37.75$_{\pm 0.46}$	35.64$_{\pm 0.59}$	44.81$_{\pm 1.35}$
	PAIE† (Ma et al., 2022)	73.27$_{\pm 0.52}$	63.03$_{\pm 0.20}$	43.92$_{\pm 0.22}$	42.06$_{\pm 0.34}$	47.17$_{\pm 0.92}$
	Tagprime (Hsu et al., 2023)	73.27$_{\pm 0.52}$	63.03$_{\pm 0.20}$	44.67$_{\pm 0.13}$	42.69$_{\pm 0.32}$	47.72$_{\pm 0.32}$
	DEEIA† (Liu et al., 2024)	73.27$_{\pm 0.52}$	63.03$_{\pm 0.20}$	34.86$_{\pm 0.81}$	33.30$_{\pm 0.75}$	32.80$_{\pm 1.67}$
Generative	BartGen† (Li et al., 2021)	73.27$_{\pm 0.52}$	63.03$_{\pm 0.20}$	39.85$_{\pm 0.52}$	37.81$_{\pm 0.39}$	42.75$_{\pm 0.11}$
	DEGREE (Hsu et al., 2022)	65.72$_{\pm 0.70}$	53.52$_{\pm 0.75}$	28.40$_{\pm 0.88}$	26.32$_{\pm 0.79}$	29.53$_{\pm 0.96}$
	KnowCoder (Li et al., 2024)	69.88$_{\pm 0.61}$	52.02$_{\pm 0.34}$	35.24$_{\pm 0.11}$	33.43$_{\pm 0.27}$	34.54$_{\pm 0.82}$
EXCEEDS (Ours)		75.29$_{\pm 0.32}$	**63.74**$_{\pm 0.14}$	**44.97**$_{\pm 0.28}$	**43.20**$_{\pm 0.29}$	**48.25**$_{\pm 0.10}$
– Contextual		75.29$_{\pm 0.21}$	63.44$_{\pm 0.67}$	44.07$_{\pm 0.51}$	42.14$_{\pm 0.39}$	47.64$_{\pm 0.85}$
– Grid Refiner		75.36$_{\pm 0.27}$	63.41$_{\pm 0.67}$	44.30$_{\pm 0.51}$	42.44$_{\pm 0.59}$	48.04$_{\pm 1.07}$

Table 3: Overall F1-score (%) on SciEvents. For †EAE-only models, trigger predictions are derived from Tagprime, which achieves the best ED performance among all baseline methods.

by reconstructing nuggets and linking arguments to triggers. Specifically, we (1) recover nugget spans by traversing HTL and validating them with a closing THL-type, and (2) attach argument nuggets to trigger nuggets using EAL and schema constraints. Appendix A presents the detailed decoding algorithm.

6 Experiment

6.1 Experiment Settings

Evaluation Metrics Four standard metrics are adopted: trigger identification (TI), trigger classification (TC), argument identification (AI), and argument classification (AC). In addition, we introduce event correlation (EC) to evaluate the extraction of hierarchical sub-event relations. Specifically, when the trigger of one event appears as an argument of another event, the two events are considered correlated through their triggers. An evaluation example is provided in Appendix B.

Baselines We conduct a comprehensive evaluation of state-of-the-art and recent EE models, which can be broadly categorized into three groups: (1) Global information extraction models that jointly model entities, relations, and events within a unified framework; (2) Discriminative EE models that formulate EE as token classification or sequential labeling problems; (3) Generative EE models that generate extractions via question answering or through a well-designed generative schema.

For a fair comparison, when evaluating EAE-only methods, we first apply a best-performing ED method to extract event triggers. The EAE-only methods then perform argument extraction conditioned on these predicted triggers.

Implementations We randomly split SciEvents into training, development, and test sets with a ratio of 80%/10%/10%. Each model is evaluated over three independent runs, and the average performance is reported. For models with the same architecture, we use the same pretrained backbone. Additional details are provided in Appendix C.

6.2 Experiment Results

Overall Results Table 3 presents the overall performance of different models on SciEvents. Results with precision and recall score can be found in Appendix E. Overall, EXCEEDS achieves the strongest performance on most evaluation metrics, particularly on AI, AC and EC. Notably, EXCEEDS outperforms the second-best model by 0.51% on AC and 0.53% on EC, indicating its advantage in extracting arguments and capturing hierarchical sub-event relations in scientific documents.

Among baseline families, global models show competitive performance on TI and TC, which can be attributed to their use of entity information during training. Discriminative models generally perform better on EAE than global and generative models, while generative approaches exhibit larger performance variance across metrics.

Ablation study shows that removing the contextual modeling module results in the most pronounced performance drop on EAE, with AC decreasing by 1.06%, indicating that grid-based modeling critically relies on high-quality contextual token representations. Excluding the grid refiner also leads to consistent degradation across metrics. These results suggest the effectiveness of contextual encoding and grid refinement in EX-

Model	Discontinuous	Overlapping		Reverse-order	Sub-event		
	AC	TC	AC	AC	TC	AC	EC
OneIE	–	62.20$_{\pm 1.07}$	17.33$_{\pm 0.98}$	–	51.19$_{\pm 0.69}$	37.41$_{\pm 1.51}$	41.39$_{\pm 2.72}$
ScentedEAE	–	55.77$_{\pm 2.40}$	11.05$_{\pm 1.23}$	–	43.23$_{\pm 2.35}$	38.00$_{\pm 2.19}$	33.04$_{\pm 1.45}$
Tagprime	–	55.03$_{\pm 1.59}$	18.11$_{\pm 1.13}$	–	53.84$_{\pm 0.81}$	47.89$_{\pm 0.39}$	48.11$_{\pm 2.45}$
EEQA	–	17.02$_{\pm 0.80}$	2.33$_{\pm 0.02}$	–	53.50$_{\pm 0.92}$	44.81$_{\pm 1.35}$	45.16$_{\pm 1.21}$
PAIE	–	49.62$_{\pm 1.71}$	13.18$_{\pm 0.67}$	–	53.66$_{\pm 1.75}$	47.34$_{\pm 1.01}$	49.08$_{\pm 1.75}$
DEEIA	–	51.31$_{\pm 2.84}$	13.13$_{\pm 1.21}$	–	42.73$_{\pm 1.20}$	32.80$_{\pm 1.67}$	31.30$_{\pm 3.73}$
BartGen	2.74$_{\pm 0.18}$	31.98$_{\pm 1.29}$	10.58$_{\pm 0.44}$	0.00$_{\pm 0.00}$	52.25$_{\pm 0.13}$	43.61$_{\pm 0.31}$	40.19$_{\pm 2.46}$
DEGREE	0.00$_{\pm 0.00}$	24.93$_{\pm 1.74}$	7.16$_{\pm 0.54}$	0.00$_{\pm 0.00}$	39.24$_{\pm 0.51}$	30.16$_{\pm 0.91}$	19.97$_{\pm 1.23}$
KnowCoder	0.00$_{\pm 0.00}$	26.18$_{\pm 1.12}$	6.93$_{\pm 0.39}$	0.00$_{\pm 0.00}$	42.36$_{\pm 1.32}$	34.81$_{\pm 0.89}$	40.33$_{\pm 2.41}$
EXCEEDS	13.86$_{\pm 2.29}$	**62.46**$_{\pm 3.86}$	**22.46**$_{\pm 2.01}$	7.27$_{\pm 5.14}$	55.13$_{\pm 0.32}$	**48.32**$_{\pm 0.18}$	**51.15**$_{\pm 0.56}$
– Contextual	13.41$_{\pm 1.34}$	59.72$_{\pm 7.56}$	20.43$_{\pm 3.20}$	9.55$_{\pm 4.41}$	54.72$_{\pm 1.13}$	47.64$_{\pm 0.85}$	50.18$_{\pm 1.30}$
– Grid Refiner	**14.98**$_{\pm 1.19}$	61.81$_{\pm 2.86}$	22.07$_{\pm 0.80}$	**11.99**$_{\pm 2.12}$	**55.28**$_{\pm 0.91}$	48.04$_{\pm 1.07}$	49.11$_{\pm 0.98}$

Table 4: F1-score (%) on complex nuggets and events. – indicates that the method cannot be evaluated.

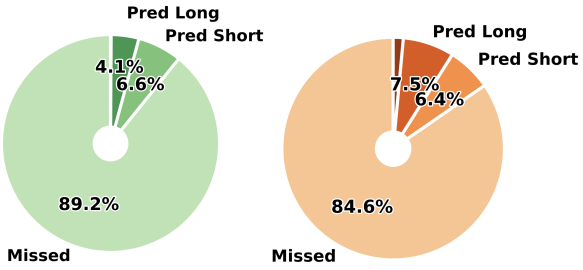
CEEDS. Despite these improvements, the overall performance on AC remains modest, indicating that SciEvents remains a challenging EE dataset.

Complex Scenarios Results Table 4 reports model performance on complex scenarios, where evaluation is conducted on subsets filtered by specific information forms. Overall, EXCEEDS consistently outperforms all baselines across complex scenarios, demonstrating its effectiveness in extracting complex nuggets and events. However, performance on discontinuous, overlapping, and reverse-order scenarios is substantially lower than the overall results, indicating that these forms remain particularly challenging for current EE models, even with specialized modeling designs.

Notably, generative models do not exhibit advantages under complex scenarios. In particular, they show pronounced performance degradation on overlapping nuggets, a trend also observed in EEQA, which adopts a generation-style formulation. This suggests that directly generating textual outputs is insufficient for accurately capturing complex nugget structures.

Error Analysis Figure 4 shows the identification error distributions for TI and AI, where errors are categorized into missed, predicted-long, predicted-short, and other overlap cases. Missed predictions dominate identification errors, accounting for 89.2% of TI errors and 84.6% of AI errors. This suggests that the primary challenge lies in failing to detect the presence of event mentions, rather than in inaccurately predicting their span boundaries. By contrast, boundary-related errors constitute a substantially smaller portion of the overall errors. This observation indicates nugget boundary imprecision is not the main bottleneck for identification performance. Overall, these results highlight that improving recall in dense scientific contexts remains a key challenge for event identification.

Figure 5 presents the misclassification patterns in EXCEEDS for TC and AC. Overall, errors in both TC and AC are dominated by confusions between semantically similar types. For TC, frequent confusions occur between closely related event types, notably *MDS* and *WKS*. For AC, the most common errors also arise from fine-grained role distinctions, such as *TriedC. vs. BaseC.* and *Subject vs. Object*. These patterns suggest that improving classification on SciEvents likely requires more fine-grained modeling of subtle semantic distinctions, *e.g.*, specialized disambiguation components, or incorporating schema/template-level cues (Ma et al., 2022; Liu et al., 2024) to better differentiate similar types.



(a) TI Error Distribution (b) AI Error Distribution

Figure 4: Identification Error Distribution

7 Conclusion

We address the challenge of understanding scientific domain with complex event structures through EE by introducing SciEvents, a large-scale document-level dataset with a refined schema tai-

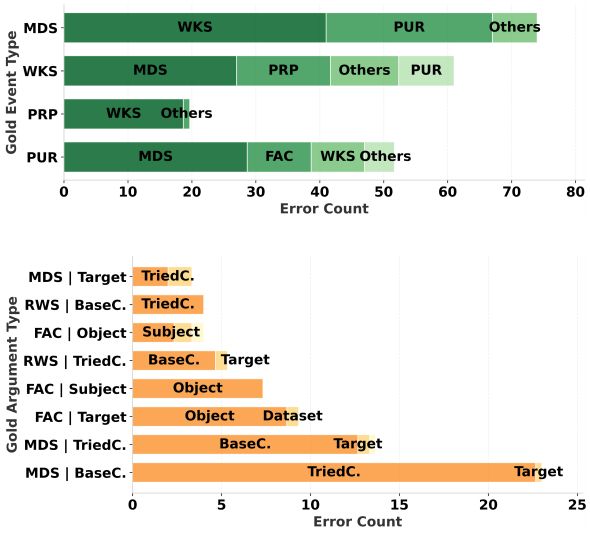


Figure 5: Classification Error Distribution. Y-axis denotes the misclassified gold type and each stacked bar indicates an incorrect predicted type and its frequency.

lored to the scientific domain, and EXCEEDS, a nugget-based grid modeling EE approach. Experiments show that SciEvents reflects the information-dense and structurally complex nature of scientific texts, while EXCEEDS achieves strong performance. Further analysis indicates that SciEvents remains a challenging dataset and suggests directions for future improvements.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (No. 2024YFF0908200).

Limitations

Abstract-Level Data Scope SciEvents is constructed from paper abstracts, rather than full-length scientific articles. While abstracts typically provide concise and information-dense summaries of scientific contributions, they do not capture all event mentions that may appear in the main body of a paper. In particular, important information conveyed through figures, tables, equations, or cross-section references is not covered in our current dataset. As a result, SciEvents does not account for multimodal or long-range contextual signals that may be essential for comprehensive scientific event understanding. Extending the dataset to full-text articles and incorporating multimodal information remains an important direction for future work.

Limited Domain Coverage SciEvents is constructed from ACL conference abstracts, which primarily represent the NLP sub-domain of scientific literature. As a result, the dataset does not cover the full diversity of writing styles, terminologies, and event structures present in other scientific disciplines.

This design choice is intended to provide a relatively controlled setting for studying dense and structurally complex event patterns, while reducing variability introduced by cross-domain differences. Nevertheless, the restricted domain coverage may limit the generalizability of models trained on SciEvents to broader scientific contexts. Extending the dataset to a wider range of scientific domains remains an important direction for future work.

References

Jacqueline Aguilar, Charley Beller, Paul McNamee, Benjamin Van Durme, Stephanie Strassel, Zhiyi Song, and Joe Ellis. 2014. A comparison of the events and relations across ACE, ERE, TAC-KBP, and FrameNet annotation standards. In *Proceedings of the Second Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 45–53, Baltimore, Maryland, USA. Association for Computational Linguistics.

Yunmo Chen, Tongfei Chen, Seth Ebner, Aaron Steven White, and Benjamin Van Durme. 2020. Reading the manual: Event extraction as definition comprehension. In *Proceedings of the Fourth Workshop on Structured Prediction for NLP*, pages 74–83, Online. Association for Computational Linguistics.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683, Online. Association for Computational Linguistics.

Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2020. Multi-sentence argument linking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8057–8077, Online. Association for Computational Linguistics.

Jun Gao, Huan Zhao, Changlong Yu, and Ruifeng Xu. 2023a. Exploring the feasibility of chatgpt for event extraction. *arXiv preprint arXiv:2303.03836*.

Jun Gao, Huan Zhao, Yice Zhang, Wei Wang, Changlong Yu, and Ruifeng Xu. 2023b. Benchmarking large language models with augmented instructions for fine-grained information extraction. *arXiv preprint arXiv:2310.05092*.

I-Hung Hsu, Kuan-Hao Huang, Elizabeth Boschee, Scott Miller, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2022. DEGREE: A data-efficient generation-based event extraction model. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1890–1908, Seattle, United States. Association for Computational Linguistics.

I-Hung Hsu, Kuan-Hao Huang, Shuning Zhang, Wenxin Cheng, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2023. TAGPRIME: A unified framework for relational structure extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12917–12932, Toronto, Canada. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Kuan-Hao Huang, I-Hung Hsu, Tanmay Parekh, Zhiyu Xie, Zixuan Zhang, Prem Natarajan, Kai-Wei Chang, Nanyun Peng, and Heng Ji. 2024. TextEE: Benchmark, reevaluation, reflections, and future challenges in event extraction. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12804–12825, Bangkok, Thailand. Association for Computational Linguistics.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *Preprint*, arXiv:1508.01991.

Jin-Dong Kim, Yue Wang, Toshihisa Takagi, and Akinori Yonezawa. 2011. Overview of Genia event task in BioNLP shared task 2011. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 7–15, Portland, Oregon, USA. Association for Computational Linguistics.

Jin-Dong Kim, Yue Wang, and Yamamoto Yasunori. 2013. The Genia event extraction shared task, 2013 edition - overview. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 8–15, Sofia, Bulgaria. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Fayuan Li, Weihua Peng, Yuguang Chen, Quan Wang, Lu Pan, Yajuan Lyu, and Yong Zhu. 2020a. Event extraction as multi-turn question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 829–838.

Manling Li, Alireza Zareian, Qi Zeng, Spencer Whitehead, Di Lu, Heng Ji, and Shih-Fu Chang. 2020b. Cross-media structured common space for multimedia event extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2557–2568, Online. Association for Computational Linguistics.

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.

Xinyu Li, Fayuan Li, Lu Pan, Yuguang Chen, Weihua Peng, Quan Wang, Yajuan Lyu, and Yong Zhu. 2020c. Duee: a large-scale dataset for chinese event extraction in real-world scenarios. In *Natural Language Processing and Chinese Computing: 9th CCF International Conference, NLPCC 2020, Zhengzhou, China, October 14–18, 2020, Proceedings, Part II 9*, pages 534–545. Springer.

Zixuan Li, Yutao Zeng, Yuxin Zuo, Weicheng Ren, Wenxuan Liu, Miao Su, Yucan Guo, Yantao Liu, Xiang Li, Zhilei Hu, Long Bai, Wei Li, Yidan Liu, Pan Yang, Xiaolong Jin, Jiafeng Guo, and Xueqi Cheng. 2024. 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, Bangkok, Thailand. Association for Computational Linguistics.

Xin Liang, Dawei Cheng, Fangzhou Yang, Yifeng Luo, Weining Qian, and Aoying Zhou. 2020. F-hmtc: Detecting financial events for investment decisions based on neural hierarchical multi-label text classification. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4490–4496. International Joint Conferences on Artificial Intelligence Organization. Special Track on AI in FinTech.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, Online. Association for Computational Linguistics.

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

2023, pages 15342–15359, Singapore. Association for Computational Linguistics.

Ruibo Liu, Jason Wei, Chenyan Jia, and Soroush Vosoughi. 2021. Modulating language models with emotions. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4332–4339, Online. Association for Computational Linguistics.

Wanlong Liu, Li Zhou, DingYi Zeng, Yichen Xiao, Shaohuan Cheng, Chen Zhang, Grandee Lee, Malu Zhang, and Wenyu Chen. 2024. Beyond single-event extraction: Towards efficient document-level multi-event argument extraction. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9470–9487, Bangkok, Thailand. Association for Computational Linguistics.

Xiao Liu, Heyan Huang, Ge Shi, and Bo Wang. 2022. Dynamic prefix-tuning for generative template-based event extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5216–5228, Dublin, Ireland. Association for Computational Linguistics.

Xiao Liu, Heyan Huang, and Yue Zhang. 2019a. Open domain event extraction using neural latent variable models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2860–2871, Florence, Italy. Association for Computational Linguistics.

Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018. Jointly multiple events extraction via attention-based graph information aggregation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1247–1256, Brussels, Belgium. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Jie Lou, Yaojie Lu, Dai Dai, Wei Jia, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2023. Universal information extraction as unified semantic matching. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(11):13318–13326.

Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021. Text2Event: Controllable sequence-to-structure generation for end-to-end event extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2795–2806, Online. Association for Computational Linguistics.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5755–5772, Dublin, Ireland. Association for Computational Linguistics.

Mingyu Derek Ma, Alexander Taylor, Wei Wang, and Nanyun Peng. 2023. 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, Toronto, Canada. Association for Computational Linguistics.

Yubo Ma, Zehao Wang, Yixin Cao, Mukai Li, Meiqi Chen, Kun Wang, and Jing Shao. 2022. Prompt for extraction? PAIE: Prompting argument interaction for event argument extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6759–6774, Dublin, Ireland. Association for Computational Linguistics.

Hieu Man Duc Trong, Duc Trong Le, Amir Pouran Ben Veyseh, Thuat Nguyen, and Thien Huu Nguyen. 2020. Introducing a new dataset for event detection in cybersecurity texts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5381–5390, Online. Association for Computational Linguistics.

Minh Van Nguyen, Viet Dac Lai, and Thien Huu Nguyen. 2021. Cross-task instance representation interactions and label dependencies for joint information extraction with graph convolutional networks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 27–38, Online. Association for Computational Linguistics.

Tanmay Parekh, I-Hung Hsu, Kuan-Hao Huang, Kai-Wei Chang, and Nanyun Peng. 2023. GENEVA: Benchmarking generalizability for event argument extraction with hundreds of event types and argument roles. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3664–3686, Toronto, Canada. Association for Computational Linguistics.

Hao Peng, Xiaozhi Wang, Feng Yao, Kaisheng Zeng, Lei Hou, Juanzi Li, Zhiyuan Liu, and Weixing Shen. 2023. The devil is in the details: On the pitfalls of event extraction evaluation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9206–9227, Toronto, Canada. Association for Computational Linguistics.

Amir Pouran Ben Veyseh, Minh Van Nguyen, Franck Dernoncourt, and Thien Nguyen. 2022. MINION: a large-scale and diverse dataset for multilingual event detection. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, pages 2286–2299, Seattle, United States. 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.

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. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8749–8757.

Shirong Shen, Guilin Qi, Zhen Li, Sheng Bi, and Lusheng Wang. 2020. Hierarchical Chinese legal event extraction via pedal attention mechanism. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 100–113, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Matthew Sims, Jong Ho Park, and David Bamman. 2019. Literary event detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3623–3634, Florence, Italy. Association for Computational Linguistics.

Zhiyi Song, Ann Bies, Stephanie Strassel, Tom Riese, Justin Mott, Joe Ellis, Jonathan Wright, Seth Kulick, Neville Ryant, and Xiaoyi Ma. 2015. From light to rich ERE: Annotation of entities, relations, and events. In *Proceedings of the The 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 89–98, Denver, Colorado. Association for Computational Linguistics.

Jianlin Su, Mingren Zhu, Ahmed Murtadha, Shengfeng Pan, Bo Wen, and Yunfeng Liu. 2022. Zlpr: A novel loss for multi-label classification. *Preprint*, arXiv:2208.02955.

Zhaoyue Sun, Jiazheng Li, Gabriele Pergola, Byron Wallace, Bino John, Nigel Greene, Joseph Kim, and Yulan He. 2022. PHEE: A dataset for pharmacovigilance event extraction from text. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5571–5587, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Beth M. Sundheim. 1992. Overview of the fourth Message Understanding Evaluation and Conference. In *Fourth Message Understanding Conference (MUC-4): Proceedings of a Conference Held in McLean, Virginia, June 16-18, 1992*.

MeiHan Tong, Bin Xu, Shuai Wang, Meihuan Han, Yixin Cao, Jiangqi Zhu, Siyu Chen, Lei Hou, and Juanzi Li. 2022. DocEE: A large-scale and fine-grained benchmark for document-level event extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3970–3982, Seattle, United States. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. *Techbical report, Linguistic Data Consortium*.

Bo Wang, Heyan Huang, Xiaochi Wei, Ge Shi, Xiao Liu, Chong Feng, Tong Zhou, Shuaiqiang Wang, and Dawei Yin. 2023a. Boosting event extraction with denoised structure-to-text augmentation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11267–11281, Toronto, Canada. Association for Computational Linguistics.

Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, et al. 2023b. 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.

Kaiwen Wei, Xian Sun, Zequn Zhang, Jingyuan Zhang, Guo Zhi, and Li Jin. 2021. Trigger is not sufficient: Exploiting frame-aware knowledge for implicit event

argument extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4672–4682.

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, et al. 2023. Zero-shot information extraction via chatting with chatgpt. *arXiv preprint arXiv:2302.10205*.

Hang Yang, Yubo Chen, Kang Liu, Yang Xiao, and Jun Zhao. 2018. DCFEE: A document-level Chinese financial event extraction system based on automatically labeled training data. In *Proceedings of ACL 2018, System Demonstrations*, pages 50–55, Melbourne, Australia. Association for Computational Linguistics.

Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan, and Dongsheng Li. 2019. Exploring pre-trained language models for event extraction and generation. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 5284–5294.

Yu Yang, Jinyu Guo, Kai Shuang, and Chenrui Mao. 2024. Scented-EAE: Stage-customized entity type embedding for event argument extraction. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5222–5235, Bangkok, Thailand. Association for Computational Linguistics.

Feng Yao, Chaojun Xiao, Xiaozhi Wang, Zhiyuan Liu, Lei Hou, Cunchao Tu, Juanzi Li, Yun Liu, Weixing Shen, and Maosong Sun. 2022. LEVEN: A large-scale Chinese legal event detection dataset. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 183–201, Dublin, Ireland. Association for Computational Linguistics.

Yang Zhou, Yubo Chen, Jun Zhao, Yin Wu, Jiexin Xu, and Jinlong Li. 2021. What the role is vs. what plays the role: Semi-supervised event argument extraction via dual question answering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 14638–14646.

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

A Word-word Event Grid Decoding

To ensure the structural validity of decoded nuggets and events, we apply two pruning heuristics: (1) an HTL chain is kept only if it can be closed by a THL-type edge; (2) an argument nugget is kept

Algorithm 1: Decoding grid into events

Input: Binary grid $\hat{\mathbf{M}} \in \{0,1\}^{l \times l \times |R|}$; label vocabulary $\mathcal{V}$ (including HTL, EAL, and THL-types); ontology checker $\textsc{Valid}(t_e, t_a)$.
Output: Event set E in the target format.

Initialize `Forward` $\leftarrow$ empty map ; `// HTL: head` $\rightarrow$ `next tokens`
Initialize `Tails` $\leftarrow$ empty map ; `// THL-type: head` $\rightarrow$ `possible tails`
Initialize `Links` $\leftarrow$ empty set ; `// EAL: (trigger-head, argument-head)`
for $i \leftarrow 1$ **to** l **do**
  for $j \leftarrow 1$ **to** l **do**
    $\mathcal{R}_{i,j} \leftarrow \{r \in R \mid \hat{\mathbf{M}}_{i,j,r} = 1\}$;
    foreach $r \in \mathcal{R}_{i,j}$ **do**
      if $r = \text{HTL}$ ***and*** $i \neq j$ **then**
        `Forward`$[i] \leftarrow$ `Forward`$[i] \cup \{j\}$;
      else if $r = \text{EAL}$ **then**
        `Links` $\leftarrow$ `Links` $\cup \{(i, j)\}$;
      else
        `// r is a THL-type label indicating mention type`
        `Tails`$[j] \leftarrow$ `Tails`$[j] \cup \{i\}$;

`// Step 1: recover nuggets by DFS over HTL and close with THL-type`
Initialize `Mentions` $\leftarrow \emptyset$;
foreach *head* h *in* `Tails` **do**
  Run DFS starting from h following `Forward` edges to enumerate paths $p = [h, \ldots, t]$;
  Keep p only if $t \in$ `Tails`$[h]$; `// Heuristic (1): must be closed by THL-type`
  Add each kept path as a mention span into `Mentions`;

`// Step 2: assign mention types via closing THL-type and split triggers/arguments`
Initialize `Triggers` $\leftarrow \emptyset$, `Args` $\leftarrow \emptyset$;
foreach *mention span* $p = [h, \ldots, t]$ *in* `Mentions` **do**
  $\mathcal{T} \leftarrow \{r \in R \mid \hat{\mathbf{M}}_{t,h,r} = 1 \ \wedge \ r \neq \text{HTL} \ \wedge \ r \neq \text{EAL}\}$;
  foreach $\tau \in \mathcal{T}$ **do**
    if $\tau \in T_E$ **then**
      `Triggers` $\leftarrow$ `Triggers` $\cup \{(p, \tau)\}$
    else
      `Args` $\leftarrow$ `Args` $\cup \{(p, \tau)\}$

`// Step 3: build events by linking arguments to triggers via EAL + ontology constraints`
Initialize $E \leftarrow \{e = (t_e, \text{trigger} = p, A = \emptyset) \mid (p, t_e) \in$ `Triggers`$\}$;
foreach $(p_a, t_a) \in$ `Args` **do**
  Let h_a be the head (first token) of p_a;
  Find all triggers (p_t, t_e) such that $(h_t, h_a) \in$ `Links` and $\textsc{Valid}(t_e, t_a)$;
  if *no such trigger exists* **then**
    continue ; `// Heuristic (2): drop arguments not attachable to any trigger`
  foreach *matched trigger event* e **do**
    Add (t_a, p_a) into $e.A$;

return E;

only if it can be linked to at least one trigger nugget via EAL and passes the ontology constraint.

With these two pruning heuristics, Algorithm 1 summarizes the detailed decoding process.

Notably, to prevent potential excessively long decoding time during early training stages, we adopt a conservative training strategy. In early epochs, model predictions may be unstable and could assign a large number of labels to grid cells, which in the worst case may lead to an exponential number of candidate HTL chains during DFS-based decoding and significantly slow down validation. To avoid such degenerate cases, we skip validation in the initial training phase (typically the first few epochs) and enable regular validation after this phase.

B Evaluation Metrics Demonstration

In SciEvents, there are 3 tasks and 5 kinds of metrics: (1) Trigger Extraction, also known as Event Detection, includes Trigger Identification (TI) and Trigger Classification (TC). (2) Event Argument Extraction includes Argument Identification (AI) and Argument Classification (AC). (3) Sub-Event Extraction includes Event Correlation (EC).

In SciEvents, a nugget serves as the basic unit for evaluation, and an exact-match criterion is applied when assessing nugget token spans. Table 5 shows an example with two prediction events. Table 6 shows how to evaluate these two prediction events.

	Event 1	Event 2
trigger	A (Type M)	C (Type N)
argument	B (Type BT) C (Type CT)	D (Type DT)

Table 5: An example with two prediction events. A, B, C and D are token spans. M and N are event types. BT, CT and DT are argument types.

Metric	What elements should match ground-truth
TI	A C
TC	A+M C+N
AI	A+M+B A+M+C C+N+D
AC	A+M+B+BT A+M+C+CT C+N+D+DT
EC	A+M+C+CT+N

Table 6: An evaluation example for two prediction events. A, B, C and D are token spans. M and N are event types. BT, CT and DT are argument types.

C Implementation Details

Pretrained Backbone. For all evaluated models, we adopt either BART-large (Lewis et al., 2020) or RoBERTa-large (Liu et al., 2019b) as the pretrained backbone to ensure a fair and controlled comparison across architectures. KnowCoder (Li et al., 2024), which is based on large language models, employs LLaMA 2-7B (Touvron et al., 2023) as its pretrained backbone.

Hyperparameter Settings. For EXCEEDS, the hyperparameter settings used in our implementation are reported in Table 7. For the remaining models, we primarily apply a unified event extraction framework, TextEE (Huang et al., 2024), and adopt the hyperparameters recommended by TextEE. For models that are not supported by TextEE, we adapt SciEvents to their official code, and use the hyperparameter settings suggested in their implementations.

Hyperparameter	**Value**
Roberta-large Learning Rate	1e-5
Warm Up Ratio	0.1
Other Learning Rate	1e-3
Batch Size	2
Epoch	20
Distance Embedding Size	20
Bi-LSTM Hidden Size	1024
Grid Channels (C_g)	256
Grid Refiner Dropout Rate	0.1
Other Dropout Rate	0.5
Grid Refiner Layers (K)	2
Grid Refiner Kernel	3

Table 7: Hyperparameter used in EXCEEDS.

Architecture-Specific Implementations. For global information extraction models, we leverage the nugget type information provided in SciEvents (described in Appendix F.1) to facilitate entity training, following their original modeling assumptions.

For discriminative models, as most models rely on explicit start and end offsets for training and inference, discontinuous and reverse-order nuggets are not directly applicable under their modeling assumptions. Accordingly, these models are evalu-

ated only on nugget instances with contiguous span representations.

For generative models, input-output interfaces and evaluation scripts are modified to support raw text, without relying on explicit span offsets, during training, inference, and evaluation. This adaptation ensures that generative approaches can be fairly evaluated on SciEvents under an offset-free setting.

Training and Inference Cost Most models are trained and evaluated on NVIDIA RTX 3090 GPUs. Experiments involving large language models are conducted on NVIDIA A800 80GB PCIe GPUs, where parameter-efficient fine-tuning with LoRA (Hu et al., 2021) is adopted. Table 8 reports the average training time **per epoch** and inference time for each model, measured in GPU hours, providing a reference for their computational cost. Appendix D provides a theoretical analysis of the computational complexity and memory overhead of EXCEEDS.

	Model	Training	Inference
Global	OneIE	0.1816	0.0036
	ScentedEAE	0.5249	0.0440
ED & EAE	EEQA-ED	0.0259	0.0056
	EEQA-EAE	5.2138	3.7292
	Tagprime-ED	0.2510	0.0014
	Tagprime-EAE	0.3910	0.0083
	DEGREE	0.3639	0.0067
	KnowCoder-ED†	0.2381	0.2150
	KnowCoder-EAE	1.5869	0.2897
	EXCEEDS (Ours)	0.0609	0.1692
EAE-only	PAIE	0.7536	0.0033
	DEEIA	0.1237	0.0165
	BartGen	0.2950	0.0236

Table 8: Average GPU hours required per training epoch and for inference across different models. †is based on a large language model and is fine-tuned using parameter-efficient adaptation.

D Computational Complexity and Memory of EXCEEDS

Let l denote the input sequence length, d the encoder hidden size, C_g the grid channel size, K the number of grid refinement layers, and $|R|$ the number of relation types of the word-word event grid. EXCEEDS consists of:

- *Contextual token encoding*: a pretrained transformer encoder (RoBERTa), followed by a BiLSTM and CLN. The transformer encoding has the standard complexity $O(l^2 d)$, while BiLSTM and CLN are $O(ld^2)$ and $O(ld)$, respectively.
- *Pair-wise grid construction*: explicit construction of an $l \times l$ token-pair grid, followed by applying an MLP to each token pair to obtain grid features. This step scales as $O(l^2)$, with constants determined by the MLP width and feature dimensions.
- *Grid refinement and classification*: application of K lightweight 2D convolutional refinement blocks on the $l \times l$ grid, scaling as $O(Kl^2)$, with constants determined by kernel size and channel width. The final classifier head projects each grid cell from C_g to $|R|$ relation logits, costing $O(l^2 C_g |R|)$ for a linear head.

Overall, the complexity of EXCEEDS is dominated by $O(l^2)$ terms. The memory footprint is dominated by storing the grid features and logits, *i.e.*, $O(l^2 C_g + l^2 |R|)$, in addition to encoder activations.

E Full Experiment Results

Table 20, 21 and 22 presents the full experiment results, including precision and recall scores.

F Schema of SciEvents

The schema of SciEvents consists of nugget types and event types. In this section, we will introduce the description of each nugget type and the template of each event type.

F.1 Nugget Types

There are 10 nugget types in SciEvents as follows:

Research Organization / Group (OG) refers to a research team composed of people. Typical examples include: *We*; *Li et al., 2013*; *They*.

Approach (APP) refers to nouns, pronouns, and corresponding phrases that denote a complete method or algorithm with concrete inputs and outputs. Typical examples include: *... work*; *... model*; *... method*; *... framework*; *... network*; *... algorithm*; *baselines*; *state-of-the-art*.

Module (MOD) refers to nouns, pronouns, and corresponding phrases that denote components of a method or architecture, such as modules or algorithmic elements. A single MOD item is usually

not detailed enough to constitute a full APP. Typical examples include: ... *encoders*; ... *decoders*; ... *module*; ... *process*; *message propagation process*; *beam search*.

Feature (FEA) refers to nouns, pronouns, and corresponding phrases that denote features. Typical examples include: ... *information*; *the first-order adjacency information*; *the relationships between labeled edges*.

Note the difference among APP, MOD, and FEA: **APP** refers to a complete method with concrete inputs (*e.g.*, a task) and outputs (*e.g.*, the desired results of the task). **MOD** refers to a sub-process or component within the overall APP framework, such as a module or algorithmic element. **FEA** refers to features utilized during the execution of an APP or a MOD, such as *positional information*, *vector representations of part-of-speech tags*, or *sentence length*.

Task (TAK) refers to phrases that denote the intention or objective of a task optimization, *i.e.*, the research focus or target point. These expressions are neutral. Typical examples include: *graph-to-sequence modeling*; *performance unimodal*; *performance multimodal*; *accuracy*; *F1 score*; *robustness*; *reproducibility*; *zero-shot translation quality*.

Dataset (DST) refers to nouns, pronouns, and corresponding phrases that denote datasets used or relied upon when describing an artifact, research objective, or experimental conclusion. Typical examples include: *TAC-KBP 2017 datasets*; *Chinese multimodal NER dataset*; *CNERTA*; *training data*.

Limit (LIM) refers to phrases that denote conditional or environmental limitations, often introduced with prepositions. Typical examples include: *for a small number of confusing type pairs*; *in existing verb metaphor detection benchmarks*; *of the dynamic self-attention*.

Strength (STR) refers to phrases that describe the advantages or strengths of an artifact, often with an evaluative or positive connotation. Typical examples include: *state-of-the-art performance*.

Weakness (WEA) refers to phrases that describe the disadvantages, shortcomings, or weaknesses of an artifact, often with an evaluative or negative connotation. Typical examples include: *most of the mislabeling*; *biases and failure cases of beam search*.

Degree (DEG) refers to adjectives, adverbs, numerals, or other expressions that describe the degree or quantity of an event. Typical examples include: *only*; *not fully*; *1.5%*.

F.2 Event Types

There are 10 event types classified by four different rhetorical components as follows:

(1) *General*. *General* events occur in all four components.

Argument Type	Constrained Types
Aim	APP / MOD / FEA / DST / STR / WEA / TAK
Condition	LIM
Dataset	DST

Table 9: Schema of Purpose (PUR) Event

Purpose (PUR). As the schema shown in Table 9, the Purpose event describes: *In order to deal with <Aim:arg1> under <Condition:arg2> circumstance on <Dataset:arg3> datasets.*

(2) *Background* includes one kind of event type:

Argument Type	Constrained Types
Target	APP / MOD / FEA / DST / STR / WEA / TAK
Condition	LIM
Dataset	DST

Table 10: Schema of IntroduceTarget (ITT) Event

IntroduceTarget (ITT). As the schema shown in Table 10, the IntroduceTarget event describes: *<Target:arg1> is the abstract research target under <Condition:arg2> circumstance on <Dataset:arg3> datasets in this paper.*

(3) *Related Work* includes two kinds of event type:

Argument Type	Constrained Types
Subject	APP / MOD / FEA / DST
BaseComponent	APP / MOD / FEA / DST
TriedComponent	APP / MOD / FEA / DST
Condition	LIM / E-RWS
Dataset	DST
Target	E-PUR / TAK / STR / WEA / APP / FEA / MOD

Table 11: Schema of RelatedWorkStep (RWS) Event

RelatedWorkStep (RWS). As the schema shown in Table 11, the RelatedWorkStep event

describes: *Previously <Subject:arg1> on <Target:arg2> are mostly based on <BaseComponent:arg3> with <TriedComponent:arg4> under <Condition:arg5> circumstance on <Dataset:arg6> datasets.*

Argument Type	Constrained Types
Concern	APP / FEA / STR / WEA / MOD / DST
Fault	APP / FEA / STR / WEA / MOD / DST
Condition	LIM / E-RWF / E-RWS
Dataset	DST
Target	E-PUR / TAK / STR / WEA
Extent	DEG

Table 12: Schema of RelatedWorkFault (RWF) Event

RelatedWorkFault (RWF). As the schema shown in Table 12, the RelatedWorkFault event describes: *Aiming to <Target:arg1>, to <Extent:arg5> degree, <Concern:arg2> has some <Fault:arg6> faults under <Condition:arg3> circumstance on <Dataset:arg4> datasets.*

(4) *Methodology* includes three kinds of event type:

Argument Type	Constrained Types
Proposer	OG
Content	APP / FEA / MOD / DST / TAK
Target	E-PUR / TAK / FEA / WEA

Table 13: Schema of Propose (PRP) Event

Propose (PRP). As the schema shown in Table 13, the Propose event describes: *In this paper, <Proposer:arg1> propose <Content:arg2> for <Target:arg3>.*

Argument Type	Constrained Types
Researcher	OG
Content	APP / MOD / FEA / DST / STR / WEA / TAK
Condition	LIM
Dataset	DST
Target	E-PUR / TAK / STR / WEA / APP / FEA / MOD

Table 14: Schema of WorkStatement (WKS) Event

WorkStatement (WKS). As the schema shown in Table 14, the WorkStatement event describes: *<Researcher:arg1> report <Content:arg2> under <Condition:arg3> circumstance on <Dataset:arg4> datasets for <Target:arg5>.*

Argument Type	Constrained Types
BaseComponent	APP / MOD / FEA / DST
TriedComponent	APP / MOD / FEA / DST
Condition	LIM / E-MDS
Dataset	DST
Target	E-PUR / TAK / STR / WEA / APP / FEA / MOD

Table 15: Schema of MethodStep (MDS) Event

MethodStep (MDS). As the schema shown in Table 15, the MethodStep event describes: *Our approach adopt <BaseComponent:arg1> with <TriedComponent:arg2> under <Condition:arg3> circumstance on <Dataset:arg4> datasets for <Target:arg5>.*

(5) *Results* includes three kinds of event type:

Argument Type	Constrained Types
Finder	OG
Content	E-FAC / E-CMP

Table 16: Schema of Finding (FIN) Event

Finding (FIN). As the schema shown in Table 16, the Finding event describes: *In experiments, <Finder:arg1> find or demostrate findings that <Content:arg2>.*

Argument Type	Constrained Types
Arg1	E-FAC / APP / MOD / FEA / DST
Arg2	E-FAC / APP / MOD / FEA / DST
Condition	LIM / E-FAC
Dataset	DST
Result	STR / WEA
Metrics	TAK
Extent	DEG

Table 17: Schema of ExperimentCompare (CMP) Event

ExperimentCompare (CMP). As the schema shown in Table 17, the ExperimentCompare event describes: *Experimental results show that the <Metrics:arg6> of <Arg1:arg1> is <Extent:arg2> <Result:arg3> than <Arg2:arg4> under <Condition:arg5> circumstance on <Dataset:arg7> datasets.*

OutcomeFact (FAC). As the schema shown in Table 18, the OutcomeFact event describes: *Experimental results show that <Subject:arg1> can <Extent:arg2> provide <Object:arg3> for <Target:arg4> under <Condition:arg5> circumstance on <Dataset:arg6> datasets because <Reason:arg7> reasons.*

Argument Type	Constrained Types
Subject	APP / MOD / FEA / STR / WEA / TAK / DST
Object	APP / MOD / FEA / STR / WEA / TAK / DST
Condition	LIM / E-FAC
Reason	LIM / E-FAC
Dataset	DST
Target	E-PUR / TAK / STR / WEA
Extent	DEG

Table 18: Schema of OutcomeFact (FAC) Event

G Example of Document-Level Event Annotation

Table 19 presents a fully annotated example document from SciEvents, illustrating all event instances annotated within a single document. For each event, we show the corresponding trigger nugget, argument nuggets, and their semantic roles, as well as hierarchical sub-event relations when applicable. This example is provided to demonstrate the density and structural complexity of event annotations in scientific documents, and to facilitate a clearer understanding of the annotation schema and evaluation setup.

H Distributions in SciEvents

In this section, we will present comprehensive distributions in SciEvents.

Nugget Type Distribution. Table 23 and Figure 7 present distribution of nugget types across train, develop and test splits.

Event Type Distribution. Table 24 and Figure 8 present distribution of event types across train, develop and test splits.

Argument Type Distribution. Table 25 and Figure 9 present distribution of argument types across train, develop and test splits.

Document Length-Event Instance Distribution. Figure 10 present the distribution of document length versus event instance.

Discontinuous Nugget Distribution. Figures 11a, 11b, and 11c present the distribution of discontinuous nuggets over nugget types, event types and argument types, respectively.

Overlapping Nugget Distribution. Figures 11d, 11e, and 11f present the distribution of overlapping nuggets over nugget types, event types and argument types, respectively.

Reverse-order Nugget Distribution. Figures 11g, 11h, and 11i present the distribution of reverse-order nuggets over nugget types, event types and argument types, respectively.

Sub-event Distribution. Figures 11j, 11k, and 11l present the distribution of sub-events over event types, argument types and sub-event types, respectively.

I Dataset Annotation Protocol and Reproducibility Details

To facilitate reproducibility and provide practical guidance for future research, we detail the annotation protocol of SciEvents, which begins with the finalization of the schema and ends with the completion of the dataset.

With the defined schema of SciEvents, we hire a professional annotation company to support the annotation work of SciEvents. The entire annotation process is organized as a formal project by the annotation company, comprising the following four stages: project familiarization, annotator selection, annotator training, and formal annotation. We provide the details of each stage as follows:

Project Familiarization Stage. In this stage, we engage in in-depth discussions with senior staff from the annotation company. Specifically, we communicate closely with three senior staff members to clarify the input and output formats, data sources, schema, and annotation guidelines. These staff members are referred to as supervisors, as they will play leading roles in subsequent stages of the project.

To prepare the supervisors for leading the subsequent annotation process, we further conduct an iterative annotation and alignment procedure. In each round, the three supervisors independently annotate the same set of 10 documents, followed by a joint discussion with our team to align their understanding of the annotation guidelines and refine the guidelines accordingly. This process is repeated until a consistent understanding is reached. In total, the procedure is conducted for approximately five rounds, with each supervisor annotating 50 documents.

Annotator Selection Stage. This stage is primarily led by the three supervisors. Specifically, they organize an internal project briefing within the company to recruit candidates for the pre-annotation phase, resulting in 21 participants. Based on two

criteria, the basic understanding of the annotation guidelines and the ability to correctly identify event occurrences, they select 7 candidates as qualified annotators for the subsequent stages.

Annotator Training. This stage is conducted in parallel with the formal annotation stage. It consists of three components: (1) *One-on-one training:* Before formal annotation, each annotator receives approximately three days of one-on-one training from the supervisors, during which annotators are required to annotate at least 15 documents; (2) *On-demand support:* When annotators encounter ambiguous or difficult cases during annotation, they can directly consult the supervisors for clarification; (3) *Regular group sessions:* At least once per week, supervisors organize group sessions to summarize common issues identified during quality inspection and provide unified explanations.

Formal Annotation Stage. This stage includes both the annotation process and quality inspection. Most details are provided in the main paper. The annotation tool is independently developed and customized by the annotation company. Figure 6a and Figure 6b illustrate representative interfaces for annotation and quality inspection, respectively.

J Dataset Annotation Remunerations

During the official annotation stage, annotators spend approximately 364 minutes to annotate 19 consecutive documents, corresponding to an average of 19.1 minutes per document. Annotators are compensated at approximately $4.5 per document, which is aligned with local wage standards and ensures fair remuneration for the annotation work.

(a) Annotation interface.

(b) Quality inspection interface.

Figure 6: Representative screenshots of the annotation and quality inspection tools developed by the annotation company (presented in Chinese). (a) The annotation interface supports adding events, selecting event types, annotating spans, assigning nugget and argument types, and submitting annotations. (b) The quality inspection interface allows inspectors to modify annotations, provide feedback, and mark documents as approved or returned for revision.

Adaptive Compression of Word Embeddings					
Document ID: f63de5c23cce0cc5bb67d42ab12e7bed					
Abstract: Distributed representations of words have been an indispensable **component**$_{\mathrm{E1}}$ for natural language processing (NLP) tasks. However, the **large memory footprint**$_{\mathrm{E2}}$ of word embeddings makes it challenging to deploy NLP models to memory-constrained devices (e.g., self-driving cars, mobile devices). In this paper, we **propose**$_{\mathrm{E3}}$ a novel method to **adaptively compress**$_{\mathrm{E4}}$ word embeddings. We fundamentally **follow**$_{\mathrm{E5}}$ a code-book approach that **represents**$_{\mathrm{E6}}$ words as discrete codes such as (8, 5, 2, 4). However, unlike prior works that assign the same length of codes to all words, we adaptively **assign**$_{\mathrm{E8}}$ different lengths of codes to each word by **learning**$_{\mathrm{E7}}$ downstream tasks. The proposed method works in two steps. First, each word directly learns to **select**$_{\mathrm{E10}}$ its code length in an end-to-end manner by **applying**$_{\mathrm{E9}}$ the Gumbel-softmax tricks. After selecting the code length, each word **learns**$_{\mathrm{E12}}$ discrete codes **through**$_{\mathrm{E11}}$ a neural network with a binary constraint. To **showcase**$_{\mathrm{E14}}$ the general applicability of the proposed method, we **evaluate**$_{\mathrm{E13}}$ the performance on four different downstream tasks. Comprehensive evaluation results clearly **show**$_{\mathrm{E15}}$ that our method is **effective**$_{\mathrm{E16}}$ and **makes**$_{\mathrm{E17}}$ the highly compressed word embeddings without hurting the task accuracy. Moreover, we **show**$_{\mathrm{E18}}$ that our model **assigns**$_{\mathrm{E20}}$ word to each code-book by **considering**$_{\mathrm{E19}}$ the significance of tasks.					
Event ID	Event Type	Trigger	Arg Type	Arg Text	Nugget Type
E1	ITT	component	Target	natural language processing	TAK
E2	RWF	large memory footprint	Concern	word embeddings	MOD
E3	PRP	propose	Proposer	we	OG
			Content	method	APP
			Target	adaptively compress	E-PUR†
E4	PUR*	adaptively compress	Aim	word embeddings	MOD
E5	WKS	follow	Researcher	we	OG
			Content	code - book approach	APP
			Target	represents	E-PUR†
E6	PUR*	represents	Aim	words	FEA
			Condition	as discrete codes	LIM
E7	WKS	learning	Content	downstream tasks	TAK
			Researcher	we	OG
			Target	assign	E-PUR†
E8	PUR*	assign	Aim	different lengths of codes	FEA
			Condition	to each word	LIM
E9	MDS	applying	Target	select	E-PUR†
			TriedComponent	gumbel - softmax tricks	APP
			BaseComponent	word	FEA
E10	PUR*	select	Aim	code length	TAK
			Condition	in an end - to - end manner	LIM
E11	MDS	through	Target	learns	E-PUR†
			BaseComponent	word	FEA
			TriedComponent	neural network with a binary constraint	APP
			Condition	after selecting the code length	LIM
E12	PUR*	learns	Aim	discrete codes	TAK
E13	WKS	evaluate	Researcher	we	OG
			Content	performance	TAK
			Condition	on four different downstream tasks	LIM
			Target	showcase	E-PUR†
E14	PUR*	showcase	Aim	general applicability	TAK
E15	FIN	show	Content	effective	E-FAC†
			Content	makes	E-FAC†
E16	FAC*	effective	Subject	method	APP
E17	FAC*	makes	Condition	without hurting the task accuracy	LIM
			Subject	method	APP
			Object	highly compressed word embeddings	STR
E18	FIN	show	Finder	we	OG
			Content	considering	E-FAC†
E19	FAC*	considering	Object	significance of tasks	TAK
			Target	assigns	E-PUR†
			Subject	model	APP
E20	PUR	assigns	Aim	word	FEA
			Condition	to each code - book	LIM

Table 19: Event extraction annotations for the paper *Adaptive Compression of Word Embeddings*. * indicates that the event is a sub-event; † indicates that the argument is a sub-event argument (nugget_type starts with E-).

Model		TI(%) P	TI(%) R	TI(%) F1	TC(%) P	TC(%) R	TC(%) F1
Global	OneIE	$74.18_{\pm 0.24}$	$77.33_{\pm 0.38}$	$75.72_{\pm 0.14}$	$61.68_{\pm 0.06}$	$64.23_{\pm 0.42}$	$62.93_{\pm 0.17}$
Discriminative	EEQA	$77.04_{\pm 2.13}$	$72.96_{\pm 3.19}$	$74.85_{\pm 0.78}$	$64.03_{\pm 1.87}$	$60.53_{\pm 2.64}$	$62.15_{\pm 0.73}$
	Tagprime	$75.72_{\pm 1.16}$	$71.04_{\pm 1.98}$	$73.27_{\pm 0.52}$	$64.85_{\pm 1.54}$	$61.36_{\pm 1.05}$	$63.03_{\pm 0.20}$
Generative	DEGREE	$64.98_{\pm 1.72}$	$66.53_{\pm 1.08}$	$65.72_{\pm 0.70}$	$51.93_{\pm 1.80}$	$55.27_{\pm 0.42}$	$53.52_{\pm 0.75}$
	KnowCoder	$69.83_{\pm 0.58}$	$69.95_{\pm 0.86}$	$69.88_{\pm 0.61}$	$52.02_{\pm 0.30}$	$52.03_{\pm 0.55}$	$52.02_{\pm 0.34}$
EXCEEDS (Ours)		$73.90_{\pm 0.78}$	$76.76_{\pm 1.44}$	$75.29_{\pm 0.32}$	$62.28_{\pm 0.78}$	$65.31_{\pm 1.14}$	$63.74_{\pm 0.14}$
– Contextual		$73.02_{\pm 1.38}$	$77.75_{\pm 1.38}$	$75.29_{\pm 0.21}$	$61.34_{\pm 1.46}$	$65.74_{\pm 1.26}$	$63.44_{\pm 0.67}$
– Grid Refiner		$73.73_{\pm 0.73}$	$77.09_{\pm 1.01}$	$75.36_{\pm 0.27}$	$61.76_{\pm 0.97}$	$65.16_{\pm 1.02}$	$63.41_{\pm 0.67}$

Table 20: Precision, recall and F1-score (%) of trigger identification (TI) and trigger classification (TC) on SciEvents. EAE-only models are not presented.

Model		AI(%) P	AI(%) R	AI(%) F1	AC(%) P	AC(%) R	AC(%) F1
Global	OneIE	$28.71_{\pm 1.80}$	$32.09_{\pm 1.90}$	$30.30_{\pm 1.85}$	$27.29_{\pm 1.73}$	$30.49_{\pm 1.82}$	$28.81_{\pm 1.77}$
	ScentedEAE	$50.73_{\pm 1.66}$	$29.07_{\pm 4.12}$	$36.70_{\pm 2.99}$	$49.48_{\pm 2.32}$	$28.29_{\pm 3.61}$	$35.74_{\pm 2.41}$
Discriminative	EEQA	$32.65_{\pm 0.97}$	$44.93_{\pm 2.45}$	$37.75_{\pm 0.46}$	$30.75_{\pm 0.78}$	$42.55_{\pm 2.53}$	$35.64_{\pm 0.59}$
	PAIE	$46.86_{\pm 0.26}$	$41.34_{\pm 0.59}$	$43.92_{\pm 0.22}$	$44.82_{\pm 0.31}$	$39.63_{\pm 0.65}$	$42.06_{\pm 0.34}$
	Tagprime	$47.79_{\pm 0.71}$	$41.95_{\pm 0.47}$	$44.67_{\pm 0.13}$	$45.67_{\pm 0.55}$	$40.09_{\pm 0.67}$	$42.69_{\pm 0.32}$
	DEEIA	$43.06_{\pm 2.41}$	$29.33_{\pm 0.37}$	$34.86_{\pm 0.81}$	$41.05_{\pm 2.23}$	$28.06_{\pm 0.36}$	$33.30_{\pm 0.75}$
Generative	BartGen	$41.21_{\pm 1.34}$	$38.60_{\pm 0.58}$	$39.85_{\pm 0.52}$	$39.10_{\pm 1.21}$	$36.63_{\pm 0.48}$	$37.81_{\pm 0.39}$
	DEGREE	$29.03_{\pm 1.33}$	$27.81_{\pm 0.50}$	$28.40_{\pm 0.88}$	$26.87_{\pm 1.21}$	$25.79_{\pm 0.43}$	$26.32_{\pm 0.79}$
	KnowCoder	$37.82_{\pm 0.17}$	$33.00_{\pm 0.17}$	$35.24_{\pm 0.11}$	$35.89_{\pm 0.35}$	$31.29_{\pm 0.24}$	$33.43_{\pm 0.27}$
EXCEEDS (Ours)		$50.86_{\pm 2.68}$	$40.48_{\pm 1.92}$	$44.97_{\pm 0.28}$	$48.82_{\pm 2.81}$	$38.91_{\pm 1.66}$	$43.20_{\pm 0.29}$
– Contextual		$48.64_{\pm 2.16}$	$40.40_{\pm 1.70}$	$44.07_{\pm 0.51}$	$46.46_{\pm 2.35}$	$38.68_{\pm 1.43}$	$42.14_{\pm 0.39}$
– Grid Refiner		$49.64_{\pm 1.24}$	$40.03_{\pm 0.94}$	$44.30_{\pm 0.51}$	$47.51_{\pm 1.37}$	$38.39_{\pm 0.87}$	$42.44_{\pm 0.59}$

Table 21: Precision, recall and F1-score (%) of argument identification (AI) and argument classification (AC) on SciEvents. For EAE-only models, trigger predictions are derived from Tagprime.

Model		EC(%) P	EC(%) R	EC(%) F1
Global	OneIE	$35.26_{\pm 1.48}$	$39.85_{\pm 1.61}$	$37.41_{\pm 1.51}$
	ScentedEAE	$51.21_{\pm 2.24}$	$30.12_{\pm 2.32}$	$37.88_{\pm 2.10}$
Discriminative	EEQA	$44.45_{\pm 3.77}$	$45.57_{\pm 2.38}$	$44.81_{\pm 1.35}$
	PAIE	$49.98_{\pm 2.11}$	$44.70_{\pm 0.06}$	$47.17_{\pm 0.92}$
	Tagprime	$50.22_{\pm 1.42}$	$45.53_{\pm 1.46}$	$47.72_{\pm 0.32}$
	DEEIA	$40.64_{\pm 3.35}$	$27.53_{\pm 0.79}$	$32.80_{\pm 1.67}$
Generative	BartGen	$46.93_{\pm 1.37}$	$39.30_{\pm 0.87}$	$42.75_{\pm 0.11}$
	DEGREE	$39.60_{\pm 1.01}$	$23.55_{\pm 0.86}$	$29.53_{\pm 0.96}$
	KnowCoder	$41.03_{\pm 0.94}$	$29.85_{\pm 1.22}$	$34.54_{\pm 0.82}$
EXCEEDS (Ours)		$48.28_{\pm 1.05}$	$48.29_{\pm 1.22}$	$48.25_{\pm 0.10}$
– Contextual		$47.05_{\pm 1.10}$	$48.32_{\pm 2.13}$	$47.64_{\pm 0.85}$
– Grid Refiner		$47.33_{\pm 0.86}$	$48.80_{\pm 1.80}$	$48.04_{\pm 1.07}$

Table 22: Precision, recall and F1-score (%) of event correlation (EC) on SciEvents. For EAE-only models, trigger predictions are derived from Tagprime.

Nugget Type	Train	Develop	Test	Total
APP	8,860 (22.16%)	1,068 (21.38%)	1,146 (22.50%)	11,074 (22.12%)
TAK	8,219 (20.56%)	993 (19.88%)	1,042 (20.46%)	10,254 (20.48%)
FEA	6,147 (15.38%)	801 (16.03%)	764 (15.00%)	7,712 (15.40%)
OG	4,663 (11.67%)	567 (11.35%)	564 (11.07%)	5,794 (11.57%)
LIM	3,933 (9.84%)	536 (10.73%)	505 (9.92%)	4,974 (9.94%)
STR	1,986 (4.97%)	247 (4.94%)	249 (4.89%)	2,482 (4.96%)
WEA	1,779 (4.45%)	216 (4.32%)	246 (4.83%)	2,241 (4.48%)
DST	1,736 (4.34%)	219 (4.38%)	238 (4.67%)	2,193 (4.38%)
MOD	1,652 (4.13%)	240 (4.80%)	206 (4.04%)	2,098 (4.19%)
DEG	998 (2.50%)	109 (2.18%)	133 (2.61%)	1,240 (2.48%)
Total	**39,973**	**4,996**	**5,093**	**50,062**

Table 23: Distribution of Nugget Types across Train, Develop and Test Splits

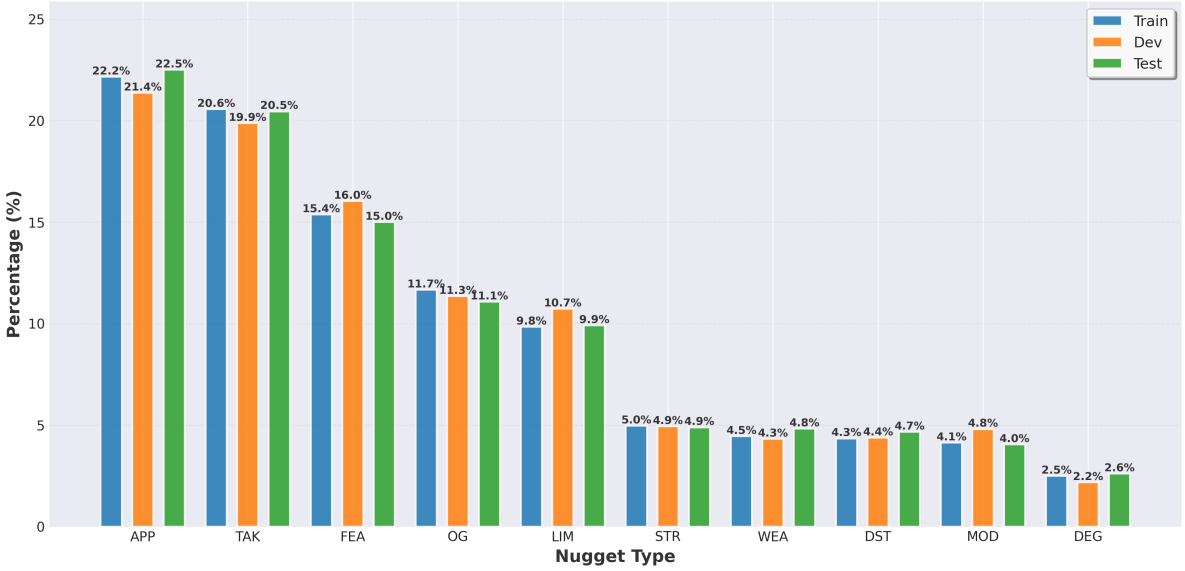


Figure 7: Percentage Distribution of Nugget Types across Train, Develop and Test Splits

Event Type	Train	Develop	Test	Total
PUR	3,320 (17.04%)	395 (16.16%)	416 (16.99%)	4,131 (16.94%)
WKS	2,733 (14.02%)	354 (14.48%)	332 (13.56%)	3,419 (14.02%)
FAC	2,528 (12.97%)	313 (12.80%)	324 (13.24%)	3,165 (12.98%)
MDS	2,385 (12.24%)	303 (12.39%)	300 (12.25%)	2,988 (12.26%)
PRP	2,094 (10.75%)	235 (9.61%)	257 (10.50%)	2,586 (10.61%)
RWF	1,549 (7.95%)	201 (8.22%)	205 (8.37%)	1,955 (8.02%)
CMP	1,536 (7.88%)	192 (7.85%)	195 (7.97%)	1,923 (7.89%)
FIN	1,480 (7.59%)	192 (7.85%)	175 (7.15%)	1,847 (7.58%)
ITT	1,400 (7.18%)	190 (7.77%)	174 (7.11%)	1,764 (7.24%)
RWS	463 (2.38%)	70 (2.86%)	70 (2.86%)	603 (2.47%)
Total	**19,488**	**2,445**	**2,448**	**24,381**

Table 24: Distribution of Event Types across Train, Develop and Test Splits

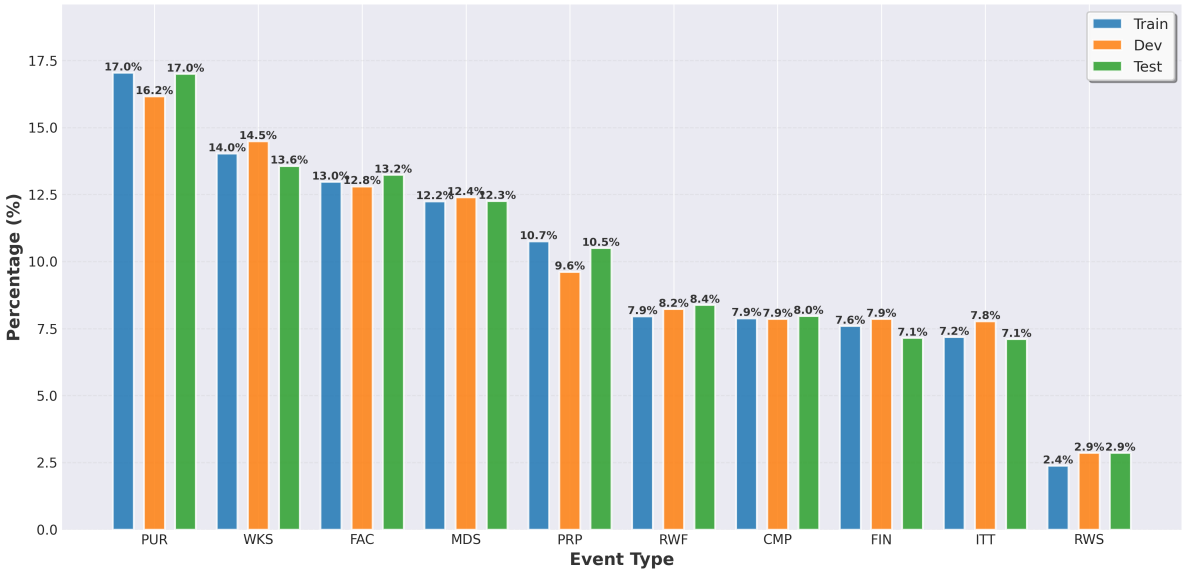


Figure 8: Percentage Distribution of Event Types across Train, Develop and Test Splits

Argument Type	Train	Develop	Test	Total
Content	7,010 (15.55%)	877 (15.61%)	853 (14.92%)	8,740 (15.49%)
Target	6,982 (15.49%)	844 (15.02%)	876 (15.32%)	8,702 (15.43%)
Condition	3,919 (8.69%)	534 (9.51%)	506 (8.85%)	4,959 (8.79%)
Aim	3,574 (7.93%)	424 (7.55%)	443 (7.75%)	4,441 (7.87%)
BaseComponent	2,858 (6.34%)	377 (6.71%)	366 (6.40%)	3,601 (6.38%)
Subject	2,662 (5.91%)	326 (5.80%)	365 (6.38%)	3,353 (5.94%)
TriedComponent	2,143 (4.75%)	274 (4.88%)	279 (4.88%)	2,696 (4.78%)
Researcher	2,067 (4.59%)	256 (4.56%)	250 (4.37%)	2,573 (4.56%)
Object	2,009 (4.46%)	250 (4.45%)	256 (4.48%)	2,515 (4.46%)
Proposer	1,901 (4.22%)	217 (3.86%)	240 (4.20%)	2,358 (4.18%)
Fault	1,577 (3.50%)	191 (3.40%)	231 (4.04%)	1,999 (3.54%)
Arg1	1,329 (2.95%)	164 (2.92%)	155 (2.71%)	1,648 (2.92%)
Result	1,302 (2.89%)	170 (3.03%)	166 (2.90%)	1,638 (2.90%)
Arg2	1,250 (2.77%)	155 (2.76%)	152 (2.66%)	1,557 (2.76%)
Concern	1,066 (2.36%)	149 (2.65%)	144 (2.52%)	1,359 (2.41%)
Extent	998 (2.21%)	109 (1.94%)	133 (2.33%)	1,240 (2.20%)
Dataset	914 (2.03%)	115 (2.05%)	136 (2.38%)	1,165 (2.07%)
Metrics	794 (1.76%)	90 (1.60%)	92 (1.61%)	976 (1.73%)
Finder	695 (1.54%)	94 (1.67%)	74 (1.29%)	863 (1.53%)
Reason	26 (0.06%)	2 (0.04%)	0 (0.00%)	28 (0.05%)
Total	**45,076**	**5,618**	**5,717**	**56,411**

Table 25: Distribution of Argument Types across Train, Develop and Test Splits

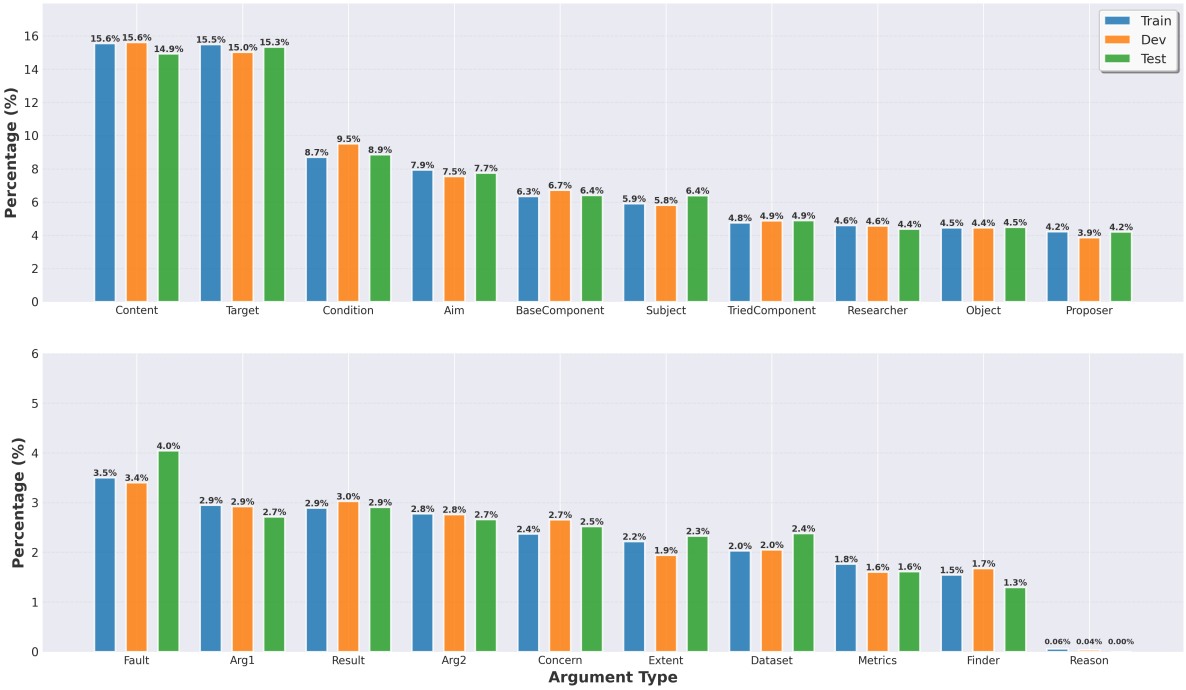


Figure 9: Percentage Distribution of Argument Types across Train, Develop and Test Splits

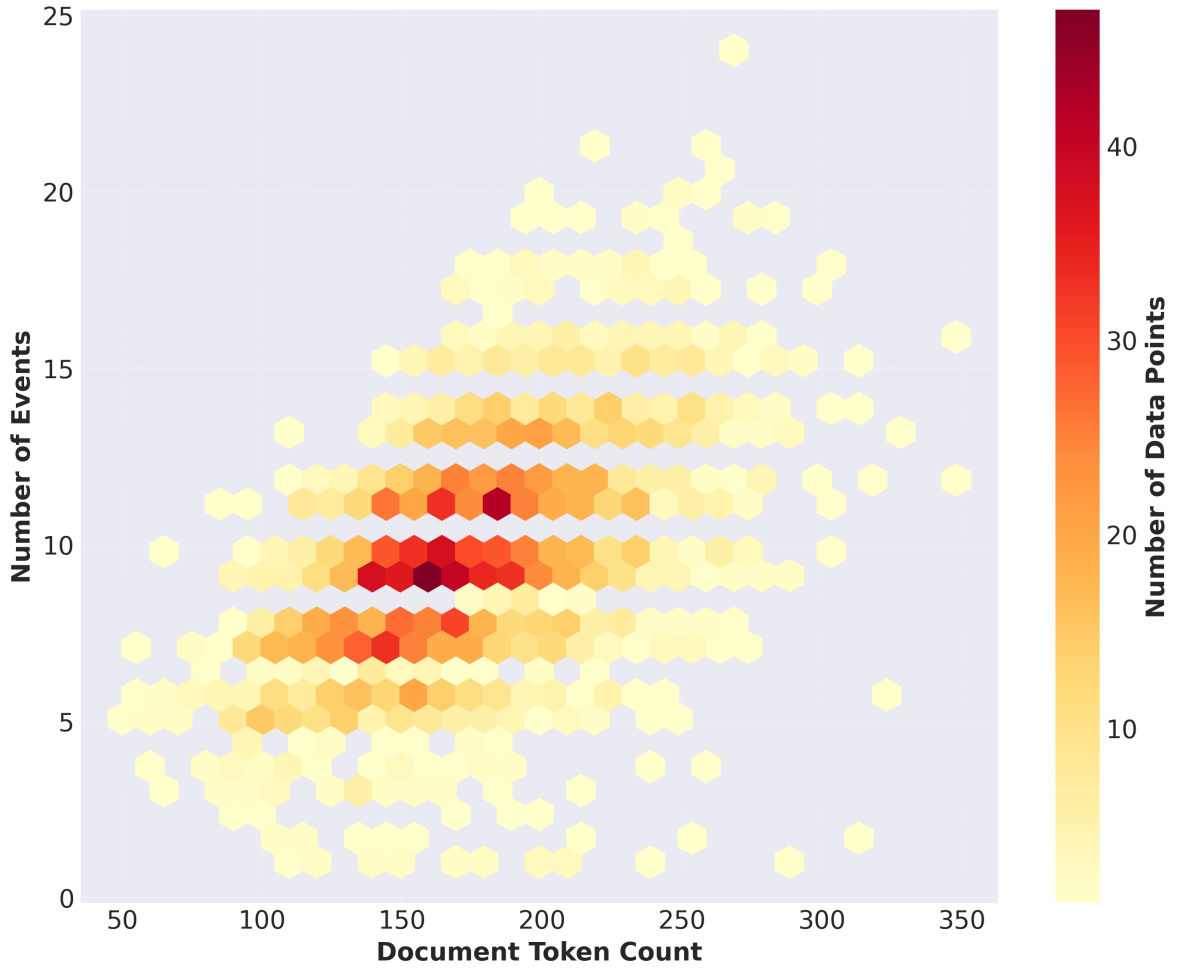
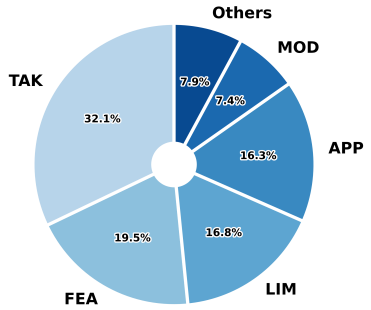
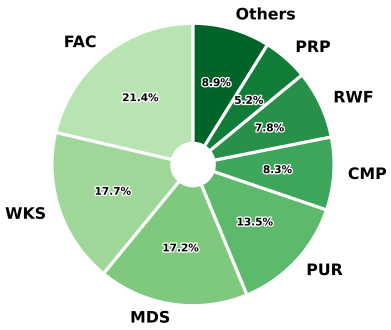


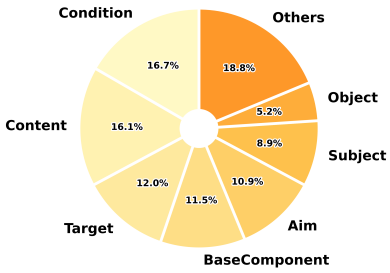
Figure 10: Document Length-Event Instance Distribution



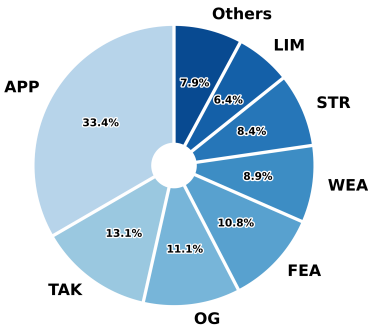
(a) Discontinuous Nugget: Nugget Type



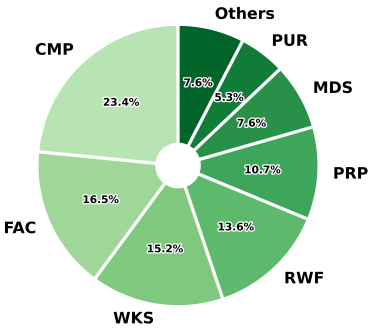
(b) Discontinuous Nugget: Event Type



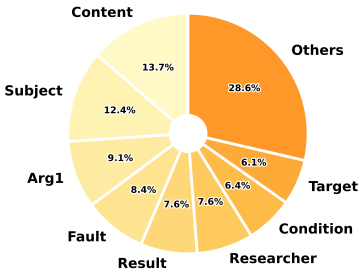
(c) Discontinuous Nugget: Arg Type



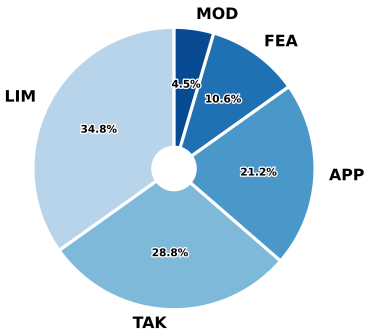
(d) Overlap Nugget: Nugget Type



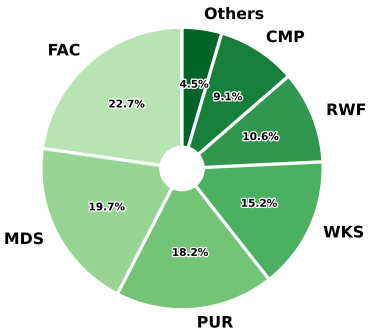
(e) Overlap Nugget: Event Type



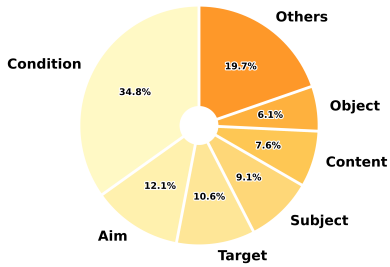
(f) Overlap Nugget: Arg Type



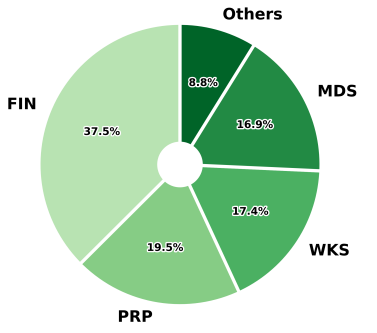
(g) Reverse-Order Nugget: Nugget Type



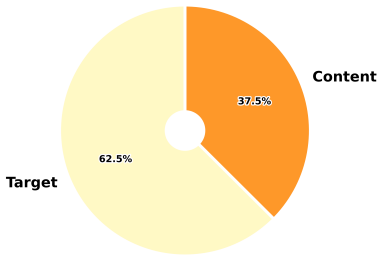
(h) Reverse-Order Nugget: Event Type



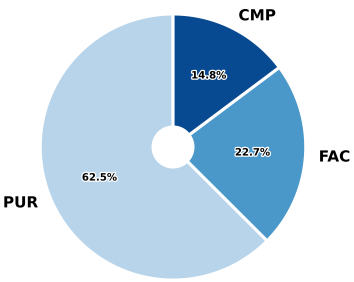
(i) Reverse-Order Nugget: Arg Type



(j) Sub-Event: Event Type



(k) Sub-Event: Argument Type



(l) Sub-Event: Sub-Event Type

Figure 11: Distributions of Complex Nuggets and Events