TextEE: Benchmark, Reevaluation, Reflections, and Future Challenges in Event Extraction

Anonymous ACL submission

Abstract

Event extraction has gained considerable interest due to its wide-ranging applications. However, recent studies draw attention to evaluation issues, suggesting that reported scores may not accurately reflect the true performance. In this work, we identify and address evaluation challenges, including *inconsistency* due to varying data assumptions or preprocessing steps, the *insufficiency* of current evaluation frameworks that may introduce dataset or data split bias, and the *low reproducibility* of some previous approaches. To address these challenges, we present TextEE, a standardized, fair, and reproducible benchmark for event extraction. TextEE comprises standardized data preprocessing scripts and splits for 14 datasets spanning seven diverse domains and includes 14 recent methodologies, conducting a comprehensive benchmark reevaluation. We also evaluate five varied large language models on our TextEE benchmark and demonstrate how they struggle to achieve satisfactory performance. Inspired by our reevaluation results and findings, we discuss the role of event extraction in the current NLP era, as well as future challenges and insights derived from TextEE. We believe TextEE, the first standardized comprehensive benchmarking tool, will significantly facilitate future event extraction research.[1]

1 Introduction

Event extraction (Ji and Grishman, 2008) has been always a challenging task in the field of natural language processing (NLP) due to its demand for a high-level comprehension of texts. Since event extraction benefits many applications (Zhang et al., 2020; Han et al., 2021), it has attracted increasing attention in recent years (Luan et al., 2019; Lin et al., 2020; Nguyen et al., 2021; Hsu et al., 2022; Ma et al., 2022). However, due to the complicated nature of event extraction datasets and systems, fairly evaluating and comparing different event extraction approaches is not straightforward. Recent attempts (Zheng et al., 2021; Peng et al., 2023a,b) point out that the reported scores in previous work might not reflect the true performance in real-world applications because of various shortcomings and issues during the evaluation process. This poses a potential obstacle to the development of robust techniques for research in event extraction.

Motivated by the evaluation concern, this work aims to establish a standardized, fair, and reproducible benchmark for assessing event extraction approaches. We start by identifying and discussing several significant issues in the current evaluation process. First, we discuss the *inconsistency* issue caused by discrepant assumptions about data, different preprocessing steps, and additional external resources. Next, we highlight the *insufficiency* problem of existing evaluation pipelines that cover limited datasets and rely on a fixed data split, which potentially introduces bias when evaluating performance. Finally, we emphasize the importance of *reproducibility*, which indirectly causes the aforementioned inconsistency and insufficiency issues.

To address these evaluation concerns, we propose TextEE, an evaluation platform that covers 14 datasets spanning diverse domains. To ensure fairness in comparisons, we standardize data preprocessing procedures and introduce five standardized data splits. Furthermore, we aggregate and re-implement 14 event extraction approaches published in recent years and conduct a comprehensive reevaluation. TextEE offers the benefits of *consistency*, *sufficiency*, *reproducibility* in evaluation. Additionally, we benchmark several large language models (LLMs) (Touvron et al., 2023; Tunstall et al., 2023; Jiang et al., 2024) for event extraction with TextEE and show the unsatisfactory performance of LLMs for this task.

Based on our reevaluation results and findings, we discuss the role of event extraction in the current

[1] We will open-source TextEE after paper acceptance.

era of LLMs, along with challenges and insights gleaned from TEXTEE. Specifically, we discuss how event extraction systems can be optional tools for LLMs to utilize, as well as highlight future challenges, including enhancing generalization, expanding event coverage, and improving efficiency.

In summary, our contributions are as follows: (1) We highlight and overcome the difficulties of fair evaluation for event extraction tasks. (2) We present TEXTEE as a benchmark platform for event extraction research and conduct a thorough reevaluation of recent approaches as well as LLMs. (3) Based on our results and findings, we discuss limitations and future challenges in event extraction.

2 Background and Related Work

2.1 Event Extraction

Event extraction (EE) aims to identify structured information from texts. Each event consists of an event type, a trigger span, and several arguments as well as their roles.[2] Figure 1 shows an example of a *Justice-Execution* event extracted from the text. This event is triggered by the text span *execution* and contains two argument roles, including *Indonesia (Agent)* and *convicts (Person)*.

Previous work can be categorized into two types. (1) **End-to-end (E2E)** approaches extract event types, triggers, and argument roles in an end-to-end manner. (2) Pipeline approaches first solve the **event detection (ED)** task, which detects trigger spans and the corresponding event types, then deal with the **event argument extraction (EAE)** task, which extracts arguments and the corresponding roles given an event type and a trigger span.

2.2 Related Work

Event extraction. Most end-to-end approaches construct graphs to model the relations between entities and extract triggers and argument roles accordingly (Luan et al., 2019; Wadden et al., 2019; Lin et al., 2020; Nguyen et al., 2021; Zhang and Ji, 2021). There is a recent focus on employing generative models to generate summaries for extracting events (Lu et al., 2021; Hsu et al., 2022). Unlike end-to-end approaches, pipeline methods train two separate models for event detection and event argument extraction. Different techniques are introduced, such as question answering (Du and Cardie, 2020; Liu et al., 2020; Li et al., 2020a; Lu et al., 2023), language generation (Paolini et al., 2021; Hsu et al., 2022), querying and extracting (Wang et al., 2022), pre-training (Wang et al., 2021), and multi-tasking (Lu et al., 2022; Wang et al., 2023b). Some works focus on zero-shot or few-shot settings (Huang et al., 2018; Hsu et al., 2022).

[2]In this work, we only cover closed domain EE with a given ontology. We consider event mentions as events and do not consider event coreference resolution.

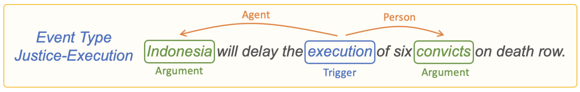

Figure 1: An example of a *Justice-Execution* event. One trigger span (*execution*) and two argument roles, *Indonesia (Agent)* and *convicts (Person)*, are identified.

Event detection. There are many prior studies focusing on extracting triggers only. Most works pay attention to the standard supervised setting (Liu et al., 2018; Lai et al., 2020; Veyseh et al., 2021; Li et al., 2021a; Huang et al., 2022a; Liu et al., 2022a; Li et al., 2023b). Some others study the few-shot setting (Deng et al., 2021; Zhao et al., 2022; Zhang et al., 2022; Ma et al., 2023; Wang et al., 2023a)

Event argument extraction. Event argument extraction has caught much attention in recent years (Veyseh et al., 2022b; Li et al., 2021b; Hsu et al., 2023a; Zeng et al., 2022; Ma et al., 2022; Huang et al., 2022b; Xu et al., 2022; Hsu et al., 2023b; Nguyen et al., 2023; He et al., 2023; Huang et al., 2023; Parekh et al., 2023a). Some works focus on training models with only a few examples (Sainz et al., 2022a; Yang et al., 2023; Wang et al., 2023c).

Event extraction datasets. Most of event extraction datasets come from Wikipedia and the news domain (Doddington et al., 2004; Song et al., 2015; Ebner et al., 2020; Li et al., 2020b, 2021b; Veyseh et al., 2022a; Li et al., 2022). To increase the event type coverage, some works focus on general domain datasets (Wang et al., 2020; Deng et al., 2020; Parekh et al., 2023b; Li et al., 2023b). Recently, datasets in specific domains have been proposed, including cybersecurity (Satyapanich et al., 2020; Trong et al., 2020), pharmacovigilance (Sun et al., 2022), and historical text (Lai et al., 2021).

Event extraction evaluation and analysis. Recently, some works point out several pitfalls when training event extraction models and attempt to provide solutions (Zheng et al., 2021; Peng et al., 2023a,b). Our observation partially echos their findings, while our proposed TEXTEE covers more diverse datasets and includes more recent approaches. On the other hand, some studies discuss ChatGPT's performance on event extraction but only for one dataset (Li et al., 2023a; Gao et al., 2023).

3 Issues in Past Evaluation

Despite a wide range of works in EE, we identify several major issues of the past evaluation. We classify those issues into three categories: *inconsistency*, *insufficiency*, and *low reproducibility*.

Inconsistency. Due to the lack of a standardized evaluation framework, we notice that many studies utilize varied experimental setups while comparing their results with reported numbers in the literature. This leads to unfair comparisons and makes the evaluation less reliable and persuasive. We identify and summarize the underlying reasons as follows:

- **Different assumptions about data.** In the past, different approaches tend to have their own assumptions of data. For instance, some works allow trigger spans consisting of multiple words (Lin et al., 2020; Hsu et al., 2022, 2023a), whereas others consider only single-word triggers (Liu et al., 2020; Du and Cardie, 2020; Wang et al., 2022); some studies assume that there are no overlapping argument spans (Zhang and Ji, 2021), while others can handle overlapping spans (Wadden et al., 2019; Huang et al., 2022b); some methods filter out testing data when the texts are too long (Liu et al., 2022a) but others do not (Hsu et al., 2023b; Ma et al., 2022). Due to these discrepancies in assumptions, the reported numbers from the original papers are actually not directly comparable.
- **Different data preprocessing steps.** Many previous works benchmark on the ACE05 (Doddington et al., 2004) and RichERE (Song et al., 2015) datasets. Since these datasets are behind the paywall and not publicly accessible, people can only share the data preprocessing scripts. Unfortunately, we observe that some popular preprocessing scripts can generate very different data. For instance, the processed ACE05 datasets from Wadden et al. (2019), Li et al. (2020a), and Veyseh et al. (2022b) have varying numbers of role types (22, 36, and 35 respectively). In addition, it is crucial to note that variations in Python package versions can lead to different generated data even when using the same script. For example, different versions of `nltk` packages may have discrepancies in sentence tokenization and word tokenization, resulting in different processed data. Such differences in preprocessing largely affect model evaluation, leading to significant discrepancies (e.g., over 4 F1 score), thereby reducing persuasiveness (Peng et al., 2023b).
- **Different external resources.** We notice that many approaches utilize additional resources without clearly describing the experimental setting difference. For example, Wang et al. (2023a) employs part-of-speech tags for event detection; Sainz et al. (2022b) and Wang et al. (2022) consider gold entity annotations for event argument extraction. The setting difference can lead to potential unfair comparisons.

Insufficiency. We argue that the existing evaluation process used by the majority of approaches cannot thoroughly evaluate the capabilities of event extraction models due to the following aspects:

- **Limited dataset coverage.** Early works usually utilize ACE05 (Doddington et al., 2004) and RichERE (Song et al., 2015) as the evaluation datasets. Consequently, most follow-up works adopt the same two datasets for comparison regardless that several new datasets across diverse domains are proposed (Li et al., 2021b; Sun et al., 2022; Tong et al., 2022; Parekh et al., 2023b). The limited dataset coverage may introduce domain bias and lead to biased evaluations.
- **Data split bias.** Although many works address model randomness by averaging multiple experimental runs (Zhang and Ji, 2021; Hsu et al., 2022; Wang et al., 2022), they often overlook randomness in data splits and report numbers only for a *single* and *fixed* split for train, dev, and test sets. This can lead to a notable bias, especially for event extraction where there is a high variance of annotation density across sentences or documents. For example, following the preprocessing step of Wadden et al. (2019) applied to ACE05, the resulting processed dataset has 33 event types in the train set, 21 event types in the dev set, and 31 event types in the test set. Accordingly, it is likely to have a significant performance discrepancy between the dev and the test set, making the reported numbers biased.

Low reproducibility. Because of the complex nature of the event extraction tasks, the event extraction models have become increasingly complicated. Releasing code and checkpoints for reproducing results has become essential as many details and tricks need to be taken into account during the re-implementation process. However, many promis-

Dataset	Task	#Docs	#Inst	#ET	#Evt	#RT	#Arg	Event	Entity	Relation	Domain
ACE05 (Doddington et al., 2004)	E2E, ED, EAE	599	20920	33	5348	22	8097	✓	✓	✓	News
RichERE (Song et al., 2015)	E2E, ED, EAE	288	11241	38	5709	21	8254	✓	✓	✓	News
MLEE (Pyysalo et al., 2012)	E2E, ED, EAE	262	286	29	6575	14	5958	✓	✓	✓	Biomedical
Genia2011 (Kim et al., 2011)	E2E, ED, EAE	960	1375	9	13537	10	11865	✓	✓		Biomedical
Genia2013 (Kim et al., 2013)	E2E, ED, EAE	20	664	13	6001	7	5660	✓	✓	✓	Biomedical
M^2E^2 (Li et al., 2020b)	E2E, ED, EAE	6013	6013	8	1105	15	1659	✓	✓		Multimedia
CASIE (Satyapanich et al., 2020)	E2E, ED, EAE	999	1483	5	8469	26	22575	✓			Cybersecurity
PHEE (Sun et al., 2022)	E2E, ED, EAE	4827	4827	2	5019	16	25760	✓			Pharmacovigilance
MAVEN (Wang et al., 2020)	ED	3623	40473	168	96897	–	–	✓			General
FewEvent (Deng et al., 2020)	ED	12573	12573	100	12573	–	–	✓			General
MEE (Veyseh et al., 2022a)	ED	13000	13000	16	17257	–	–	✓	✓		Wikipedia
WikiEvents (Li et al., 2021b)	EAE	245	565	50	3932	58	5501	✓	✓		Wikipedia
RAMS (Ebner et al., 2020)	EAE	9647	9647	139	9647	65	21206	✓	✓		News
GENEVA (Parekh et al., 2023b)	EAE	262	3684	115	7505	220	12314	✓	✓		General

Table 1: TEXTEE supports fourteen datasets across various domains. *#Docs, #Inst, #ET, #EvT, #RT, and #Arg represent the number of documents, instances, event types, events, roles, and arguments, respectively. Event, Entity, and Relation indicate if the dataset contains the corresponding annotations.*

ing approaches do not provide an official codebase (Li et al., 2020a; Nguyen et al., 2021; Wei et al., 2021; Liu et al., 2022b), which potentially impedes the progress of research in event extraction.

4 Benchmark and Reevaluation

To address the issues listed in Section 3, we present TEXTEE, a framework aiming to standardize and benchmark the evaluation process of event extraction. TEXTEE has several advantages as follows.

Better Consistency. We propose a standardized experimental setup for fair comparisons.

- **Normalizing assumptions about data.** We adopt the loosest assumption about data to align with real-world cases effectively. This includes allowing multiple-word triggers, considering overlapping argument spans, and retaining all instances without filtering.
- **Standardizing data preprocessing steps.** We provide a standard script for data preprocessing, including tokenization and label offset mapping. To avoid the difference caused by variations in Python package versions, we use `stanza 1.5.0` for tokenization and save all the offsets. Our script will load the saved offsets during preprocessing, ensuring that everyone can generate exactly the same data.
- **Specifying additional resources.** We clearly specify the resources utilized by all baselines (Table 2). For the approaches that require additional gold annotations (such as POS tags, AMR, and gold entities), considering the purpose of fair comparisons, we either train a new predictor from training annotations (for entities) or use a pre-trained model (for POS tags and AMR), and consider the predicted labels as a substitute for the gold annotations.

Improved Sufficiency. We improve the sufficiency of the evaluation process as follows.

- **Increasing dataset coverage.** As listed in Table 1, we increase the dataset coverage by including *fourteen* event extraction datasets that cover various domains.
- **Providing standard data splits.** For each dataset, we merge all the labeled data and regenerate data splits. To mitigate the data split bias, we offer *five* split for each dataset and report the average results. To reduce the distribution gap among the train, dev, and test sets, we select splits that these sets share the most similar statistics, such as the number of event types and role types, as well as the number of events and arguments. Table 8 in Appendix A lists the detailed statistics of each split for each dataset.
- **New evaluation metrics.** Most prior works follow Lin et al. (2020) and consider Trigger F1-score and Argument F1-score as the evaluation metrics. Specifically, they calculate F1-scores regarding the following: (1) **TI**: if the *(start_idx, end_idx)* of a predicted trigger match the gold ones. (2) **TC**: if the *(start_idx, end_idx, event_type)* of a predicted trigger match the gold ones. (3) **AI**: if the *(start_idx, end_idx, event_type)* of a predicted argument match the gold ones. (4) **AC**: if the *(start_idx, end_idx, event_type, role_type)* of a predicted argument match the gold ones. However, we notice that AI and AC cannot precisely evaluate the quality of predicted arguments. There can be multiple triggers sharing the same event type in an instance,

Model	Task	Event	Entity	Relation	POS Tags	AMR	Verbalization	Template
Classification-Based Models								
DyGIE++ (Wadden et al., 2019)	E2E	✓	✓	✓				
OneIE (Lin et al., 2020)	E2E	✓	✓	✓				
AMR-IE (Zhang and Ji, 2021)	E2E	✓	✓	✓		✓		
EEQA (Du and Cardie, 2020)	ED, EAE	✓						✓
RCEE (Liu et al., 2020)	ED, EAE	✓						✓
Query&Extract (Wang et al., 2022)	ED, EAE	✓			✓		✓	
TagPrime-C (Hsu et al., 2023a)	ED, EAE	✓					✓	
TagPrime-CR (Hsu et al., 2023a)	EAE	✓					✓	
UniST (Huang et al., 2022a)	ED	✓					✓	
CEDAR (Li et al., 2023b)	ED	✓					✓	
Generation-Based Models								
DEGREE (Hsu et al., 2022)	E2E, ED, EAE	✓					✓	✓
BART-Gen (Li et al., 2021b)	EAE	✓						✓
X-Gear (Huang et al., 2022b)	EAE	✓						
PAIE (Ma et al., 2022)	EAE	✓					✓	✓
AMPERE (Hsu et al., 2023b)	EAE	✓				✓	✓	✓

Table 2: TEXTEE supports various models with different assumptions. *Event, Entity, Relation, POS Tags, and AMR indicate if the model considers the corresponding annotations. Verbalization: if the model requires verbalized type strings. Template: if the model needs a human-written template to connect the semantics of triggers and arguments.*

but the current score *does not* evaluate if the predicted argument attaches to the correct trigger. Accordingly, we propose two new scores to evaluate this attachment: (5) **AI+**: if the *(start_idx, end_idx, event_type, attached_trigger_offsets)* of a predicted argument match the gold ones. (6) **AC+**: if the *(start_idx, end_idx, event_type, attached_trigger_offsets, role_type)* of a predicted argument match the gold ones.

Reproducibility. We open-source the proposed TEXTEE framework for better reproducibility. Additionally, we encourage the community to contribute their datasets and codebases to further the research in event extraction.

4.1 TEXTEE Benchmark

TEXTEE supports 14 datasets across various domains and 14 models proposed in recent years.

Dataset. In addition to the two most common datasets, **ACE05** (Doddington et al., 2004) and **RichERE** (Song et al., 2015), which particularly focus on the news domain, we consider as many other event extraction datasets across diverse domains as possible, including **MLEE** (Pyysalo et al., 2012), **Genia2011** (Kim et al., 2011), and **Genia2013** (Kim et al., 2013) from the biomedical domain, **CASIE** (Satyapanich et al., 2020) from the cybersecurity domain, **PHEE** (Sun et al., 2022) from the pharmacovigilance domain, **M^{2}E^2** (Li et al., 2020b) and **RAMS** (Ebner et al., 2020) from the news domain, **MEE** (Veyseh et al., 2022a) and **WikiEvents** (Li et al., 2021b) from Wikipedia, **MAVEN** (Wang et al., 2020), **FewEvent** (Deng et al., 2020), and **GENEVA** (Parekh et al., 2023b) from the general domain. We also notice that there are other valuable datasets, such as GLEN (Li et al., 2023b) and VOANews (Li et al., 2022), but we do not include them as their training examples are not all annotated by humans. Table 1 summarizes the statistics for each dataset after our preprocessing steps. Appendix A describes the details of the preprocessing steps and our assumptions.

Models. We do our best to aggregate as many models as possible into TEXTEE. For those works having public codebases, we adapt their code to fit our evaluation framework. We also re-implement some models based on the description from the original papers. Currently, TEXTEE supports the following models: (1) *Joint training models* that train ED and EAE together in an end-to-end manner, including **DyGIE** (Wadden et al., 2019), **OneIE** (Lin et al., 2020), and **AMR-IE** (Zhang and Ji, 2021). (2) *Classification-based models* that formulate the event extraction task as a token classification problem, a sequential labeling problem, or a question answering problem, including **EEQA** (Du and Cardie, 2020), **RCEE** (Liu et al., 2020), **Query&Extract** (Wang et al., 2022), **TagPrime** (Hsu et al., 2023a), **UniST** (Huang et al., 2022a), and **CEDAR** (Li et al., 2023b). (3) *Generation-based models* that convert the event extraction task to a conditional generation problem, including **DEGREE** (Hsu et al., 2022), **BART-Gen** (Li et al., 2021b), **X-Gear** (Huang et al., 2022b), **PAIE** (Ma

Model	ACE05		RichERE		MLEE		Genia2011		Genia2013		M^2E^2		ACE05		CASIE	
	TC	AC+	TC	AC+	TC	AC+	TC	AC+	TC	AC+	TC	AC+	TC	AC+	TC	AC+
DyGIE++	71.3	51.8	59.8	38.3	78.2	54.4	70.3	52.1	72.9	57.2	51.0	30.8	44.7	29.5	70.4	45.7
OneIE	71.1	54.7	62.5	45.2	78.5	13.1	72.1	33.6	74.3	32.9	50.6	32.1	70.6	22.1	70.0	29.8
AMR-IE	71.1	54.6	62.3	44.7	78.2	4.7	72.4	29.0	74.5	23.1	50.5	31.9	70.8	3.1	69.4	34.1
EEQA	70.0	50.4	60.2	41.9	76.9	38.1	71.3	38.4	69.4	35.7	51.0	30.2	42.8	26.2	70.3	32.0
RCEE	70.5	51.0	60.0	42.1	77.2	35.4	70.1	37.2	68.0	31.6	48.1	28.0	42.1	23.7	70.9	33.1
Query&Extract	65.1	49.0	59.8	44.5	–	–	–	–	–	–	49.4	28.8	–	–	55.5	31.8
TagPrime	69.9	54.6	63.5	48.4	79.0	60.3	72.2	57.8	73.0	57.4	50.2	32.4	69.3	49.1	71.1	40.6
DEGREE-E2E	66.8	49.1	60.5	43.7	70.2	23.3	59.2	25.4	62.6	24.8	49.5	30.0	60.7	14.6	69.1	36.5
DEGREE-PIPE	68.4	50.7	61.7	44.8	70.4	42.7	60.5	39.8	61.0	41.9	48.3	30.1	57.1	33.7	69.1	36.7

Table 3: Reevaluation results for end-to-end event extraction (E2E). All the numbers are the average score of 5 data splits. Darker cells imply higher scores. We use "–" to denote the cases that models are not runnable.

Model	ACE05	Rich-ERE	MLEE	Genia 2011	Genia 2013	M^2E^2	CASIE	PHEE	MAVEN	Few-Event	MEE
	TC	TC	TC	TC	TC	TC	TC	TC	TC	TC	TC
DyGIE++	71.3	59.8	78.2	70.3	72.9	51.0	44.7	70.4	65.3	65.2	79.8
OneIE	71.1	62.5	78.5	72.1	74.3	50.6	70.6	70.0	65.5	65.4	78.8
AMR-IE	71.1	62.3	78.2	72.4	74.5	50.5	70.8	69.4	–	65.2	–
EEQA	70.0	60.2	77.4	69.6	71.1	51.0	43.2	70.3	64.4	65.1	79.5
RCEE	70.5	60.0	77.3	69.3	70.8	48.1	43.3	70.9	64.6	65.0	79.1
Query&Extract	65.1	59.8	74.9	68.9	70.1	49.4	51.5	55.5	–	63.8	78.1
TagPrime-C	69.9	63.5	79.0	72.2	73.0	50.2	69.3	71.1	66.1	65.6	79.8
UniST	69.8	60.7	74.9	70.3	69.9	49.0	68.1	69.6	63.4	63.1	78.3
CEDAR	62.6	52.3	65.5	66.8	67.1	48.0	67.6	70.3	54.5	52.1	78.6
DEGREE	68.4	61.7	70.4	60.5	61.0	48.3	61.3	69.1	65.5	65.5	78.2

Table 4: Reevaluation results for event detection (ED). All the numbers are the average score of 5 data splits. Darker cells imply higher scores. We use "–" to denote the cases that models are not runnable.

et al., 2022), and **AMPERE** (Hsu et al., 2023b). Table 2 presents the different assumptions and requirements for each model. It is worth noting that some models need additional annotations or information, as indicated in the table. Appendix B lists more details about implementations.

Reevalutation results. For a fair comparison, we utilize RoBERTa-large (Liu et al., 2019) for all the classification-based models and use BART-large (Lewis et al., 2020) for all the generation-based models. Table 3, 4, and 5 present the reevaluation results of end-to-end EE, ED, and EAE, respectively. Appendix C lists more detailed results. We first notice that for end-to-end EE and ED, there is no obvious dominant approach. It suggests that the reported improvements from previous studies may be influenced by dataset bias, data split bias, or data processing. This verifies the importance of a comprehensive evaluation framework that covers various domains of datasets and standardized data splits. TagPrime (Hsu et al., 2023a) and PAIE (Ma et al., 2022) seem to be the two dominant approaches across different types of datasets for EAE. These results validate the effectiveness of those two models, aligning with our expectations for guiding reliable and reproducible research in event extraction with TEXTEE.

In addition, we observe a gap between the established evaluation metrics (AI and AC) and the proposed ones (AI+ and AC+). This implies a potential mismatch between the earlier metrics and the predictive quality. We strongly recommend reporting the attaching score (AI+ and AC+) for future research in event extraction to provide a more accurate assessment of performance.

5 Have LLMs Solved Event Extraction?

Given the demonstrated potential of large language models (LLMs) across various NLP tasks, we discuss the capability of LLMs in solving event extraction tasks. In contrast to previous studies (Li et al., 2023a; Gao et al., 2023), which evaluate a *single* LLM on a *single* EE dataset, we investigate multiple popular LLMs across multiple datasets provided by TEXTEE. We consider **GPT-3.5-Turbo** as well as some open-source LLMs that achieve strong performance on Chatbot Arena (Zheng et al., 2023), including **Llama-2-13b-chat-hf** and **Llama-2-70b-chat-hf** (Touvron et al., 2023), **Zephyr-7b-alpha** (Tunstall et al., 2023), and **Mixtral-8x7B-Instruct** (Jiang et al., 2024), with vLLM framework (Kwon et al., 2023). We evaluate them on the pipelined

Model	ACE05			RichERE			MLEE			Genia2011			Genia2013			M^2E^2		
	AI	AC	AC+	AI	AC	AC+	AI	AC	AC+	AI	AC	AC+	AI	AC	AC+	AI	AC	AC+
DyGIE++	66.9	61.5	60.0	58.5	49.4	47.3	67.9	64.8	62.4	66.1	63.7	61.0	71.7	69.3	66.9	41.7	38.9	38.5
OneIE	75.4	71.5	70.2	71.6	65.8	63.7	31.0	28.9	15.7	62.9	60.3	38.9	57.2	55.7	38.7	59.0	55.2	53.3
AMR-IE	76.2	72.6	70.9	72.8	65.8	63.0	23.2	16.6	6.1	49.1	47.6	35.3	38.9	38.1	26.4	56.0	51.3	50.4
EEQA	73.8	71.4	69.6	73.3	67.3	64.9	64.8	62.1	49.5	63.2	60.8	49.4	64.7	61.1	47.5	57.6	55.9	55.3
RCEE	73.7	71.2	69.4	72.8	67.0	64.5	61.1	58.2	45.1	62.3	59.9	49.6	60.7	57.4	42.7	57.9	56.4	55.8
Query&Extract	77.3	73.6	72.0	76.4	70.9	69.2	–	–	–	–	–	–	–	–	–	59.9	56.2	54.2
TagPrime-C	80.0	76.0	74.5	78.8	73.3	71.4	78.9	76.6	74.5	79.6	77.4	75.8	79.8	77.4	74.9	63.4	60.1	59.0
TagPrime-CR	80.1	77.8	76.2	78.7	74.3	72.5	79.2	77.3	74.6	78.0	76.2	74.5	76.6	74.5	72.3	63.2	60.8	59.9
DEGREE	76.4	73.3	71.8	75.1	70.2	68.8	67.6	65.3	61.5	68.2	65.7	62.4	68.4	66.0	62.5	62.3	59.8	59.2
BART-Gen	76.0	72.6	71.2	74.4	68.8	67.7	73.1	69.8	68.7	73.4	70.9	69.5	76.4	73.6	72.2	62.5	60.0	59.6
X-Gear	76.1	72.4	70.8	75.0	68.7	67.2	64.8	63.3	59.4	68.4	66.2	63.1	64.1	61.9	58.6	62.7	59.8	59.0
PAIE	77.2	74.0	72.9	76.6	71.1	70.0	76.0	73.5	72.4	76.8	74.6	73.4	77.8	75.2	74.2	62.9	60.6	60.4
Ampere	75.5	72.0	70.6	73.8	69.2	67.7	69.2	67.1	62.6	69.5	67.1	63.8	73.2	71.0	67.7	62.1	59.1	58.4

Model	CASIE			PHEE			WikiEvents			RAMS			GENEVA			–
	AI	AC	AC+	AI	AC	AC+	AI	AC	AC+	AI	AC	AC+	AI	AC	AC+	–
DyGIE++	58.0	56.0	51.5	63.4	54.6	54.2	39.8	35.3	34.7	44.3	35.3	35.3	66.0	62.5	62.3	
OneIE	58.3	55.3	27.7	55.9	40.6	40.4	17.5	15.0	7.9	48.0	40.7	40.7	38.9	37.1	36.9	
AMR-IE	35.5	11.0	4.0	60.4	45.3	44.9	17.8	16.0	10.4	49.6	42.3	42.3	23.7	16.6	16.4	
EEQA	56.1	54.0	49.0	53.7	45.6	45.4	54.3	51.7	46.1	48.9	44.7	44.7	69.7	67.3	67.0	
RCEE	47.6	45.3	39.5	54.1	45.8	45.6	53.7	50.9	44.0	45.4	41.5	41.5	66.2	63.8	63.4	
Query&Extract	–	–	–	64.6	54.8	54.4	–	–	–	–	–	–	52.2	50.3	50.0	
TagPrime-C	71.9	69.1	66.1	66.0	55.6	55.3	70.4	65.7	64.0	54.4	48.3	48.3	83.0	79.2	79.0	
TagPrime-CR	71.1	69.2	66.1	65.8	56.0	55.7	70.3	67.2	65.5	54.1	49.7	49.7	82.8	80.4	80.1	
DEGREE	61.0	59.0	54.7	61.7	52.5	52.3	60.4	57.3	53.9	50.5	45.5	45.5	67.2	64.1	63.9	
BART-Gen	63.7	60.0	58.3	57.1	47.7	47.5	68.5	64.2	63.9	50.4	45.4	45.4	67.3	64.4	64.3	
X-Gear	65.7	63.4	59.3	67.6	58.3	58.2	58.7	55.6	52.4	52.1	46.2	46.2	78.9	75.1	74.9	
PAIE	68.1	65.7	64.0	74.9	73.3	73.1	69.8	65.5	65.2	55.2	50.5	50.5	73.5	70.4	70.3	
Ampere	61.1	58.4	53.9	61.4	51.7	51.6	59.9	56.7	53.3	52.0	46.8	46.8	67.8	65.0	64.8	

Table 5: Reevaluation results for event argument extraction (EAE). All the numbers are the average score of 5 data splits. Darker cells imply higher scores. We use "–" to denote the cases that models are not runnable.

tasks of event detection (ED) and event argument extraction (EAE). As part of the prompt, we provide LLMs with the task instructions, a few demonstration examples (positive and negative), and the query text. It is worth noting that the number of demonstration examples will be limited by the maximum length supported by LLMs. Appendix D illustrates the best prompt we use.

Results. Due to the cost and time of running LLMs, we evaluate only on sampled 250 documents for each dataset. Table 6 and 7 list the average results of LLMs as well as some well-performed models selected from TEXTEE. Unlike other NLP tasks such as named entity recognition and commonsense knowledge, where LLMs can achieve competitive performance with fine-tuning models using only a few in-context demonstrations (Wei et al., 2022; Qin et al., 2023), it is noteworthy that there is a large gap between LLMs and the baselines for both the ED and EAE tasks. Our hypothesis is that event extraction requires more recognition of abstract concepts and relations, which is harder compared to other NLP tasks (Li et al., 2023a).

We also manually examine the cases where LLMs make mistakes. For ED, we notice that LLMs usually can recognize trigger spans but fail to predict correct event types, therefore causing many false positives. In contrast, LLMs demonstrate relatively improved performance for EAE. However, they struggle with predicting accurate text spans. Sometimes, LLMs capture the right entities but fail to predict exact offsets as the ground truths. The results suggest that there is still room for improving LLMs' performance.

6 Future Challenges and Opportunities

In this section, we discuss the role of event extraction in the current NLP era, as well as some challenges and insights derived from TEXTEE.

How should we position event extraction in the era of LLMs? Based on the findings in Section 5, LLMs struggle with extracting and comprehending complicated structured semantic concepts. This indicates the need for a dedicated system with specialized design to effectively recognize and extract abstract concepts and relations from texts. We believe that a good event extractor, capable of identifying a wide range of events, could serve as a tool that provides grounded structured information about texts for LLMs. Accordingly, LLMs can flexibly decide whether they require this information

Model	TI	TC
OneIE (Lin et al., 2020)	73.5	69.5
TagPrime-C (Hsu et al., 2023a)	72.5	69.5
Llama-2-13b-chat-hf *(2-shot)*	23.5	9.3
Llama-2-13b-chat-hf *(6-shot)*	28.0	10.4
Llama-2-70b-chat-hf *(2-shot)*	30.6	11.3
Llama-2-70b-chat-hf *(6-shot)*	32.2	12.4
Zephyr-7b-alpha *(2-shot)*	25.0	6.6
Zephyr-7b-alpha *(6-shot)*	26.1	8.0
Zephyr-7b-alpha *(16-shot)*	26.1	9.1
Zephyr-7b-alpha *(32-shot)*	25.2	10.1
Zephyr-7b-alpha *(64-shot)*	23.8	9.7
Mixtral-8x7B-Instruct-v0.1 *(2-shot)*	30.4	10.2
Mixtral-8x7B-Instruct-v0.1 *(6-shot)*	34.4	10.6
Mixtral-8x7B-Instruct-v0.1 *(16-shot)*	35.4	12.1
Mixtral-8x7B-Instruct-v0.1 *(32-shot)*	36.7	13.8
Mixtral-8x7B-Instruct-v0.1 *(64-shot)*	37.5	14.6
gpt-3.5-turbo-1106 *(2-shot)*	33.9	11.8
gpt-3.5-turbo-1106 *(16-shot)*	35.2	12.3

Table 6: Average results over all datasets for event detection (ED) on sampled 250 documents.

Model	AI	AC	AI+	AC+
TagPrime-CR (Hsu et al., 2023a)	73.3	69.5	71.9	68.1
PAIE (Ma et al., 2022)	72.0	68.9	71.3	68.1
Llama-2-13b-chat-hf *(2-shot)*	26.5	19.0	24.1	17.1
Llama-2-13b-chat-hf *(4-shot)*	25.0	18.7	22.8	17.0
Llama-2-70b-chat-hf *(2-shot)*	30.6	24.4	28.5	22.8
Llama-2-70b-chat-hf *(4-shot)*	30.1	23.6	28.3	22.3
Zephyr-7b-alpha *(2-shot)*	28.9	22.6	27.0	21.3
Zephyr-7b-alpha *(4-shot)*	29.3	23.9	27.0	22.4
Zephyr-7b-alpha *(8-shot)*	29.7	25.2	27.7	23.5
Zephyr-7b-alpha *(16-shot)*	27.2	22.5	26.3	21.8
Zephyr-7b-alpha *(32-shot)*	24.3	19.7	23.7	19.3
Mixtral-8x7B-Instruct-v0.1 *(2-shot)*	28.5	23.6	26.7	22.2
Mixtral-8x7B-Instruct-v0.1 *(4-shot)*	30.5	24.7	28.4	23.4
Mixtral-8x7B-Instruct-v0.1 *(8-shot)*	32.9	27.2	30.4	25.4
Mixtral-8x7B-Instruct-v0.1 *(16-shot)*	34.1	28.1	31.4	25.8
Mixtral-8x7B-Instruct-v0.1 *(32-shot)*	35.1	29.2	32.0	26.5
gpt-3.5-turbo-1106 *(2-shot)*	33.2	25.9	30.5	23.8
gpt-3.5-turbo-1106 *(8-shot)*	34.9	26.9	31.8	24.7

Table 7: Average results over all datasets for event argument extraction (EAE) on sampled 250 documents.

for the following reasoning steps or inference process. To achieve this goal, we expect event extractors to be universal, efficient, and accurate, which introduces the following research challenges.

Broader event coverage and generalizablity. We anticipate that a strong event extractor can recognize a wide range of events and even identify new event concepts that may not have appeared during training. This requires two efforts: (1) *Expanding domain coverage in datasets*. Most existing event extraction datasets suffer from a restricted coverage of event types. For instance, all the datasets incorporated by TEXTEE have no more than 200 event types, which is significantly below the amount of human concepts encountered in daily life. Although some recent studies have attempted to tackle this issue (Li et al., 2023b), their data often contains label noise and lacks detailed role annotations. We believe that efficiently collecting or synthesizing high-quality data that covers a wild range of events is crucial for enhancing the emerging ability to generalize event recognition. (2) *Better model design for generalization*. Most existing event extraction models focus on in-domain performance. Therefore, their design can fail when encountering novel events. While exploring prompting in LLMs shows promise, as discussed in Section 5, the results remain unsatisfactory. Some recent works (Lu et al., 2022; Ping et al., 2023) explore learning a unified model across multiple information extraction tasks for improved generalization, but their integration is constrained by limited domains. We expect that TEXTEE can serve as a starting point for aggregating diverse datasets and training more robust unified models.

Enhanced model efficiency. Inference time can pose a bottleneck for effective event extraction, especially when the number of event (role) types increases. For instance, well-performing methods in TEXTEE (e.g., TagPrime and PAIE) require enumerating all the event (role) types, resulting in multiple times of model inference, which significantly slows down as more events (roles) are considered. Similar challenges arise with LLMs, as we have to prompt them per event. Therefore, there is a critical necessity for model designs that not only prioritize performance but also optimize efficiency.

7 Conclusion

In this work, we identify and discuss several evaluation issues for event extraction, including inconsistent comparisons, insufficiency, and low reproducibility. To address these challenges, we propose TEXTEE, a consistent, sufficient, and reproducible benchmark for event extraction. We also study and benchmark the capability of five large language models in event extraction. Additionally, we discuss the role of event extraction in the current NLP era, as well as challenges and insights derived from TEXTEE. We expect TEXTEE and our reevaluation results will serve as a reliable benchmark for research in event extraction.

Limitations

In this work, we make efforts to incorporate as many event extraction datasets as possible. However, for some datasets, it is hard for us to obtain the raw files. Moreover, there is a possibility that we may overlook some datasets. Similarly, we aim to include a broad range of event extraction approaches, but we acknowledge that it is not feasible to cover all works in the field. We do our best to consider representative methods that published in recent years. Additionally, for works without released codebases, we make efforts to reimplement their proposed methods based on the descriptions in the original papers. There can be discrepancies between our implementation and theirs due to differences in packages and undisclosed techniques. We will continue to maintain our proposed library and welcome contributions and updates from the community.

References

Shumin Deng, Ningyu Zhang, Jiaojian Kang, Yichi Zhang, Wei Zhang, and Huajun Chen. 2020. Meta-learning with dynamic-memory-based prototypical network for few-shot event detection. In *The Thirteenth ACM International Conference on Web Search and Data Mining (WSDM)*.

Shumin Deng, Ningyu Zhang, Luoqiu Li, Hui Chen, Huaixiao Tou, Mosha Chen, Fei Huang, and Huajun Chen. 2021. Ontoed: Low-resource event detection with ontology embedding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL/IJCNLP)*.

George R. Doddington, Alexis Mitchell, Mark A. Przybocki, Lance A. Ramshaw, Stephanie M. Strassel, and Ralph M. 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)*.

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)*.

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 (ACL)*.

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

Rujun Han, I-Hung Hsu, Jiao Sun, Julia Baylon, Qiang Ning, Dan Roth, and Nanyun Peng. 2021. ESTER: A machine reading comprehension dataset for reasoning about event semantic relations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Yuxin He, Jingyue Hu, and Buzhou Tang. 2023. Revisiting event argument extraction: Can EAE models learn better when being aware of event co-occurrences? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*.

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 (NAACL-HLT)*.

I-Hung Hsu, Kuan-Hao Huang, Shuning Zhang, Wenxin Cheng, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2023a. TAGPRIME: A unified framework for relational structure extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*.

I-Hung Hsu, Zhiyu Xie, Kuan-Hao Huang, Prem Natarajan, and Nanyun Peng. 2023b. AMPERE: amr-aware prefix for generation-based event argument extraction model. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*.

James Y. Huang, Bangzheng Li, Jiashu Xu, and Muhao Chen. 2022a. Unified semantic typing with meaningful label inference. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*.

Kuan-Hao Huang, I-Hung Hsu, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2022b. Multilingual generative language models for zero-shot cross-lingual event argument extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Lifu Huang, Heng Ji, Kyunghyun Cho, Ido Dagan, Sebastian Riedel, and Clare R. Voss. 2018. Zero-shot transfer learning for event extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Quzhe Huang, Yanxi Zhang, and Dongyan Zhao. 2023. From simple to complex: A progressive framework for document-level informative argument extraction. In *Findings of the Association for Computational Linguistics: EMNLP*.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *arXiv preprint: arXiv:2401.04088*.

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*.

Jin-Dong Kim, Yue Wang, and Yasunori Yamamoto. 2013. The genia event extraction shared task, 2013 edition - overview. In *Proceedings of the BioNLP Shared Task 2013 Workshop*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP)*.

Viet Dac Lai, Minh Van Nguyen, Heidi Kaufman, and Thien Huu Nguyen. 2021. Event extraction from historical texts: A new dataset for black rebellions. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP*.

Viet Dac Lai, Tuan Ngo Nguyen, and Thien Huu Nguyen. 2020. Event detection: Gate diversity and syntactic importance scores for graph convolution neural networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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 (ACL)*.

Bo Li, Gexiang Fang, Yang Yang, Quansen Wang, Wei Ye, Wen Zhao, and Shikun Zhang. 2023a. Evaluating chatgpt's information extraction capabilities: An assessment of performance, explainability, calibration, and faithfulness. *arXiv preprint arXiv:2304.11633*.

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*.

Manling Li, Ruochen Xu, Shuohang Wang, Luowei Zhou, Xudong Lin, Chenguang Zhu, Michael Zeng, Heng Ji, and Shih-Fu Chang. 2022. Clip-event: Connecting text and images with event structures. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR))*.

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 (ACL)*.

Rui Li, Wenlin Zhao, Cheng Yang, and Sen Su. 2021a. Treasures outside contexts: Improving event detection via global statistics. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Sha Li, Heng Ji, and Jiawei Han. 2021b. 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 (NAACL-HLT)*.

Sha Li, Qiusi Zhan, Kathryn Conger, Martha Palmer, Heng Ji, and Jiawei Han. 2023b. GLEN: general-purpose event detection for thousands of types. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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 (ACL)*.

Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. 2020. Event extraction as machine reading comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Jian Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2018. Event detection via gated multilingual attention mechanism. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*.

Jian Liu, Yufeng Chen, and Jinan Xu. 2022a. Saliency as evidence: Event detection with trigger saliency attribution. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Xiao Liu, Heyan Huang, Ge Shi, and Bo Wang. 2022b. Dynamic prefix-tuning for generative template-based event extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*.

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

Di Lu, Shihao Ran, Joel R. Tetreault, and Alejandro Jaimes. 2023. Event extraction as question generation and answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*.

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 (ACL/IJCNLP)*.

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 (ACL)*.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

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 (ACL)*.

Yubo Ma, Zehao Wang, Yixin Cao, and Aixin Sun. 2023. Few-shot event detection: An empirical study and a unified view. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*.

Chien Van Nguyen, Hieu Man, and Thien Huu Nguyen. 2023. Contextualized soft prompts for extraction of event arguments. In *Findings of the Association for Computational Linguistics: ACL 2023*.

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 (NAACL-HLT)*.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cícero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *9th International Conference on Learning Representations (ICLR)*.

Tanmay Parekh, I-Hung Hsu, Kuan-Hao Huang, Kai-Wei Chang, and Nanyun Peng. 2023a. Contextual label projection for cross-lingual structure extraction. *arXiv preprint arXiv:2309.08943*.

Tanmay Parekh, I-Hung Hsu, Kuan-Hao Huang, Kai-Wei Chang, and Nanyun Peng. 2023b. 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 (ACL)*.

Hao Peng, Xiaozhi Wang, Feng Yao, Zimu Wang, Chuzhao Zhu, Kaisheng Zeng, Lei Hou, and Juanzi Li. 2023a. Omnievent: A comprehensive, fair, and easy-to-use toolkit for event understanding. *arXiv preprint arXiv:2309.14258*.

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

Yang Ping, Junyu Lu, Ruyi Gan, Junjie Wang, Yuxiang Zhang, Pingjian Zhang, and Jiaxing Zhang. 2023. Uniex: An effective and efficient framework for unified information extraction via a span-extractive perspective. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*.

Sampo Pyysalo, Tomoko Ohta, Makoto Miwa, Han-Cheol Cho, Junichi Tsujii, and Sophia Ananiadou. 2012. Event extraction across multiple levels of biological organization. *Bioinformatics*, 28(18):575–581.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is chatgpt a general-purpose natural language processing task solver? *arXiv preprint arXiv:2302.06476*.

Oscar Sainz, Itziar Gonzalez-Dios, Oier Lopez de Lacalle, Bonan Min, and Eneko Agirre. 2022a. Textual entailment for event argument extraction: Zero- and few-shot with multi-source learning. In *Findings of the Association for Computational Linguistics: (NAACL)*.

Oscar Sainz, Itziar Gonzalez-Dios, Oier Lopez de Lacalle, Bonan Min, and Eneko Agirre. 2022b. Textual entailment for event argument extraction: Zero- and few-shot with multi-source learning. In *Findings of the Association for Computational Linguistics (NAACL)*.

Taneeya Satyapanich, Francis Ferraro, and Tim Finin. 2020. CASIE: extracting cybersecurity event information from text. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*.

Zhiyi Song, Ann Bies, Stephanie M. 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, EVENTS@HLP-NAACL*.

Zhaoyue Sun, Jiazheng Li, Gabriele Pergola, Byron C. Wallace, Bino John, Nigel Greene, Joseph Kim, and Yulan He. 2022. PHEE: A dataset for pharmacovigilance event extraction from text. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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 (NAACL)*.

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, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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)*.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2023. Zephyr: Direct distillation of LM alignment. *arXiv preprint: arXiv:2310.16944*.

Amir Pouran Ben Veyseh, Javid Ebrahimi, Franck Dernoncourt, and Thien Nguyen. 2022a. MEE: A novel multilingual event extraction dataset. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Amir Pouran Ben Veyseh, Minh Van Nguyen, Franck Dernoncourt, Bonan Min, and Thien Huu Nguyen. 2022b. Document-level event argument extraction via optimal transport. In *Findings of the Association for Computational Linguistics: ACL 2022*.

Amir Pouran Ben Veyseh, Minh Van Nguyen, Nghia Trung Ngo, Bonan Min, and Thien Huu Nguyen. 2021. Modeling document-level context for event detection via important context selection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Sijia Wang, Mo Yu, Shiyu Chang, Lichao Sun, and Lifu Huang. 2022. Query and extract: Refining event extraction as type-oriented binary decoding. In *Findings of the Association for Computational Linguistics: ACL 2022*.

Sijia Wang, Mo Yu, and Lifu Huang. 2023a. The art of prompting: Event detection based on type specific prompts. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*.

Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, Jihua Kang, Jingsheng Yang, Siyuan Li, and Chunsai Du. 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)*.

Xingyao Wang, Sha Li, and Heng Ji. 2023c. Code4struct: Code generation for few-shot event structure prediction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*.

Ziqi Wang, Xiaozhi Wang, Xu Han, Yankai Lin, Lei Hou, Zhiyuan Liu, Peng Li, Juanzi Li, and Jie Zhou. 2021. CLEVE: contrastive pre-training for 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 (ACL/IJCNLP).*

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.

Kaiwen Wei, Xian Sun, Zequn Zhang, Jingyuan Zhang, Zhi Guo, 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 (ACL).*

Runxin Xu, Peiyi Wang, Tianyu Liu, Shuang Zeng, Baobao Chang, and Zhifang Sui. 2022. A two-stream amr-enhanced model for document-level event argument extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL).*

Xianjun Yang, Yujie Lu, and Linda R. Petzold. 2023. Few-shot document-level event argument extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL).*

Qi Zeng, Qiusi Zhan, and Heng Ji. 2022. Ea^2e: Improving consistency with event awareness for document-level argument extraction. In *Findings of the Association for Computational Linguistics: (NAACL).*

Hongming Zhang, Xin Liu, Haojie Pan, Yangqiu Song, and Cane Wing-Ki Leung. 2020. ASER: A large-scale eventuality knowledge graph. In *The Web Conference 2020 (WWW).*

Hongming Zhang, Wenlin Yao, and Dong Yu. 2022. Efficient zero-shot event extraction with context-definition alignment. In *Findings of the Association for Computational Linguistics (EMNLP).*

Zixuan Zhang and Heng Ji. 2021. Abstract meaning representation guided graph encoding and decoding for joint information extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, (NAACL-HLT).*

Kailin Zhao, Xiaolong Jin, Long Bai, Jiafeng Guo, and Xueqi Cheng. 2022. Knowledge-enhanced self-supervised prototypical network for few-shot event detection. In *Findings of the Association for Computational Linguistics: (EMNLP).*

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685.*

Shun Zheng, Wei Cao, Wei Xu, and Jiang Bian. 2021. Revisiting the evaluation of end-to-end event extraction. In *Findings of the Association for Computational Linguistics: (ACL).*

A Details of Dataset Preprocessing

We describe the detailed preprocessing steps for each dataset in the following.

ACE05-en (Doddington et al., 2004). We download the ACE05 dataset from LDC[3] and consider the data in English. The original text in ACE05 dataset is document-based. We follow most prior usage of the dataset (Lin et al., 2020; Wadden et al., 2019) to split each document into sentences and making it a sentence-level benchmark on event extraction. We use Stanza (Qi et al., 2020) to perform sentence splitting and discard any label (entity mention, relation mention, event arguments, etc.) where its span is not within a single sentence. Similar to prior works (Lin et al., 2020; Wadden et al., 2019), we consider using head span to represent entity mentions and only include event arguments that are entities (i.e., remove time and values in the ACE05 annotation). The original annotation of the dataset is character-level. However, to make the dataset consistent with others, we perform tokenization through Stanza and map the character-level annotation into token-level. We merge the original train, dev, and test sets, and split them into the new train, dev, and test sets based on documents with the ratio 80%, 10%, and 10%.

RichERE (Song et al., 2015). Considering the unavailability of the RichERE dataset used in prior works (Lin et al., 2020; Hsu et al., 2022), we download the latest RichERE dataset from LDC[4] and only consider the 288 documents labeled with Rich ERE annotations. Similar to the pre-processing step in ACE05-en, we use Stanza (Qi et al., 2020) to perform sentence splitting and making it a sentence-level benchmark. Following the strategy in (Lin et al., 2020), we use head span to represent entity mentions and only consider named entities, weapons and vehicles as event argument candidates. Again, the original annotation of the dataset is character-level, and we perform tokenization through Stanza and map the annotation into token-level, forming the final RichERE dataset we use. We merge the original train, dev, and test sets, and split them into the new train, dev, and test sets based on documents with the ratio 80%, 10%, and 10%.

[3] https://catalog.ldc.upenn.edu/LDC2006T06

[4] https://catalog.ldc.upenn.edu/LDC2023T04

MLEE (Pyysalo et al., 2012). The original MLEE dataset is document-level.[5] We use Stanza (Qi et al., 2020) to do the sentence tokenization and the word tokenization. For the purpose of evaluating most baselines, we divide the documents into several segment-level instances with a sub-token window size being 480 based on the RoBERTa-large tokenizer (Liu et al., 2019). We split the train, dev, and test sets based on documents with the ratio 70%, 15%, and 15%.

Genia2011 (Kim et al., 2011). The original Genia2011 dataset is document-level.[6] We use Stanza (Qi et al., 2020) to do the sentence tokenization and the word tokenization. For the purpose of evaluating most baselines, we divide the documents into several segment-level instances with a sub-token window size being 480 based on the RoBERTa-large tokenizer (Liu et al., 2019). We split the train, dev, and test sets based on documents with the ratio 60%, 20%, and 20%.

Genia2013 (Kim et al., 2013). The original Genia2013 dataset is document-level.[7] We use Stanza (Qi et al., 2020) to do the sentence tokenization and the word tokenization. For the purpose of evaluating most baselines, we divide the documents into several segment-level instances with a sub-token window size being 480 based on the RoBERTa-large tokenizer (Liu et al., 2019). We split the train, dev, and test sets based on documents with the ratio 60%, 20%, and 20%.

M^2E^2 (Li et al., 2020b). The M^2E^2 dataset contains event argument annotations from both texts and images.[8] We consider only the text annotations in our benchmark. We directly use the tokenized words without any modifications. We merge the original train, dev, and test sets, and split them into the new train, dev, and test sets based on documents with the ratio 70%, 15%, and 15%.

CASIE (Satyapanich et al., 2020). The original CASIE dataset is document-level.[9] We use Stanza (Qi et al., 2020) to do the sentence tokenization and the word tokenization. For the purpose of evaluating most baselines, we divide the documents into

[5] https://www.nactem.ac.uk/MLEE/

[6] https://bionlp-st.dbcls.jp/GE/2011/downloads/

[7] https://2013.bionlp-st.org/tasks/

[8] https://blender.cs.illinois.edu/software/m2e2

[9] https://github.com/Ebiquity/CASIE

several segment-level instances with a sub-token window size being 480 based on the RoBERTa-large tokenizer (Liu et al., 2019). We split the train, dev, and test sets based on documents with the ratio 70%, 15%, and 15%.

PHEE (Sun et al., 2022). We download the PHEE dataset from the official webpage.[10] We directly use the tokenized words without any modifications. We merge the original train, dev, and test sets, and split them into the new train, dev, and test sets based on documents with the ratio 60%, 20%, and 20%.

MAVEN (Wang et al., 2020). We consider the sentence-level annotations from the original data.[11] We directly use the tokenized words without any modifications. Because the labels of the original test set are not publicly accessible, we merge the original train and dev sets and split it into new train, dev, and test sets by documents with the ratio 70%, 15%, and 15%.

MEE-en (Veyseh et al., 2022a). We download the MEE dataset[12] and consider the English annotations. We use the annotations for event detection only because we observe that the quality of the annotations for event argument extraction is not good and many important arguments are actually missing. We directly use the tokenized words without any modifications. We merge the original train, dev, and test sets, and split them into the new train, dev, and test sets based on documents with the ratio 80%, 10%, and 10%.

FewEvent (Deng et al., 2020). We download the FewEvent dataset from the official webpage.[13] Notice that we consider FewEvent as a normal supervised event detection dataset. We use Stanza (Qi et al., 2020) to do the word tokenization. For the purpose of evaluating most baselines, we discard the instances with the length longer than 300. We split the train, dev, and test sets based on documents with the ratio 60%, 20%, and 20%.

RAMS (Ebner et al., 2020). We use the latest version of the RAMS dataset.[14] We directly use the tokenized words without any modifications. For the purpose of evaluating most baselines, we discard the instances with the sub-token length larger than 500 based on the RoBERTa-large tokenizer (Liu et al., 2019). We merge the original train, dev, and test sets, and split them into the new train, dev, and test sets based on documents with the ratio 80%, 10%, and 10%.

WikiEvents (Li et al., 2021b). We download the WikiEvents dataset from the official webpage.[15] We directly use the tokenized words without any modifications. For the purpose of evaluating most baselines, we divide the documents into several segment-level instances with a sub-token window size being 480 based on the RoBERTa-large tokenizer (Liu et al., 2019). We split the train, dev, and test sets based on documents with the ratio 80%, 10%, and 10%.

GENEVA (Parekh et al., 2023b). We download the GENEVA dataset from the officail webpage.[16] We directly use the tokenized words without any modifications. We split the train, dev, and test sets based on documents with the ratio 70%, 15%, and 15%.

B Details of Model Implementations

We utilize RoBERTa-large (Liu et al., 2019) for all the classification-based models and use BART-large (Lewis et al., 2020) for all the generation-based models to have a consistent comparison.

DyGIE++ (Wadden et al., 2019). We re-implement the model based on the original codebase.[17]

OneIE (Lin et al., 2020). We adapt the code from the original codebase.[18]

AMR-IE (Zhang and Ji, 2021). We adapt the code from the original codebase.[19]

EEQA (Du and Cardie, 2020). We re-implement the model based on the original codebase.[20] Notice that EEQA requires some human-written queries for making predictions. For

[10] https://github.com/ZhaoyueSun/PHEE
[11] https://github.com/THU-KEG/MAVEN-dataset
[12] http://nlp.uoregon.edu/download/MEE/MEE.zip
[13] https://github.com/231sm/Low_Resource_KBP
[14] https://nlp.jhu.edu/rams/RAMS_1.0c.tar.gz
[15] s3://gen-arg-data/wikievents/
[16] https://github.com/PlusLabNLP/GENEVA
[17] https://github.com/dwadden/dygiepp
[18] https://blender.cs.illinois.edu/software/oneie/
[19] https://github.com/zhangzx-uiuc/AMR-IE
[20] https://github.com/xinyadu/eeqa

Dataset	Task	Split	Train						Dev						Test					
			#Docs	#Inst	#ET	#Evt	#RT	#Arg	#Docs	#Inst	#ET	#Evt	#RT	#Arg	#Docs	#Inst	#ET	#Evt	#RT	#Arg
		1	481	16531	33	4309	22	6503	59	1870	30	476	22	766	59	2519	30	563	22	828
	E2E	2	481	17423	33	4348	22	6544	59	1880	29	555	22	894	59	1617	30	445	22	659
ACE05-en	ED	3	481	17285	33	4331	22	6484	59	2123	30	515	22	799	59	1512	30	502	22	814
	EAE	4	481	16842	33	4437	22	6711	59	1979	30	460	22	728	59	2099	29	451	22	658
		5	481	16355	33	4198	22	6392	59	1933	30	509	22	772	59	2632	31	641	22	933
		1	232	9198	38	4549	21	6581	28	876	35	488	21	737	28	1167	34	672	21	936
	E2E	2	232	8886	38	4444	21	6520	28	1299	36	688	21	978	28	1056	37	577	21	756
RichERE	ED	3	232	9094	38	4490	21	6517	28	1081	36	678	21	942	28	1066	35	541	21	795
	EAE	4	232	9105	38	4541	21	6647	28	973	34	571	21	804	28	1163	37	597	21	803
		5	232	9169	38	4682	21	6756	28	1135	34	487	21	692	28	937	35	540	21	806
		1	184	199	29	4705	14	4237	39	45	21	1003	9	895	39	42	21	867	12	826
	E2E	2	184	202	29	4733	14	4258	39	42	19	898	10	854	39	42	21	944	11	846
MLEE	ED	3	184	200	29	4627	14	4165	39	42	20	1029	10	944	39	44	20	919	10	849
	EAE	4	184	203	29	4629	14	4236	39	40	20	980	11	872	39	43	20	966	11	850
		5	184	201	29	4653	14	4200	39	42	21	887	11	843	39	43	20	1035	11	915
		1	576	773	9	7396	10	6495	192	348	9	3773	9	3352	192	254	9	2368	8	2018
	E2E	2	576	843	9	8455	10	7397	192	266	9	2713	9	2358	192	266	9	2369	9	2110
Genia2011	ED	3	576	901	9	8638	10	7687	192	233	9	2042	8	1743	192	241	9	2857	9	2435
	EAE	4	576	808	9	7836	10	7037	192	277	9	2842	9	2319	192	290	9	2859	9	2509
		5	576	853	9	8460	10	7464	192	240	9	2368	9	2061	192	282	9	2709	9	2340
		1	12	420	13	4077	7	3921	4	105	10	950	7	858	4	139	11	974	7	881
	E2E	2	12	388	13	3578	7	3561	4	128	11	1284	6	1134	4	148	10	1149	6	965
Genia2013	ED	3	12	381	13	3816	7	3674	4	143	10	1174	7	1079	4	140	11	1011	6	907
	EAE	4	12	441	13	3971	7	3993	4	111	9	785	7	616	4	112	11	1245	6	1051
		5	12	427	13	4225	7	4112	4	120	10	809	6	717	4	117	10	967	7	831
		1	4211	4211	8	748	15	1120	901	901	8	183	15	280	901	901	8	174	15	259
	E2E	2	4211	4211	8	794	15	1171	901	901	8	148	14	232	901	901	8	163	15	256
M^2E^2	ED	3	4211	4211	8	760	15	1138	901	901	8	160	15	252	901	901	8	185	15	269
	EAE	4	4211	4211	8	770	15	1137	901	901	8	178	15	276	901	901	8	157	15	246
		5	4211	4211	8	747	15	1122	901	901	8	164	14	258	901	901	8	194	15	279
		1	701	1047	5	5980	26	15869	149	218	5	1221	26	3175	149	218	5	1268	26	3531
	E2E	2	701	1046	5	6010	26	15986	149	223	5	1294	26	3492	149	214	5	1165	26	3097
CASIE	ED	3	701	1044	5	6009	26	16090	149	210	5	1286	26	3344	149	229	5	1174	26	3141
	EAE	4	701	1040	5	6034	26	15962	149	229	5	1172	26	3211	149	214	5	1263	26	3402
		5	701	1043	5	5831	26	15544	149	218	5	1288	26	3369	149	222	5	1350	26	3662
		1	2897	2897	2	3003	16	15482	965	965	2	1011	16	5123	965	965	2	1005	16	5155
	E2E	2	2897	2897	2	3014	16	15576	965	965	2	1002	16	5090	965	965	2	1003	16	5094
PHEE	ED	3	2897	2897	2	3009	16	15230	965	965	2	1001	16	5200	965	965	2	1009	16	5330
	EAE	4	2897	2897	2	3020	16	15496	965	965	2	996	16	5124	965	965	2	1003	16	5140
		5	2897	2897	2	3011	16	15498	965	965	2	1000	16	5049	965	965	2	1008	16	5213
		1	2537	28734	168	69069	–	–	543	5814	167	13638	–	–	543	5925	168	14190	–	–
		2	2537	28341	168	68162	–	–	543	5982	167	14233	–	–	543	6150	168	14502	–	–
MAVEN	ED	3	2537	28348	168	67832	–	–	543	6049	167	14185	–	–	543	6076	168	14880	–	–
		4	2537	28172	168	67450	–	–	543	6190	167	14637	–	–	543	6111	167	14810	–	–
		5	2537	28261	168	67826	–	–	543	6190	167	14493	–	–	543	6022	168	14578	–	–
		1	10400	10400	16	13748	–	–	1300	1300	16	1764	–	–	1300	1300	16	1745	–	–
		2	10400	10400	16	13801	–	–	1300	1300	16	1731	–	–	1300	1300	16	1725	–	–
MEE-en	ED	3	10400	10400	16	13847	–	–	1300	1300	16	1722	–	–	1300	1300	16	1688	–	–
		4	10400	10400	16	13855	–	–	1300	1300	16	1701	–	–	1300	1300	16	1701	–	–
		5	10400	10400	16	13802	–	–	1300	1300	16	1734	–	–	1300	1300	16	1721	–	–
		1	7579	7579	100	7579	–	–	2513	2513	98	2513	–	–	2541	2541	99	2541	–	–
		2	7579	7579	100	7579	–	–	2513	2513	98	2513	–	–	2541	2541	99	2541	–	–
FewEvent	ED	3	7579	7579	100	7579	–	–	2513	2513	98	2513	–	–	2541	2541	99	2541	–	–
		4	7579	7579	100	7579	–	–	2513	2513	98	2513	–	–	2541	2541	99	2541	–	–
		5	7579	7579	100	7579	–	–	2513	2513	98	2513	–	–	2541	2541	99	2541	–	–
		1	7827	7827	139	7287	65	16951	910	910	136	910	64	2132	910	910	135	910	63	2123
		2	7827	7827	139	7287	65	16946	910	910	135	910	65	2113	910	910	137	910	65	2147
RAMS	EAE	3	7827	7827	139	7287	65	16937	910	910	135	910	64	2168	910	910	135	910	64	2101
		4	7827	7827	139	7287	65	17014	910	910	136	910	62	2093	910	910	137	910	63	2099
		5	7827	7827	139	7287	65	17003	910	910	135	910	63	2130	910	910	137	910	65	2073
		1	197	450	50	3131	57	4393	24	53	39	422	43	592	24	62	38	379	46	516
		2	197	439	50	2990	57	4234	24	57	39	405	42	571	24	69	37	537	38	696
WikiEvents	EAE	3	197	435	50	3014	56	4228	24	78	36	471	43	623	24	52	37	447	47	650
		4	197	454	50	3143	57	4391	24	46	36	431	43	606	24	65	40	358	47	504
		5	197	441	50	3142	57	4370	24	57	38	394	43	562	24	67	40	396	45	569
		1	96	2582	115	5290	220	8618	82	509	115	1016	159	1683	84	593	115	1199	171	2013
		2	97	2583	115	5268	220	8660	85	509	114	1014	158	1615	85	592	115	1223	164	1994
GENEVA	EAE	3	97	2582	115	5294	220	8638	85	509	115	1010	156	1642	81	593	115	1201	170	1989
		4	96	2582	115	5293	220	8705	79	509	115	1003	164	1636	88	593	115	1209	166	1928
		5	97	2582	115	5337	220	8673	88	509	115	1004	161	1680	86	593	115	1164	161	1916

Table 8: Detailed statistics of each data split for each dataset. *#Docs, #Inst, #ET, #EvT, #RT, and #Arg represent the number of documents, instances, event types, events, roles, and arguments, respectively.*

those datasets that EEQA provides queries, we directly use those queries. For other datasets, we follow the suggestion from the paper and use "arg" style queries like "{*role_name*} in [Trigger]".

RCEE (Liu et al., 2020). We re-implement the model based on the description in the original paper. Notice that RCEE requires a question generator to generate queries for making predictions. Alternatively, we re-use the queries from EEQA as the generated queries.

Query&Extract (Wang et al., 2022). We adapt the code from the original codebase.[21] We use the event type names as the verbalized string for each event. Since the origin model supports event argument role labeling rather than event argument extraction, we learn an additional NER sequential labeling model during training and use the predicted entities for event argument role labeling during testing.

TagPrime (Hsu et al., 2023a). We adapt the code from the original codebase.[22]

PAIE (Ma et al., 2022). We adapt the code from the original codebase.[23] Notice that PAIE requires some human-written templates for making predictions. For those datasets that PAIE provides templates, we directly use them. For other datasets, we create automated templates like "{*role_1_name*} [argument_1] {*role_2_name*} [argument_2] ... {*role_k_name*} [argument_k] ".

DEGREE (Hsu et al., 2022). We adapt the code from the original codebase.[24] Notice that DEGREE requires some human-written templates for making predictions. For those datasets that DEGREE provides templates, we directly use them. For other datasets, we re-use the templates generated by PAIE.

BART-Gen (Li et al., 2021b). We re-implement the model from the original codebase.[25] We replace the original pure copy mechanism with a copy-generator since we observe this works better. Notice that BART-Gen requires some human-written templates for making predictions. For those datasets that BART-Gen provides templates, we directly use them. For other datasets, we re-use the templates generated by PAIE.

X-Gear (Huang et al., 2022b). We adapt the code from the original codebase.[26]

AMPERE (Hsu et al., 2023b). We adapt the code from the original codebase.[27] Notice that AMPERE requires some human-written templates for making predictions. For those datasets that AMPERE provides templates, we directly use them. For other datasets, we re-use the templates generated by PAIE.

UniST (Huang et al., 2022a). We re-implement the model from the original codebase.[28] Since the origin model supports semantic typing only, we learn an additional span recognition model during training and use the predicted trigger spans for trigger span typing during testing.

CEDAR (Li et al., 2023b). We re-implement the model from the original codebase.[29] Notice that in the original paper, they consider *self-labeling* during training as the dataset they consider is noisy. Our implementation currently ignore the *self-labeling* part.

C Detailed Results

Table 9, 10, 11 demonstrate the detailed reevaluation results for end-to-end event extraction, event detection, and event argument extraction, respectively.

D Prompts for LLMs

Table 12 illustrates the prompts we use for testing the ability of LLMs in event detection and event argument extraction.

[21] https://github.com/VT-NLP/Event_Query_Extract

[22] https://github.com/PlusLabNLP/TagPrime

[23] https://github.com/mayubo2333/PAIE

[24] https://github.com/PlusLabNLP/DEGREE

[25] https://github.com/raspberryice/gen-arg

[26] https://github.com/PlusLabNLP/X-Gear

[27] https://github.com/PlusLabNLP/AMPERE

[28] https://github.com/luka-group/unist

[29] https://github.com/ZQS1943/GLEN

Model	ACE05						RichERE						MLEE					
	TI	TC	AI	AC	AI+	AC+	TI	TC	AI	AC	AI+	AC+	TI	TC	AI	AC	AI+	AC+
DyGIE++	74.7	71.3	59.1	56.0	54.5	51.8	69.7	59.8	47.1	42.0	43.1	38.3	82.6	78.2	60.4	57.8	56.6	54.4
OneIE	75.0	71.1	62.4	59.9	56.9	54.7	71.0	62.5	53.9	50.0	48.4	45.2	82.7	78.5	28.7	26.9	13.6	13.1
AMR-IE	74.6	71.1	63.1	60.6	56.9	54.6	70.5	62.3	53.7	49.5	48.1	44.7	82.4	78.2	21.3	15.2	6.0	4.7
EEQA	73.8	70.0	57.0	55.3	51.9	50.4	69.3	60.2	49.2	45.8	44.7	41.9	81.4	76.9	52.9	51.1	39.0	38.1
RCEE	74.0	70.5	57.2	55.5	52.5	51.0	68.6	60.0	49.8	46.2	45.1	42.1	81.3	77.2	52.0	49.3	36.9	35.4
Query&Extract	68.6	65.1	57.4	55.0	51.2	49.0	67.5	59.8	52.3	48.9	47.5	44.5	–	–	–	–	–	–
TagPrime	73.2	69.9	61.6	59.8	56.1	54.6	69.6	63.5	56.0	52.8	51.1	48.4	81.8	79.0	66.6	65.2	61.4	60.3
DEGREE-E2E	70.3	66.8	57.6	55.1	51.3	49.1	67.7	60.5	52.2	48.7	46.6	43.7	74.7	70.2	38.6	33.8	25.9	23.3
DEGREE-PIPE	72.0	68.4	58.6	56.3	52.9	50.7	68.3	61.7	52.5	48.9	47.8	44.8	74.0	70.4	50.9	49.6	43.6	42.7

Model	Genia2011						Genia2013						M^2E^2					
	TI	TC	AI	AC	AI+	AC+	TI	TC	AI	AC	AI+	AC+	TI	TC	AI	AC	AI+	AC+
DyGIE++	74.2	70.3	58.9	56.9	53.7	52.1	76.3	72.9	62.7	60.5	58.8	57.2	53.1	51.0	34.6	33.4	31.7	30.8
OneIE	76.1	72.1	59.0	57.0	34.2	33.6	78.0	74.3	52.3	51.0	33.7	32.9	52.4	50.6	37.8	36.1	33.4	32.1
AMR-IE	76.4	72.4	44.1	42.8	29.8	29.0	78.0	74.5	35.4	34.8	23.3	23.1	52.4	50.5	37.1	35.5	33.1	31.9
EEQA	74.4	71.3	52.6	50.6	39.5	38.4	72.4	69.4	50.7	48.1	37.6	35.7	53.6	51.0	33.7	32.6	31.1	30.2
RCEE	73.3	70.1	50.9	49.0	38.2	37.2	71.4	68.0	48.0	45.8	33.0	31.6	50.1	48.1	32.0	31.0	28.8	28.0
Query&Extract	–	–	–	–	–	–	–	–	–	–	–	–	51.4	49.4	35.5	33.9	30.2	28.8
TagPrime	74.9	72.2	64.1	62.8	58.8	57.8	75.7	73.0	61.8	60.8	58.2	57.4	52.2	50.2	36.5	35.5	33.2	32.4
DEGREE-E2E	61.6	59.2	40.0	35.6	27.7	25.4	66.4	62.6	37.1	33.3	27.0	24.8	50.9	49.5	33.7	32.5	30.9	30.0
DEGREE-PIPE	63.7	60.5	51.1	49.3	40.8	39.8	64.9	61.0	51.0	49.4	43.0	41.9	50.4	48.3	34.0	33.1	30.9	30.1

Model	CASIE						PHEE						–
	TI	TC	AI	AC	AI+	AC+	TI	TC	AI	AC	AI+	AC+	–
DyGIE++	44.9	44.7	37.5	36.4	30.4	29.5	71.4	70.4	69.9	60.8	52.4	45.7	
OneIE	70.8	70.6	57.2	54.2	23.1	22.1	70.9	70.0	51.5	37.5	40.1	29.8	
AMR-IE	71.1	70.8	34.5	10.7	10.0	3.1	70.2	69.4	57.1	45.7	42.2	34.1	
EEQA	43.2	42.8	36.2	35.1	27.0	26.2	70.9	70.3	48.5	40.4	38.1	32.0	
RCEE	42.3	42.1	34.1	32.8	24.6	23.7	71.6	70.9	49.1	41.6	38.7	33.1	
Query&Extract	–	–	–	–	–	–	66.2	55.5	48.1	41.4	36.7	31.8	
TagPrime	69.5	69.3	63.3	61.0	50.9	49.1	71.7	71.1	60.9	51.7	47.4	40.6	
DEGREE-E2E	60.9	60.7	36.0	27.0	18.5	14.6	70.0	69.1	57.5	49.3	42.4	36.5	
DEGREE-PIPE	57.4	57.1	49.7	48.0	34.8	33.7	69.8	69.1	59.0	50.2	42.8	36.7	

Table 9: Reevaluation results for end-to-end event extraction (E2E). All the numbers are the average score of 5 data splits. Darker cells imply higher scores. We use “–” to denote the cases that models are not runnable.

Model	ACE05		RichERE		MLEE		Genia2011		Genia2013		M^2E^2	
	TI	TC	TI	TC	TI	TC	TI	TC	TI	TC	TI	TC
DyGIE++	74.7	71.3	69.7	59.8	82.6	78.2	74.2	70.3	76.3	72.9	53.1	51.0
OneIE	75.0	71.1	71.0	62.5	82.7	78.5	76.1	72.1	78.0	74.3	52.4	50.6
AMR-IE	74.6	71.1	70.5	62.3	82.4	78.2	76.4	72.4	78.0	74.5	52.4	50.5
EEQA	73.8	70.0	69.3	60.2	82.0	77.4	73.3	69.6	74.7	71.1	53.6	51.0
RCEE	74.0	70.5	68.6	60.0	82.0	77.3	73.1	69.3	74.6	70.8	50.1	48.1
Query&Extract	68.6	65.1	67.5	59.8	78.0	74.9	71.6	68.9	73.0	70.1	51.4	49.4
TagPrime-C	73.2	69.9	69.6	63.5	81.8	79.0	74.9	72.2	75.7	73.0	52.2	50.2
UniST	73.9	69.8	69.6	60.7	80.2	74.9	73.8	70.3	73.7	69.9	51.1	49.0
CEDAR	71.9	62.6	67.3	52.3	71.0	65.5	70.2	66.8	73.6	67.1	50.9	48.0
DEGREE	72.0	68.4	68.3	61.7	74.0	70.4	63.7	60.5	64.9	61.0	50.4	48.3

Model	CASIE		PHEE		MAVEN		FewEvent		MEE-en		–
	TI	TC	TI	TC	TI	TC	TI	TC	TI	TC	–
DyGIE++	44.9	44.7	71.4	70.4	75.9	65.3	67.7	65.2	81.7	79.8	
OneIE	70.8	70.6	70.9	70.0	76.4	65.5	67.5	65.4	80.7	78.8	
AMR-IE	71.1	70.8	70.2	69.4	–	–	67.4	65.2	–	–	
EEQA	43.4	43.2	70.9	70.3	75.2	64.4	67.0	65.1	81.4	79.5	
RCEE	43.5	43.3	71.6	70.9	75.2	64.6	67.0	65.0	81.1	79.1	
Query&Extract	51.6	51.5	66.2	55.5	–	–	66.3	63.8	80.2	78.1	
TagPrime-C	69.5	69.3	71.7	71.1	74.7	66.1	67.2	65.6	81.5	79.8	
UniST	68.4	68.1	70.7	69.6	76.7	63.4	67.5	63.1	80.5	78.3	
CEDAR	68.7	67.6	71.2	70.3	76.5	54.5	66.9	52.1	81.5	78.6	
DEGREE	61.5	61.3	69.8	69.1	76.2	65.5	67.9	65.5	80.2	78.2	

Table 10: Reevaluation results for event detection (ED). All the numbers are the average score of 5 data splits. Darker cells imply higher scores. We use “–” to denote the cases that models are not runnable.

Model	ACE05				RichERE				MLEE				Genia2011			
	AI	AC	AI+	AC+	AI	AC	AI+	AC+	AI	AC	AI+	AC+	AI	AC	AI+	AC+
DyGIE++	66.9	61.5	65.2	60.0	58.5	49.4	56.2	47.3	67.9	64.8	65.2	62.4	66.1	63.7	63.0	61.0
OneIE	75.4	71.5	74.0	70.2	71.6	65.8	69.3	63.7	31.0	28.9	16.4	15.7	62.9	60.3	40.1	38.9
AMR-IE	76.2	72.6	74.5	70.9	72.8	65.8	69.6	63.0	23.2	16.6	8.0	6.1	49.1	47.6	36.1	35.3
EEQA	73.8	71.4	71.9	69.6	73.3	67.3	70.8	64.9	64.8	62.1	51.4	49.5	63.2	60.8	51.2	49.4
RCEE	73.7	71.2	71.8	69.4	72.8	67.0	70.2	64.5	61.1	58.2	47.3	45.1	62.3	59.9	51.4	49.6
Query&Extract	77.3	73.6	75.7	72.0	76.4	70.9	74.7	69.2	–	–	–	–	–	–	–	–
TagPrime-C	80.0	76.0	78.5	74.5	78.8	73.3	76.7	71.4	78.9	76.6	76.5	74.5	79.6	77.4	77.7	75.8
TagPrime-CR	80.1	77.8	78.5	76.2	78.7	74.3	76.6	72.5	79.2	77.3	76.4	74.6	78.0	76.2	76.2	74.5
DEGREE	76.4	73.3	74.9	71.8	75.1	70.2	73.6	68.8	67.6	65.3	63.4	61.5	68.2	65.7	64.5	62.4
BART-Gen	76.0	72.6	74.8	71.2	74.4	68.8	73.1	67.7	73.1	69.8	71.8	68.7	73.4	70.9	71.8	69.5
X-Gear	76.1	72.4	74.4	70.8	75.0	68.7	73.4	67.2	64.8	63.3	60.7	59.4	68.4	66.2	65.0	63.1
PAIE	77.2	74.0	76.0	72.9	76.6	71.1	75.3	70.0	76.0	73.5	74.7	72.4	76.8	74.6	75.5	73.4
Ampere	75.5	72.0	73.9	70.6	73.8	69.2	72.2	67.7	69.2	67.1	64.4	62.6	69.5	67.1	66.0	63.8

Model	Genia2013				M^2E^2				CASIE				PHEE			
	AI	AC	AI+	AC+	AI	AC	AI+	AC+	AI	AC	AI+	AC+	AI	AC	AI+	AC+
DyGIE++	71.7	69.3	68.7	66.9	41.7	38.9	41.0	38.5	58.0	56.0	53.4	51.5	63.4	54.6	63.0	54.2
OneIE	57.2	55.7	39.4	38.7	59.0	55.2	57.2	53.3	58.3	55.3	29.0	27.7	55.9	40.6	55.5	40.4
AMR-IE	38.9	38.1	26.7	26.4	56.0	51.3	55.3	50.4	35.5	11.0	12.8	4.0	60.4	45.3	59.9	44.9
EEQA	64.7	61.1	50.3	47.5	57.6	55.9	57.0	55.3	56.1	54.0	50.9	49.0	53.7	45.6	53.4	45.4
RCEE	60.7	57.4	45.1	42.7	57.9	56.4	57.3	55.8	47.6	45.3	41.5	39.5	54.1	45.8	53.8	45.6
Query&Extract	–	–	–	–	59.9	56.2	58.0	54.2	–	–	–	–	64.6	54.8	64.2	54.4
TagPrime-C	79.8	77.4	77.1	74.9	63.4	60.1	62.3	59.0	71.9	69.1	68.8	66.1	66.0	55.6	65.6	55.3
TagPrime-CR	76.6	74.5	74.3	72.3	63.2	60.8	62.3	59.9	71.1	69.2	67.9	66.1	65.8	56.0	65.5	55.7
DEGREE	68.4	66.0	64.6	62.5	62.3	59.8	61.7	59.2	61.0	59.0	56.5	54.7	61.7	52.5	61.4	52.3
BART-Gen	76.4	73.6	74.8	72.2	62.5	60.0	62.1	59.6	63.7	60.0	61.8	58.3	57.1	47.7	56.9	47.5
X-Gear	64.1	61.9	60.5	58.6	62.7	59.8	61.9	59.0	65.7	63.4	61.4	59.3	67.6	58.3	67.4	58.2
PAIE	77.8	75.2	76.6	74.2	62.9	60.6	62.7	60.4	68.1	65.7	66.4	64.0	74.9	73.3	74.7	73.1
Ampere	73.2	71.0	69.6	67.7	62.1	59.1	61.4	58.4	61.1	58.4	56.4	53.9	61.4	51.7	61.1	51.6

Model	WikiEvnts				RAMS				GENEVA				–
	AI	AC	AI+	AC+	AI	AC	AI+	AC+	AI	AC	AI+	AC+	–
DyGIE++	39.8	35.3	39.0	34.7	44.3	35.3	44.3	35.3	66.0	62.5	65.8	62.3	
OneIE	17.5	15.0	9.2	7.9	48.0	40.7	48.0	40.7	38.9	37.1	38.6	36.9	
AMR-IE	17.8	16.0	11.7	10.4	49.6	42.3	49.6	42.3	23.7	16.6	23.4	16.4	
EEQA	54.3	51.7	48.4	46.1	48.9	44.7	48.9	44.7	69.7	67.3	69.4	67.0	
RCEE	53.7	50.9	46.4	44.0	45.4	41.5	45.4	41.5	66.2	63.8	65.8	63.4	
Query&Extract	–	–	–	–	–	–	–	–	52.2	50.3	51.8	50.0	
TagPrime-C	70.4	65.7	68.6	64.0	54.4	48.3	54.4	48.3	83.0	79.2	82.7	79.0	
TagPrime-CR	70.3	67.2	68.4	65.5	54.1	49.7	54.1	49.7	82.8	80.4	82.5	80.1	
DEGREE	60.4	57.3	56.8	53.9	50.5	45.5	50.5	45.5	67.2	64.1	67.0	63.9	
BART-Gen	68.5	64.2	68.1	63.9	50.4	45.4	50.4	45.4	67.3	64.4	67.2	64.3	
X-Gear	58.7	55.6	55.4	52.4	52.1	46.2	52.1	46.2	78.9	75.1	78.7	74.9	
PAIE	69.8	65.5	69.5	65.2	55.2	50.5	55.2	50.5	73.5	70.4	73.4	70.3	
Ampere	59.9	56.7	56.2	53.3	52.0	46.8	52.0	46.8	67.8	65.0	67.6	64.8	

Table 11: Reevaluation results for event argument extraction (EAE). All the numbers are the average score of 5 data splits. Darker cells imply higher scores. We use "–" to denote the cases that models are not runnable.

Prompt Used for Event Detection	
Instruction	You are an event extractor designed to check for the presence of a specific event in a sentence and to locate the corresponding event trigger. Task Description: Identify all triggers related to the event of interest in the sentence. A trigger is the key word in the sentence that most explicitly conveys the occurrence of the event. If yes, please answer 'Yes, the event trigger is [trigger] in the text.'; otherwise, answer 'No.' The event of interest is Business.Collaboration. This event is related to business collaboration.
Example 1	Examples 1 Text: It is a way of coordinating different ideas from numerous people to generate a wide variety of knowledge. Answer: Yes, the event trigger is *coordinating* in the text.
Example 2	Examples 2 Text: What's going on is that union members became outraged after learning about the airline's executive compensation plan where we would have paid huge bonuses even in bankruptcy Answer: No.
...	...
Query	Question Text: Social networks permeate business culture where collaborative uses include file sharing and knowledge transfer. Answer:
Output	Yes, the event trigger is *sharing* in the text.
Prompt Used for Event Argument Extraction	
Instruction	You are an argument extractor designed to check for the presence of arguments regarding specific roles for an event in a sentence. Task Description: Identify all arguments related to the role *Agent*, *Person*, *Place* in the sentence. These arguments should have the semantic role corresponding to the given event trigger by the word span between [t] and [/t]. Follow the the format of below examples. Your answer should only contain the answer string and nothing else. The event of interest is Justice:Arrest-Jail. The event is related to a person getting arrested or a person being sent to jail. Roles of interest: *Agent*, *Person*, *Place*
Example 1	Examples 1 Text: Currently in California , 7000 people [t] serving [/t] 25 to year life sentences under the three strikes law. Agent: Person: people Place: California
Example 2	Examples 2 Text: We've been playing warnings to people to stay in their houses , and we've only [t] lifted [/t] those people we've got very good intelligence on. Agent: we Person: people Place:
...	...
Query	Question Text: A pizza delivery helped police [t] nab [/t] the suspect in the kidnapping of a 9-year-old California girl.
Output	Agent: police Person: suspect Place:

Table 12: Prompts use for testing the ability of LLMs in event extraction.