The Devil is in the Details: On the Pitfalls of Event Extraction Evaluation

Hao Peng[1*], **Xiaozhi Wang**[1*], **Feng Yao**[2*], **Kaisheng Zeng**[1],
Lei Hou[1,3], **Juanzi Li**[1,3†], **Zhiyuan Liu**[1,3], **Weixing Shen**[2]
[1]Department of Computer Science and Technology, BNRist;
[2]School of Law, Institute for AI and Law;
[3]KIRC, Institute for Artificial Intelligence,
Tsinghua University, Beijing, 100084, China
{peng-h21, wangxz20, yaof20}@mails.tsinghua.edu.cn

Abstract

Event extraction (EE) is a crucial task aiming at extracting events from texts, which includes two subtasks: event detection (ED) and event argument extraction (EAE). In this paper, we check the reliability of EE evaluations and identify three major pitfalls: (1) The **data preprocessing discrepancy** makes the evaluation results on the same dataset not directly comparable, but the data preprocessing details are not widely noted and specified in papers. (2) The **output space discrepancy** of different model paradigms makes different-paradigm EE models lack grounds for comparison and also leads to unclear mapping issues between predictions and annotations. (3) The **absence of pipeline evaluation** of many EAE-only works makes them hard to be directly compared with EE works and may not well reflect the model performance in real-world pipeline scenarios. We demonstrate the significant influence of these pitfalls through comprehensive meta-analyses of recent papers and empirical experiments. To avoid these pitfalls, we suggest a series of remedies, including specifying data preprocessing, standardizing outputs, and providing pipeline evaluation results. To help implement these remedies, we develop a consistent evaluation framework OMNIEVENT, which can be obtained from https://github.com/THU-KEG/OmniEvent.

1 Introduction

Event extraction (EE) is a fundamental information extraction task aiming at extracting structural event knowledge from plain texts. As illustrated in Figure 1, it is typically formalized as a two-stage pipeline (Ahn, 2006). The first subtask, event detection (ED), is to detect the event triggers (keywords or phrases evoking events, e.g., *quitting* in Figure 1) and classify their event types (e.g., `End-Position`). The second subtask, event argument extraction (EAE), is to extract corresponding event arguments and their roles (e.g., *Elon Musk* and its argument role `Person`) based on the first-stage ED results.

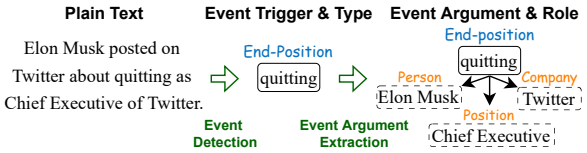

Figure 1: An illustration for the event extraction (EE) pipeline, including two stages: event detection (ED) and event argument extraction (EAE).

Since events play an important role in human language understanding and broad applications benefit from structural event knowledge (Ji and Grishman, 2011; Glavaš and Šnajder, 2014; Hogenboom et al., 2016; Zhang et al., 2020a), EE has attracted much research attention, and novel models have been continually developed. Beyond the conventional paradigms like classification (Chen et al., 2015; Wang et al., 2021) and sequence labeling (Nguyen et al., 2016; Chen et al., 2018), new model paradigms such as span prediction (Liu et al., 2020a; Du and Cardie, 2020b) and conditional generation (Lu et al., 2021; Li et al., 2021b) are proposed. These sophisticated models push evaluation results to increasingly high levels.

However, due to the complex input/output formats and task pipeline of EE, there are some hidden pitfalls in EE evaluations, which are rarely noted and discussed in EE papers (Wadden et al., 2019; Wang et al., 2020, 2022). These pitfalls make many competing EE methods actually lack grounds for comparison, and the reported scores cannot reflect real-world model performances well.

In this paper, we summarize three major pitfalls: (1) **Data preprocessing discrepancy**. If two EE works conduct evaluations on the same dataset but adopt different preprocessing methods, their results are not directly comparable. Since EE datasets have complex data formats (involving multiple heteroge-

[*] Equal contribution. Random Order.
[†] Corresponding author: J.Li

Findings of the Association for Computational Linguistics: ACL 2023, pages 9206–9227
July 9-14, 2023

neous elements including event triggers, arguments, entities, temporal expressions, etc.), data preprocessing methods of existing works often disagree on some design choices, like whether to include multi-token triggers, which results in major data discrepancies. For instance, for the widely-used English subset of ACE 2005 (Walker et al., 2006), the preprocessing of Wadden et al. (2019) gets $5,055$ event triggers, but Wang et al. (2021) have $5,349$. (2) **Output space discrepancy**. Different model paradigms have inconsistent output spaces, which makes the evaluation metrics of different-paradigm models often not calculated on the same bases. For example, the phrase *Elon Musk* is one argument candidate in the output space of conventional classification-based methods, and it is regarded as one error case when the model misclassifies it. But other model paradigms, like the sequence labeling, have more free output formats and can make two independent predictions for the two tokens *Elon* and *Musk*, which will account for two error cases in the evaluation metric calculation. Larger output spaces of the new model paradigms also result in unclear mappings between predictions and annotations in some cases, which are often overlooked in EE evaluation implementations and lead to problematic results. These details are presented in § 3.3. (3) **Absence of pipeline evaluation**. Recent works handling only the EAE subtask often evaluate the performances based on gold event triggers (Subburathinam et al., 2019; Xi et al., 2021; Ma et al., 2022). In contrast, conventional EE works often conduct pipeline evaluation, i.e., evaluate EAE performances based on triggers predicted at the ED stage. The absence of pipeline evaluation makes these EAE-only works hard to be directly compared with EE works. This has discouraged the research community from considering all the EE subareas in a holistic view. Moreover, only using gold triggers in evaluation cannot evaluate EAE models' resistance to the noise of predicted triggers, which is important in real-world application scenarios.

We conduct systematic meta-analyses of EE papers and empirical experiments, demonstrating the pitfalls' broad and significant influence. We suggest a series of remedies to avoid these pitfalls, including specifying data preprocessing methods, standardizing outputs, and providing pipeline evaluation results. To help conveniently achieve these remedies, we develop a consistent evaluation framework, OMNIEVENT, which contains implementations for data preprocessing and output standardization, and off-the-shelf predicted triggers on widely-used datasets for easier pipeline evaluation.

To summarize, our contributions are two-fold: (1) We systematically analyze the inconspicuous pitfalls of EE evaluations and demonstrate their significant influence with meta-analyses and experiments. (2) We propose corresponding remedies to avoid the pitfalls and develop a consistent evaluation framework to help implement them.

2 Related Work

Traditional methods (Ji and Grishman, 2008; Gupta and Ji, 2009; Hong et al., 2011; Li et al., 2013) rely on human-crafted features and rules to extract events. Most modern EE models automate feature learning with neural networks (Nguyen and Grishman, 2015; Nguyen et al., 2016; Nguyen and Grishman, 2018) and adopt different model paradigms to model the EE task. The most common **classification**-based methods view EE as classifying given trigger and argument candidates into different labels (Chen et al., 2015; Feng et al., 2016; Chen et al., 2017; Liu et al., 2018b; Wang et al., 2019a; Lai et al., 2020; Wang et al., 2021, 2022). **Sequence labeling** methods (Nguyen et al., 2016; Chen et al., 2018; Araki and Mitamura, 2018; Ding et al., 2019; Ma et al., 2020; Nguyen et al., 2021; Guzman-Nateras et al., 2022) do EE by labeling every word following a certain tagging schema such as BIO (Ramshaw and Marcus, 1995). Recently, some works (Du and Cardie, 2020b; Li et al., 2020a; Liu et al., 2020a, 2021b; Wei et al., 2021; Sheng et al., 2021; Zhou et al., 2022) propose to cast the task formalization of EE into resource-rich machine reading comprehension tasks and adopt the **span prediction** paradigm to predict the starting and ending positions of event trigger and argument spans. With the development of generative pre-trained language models (Lewis et al., 2020; Raffel et al., 2020; Brown et al., 2020), there have been works (Lu et al., 2021; Xi et al., 2021; Li et al., 2021b, 2022a; Liu et al., 2022c; Huang et al., 2022; Du et al., 2022; Hsu et al., 2022; Zeng et al., 2022) exploring the **conditional generation** paradigm to generate sequences indicating EE results.

A few previous works (Wadden et al., 2019; Lai et al., 2020; Wang et al., 2020, 2022) have noted that data preprocessing discrepancy may influence evaluation results, but they did not especially study its impact with in-depth analyses. To the best of

our knowledge, we are the first to study all three kinds of pitfalls of EE evaluation and propose comprehensive remedies for them.

3 Pitfalls of Event Extraction Evaluation

We first introduce our investigation setup for meta-analysis and empirical analysis (§ 3.1). Then we analyze the three pitfalls: data preprocessing discrepancy (§ 3.2), output space discrepancy (§ 3.3), and absence of pipeline evaluation (§ 3.4).

3.1 Investigation Setup

We adopt the following two investigation methods to analyze the influence of the observed pitfalls.

Meta-Analysis To comprehensively understand the research status and investigate the potential influence of the evaluation pitfalls, we analyze a broad range of recent EE studies in the meta-analysis. Specifically, we manually retrieve all published papers concerning EE, ED, and EAE tasks at four prestigious venues from 2015 to 2022 via keyword[1] matching and manual topic rechecking by the authors. The complete paper list is shown in appendix C, including 44 at ACL, 39 at EMNLP, 19 at NAACL, and 14 at COLING.

We conduct statistical analyses of these papers and their released codes (if any) from multiple perspectives. These statistics will be presented to demonstrate the existence and influence of the pitfalls in the following sections, respectively.

Empirical Analysis In addition to the meta-analysis, we conduct empirical experiments to quantitatively analyze the pitfalls' influence on EE evaluation results. We reproduce several representative models covering all four model paradigms mentioned in § 2 to systematically study the influence. Specifically, the models contain: (1) **Classification** methods, including DMCNN (Chen et al., 2015) , DMBERT (Wang et al., 2019a,b), and CLEVE (Wang et al., 2021). DMCNN and DMBERT adopt a dynamic multi-pooling operation over hidden representations of convolutional neural networks and BERT (Devlin et al., 2019), respectively. CLEVE is an event-aware pre-trained model enhanced with event-oriented contrastive pre-training. (2) **Sequence labeling** methods, including BiLSTM+CRF (Wang et al., 2020) and BERT+CRF (Wang et al., 2020), which adopt the conditional random field (Lafferty et al., 2001) as the output layer to make structural predictions. (3) **Span prediction** methods, including EEQA (Du and Cardie, 2020b) converting EE into a question-answering task, and PAIE (Ma et al., 2022), which is a prompt-tuning-based EAE method. (4) **Conditional generation** method, including Text2Event (Lu et al., 2021), which is a sequence-to-structure generative EE method with constrained decoding and curriculum learning.

	ED			EAE		
Metric	**P**	**R**	**F1**	**P**	**R**	**F1**
DMCNN	65.0	69.7	67.2	45.3	41.6	43.2
DMBERT	72.1	77.1	74.5	50.5	60.0	54.8
CLEVE	76.4	80.4	78.3	56.9	65.9	61.0
BiLSTM+CRF	72.3	79.1	75.5	27.1	32.3	29.4
BERT+CRF	69.9	74.6	72.1	41.4	43.6	42.5
EEQA	65.3	74.5	69.5	49.7	45.4	47.4
PAIE	N/A	N/A	N/A	70.6	73.2	71.8
Text2Event	66.9	72.4	69.5	48.0	54.1	50.8

Table 1: The reproduction performances (%) on ACE 2005 under respective original evaluation settings. "N/A" means not applicable as PAIE is an EAE-only model. Reproduction details are introduced in appendix A.2

The models are reproduced based on the evaluation settings described in their original papers and released open-source codes (if any). From our meta-analysis, 70% of the EE papers adopt the English subset of ACE 2005 dataset (Walker et al., 2006)[2] in their experiments. Hence we also adopt this most widely-used dataset in our empirical experiments to analyze the pitfalls without loss of generality. The reproduction performances are shown in Table 1. Following the conventional practice, we report precision (P), recall (R), and the F1 score. In the following analyses, we show the impact of three pitfalls by observing how the performances change after controlling the pitfalls' influence.

3.2 Data Preprocessing Discrepancy

Due to the inherent task complexity, EE datasets naturally involve multiple heterogeneous annotation elements. For example, besides event triggers and arguments, EE datasets often annotate entities, temporal expressions, and other spans as argument candidates. The complex data format makes the data preprocessing methods easily differ in many details, which makes the reported results on the same dataset not directly comparable. However, this pitfall has not received extensive attention.

To carefully demonstrate the differences brought by data preprocessing discrepancy, we conduct de-

[1]We use *event* and *extraction* as keywords for searching.

[2]For brevity, refer to as "ACE 2005" in the following.

	Paper%	#Token	#Trigger	#Argument	#Event Type	#Arg. Role	#Tri. Candidate	#Arg. Candidate
ACE-DYGIE	14	305,266	5,055	6,040	33	22	305,266	34,474
ACE-OneIE	19	310,020	5,311	8,055	33	22	309,709	54,650
ACE-Full	4	300,477	5,349	9,683	33	35	300,165	59,430
Unspecified	63	-	-	-	-	-	-	-

Table 2: The statistics of different ACE 2005 preprocessing scripts. "Paper%" represents the utilization rates of different scripts among surveyed papers. Unspecified includes papers (61%) that neither refer to a preprocessing method nor release their preprocessing codes and papers (2%) that release their preprocessing codes that are only used in their own papers. "Arg." and "Tri." are short for argument and trigger, respectively.

	ACE-DYGIE	ACE-OneIE	ACE-Full
NLP Toolkit	spaCy	NLTK	CoreNLP
Entity Mention	head	head	full
Multi-token Tri.	✗	✔	✔
Temporal Exp.	✗	✗	✔
Value Exp.	✗	✗	✔
Pronoun	✗	✔	✔

Table 3: The major differences between the three preprocessing scripts. NLP Toolkit: the toolkit used for sentence segmentation and tokenization. Entity Mention: using head words or full mentions as entity mentions. Multi-token Tri.: whether include multi-token triggers. Temporal Exp., Value Exp., and Pronoun: whether include temporal expressions, values, and pronouns.

	ACE-DYGIE		ACE-OneIE		ACE-Full	
Metric	ΔED F1	ΔEAE F1	ΔED F1	ΔEAE F1	ΔED F1	ΔEAE F1
DMCNN	−4.7	−9.2	−4.3	−8.0	–	–
DMBERT	−6.3	−6.7	−5.2	−7.6	–	–
CLEVE	−5.4	−6.2	−3.3	−6.3	–	–
BiLSTM+CRF	−3.8	+3.1	−4.1	+3.2	–	–
BERT+CRF	−4.2	+2.4	−4.2	+3.4	–	–
EEQA	–	–	−0.5	+0.1	+3.6	−4.1
PAIE	N/A	–	N/A	−0.7	N/A	−15.2
Text2Event	–	–	+2.5	+3.0	+4.7	−1.0

Table 4: The F1 (%) differences between using ACE 2005 preprocessed by another script and the original script. "–" indicates the model is originally trained and evaluated on this script.

tailed meta-analyses taking the most widely-used ACE 2005 as an example. From all the 116 surveyed papers, we find three repetitively used open-source preprocessing scripts: ACE-DYGIE (Wadden et al., 2019), ACE-OneIE (Lin et al., 2020), and ACE-Full (Wang et al., 2019b). In addition to these scripts, there are 6 other open-source preprocessing scripts that are only used once. The utilization rates and data statistics of the different preprocessing methods are shown in Table 2. From the statistics, we can observe that: (1) The data differences brought by preprocessing methods are significant. The differences mainly come from the different preprocessing implementation choices, as summarized in Table 3. For instance, ACE-DYGIE and ACE-OneIE ignore the annotated temporal expressions and values in ACE 2005, which results in 13 fewer argument roles compared to ACE-Full. Intuitively, the significant data discrepancy may result in inconsistent evaluation results. (2) Each preprocessing script has a certain utilization rate and the majority (63%) papers do not specify their preprocessing methods. The high preprocessing inconsistency and Unspecified rate both show that our community has not fully recognized the significance of the discrepancies resulting from differences in data preprocessing.

To further empirically investigate the influence of preprocessing, we conduct experiments on ACE 2005. Table 4 shows the F1 differences keeping all settings unchanged except for the preprocessing scripts. We can observe that the influence of different preprocessing methods is significant and varies from different models. It indicates that the evaluation results on the same dataset are not necessarily comparable due to the unexpectedly large influence of different preprocessing details.

Moreover, besides ACE 2005, there are also data preprocessing discrepancies in other datasets. For example, in addition to the implementation details, the data split of the KBP dataset is not always consistent (Li et al., 2021a, 2022a), and some used LDC[3] datasets are not freely available, such as LDC2015E29. Based on all the above analyses, we suggest the community pay more attention to data discrepancies caused by preprocessing, and we propose corresponding remedies in § 4.1.

3.3 Output Space Discrepancy

As shown in Figure 3, the diversity of adopted model paradigms in EE studies has substantially increased in recent years. Figure 2 illustrates the different paradigms' workflows in EAE scenario[4]. The paradigms inherently have very different output spaces, which results in inconspicuous pitfalls

[3] https://www.ldc.upenn.edu/

[4] The ED workflows are the same as EAE or even simpler.

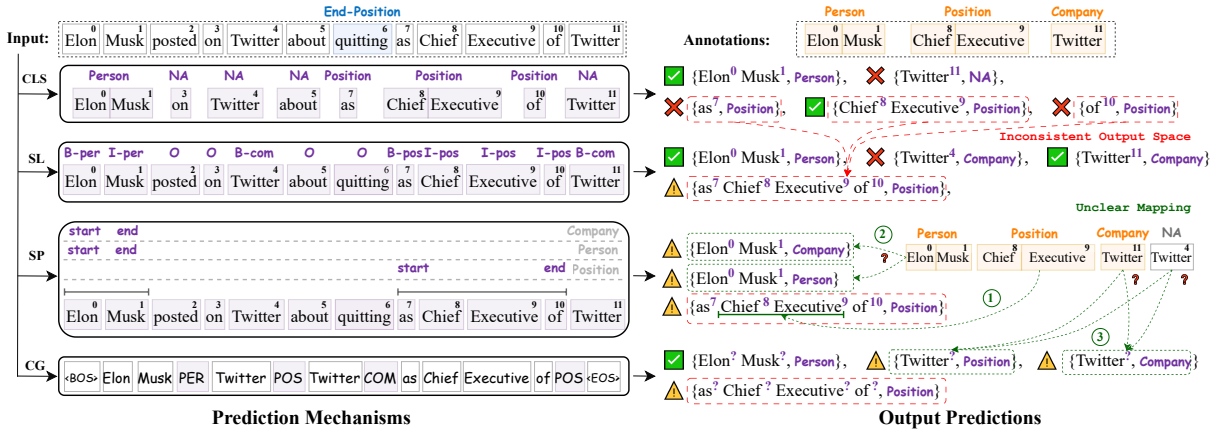

Figure 2: An illustration for the prediction mechanisms of different model paradigms and their corresponding output formats in event argument extraction task. Classification (CLS) methods do multi-class classification on the argument candidates within a pre-defined set. Sequence labeling (SL) methods predict a BIO-label for each input token. Span prediction (SP) models predict the starting and ending indices of a span for each argument role. Conditional generation (CG) models directly generate a structured sequence consisting of argument mentions and roles. Predictions marked with yellow warning signs are tricky samples for pitfalls illustration. Best viewed in color.

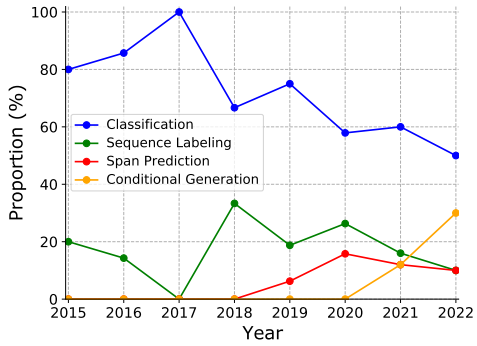

Figure 3: Proportion per year of EE papers adopting different model paradigms from 2015 to 2022.

in the comparative evaluations across paradigms.

Inconsistent Output Spaces between Different Paradigms As shown in Figure 2, there are substantial differences between the model output spaces of different paradigms. CLS-paradigm models only output a unique label for each candidate in a pre-defined set. While models of SL and SP paradigms can make predictions for any consecutive spans in the input sequence. The output space of CG-paradigm models is even larger, as their vanilla[5] output sequences are completely free, e.g., they can even involve tokens unseen in the input. The inconsistent output spaces make the evaluation metrics of different-paradigm models calculated on different bases and not directly comparable. For instance, when calculating the confusion matrices for the prediction *as Chief Executive of* in Figure 2, the CLS paradigm takes it as one true positive (TP) and two false positives (FP), while the remaining paradigms only count it as one FP. The CLS paradigm may also have an advantage in some cases since it is constrained by the pre-defined candidate sets and cannot make illegal predictions as other paradigms may have.

Unclear Mappings between Predictions and Annotations Implementing the mappings between model predictions and dataset annotations is a key component for evaluation. The larger output spaces of SL, SP, and CG paradigms often produce unclear mappings, which are easily neglected in the EE evaluation implementations and influence the final metrics. As shown in Figure 2 (bottom right), we summarize three major unclear mapping issues: ① **Prediction span overlaps the gold span.** A prediction span of non-CLS paradigm models may overlap but not strictly align with the annotated span, bringing in an unclear implementation choice. As in Figure 2, it is unclear whether the predicted role `Position` for the span *as Chief Executive of* should be regarded as a correct prediction for the contained annotated span *Chief Executive*. ② **Multiple predictions for one annotated span.** If without special constraints, models of SP and CG paradigms may make multiple predictions for one span. Figure 2 presents two contradictory predictions (`Company` and `Person`) for the annotated span *Elon Musk*. To credit the correct one only or penalize both should lead to different evaluation

[5]Indicates excluding tricks like vocabulary constraint, etc.

	ED			EAE		
Metric	**ΔP**	**ΔR**	**ΔF1**	**ΔP**	**ΔR**	**ΔF1**
BiLSTM+CRF	+2.0	−0.1	+1.0	+5.0	−0.3	+2.3
BERT+CRF	+2.8	−0.2	+1.3	+2.6	−0.3	+1.2
EEQA	−0.6	+0.8	+0.1	−3.1	−2.1	−2.6
PAIE	N/A	N/A	N/A	+4.6	−0.6	+2.0
Text2Event	+1.1	+0.0	+0.6	−1.1	−3.5	−2.3

Table 5: The precision, recall, and F1 (%) differences between evaluation with and without our output standardization. The results are evaluated on ACE-OneIE, and the results for other preprocessing methods are in appendix B. Output standardization aligns the output spaces of the other paradigms into that of the CLS paradigm, and thus we do not include the CLS-paradigm models here, whose results are unchanged.

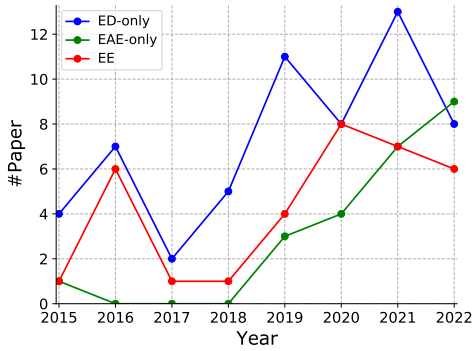

Figure 4: Number of papers concerning only ED, only EAE, and EE tasks from 2015 to 2022.

results. ③ **Predictions without positions for non-unique spans.** Vanilla CG-paradigm models make predictions by generating contents without specifying their positions. When the predicted spans are non-unique in the inputs, it is unclear how to map them to annotated spans in different positions. As in Figure 2, the CG model outputs two *Twitter* predictions, which can be mapped to two different input spans.

To quantitatively demonstrate the influence of output space discrepancy, we conduct empirical experiments. Specifically, we propose an output standardization method (details in § 4.2), which unify the output spaces of different paradigms and handle all the unclear mapping issues. We report the changes in metrics between the original evaluation implementations and the evaluation with our output standardization in Table 5. We can see the results change obviously, with the maximum increase and decrease of $+2.8$ in ED precision and -3.5 in EAE recall, respectively. It indicates the output space discrepancy can lead to highly inconsistent evaluation results. Hence, we advocate for awareness of the output space discrepancy in evaluation implementations and suggest doing output standardization when comparing models using different paradigms.

3.4 Absence of Pipeline Evaluation

The event extraction (EE) task is typically formalized as a two-stage pipeline, i.e., first event detection (ED) and then event argument extraction (EAE). In real applications, EAE is based on ED and only extracts arguments for triggers detected by the ED model. Therefore, the conventional evaluation of EAE is based on predicted triggers and considers ED prediction errors, which we call **pipeline evaluation**. It assesses the overall performance of an event extraction system and is consistent with real-world pipeline application scenarios.

However, as shown in Figure 4, more and more works have focused only on EAE in recent years. For convenience and setting a unified evaluation base between the EAE-only works, 95.45% of them only evaluate EAE taking gold triggers as inputs. We dub this evaluation setting as **gold trigger evaluation**. The conventional pipeline evaluation of EE works is absent in most EAE-only works, which poses two issues: (1) The absence of pipeline evaluation makes the results of EAE-only works hard to be directly cited and compared in EE studies. In the papers covered by our meta-analysis, there is nearly no direct comparison between EE methods and EAE-only methods. It indicates that the evaluation setting difference has created a gap between the two closely connected research tasks, which hinders the community from comprehensively understanding the research status. (2) The gold trigger evaluation may not well reflect the real-world performance since it ignores the EAE models' resistance to trigger noise. In real-world applications, the input triggers for EAE models are noisy predicted triggers. A good EAE method should be resistant to trigger noise, e.g., not extracting arguments for false positive triggers. The gold trigger evaluation neglects trigger noise.

To assess the potential influence of this pitfall, we compare experimental results under the gold trigger evaluation and pipeline evaluation of various models in Table 6. We can observe different trends from the results of gold trigger evaluation and pipeline evaluation. For example, although DMBERT performs much better than BERT+CRF under gold trigger evaluation, they perform nearly the same under pipeline evaluation (47.2 vs. 47.1). It suggests that the absence of pipeline evalua-

Metric	ED F1	Gold Tri. EAE F1	Pipeline EAE F1
DMCNN	62.8	51.6	35.2
DMBERT	69.4	67.2	47.2
CLEVE	75.0	69.6	54.7
BiLSTM+CRF	72.4	45.3	34.9
BERT+CRF	69.2	64.3	47.1
EEQA	69.1	63.9	45.0
PAIE	75.0	73.2	56.7

Table 6: EAE F1 scores (%) of gold trigger evaluation and pipeline evaluation on ACE-OneIE. Results for other preprocessing methods are in appendix B. We also report corresponding ED F1 scores to show trigger quality. PAIE adopts the triggers predicted by CLEVE. The joint model Text2Event is excluded since its trigger input cannot be controlled.

tion may bring obvious result divergence, which is rarely noticed in existing works. Based on the above discussions, we suggest also conducting the pipeline evaluation in EAE works.

4 Consistent Evaluation Framework

The above analyses show that the hidden pitfalls substantially harm the consistency and validity of EE evaluation. We propose a series of remedies to avoid these pitfalls and develop a consistent evaluation framework, OMNIEVENT. OMNIEVENT helps to achieve the remedies and eases users of handling the inconspicuous preprocessing and evaluation details. It is publicly released and continually maintained to handle emerging evaluation pitfalls. The suggested remedies include specifying data preprocessing (§ 4.1), standardizing outputs (§ 4.2), and providing pipeline evaluation results (§ 4.3). We further re-evaluate various EE models using our framework and analyze the results in § 4.4.

4.1 Specify Data Preprocessing

As analyzed in § 3.2, preprocessing discrepancies have an obvious influence on evaluation results. The research community should pay more attention to data preprocessing details and try to specify them. Specifically, we suggest future EE works adopt a consistent preprocessing method on the same dataset. Regarding the example in § 3.2, for the multiple ACE 2005 preprocessing scripts, we recommend ACE-Full since it retains the most comprehensive event annotations, e.g., multi-token triggers and the time-related argument roles, which are commonly useful in real-world applications. If a study has to use different preprocessing methods for special reasons, we suggest specifying the preprocessing method with reference to public codes. However, there are no widely-used publicly available preprocessing scripts for many EE datasets, which makes many researchers have to re-develop their own preprocessing methods. In our consistent evaluation framework, we provide preprocessing scripts for various widely-used datasets, including ACE 2005 (Walker et al., 2006), TAC KBP Event Nugget Data 2014-2016 (Ellis et al., 2014, 2015, 2016), TAC KBP 2017 (Getman et al., 2017), RichERE (Song et al., 2015), MAVEN (Wang et al., 2020), LEVEN (Yao et al., 2022), DuEE (Li et al., 2020b), and FewFC (Zhou et al., 2021). We will continually add the support of more datasets, such as RAMS (Ebner et al., 2020) and WikiEvents (Li et al., 2021b), and we welcome the community to contribute scripts for more datasets.

4.2 Standardize Outputs

Based on the discussions about output space discrepancy in § 3.3, we propose and implement an output standardization method in our framework.

To mitigate the inconsistency of output spaces between paradigms, we project the outputs of non-CLS paradigm models onto the most strict CLS-paradigm output space. Specifically, we follow strict boundary-matching rules to assign the non-CLS predictions to each trigger/argument candidate in pre-defined candidate sets of the CLS paradigm. The final evaluation metrics are computed purely on the candidate sets, and those predictions that fail to be matched are discarded. The intuition behind this operation is that given the CLS-paradigm candidate sets are automatically constructed, the illegal predictions out of this scope can also be automatically filtered in real-world applications.

Regarding the unclear mappings between predictions and annotations, we consider the scenario of real-world applications and propose several deterministic mapping rules for consistent evaluations. We respond to the issues mentioned in § 3.3 as follows. ① **Prediction span overlaps the gold span.** We follow strict boundary-matching rules and discard such overlapping predictions. For example, the SL prediction of *as Chief Executive of* cannot strictly match any candidate in the candidate set of the CLS paradigm. Hence it is discarded after output standardization. ② **Multiple predictions for one annotated span.** If the outputs are with confidence scores, we choose the prediction

	Original Evaluation						Consistent Evaluation					
	ED			EAE			ED			EAE		
Metric	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
DMCNN	75.6	63.6	69.1	62.2	46.9	53.5	65.0	69.7	67.2	45.3	41.6	43.2
DMBERT	77.6	71.8	74.6	58.8	55.8	57.2	72.1	77.1	74.5	50.5	60.0	54.8
CLEVE	78.1	81.5	79.8	55.4	68.0	61.1	76.4	80.4	78.3	56.9	65.9	61.0
BiLSTM+CRF	77.2	74.9	75.4	27.1*	32.3*	29.5*	74.2	78.9	76.5	42.8	32.4	36.9
BERT+CRF	71.3	77.1	74.1	41.4*	43.6*	42.5*	72.4	74.5	73.4	55.6	43.2	48.6
EEQA	71.1	73.7	72.4	56.9	49.8	53.1	70.5	77.3	73.6	65.8	25.5	36.4
PAIE	N/A	N/A	N/A	70.6*	73.2*	72.7	N/A	N/A	N/A	61.4	46.2	52.7
Text2Event	69.6	74.4	71.9	52.5	55.2	53.8	76.1	74.5	75.2	59.6	43.0	50.0

Table 7: Experimental results (%) under the original evaluation and our consistent evaluation on ACE-Full preprocessed dataset. The "original evaluation" results are directly taken from respective original papers, except the * results, which are missed in the original papers and from our reproduction. All the results are under pipeline evaluation, except for the "original evaluation" results of PAIE, which originally adopts the gold trigger evaluation. Experimental results for other preprocessing methods are in appendix B.

with the highest confidence as the final prediction, otherwise, we simply choose the first appearing prediction. The remaining predictions are discarded.

③ **Predictions without positions for non-unique spans.** We assign such predictions to the annotated spans simply by their appearing order in the output/input sequence to avoid information leakage. We encourage designing new models or post-processing rules to add positional information for CG predictions so that this issue can be directly solved by strict boundary-matching.

4.3 Provide Pipeline Evaluation Results

The absence of pipeline evaluation (§ 3.4) creates a gap between EE and EAE works, and may not well reflect EAE models' performance in real-world scenarios. Therefore, in addition to the common gold trigger evaluation results, we suggest future EAE-only works also provide pipeline evaluation results. However, there are two difficulties: (1) It is an extra overhead for the EAE-only works to implement an ED model and get predicted triggers on the datasets. (2) If two EAE models use different predicted triggers, their evaluation results are not directly comparable since the trigger quality influences EAE performance. To alleviate these difficulties, our consistent evaluation framework releases off-the-shelf predicted triggers for the widely-used EE datasets, which will help future EAE works conduct easy and consistent pipeline evaluations. The released predicted triggers are generated with existing top-performing ED models so that the obtained pipeline evaluation results shall help the community to understand the possible EE performance of combining top ED and EAE models.

4.4 Experimental Results

We re-evaluate various EE models with our consistent evaluation framework. The results are shown in Table 7, and we can observe that: (1) If we are not aware of the pitfalls of EE evaluation, we can only understand EE development status and compare competing models from the "Original Evaluation" results in Table 7. After eliminating the influence of the pitfalls with our framework, the consistent evaluation results change a lot in both absolute performance levels and relative model rankings. This comprehensively demonstrates the influence of the three identified evaluation pitfalls on EE research and highlights the importance of awareness of these pitfalls. Our framework can help avoid the pitfalls and save efforts in handling intensive evaluation implementation details. (2) Although the changes in F1 scores are minor for some models (e.g., CLEVE), their precision and recall scores vary significantly. In these cases, consistent evaluation is also necessary since real-world applications may have different precision and recall preferences.

5 Conclusion and Future Work

In this paper, we identify three pitfalls of event extraction evaluation, which are data preprocessing discrepancy, output space discrepancy, and absence of pipeline evaluation. Meta-analyses and empirical experiments present a huge impact of these pitfalls, which urges the attention of our research community. To avoid the pitfalls, we suggest a series of remedies, including specifying data preprocessing, standardizing outputs, and providing pipeline evaluation results. We develop a consistent

evaluation framework OMNIEVENT, to help future works implement these remedies. In the future, we will continually maintain it to well handle more emerging EE datasets, model paradigms, and other possible hidden evaluation pitfalls.

Limitations

The major limitations of our work are three-fold: (1) In the empirical experiments, we only train and evaluate models on English datasets. As the analyzed pitfalls are essentially language-independent, we believe the empirical conclusions could generalize to other languages. The developed consistent evaluation framework now includes multiple English and Chinese datasets, and we will extend it to support more languages in the future. (2) The three pitfalls analyzed in this paper are identified from our practical experiences and may not cover all the pitfalls of EE evaluation. We encourage the community to pay more attention to finding other possible hidden pitfalls of EE evaluation. We will also continually maintain the proposed consistent evaluation framework to support mitigating the influence of newly-found pitfalls. (3) Our meta-analysis only covers papers published at ACL, EMNLP, NAACL, and COLING on mainstream EE research since 2015. Although we believe that we can obtain representative observations from the 116 surveyed papers, some EE works published at other venues and at earlier times are missed.

Ethical Considerations

We discuss the ethical considerations and broader impact of this work here: (1) **Intellectual property**. The copyright of ACE 2005 belongs to LDC[6]. We access it through our LDC membership and strictly adhere to its license. We believe the established ACE 2005 dataset is desensitized. In our consistent evaluation framework, we will only provide preprocessing scripts rather than preprocessed datasets for those datasets whose licenses do not permit redistribution. The `ACE-DYGIE` preprocessing script[7] and the used code repositories for DMCNN[8], DMBERT[8], BiLSTM+CRF[8], BERT+CRF[8], EEQA[9], and Text2Event[10] are released under MIT license[11]. These are all public research resources. We use them for the research purpose in this work, which is consistent with their intended use. (2) **Intended use**. Our consistent evaluation framework implements the suggested remedies to avoid the identified pitfalls in EE evaluation. Researchers are supposed to use this framework to conduct consistent evaluations for comparing various competing EE models. (3) **Misuse risks**. The results reported in this paper and the evaluation results produced by our consistent evaluation framework **should not** be used for offensive arguments or interpreted as implying misconduct of other works. The analyzed pitfalls in this work are inconspicuous and very easy to be accidentally overlooked. Hence the community is generally unaware of them or underestimates their influence. The contribution of our work lies in raising awareness of the pitfalls and helping to avoid them in future works. (4) **Accessibility**. Many widely-used datasets (such as ACE 2005, KBP, etc.) are not freely available to everyone. The financial fairness issue may influence the broader usage of the data for EE research.

[6] https://www.ldc.upenn.edu/
[7] https://github.com/dwadden/dygiepp
[8] https://github.com/THU-KEG/MAVEN-dataset
[9] https://github.com/xinyadu/eeqa
[10] https://github.com/luyaojie/Text2Event
[11] https://opensource.org/licenses/MIT

References

David Ahn. 2006. The stages of event extraction. In *Proceedings of ACL Workshop on Annotating and Reasoning about Time and Events*, pages 1–8.

Jun Araki and Teruko Mitamura. 2018. Open-domain event detection using distant supervision. In *Proceedings of COLING*, pages 878–891.

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2019. Sub-event detection from twitter streams as a sequence labeling problem. In *Proceedings of NAACL-HLT*, pages 745–750.

Ofer Bronstein, Ido Dagan, Qi Li, Heng Ji, and Anette Frank. 2015. Seed-based event trigger labeling: How far can event descriptions get us? In *Proceedings of ACL-IJCNLP*, pages 372–376.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Proceedings of NeurIPS*, volume 33, pages 1877–1901.

Hu Cao, Jingye Li, Fangfang Su, Fei Li, Hao Fei, Shengqiong Wu, Bobo Li, Liang Zhao, and Donghong Ji. 2022. OneEE: A one-stage framework for fast overlapping and nested event extraction. In *Proceedings of COLING*, pages 1953–1964.

Pengfei Cao, Yubo Chen, Jun Zhao, and Taifeng Wang. 2020. Incremental event detection via knowledge consolidation networks. In *Proceedings of EMNLP*, pages 707–717.

Yee Seng Chan, Joshua Fasching, Haoling Qiu, and Bonan Min. 2019. Rapid customization for event extraction. In *Proceedings of ACL: System Demonstrations*, pages 31–36.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2021. Honey or poison? Solving the trigger curse in few-shot event detection via causal intervention. In *Proceedings of EMNLP*, pages 8078–8088.

Yubo Chen, Shulin Liu, Xiang Zhang, Kang Liu, and Jun Zhao. 2017. Automatically Labeled Data Generation for Large Scale Event Extraction. In *Proceedings of ACL*, pages 409–419.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of ACL-IJCNLP*, pages 167–176.

Yubo Chen, Hang Yang, Kang Liu, Jun Zhao, and Yantao Jia. 2018. Collective event detection via a hierarchical and bias tagging networks with gated multi-level attention mechanisms. In *Proceedings of EMNLP*, pages 1267–1276.

Xin Cong, Shiyao Cui, Bowen Yu, Tingwen Liu, Wang Yubin, and Bin Wang. 2021. Few-Shot Event Detection with Prototypical Amortized Conditional Random Field. In *Findings of ACL-IJCNLP*, pages 28–40.

Shiyao Cui, Bowen Yu, Tingwen Liu, Zhenyu Zhang, Xuebin Wang, and Jinqiao Shi. 2020. Edge-enhanced graph convolution networks for event detection with syntactic relation. In *Findings of EMNLP*, pages 2329–2339.

Shumin Deng, Ningyu Zhang, Luoqiu Li, Chen Hui, Tou Huaixiao, Mosha Chen, Fei Huang, and Huajun Chen. 2021. OntoED: Low-resource event detection with ontology embedding. In *Proceedings of ACL-IJCNLP*, pages 2828–2839.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Ning Ding, Ziran Li, Zhiyuan Liu, Haitao Zheng, and Zibo Lin. 2019. Event detection with trigger-aware lattice neural network. In *Proceedings of EMNLP-IJCNLP*, pages 347–356.

Xinya Du and Claire Cardie. 2020a. Document-level event role filler extraction using multi-granularity contextualized encoding. In *Proceedings of ACL*, pages 8010–8020.

Xinya Du and Claire Cardie. 2020b. Event extraction by answering (almost) natural questions. In *Proceedings of EMNLP*, pages 671–683.

Xinya Du, Sha Li, and Heng Ji. 2022. Dynamic global memory for document-level argument extraction. In *Proceedings of ACL*, pages 5264–5275.

Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2020. Multi-sentence argument linking. In *Proceedings of ACL*, pages 8057–8077.

Joe Ellis, Jeremy Getman, Dana Fore, Neil Kuster, Zhiyi Song, Ann Bies, and Stephanie M Strassel. 2015. Overview of linguistic resources for the TAC KBP 2015 evaluations: Methodologies and results. In *TAC*.

Joe Ellis, Jeremy Getman, Dana Fore, Neil Kuster, Zhiyi Song, Ann Bies, and Stephanie M Strassel. 2016. Overview of Linguistic Resources for the TAC KBP 2016 Evaluations: Methodologies and Results. In *TAC*.

Joe Ellis, Jeremy Getman, and Stephanie M Strassel. 2014. Overview of linguistic resources for the TAC KBP 2014 evaluations: Planning, execution, and results. In *TAC*.

Kurt Junshean Espinosa, Makoto Miwa, and Sophia Ananiadou. 2019. A search-based neural model for biomedical nested and overlapping event detection. In *Proceedings of EMNLP-IJCNLP*, pages 3679–3686.

Xiaocheng Feng, Lifu Huang, Duyu Tang, Heng Ji, Bing Qin, and Ting Liu. 2016. A language-independent neural network for event detection. In *Proceedings of ACL*, pages 66–71.

Tao Ge, Lei Cui, Baobao Chang, Zhifang Sui, and Ming Zhou. 2016. Event detection with burst information networks. In *Proceedings of COLING*, pages 3276–3286.

Jeremy Getman, Joe Ellis, Zhiyi Song, Jennifer Tracey, and Stephanie Strassel. 2017. Overview of linguistic resources for the tac kbp 2017 evaluations: Methodologies and results. In *TAC*.

Reza Ghaeini, Xiaoli Fern, Liang Huang, and Prasad Tadepalli. 2016. Event nugget detection with forward-backward recurrent neural networks. In *Proceedings of ACL*, pages 369–373.

Goran Glavaš and Jan Šnajder. 2014. Event graphs for information retrieval and multi-document summarization. *Expert systems with applications*, 41(15):6904–6916.

Prashant Gupta and Heng Ji. 2009. Predicting Unknown Time Arguments based on Cross-Event Propagation. In *Proceedings of ACL-IJCNLP*, pages 369–372.

Luis Guzman-Nateras, Minh Van Nguyen, and Thien Nguyen. 2022. Cross-lingual event detection via optimized adversarial training. In *Proceedings of NAACL-HLT*, pages 5588–5599.

Frederik Hogenboom, Flavius Frasincar, Uzay Kaymak, Franciska De Jong, and Emiel Caron. 2016. A survey of event extraction methods from text for decision support systems. *Decision Support Systems*, 85:12–22.

Yu Hong, Jianfeng Zhang, Bin Ma, Jianmin Yao, Guodong Zhou, and Qiaoming Zhu. 2011. Using cross-entity inference to improve event extraction. In *Proceedings of ACL-HLT*, pages 1127–1136.

Andrew Hsi, Yiming Yang, Jaime Carbonell, and Ruochen Xu. 2016. Leveraging multilingual training for limited resource event extraction. In *Proceedings of COLING*, pages 1201–1210.

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 NAACL-HLT*, pages 1890–1908.

Kuan-Hao Huang, I-Hung Hsu, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2022. Multilingual generative language models for zero-shot cross-lingual event argument extraction. In *Proceedings of ACL*, pages 4633–4646.

Kung-Hsiang Huang, Mu Yang, and Nanyun Peng. 2020a. Biomedical event extraction with hierarchical knowledge graphs. In *Findings of EMNLP*, pages 1277–1285.

Lifu Huang, Taylor Cassidy, Xiaocheng Feng, Heng Ji, Clare R. Voss, Jiawei Han, and Avirup Sil. 2016. Liberal Event Extraction and Event Schema Induction. In *Proceedings of ACL*, pages 258–268.

Lifu Huang and Heng Ji. 2020. Semi-supervised New Event Type Induction and Event Detection. In *Proceedings of EMNLP*, pages 718–724.

Peixin Huang, Xiang Zhao, Ryuichi Takanobu, Zhen Tan, and Weidong Xiao. 2020b. Joint event extraction with hierarchical policy network. In *Proceedings of COLING*, pages 2653–2664.

Yusheng Huang and Weijia Jia. 2021. Exploring sentence community for document-level event extraction. In *Findings of EMNLP*, pages 340–351.

Ander Intxaurrondo, Eneko Agirre, Oier Lopez de Lacalle, and Mihai Surdeanu. 2015. Diamonds in the rough: Event extraction from imperfect microblog data. In *Proceedings of NAACL-HLT*, pages 641–650.

Abhyuday N Jagannatha and Hong Yu. 2016. Bidirectional RNN for medical event detection in electronic health records. In *Proceedings of NAACL-HLT*, pages 473–482.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *Proceedings of ACL*, pages 254–262.

Heng Ji and Ralph Grishman. 2011. Knowledge Base Population: Successful Approaches and Challenges. In *Proceedings of ACL*, pages 1148–1158.

Alex Judea and Michael Strube. 2016. Incremental global event extraction. In *Proceedings of COLING*, pages 2279–2289.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML*, pages 282–289.

Viet Lai, Franck Dernoncourt, and Thien Huu Nguyen. 2021. Learning prototype representations across few-shot tasks for event detection. In *Proceedings of EMNLP*, pages 5270–5277.

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 EMNLP*, pages 5405–5411.

Kenton Lee, Yoav Artzi, Yejin Choi, and Luke Zettlemoyer. 2015. Event detection and factuality assessment with non-expert supervision. In *Proceedings of EMNLP*, pages 1643–1648.

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 ACL*, pages 7871–7880.

Diya Li, Lifu Huang, Heng Ji, and Jiawei Han. 2019. Biomedical event extraction based on knowledge-driven tree-LSTM. In *Proceedings of NAACL-HLT*, pages 1421–1430.

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 EMNLP*, pages 829–838.

Haochen Li, Tong Mo, Hongcheng Fan, Jingkun Wang, Jiaxi Wang, Fuhao Zhang, and Weiping Li. 2022a. KiPT: Knowledge-injected prompt tuning for event detection. In *Proceedings of COLING*, pages 1943–1952.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of ACL*, pages 73–82.

Rui Li, Wenlin Zhao, Cheng Yang, and Sen Su. 2021a. Treasures outside contexts: Improving event detection via global statistics. In *Proceedings of EMNLP*, pages 2625–2635.

Sha Li, Heng Ji, and Jiawei Han. 2021b. Document-level event argument extraction by conditional generation. In *Proceedings of NAACL-HLT*, pages 894–908.

Xinyu Li, Fayuan Li, Lu Pan, Yuguang Chen, Weihua Peng, Quan Wang, Yajuan Lyu, and Yong Zhu. 2020b. Duee: A large-scale dataset for chinese event extraction in real-world scenarios. In *Proceedings of NLPCC*, volume 12431 of *Lecture Notes in Computer Science*, pages 534–545.

Zhongqiu Li, Yu Hong, Jie Wang, Shiming He, Jianmin Yao, and Guodong Zhou. 2022b. Unregulated Chinese-to-English data expansion does NOT work for neural event detection. In *Proceedings of COLING*, pages 2633–2638.

Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. 2019. Cost-sensitive regularization for label confusion-aware event detection. In *Proceedings of ACL*, pages 5278–5283.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of ACL*, pages 7999–8009.

Anan Liu, Ning Xu, and Haozhe Liu. 2021a. Self-attention graph residual convolutional networks for event detection with dependency relations. In *Findings of EMNLP*, pages 302–311.

Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. 2020a. Event Extraction as Machine Reading Comprehension. In *Proceedings of EMNLP*, pages 1641–1651.

Jian Liu, Yubo Chen, Kang Liu, Yantao Jia, and Zhicheng Sheng. 2020b. How does context matter? On the robustness of event detection with context-selective mask generalization. In *Findings of EMNLP*, pages 2523–2532.

Jian Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2019a. Neural cross-lingual event detection with minimal parallel resources. In *Proceedings of EMNLP-IJCNLP*, pages 738–748.

Jian Liu, Yufeng Chen, and Jinan Xu. 2021b. Machine reading comprehension as data augmentation: A case study on implicit event argument extraction. In *Proceedings of EMNLP*, pages 2716–2725.

Jian Liu, Yufeng Chen, and Jinan Xu. 2022a. Saliency as evidence: Event detection with trigger saliency attribution. In *Proceedings of ACL*, pages 4573–4585.

Minqian Liu, Shiyu Chang, and Lifu Huang. 2022b. Incremental prompting: Episodic memory prompt for lifelong event detection. In *Proceedings of COLING*, pages 2157–2165.

Shaobo Liu, Rui Cheng, Xiaoming Yu, and Xueqi Cheng. 2018a. Exploiting contextual information via dynamic memory network for event detection. In *Proceedings of EMNLP*, pages 1030–1035.

Shulin Liu, Yubo Chen, Shizhu He, Kang Liu, and Jun Zhao. 2016. Leveraging FrameNet to improve automatic event detection. In *Proceedings of ACL*, pages 2134–2143.

Shulin Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2017. Exploiting Argument Information to Improve Event Detection via Supervised Attention Mechanisms. In *Proceedings of ACL*, pages 1789–1798.

Shulin Liu, Yang Li, Feng Zhang, Tao Yang, and Xinpeng Zhou. 2019b. Event detection without triggers. In *Proceedings of NAACL-HLT*, pages 735–744.

Xiao Liu, Heyan Huang, Ge Shi, and Bo Wang. 2022c. Dynamic prefix-tuning for generative template-based event extraction. In *Proceedings of ACL*, pages 5216–5228.

Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018b. Jointly multiple events extraction via attention-based graph information aggregation. In *Proceedings of EMNLP*, pages 1247–1256.

Dongfang Lou, Zhilin Liao, Shumin Deng, Ningyu Zhang, and Huajun Chen. 2021. MLBiNet: A cross-sentence collective event detection network. In *Proceedings of ACL-IJCNLP*, pages 4829–4839.

Weiyi Lu and Thien Huu Nguyen. 2018. Similar but not the same: Word sense disambiguation improves event detection via neural representation matching. In *Proceedings of EMNLP*, pages 4822–4828.

Yaojie Lu, Hongyu Lin, Xianpei Han, and Le Sun. 2019. Distilling discrimination and generalization knowledge for event detection via delta-representation learning. In *Proceedings of ACL*, pages 4366–4376.

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 ACL-IJCNLP*, pages 2795–2806.

Qing Lyu, Hongming Zhang, Elior Sulem, and Dan Roth. 2021. Zero-shot event extraction via transfer learning: Challenges and insights. In *Proceedings of ACL-IJCNLP*, pages 322–332.

Jie Ma, Shuai Wang, Rishita Anubhai, Miguel Ballesteros, and Yaser Al-Onaizan. 2020. Resource-enhanced neural model for event argument extraction. In *Findings of EMNLP*, pages 3554–3559.

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 ACL*, pages 6759–6774.

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 EMNLP*, pages 5381–5390.

Jiaxin Mi, Po Hu, and Peng Li. 2022. Event detection with dual relational graph attention networks. In *Proceedings of COLING*, pages 1979–1989.

Aakanksha Naik and Carolyn Rose. 2020. Towards open domain event trigger identification using adversarial domain adaptation. In *Proceedings of ACL*, pages 7618–7624.

Nghia Ngo Trung, Duy Phung, and Thien Huu Nguyen. 2021. Unsupervised domain adaptation for event detection using domain-specific adapters. In *Findings of ACL-IJCNLP*, pages 4015–4025.

Minh Van Nguyen, Bonan Min, Franck Dernoncourt, and Thien Nguyen. 2022. Joint extraction of entities, relations, and events via modeling inter-instance and inter-label dependencies. In *Proceedings of NAACL-HLT*, pages 4363–4374.

Minh Van Nguyen, Tuan Ngo Nguyen, Bonan Min, and Thien Huu Nguyen. 2021. Crosslingual transfer learning for relation and event extraction via word category and class alignments. In *Proceedings of EMNLP*, pages 5414–5426.

Thien Nguyen and Ralph Grishman. 2018. Graph convolutional networks with argument-aware pooling for event detection. In *Proceedings of AAAI*, pages 5900–5907.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *Proceedings of NAACL-HLT*, pages 300–309.

Thien Huu Nguyen and Ralph Grishman. 2015. Event Detection and Domain Adaptation with Convolutional Neural Networks. In *Proceedings of ACL*, pages 365–371.

Thien Huu Nguyen and Ralph Grishman. 2016. Modeling skip-grams for event detection with convolutional neural networks. In *Proceedings of EMNLP*, pages 886–891.

Walker Orr, Prasad Tadepalli, and Xiaoli Fern. 2018. Event detection with neural networks: A rigorous empirical evaluation. In *Proceedings of EMNLP*, pages 999–1004.

Haoruo Peng, Yangqiu Song, and Dan Roth. 2016. Event detection and co-reference with minimal supervision. In *Proceedings of EMNLP*, pages 392–402.

Amir Pouran Ben Veyseh, Viet Lai, Franck Dernoncourt, and Thien Huu Nguyen. 2021a. Unleash GPT-2 power for event detection. In *Proceedings of ACL-IJCNLP*, pages 6271–6282.

Amir Pouran Ben Veyseh, Minh Van Nguyen, Franck Dernoncourt, Bonan Min, and Thien Nguyen. 2022. Document-level event argument extraction via optimal transport. In *Findings of ACL*, pages 1648–1658.

Amir Pouran Ben Veyseh, Minh Van Nguyen, Nghia Ngo Trung, Bonan Min, and Thien Huu Nguyen. 2021b. Modeling document-level context for event detection via important context selection. In *Proceedings of EMNLP*, pages 5403–5413.

Amir Pouran Ben Veyseh, Tuan Ngo Nguyen, and Thien Huu Nguyen. 2020. Graph Transformer Networks with Syntactic and Semantic Structures for Event Argument Extraction. In *Findings of EMNLP*, pages 3651–3661.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Alan Ramponi, Rob van der Goot, Rosario Lombardo, and Barbara Plank. 2020. Biomedical event extraction as sequence labeling. In *Proceedings of EMNLP*, pages 5357–5367.

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*.

Yubing Ren, Yanan Cao, Fang Fang, Ping Guo, Zheng Lin, Wei Ma, and Yi Liu. 2022. CLIO: Role-interactive multi-event head attention network for document-level event extraction. In *Proceedings of COLING*, pages 2504–2514.

Oscar Sainz, Itziar Gonzalez-Dios, Oier Lopez de Lacalle, Bonan Min, and Eneko Agirre. 2022. Textual entailment for event argument extraction: Zero- and few-shot with multi-source learning. In *Findings of NAACL-HLT*, pages 2439–2455.

Lei Sha, Jing Liu, Chin-Yew Lin, Sujian Li, Baobao Chang, and Zhifang Sui. 2016. RBPB: Regularization-based pattern balancing method for event extraction. In *Proceedings of ACL*, pages 1224–1234.

Shirong Shen, Guilin Qi, Zhen Li, Sheng Bi, and Lusheng Wang. 2020. Hierarchical Chinese legal event extraction via pedal attention mechanism. In *Proceedings of COLING*, pages 100–113.

Shirong Shen, Tongtong Wu, Guilin Qi, Yuan-Fang Li, Gholamreza Haffari, and Sheng Bi. 2021. Adaptive knowledge-enhanced Bayesian meta-learning for few-shot event detection. In *Findings of ACL-IJCNLP*, pages 2417–2429.

Jiawei Sheng, Shu Guo, Bowen Yu, Qian Li, Yiming Hei, Lihong Wang, Tingwen Liu, and Hongbo Xu. 2021. CasEE: A joint learning framework with cascade decoding for overlapping event extraction. In *Findings of ACL-IJCNLP*, pages 164–174.

Matthew Sims, Jong Ho Park, and David Bamman. 2019. Literary event detection. In *Proceedings of ACL*, pages 3623–3634.

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 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 89–98.

Ananya Subburathinam, Di Lu, Heng Ji, Jonathan May, Shih-Fu Chang, Avirup Sil, and Clare Voss. 2019. Cross-lingual structure transfer for relation

and event extraction. In *Proceedings of EMNLP-IJCNLP*, pages 313–325.

Meihan Tong, Bin Xu, Shuai Wang, Yixin Cao, Lei Hou, Juanzi Li, and Jun Xie. 2020. Improving event detection via open-domain trigger knowledge. In *Proceedings of ACL*, pages 5887–5897.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, Relation, and Event Extraction with Contextualized Span Representations. In *Proceedings of EMNLP-IJCNLP*, pages 5784–5789.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. ACE 2005 multilingual training corpus. *Linguistic Data Consortium*, 57.

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 ACL*, pages 169–182.

Xiaozhi Wang, Xu Han, Zhiyuan Liu, Maosong Sun, and Peng Li. 2019a. Adversarial Training for Weakly Supervised Event Detection. In *Proceedings of NAACL-HLT*, pages 998–1008.

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 EMNLP*, pages 1652–1671.

Xiaozhi Wang, Ziqi Wang, Xu Han, Zhiyuan Liu, Juanzi Li, Peng Li, Maosong Sun, Jie Zhou, and Xiang Ren. 2019b. HMEAE: Hierarchical Modular Event Argument Extraction. In *Proceedings of EMNLP-IJCNLP*, pages 5777–5783.

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 ACL-IJCNLP*, pages 6283–6297.

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 ACL-IJCNLP*, pages 4672–4682.

Sam Wei, Igor Korostil, Joel Nothman, and Ben Hachey. 2017. English event detection with translated language features. In *Proceedings of ACL*, pages 293–298.

Yinyi Wei, Shuaipeng Liu, Jianwei Lv, Xiangyu Xi, Hailei Yan, Wei Ye, Tong Mo, Fan Yang, and Guanglu Wan. 2022. DESED: Dialogue-based explanation for sentence-level event detection. In *Proceedings of COLING*, pages 2483–2493.

Dominik Wurzer, Victor Lavrenko, and Miles Osborne. 2015. Twitter-scale new event detection via k-term hashing. In *Proceedings of EMNLP*, pages 2584–2589.

Xiangyu Xi, Wei Ye, Shikun Zhang, Quanxiu Wang, Huixing Jiang, and Wei Wu. 2021. Capturing event argument interaction via a bi-directional entity-level recurrent decoder. In *Proceedings of ACL-IJCNLP*, pages 210–219.

Jianye Xie, Haotong Sun, Junsheng Zhou, Weiguang Qu, and Xinyu Dai. 2021. Event detection as graph parsing. In *Findings of ACL-IJCNLP*, pages 1630–1640.

Runxin Xu, Tianyu Liu, Lei Li, and Baobao Chang. 2021. Document-level event extraction via heterogeneous graph-based interaction model with a tracker. In *Proceedings of ACL-IJCNLP*, pages 3533–3546.

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 NAACL-HLT*, pages 5025–5036.

Semih Yagcioglu, Mehmet Saygin Seyfioglu, Begum Citamak, Batuhan Bardak, Seren Guldamlasioglu, Azmi Yuksel, and Emin Islam Tatli. 2019. Detecting cybersecurity events from noisy short text. In *Proceedings of NAACL-HLT*, pages 1366–1372.

Haoran Yan, Xiaolong Jin, Xiangbin Meng, Jiafeng Guo, and Xueqi Cheng. 2019. Event Detection with Multi-Order Graph Convolution and Aggregated Attention. In *Proceedings of EMNLP-IJCNLP*, pages 5766–5770.

Bishan Yang and Tom M. Mitchell. 2016. Joint extraction of events and entities within a document context. In *Proceedings of NAACL-HLT*, pages 289–299.

Hang Yang, Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, and Taifeng Wang. 2021. Document-level event extraction via parallel prediction networks. In *Proceedings of ACL-IJCNLP*, pages 6298–6308.

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 ACL*, pages 5284–5294.

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 ACL*, pages 183–201.

Pengfei Yu, Heng Ji, and Prem Natarajan. 2021. Lifelong event detection with knowledge transfer. In *Proceedings of EMNLP*, pages 5278–5290.

Qi Zeng, Qiusi Zhan, and Heng Ji. 2022. EA^{2}E: Improving consistency with event awareness for document-level argument extraction. In *Findings of NAACL*, pages 2649–2655.

Hongming Zhang, Xin Liu, Haojie Pan, Yangqiu Song, and Cane Wing-Ki Leung. 2020a. ASER: A large-scale eventuality knowledge graph. In *Proceedings of WWW*, pages 201–211.

Hongming Zhang, Haoyu Wang, and Dan Roth. 2021. Zero-shot Label-aware Event Trigger and Argument Classification. In *Findings of ACL-IJCNLP*, pages 1331–1340.

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

Zhisong Zhang, Xiang Kong, Zhengzhong Liu, Xuezhe Ma, and Eduard Hovy. 2020b. A two-step approach for implicit event argument detection. In *Proceedings of ACL*, pages 7479–7485.

Zixuan Zhang and Heng Ji. 2021. Abstract Meaning Representation guided graph encoding and decoding for joint information extraction. In *Proceedings of NAACL-HLT*, pages 39–49.

Shun Zheng, Wei Cao, Wei Xu, and Jiang Bian. 2019. Doc2EDAG: An end-to-end document-level framework for Chinese financial event extraction. In *Proceedings of EMNLP-IJCNLP*, pages 337–346.

Hanzhang Zhou and Kezhi Mao. 2022. Document-level event argument extraction by leveraging redundant information and closed boundary loss. In *Proceedings of NAACL-HLT*, pages 3041–3052.

Jie Zhou, Qi Zhang, Qin Chen, Qi Zhang, Liang He, and Xuanjing Huang. 2022. A multi-format transfer learning model for event argument extraction via variational information bottleneck. In *Proceedings of COLING*, pages 1990–2000.

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 AAAI*, volume 35, pages 14638–14646.

Appendices

A Experimental Details

The section introduces the experimental details in the paper, including the data preprocessing details (appendix A.1), the reproduction details (appendix A.2), and the training details (appendix A.3).

A.1 Data Preprocessing Details

The section introduces the details of the three data preprocessing scripts for ACE 2005: `ACE-DYGIE`, `ACE-OneIE`, and `ACE-Full`.

ACE-DYGIE We adopt the released official codes[12] provided by Wadden et al. (2019) as the `ACE-DYGIE` preprocessing script. Specifically, we adopt the widely-used "default-settings" in the codes to preprocess ACE 2005. `ACE-DYGIE` uses spaCy[13] for sentence segmentation and tokenization. The version of spaCy is 2.0.18, and the used spaCy model is `en_core_web_sm`.

ACE-OneIE We adopt the released official codes[14] provided by Lin et al. (2020) as the `ACE-OneIE` preprocessing script. `ACE-OneIE` uses NLTK[15] for sentence segmentation and tokenization , and the version of NLTK is 3.5.

ACE-Full We adopt the released official codes[16] provided by Wang et al. (2019b) as the `ACE-Full` preprocessing script. `ACE-Full` uses the Stanford CoreNLP[17] toolkit for sentence segmentation and tokenization, and the version of CoreNLP is 4.4.0.

A.2 Reproduction Details

In this section, we introduce the reproduction details of all the reproduced models and provide some explanations for the results' differences between our reproduction and the originally reported results. All the reproduction experiments adopt their original evaluation settings, respectively. The number of parameters for each reproduced model is shown in Table 8.

[12] https://github.com/dwadden/dygiepp
[13] https://spacy.io/
[14] https://blender.cs.illinois.edu/software/oneie/
[15] https://www.nltk.org/
[16] https://github.com/thunlp/HMEAE
[17] https://stanfordnlp.github.io/CoreNLP/

Model	#Paramter
DMCNN	2M
DMBERT	110M
CLEVE	354M
BiLSTM+CRF	37M
BERT+CRF	110M
EEQA	110M
PAIE	406M
Text2Event	770M

Table 8: Number of parameters for each reproduced model.

DMCNN Our DMCNN implementation is mainly based on the codes[18] provided by Wang et al. (2020). The reproduced ED F1 score (67.2) is similar to the reported result (69.1) in the original paper (Chen et al., 2015) on the ACE 2005 dataset. However, there is a gap between our reproduced and the originally reported EAE F1 scores (43.2 vs. 53.5). A possible reason is that Chen et al. (2015) adopts a different EAE evaluation setting: Only the argument annotations of the predicted triggers are included in the metric calculation, while the argument annotations of the false negative trigger predictions are discarded. This setting is also adopted in some other early works like DMBERT (Wang et al., 2019b), and we call it "legacy setting". Compared to the common evaluation setting now, which includes all the argument annotations, the recall scores under the legacy setting are typically higher. When re-evaluating our reproduced DMCNN under the legacy setting, the EAE F1 score (53.9) is consistent with the originally reported result (53.5).

DMBERT Our DMBERT implementation is mainly based on the codes[18] provided by (Wang et al., 2020). The reproduced ED F1 score (74.5) is consistent with the originally reported result (74.3) on the ACE 2005 dataset. However, similar to the DMCNN case introduced in the last paragraph, the reproduced EAE F1 score (54.8) is lower than the originally reported result (57.2 in Wang et al. (2019b)) due to the "legacy setting". When re-evaluating the reproduced DMBERT under the legacy setting, the EAE F1 score is 60.6.

CLEVE We download the pre-trained CLEVE checkpoint[19] and finetune it on ACE 2005. The reproduced F1 scores of ED (78.3) and EAE (61.0) are basically consistent with the originally reported ED (79.8) and EAE (61.1) results.

[18] https://github.com/THU-KEG/MAVEN-dataset
[19] https://github.com/THU-KEG/CLEVE

BiLSTM+CRF We implement BiLSTM+CRF based on the codes[18] provided by Wang et al. (2020). The reproduced ED F1 score (75.5) is similar to the reported result (75.4) in the original paper (Wang et al., 2020) on ACE 2005. As there is no work using BiLSTM+CRF to perform EAE, we adopt all the settings used in ED and evaluate the EAE performance of BiLSTM+CRF.

BERT+CRF We implement BERT+CRF based on the codes[18] provided by Wang et al. (2020). The reproduced ED F1 score (72.1) is similar to the reported result (74.1) in the original paper (Wang et al., 2020) on ACE 2005. As there is no work using BERT+CRF to perform EAE, we implement its EAE model following all the ED settings.

EEQA We implement EEQA (Du and Cardie, 2020b) based on the released official codes[20]. When directly running the released code, we get the F1 score of 69.0 for ED and 47.3 for EAE, which are consistent with our finally reproduced ED (69.5) and EAE (47.4) results. However, there is still a gap between the reproduced and the originally reported results, which is also mentioned in several GitHub issues[21].

PAIE We implement PAIE (Ma et al., 2022) based on the released official codes[22] and evaluate it in different evaluation settings. The reproduced EAE F1 score (71.8) is basically consistent with that reported in the original paper (72.7).

Text2Event We adopt the released official codes[23] to re-evaluate Text2Event (Lu et al., 2021) in different settings. There are minor differences between the reproduced F1 results and the originally reported results (ED: 69.5 vs. 71.9, EAE: 50.8 vs. 53.8). We think the differences come from randomness. When only using the same random seed reported by the authors, the reproduction results are nearly the same as the original results.

A.3 Training Details

We run three random trials for all the experiments using three different seeds (seed=0, seed=1, seed=2). The final reported results are the average results over the three random trials. All hyperparameters are the same as those used in the original papers. The experiments of CLEVE, PAIE, and Text2Event are run on Nvidia A100 GPUs, which consume about 600 GPU hours. The other experiments are run on Nvidia GeForce RTX 3090 GPUs, which consume about 100 GPU hours.

[20] https://github.com/xinyadu/eeqa

[21] https://github.com/xinyadu/eeqa/issues/11, https://github.com/xinyadu/eeqa/issues/5

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

[23] https://github.com/luyaojie/Text2Event

B Additional Experimental Results

The section shows additional experimental results on different preprocessed ACE 2005 datasets.

Output Space Discrepancy Table 9 shows the metrics' differences with and without output standardization on the ACE-DYGIE and ACE-Full preprocessed datasets. We can observe that all evaluation metrics change obviously, which is consistent with the observations in § 3.3.

Absence of Pipeline Evaluation Table 10 shows the results using gold trigger evaluation and pipeline evaluation on the ACE-DYGIE and ACE-Full preprocessed datasets. We can observe that the phenomena are consistent with those in § 3.4.

Consistent Evaluation Framework Table 11 shows the results using our consistent evaluation on ACE-DYGIE, ACE-OneIE, and ACE-Full. We can observe that the phenomena on ACE-DYGIE and ACE-OneIE are consistent with those in § 4.4.

C Papers for Meta-Analysis

The complete list of papers surveyed in our meta-analysis is shown in Table 12.

D Authors' Contribution

Hao Peng, Feng Yao, and Kaisheng Zeng conducted the empirical experiments. Feng Yao conducted the meta-analyses. Xiaozhi Wang, Hao Peng, and Feng Yao wrote the paper. Xiaozhi Wang designed the project. Lei Hou, Juanzi Li, Zhiyuan Liu, and Weixing Shen advised the project. All authors participated in the discussion.

	ACE-DYGIE						ACE-Full					
	ED			EAE			ED			EAE		
Metric	**ΔP**	**ΔR**	**ΔF1**	**ΔP**	**ΔR**	**ΔF1**	**ΔP**	**ΔR**	**ΔF1**	**ΔP**	**ΔR**	**ΔF1**
BiLSTM+CRF	+0.4	+0.0	+0.2	+6.7	+0.2	+3.7	+1.9	−0.2	+1.0	+15.7	+0.1	+7.4
BERT+CRF	+0.6	−0.2	+0.2	+5.2	−0.1	+2.9	+2.5	−0.2	+1.3	+14.1	−0.4	+6.1
EEQA	+0.0	+0.0	+0.0	−0.7	−1.2	−1.0	−0.3	+1.3	+0.4	+19.3	−15.3	−6.9
PAIE	N/A	N/A	N/A	+4.4	−0.5	+1.9	N/A	N/A	N/A	+21.9	−1.6	+8.4
Text2Event	+0.3	+0.0	+0.1	+0.5	−2.5	−0.9	+2.0	+0.0	+1.0	+6.2	−3.7	+0.1

Table 9: The precision, recall, and F1 (%) differences between evaluation with and without our output standardization. The results are evaluated on the ACE-DYGIE and ACE-Full preprocessed datasets. Output standardization aligns the output spaces of the other paradigms into that of the CLS paradigm, and hence we do not include the CLS-paradigm models here, whose results are unchanged.

	ACE-DYGIE			ACE-Full		
Metric	**ED F1**	**Gold Tri. EAE F1**	**Pipeline EAE F1**	**ED F1**	**Gold Tri. EAE F1**	**Pipeline EAE F1**
DMCNN	62.5	50.1	34.0	67.2	61.8	43.2
DMBERT	68.3	67.3	48.1	74.5	73.1	54.8
CLEVE	72.9	71.4	54.8	78.3	76.2	61.0
BiLSTM+CRF	72.0	45.2	36.2	76.5	46.2	36.9
BERT+CRF	68.1	64.1	47.8	73.4	64.5	48.6
EEQA	69.5	63.5	46.4	73.6	46.1	36.4
PAIE	72.9	73.8	56.5	78.3	65.0	52.7

Table 10: EAE F1 scores (%) of gold trigger evaluation and pipeline evaluation on ACE-DYGIE and ACE-Full. We also report corresponding ED F1 scores to show trigger quality. PAIE adopts the triggers predicted by CLEVE. The joint model Text2Event is excluded since its trigger input cannot be controlled.

	ED			EAE		
Metric	**P**	**R**	**F1**	**P**	**R**	**F1**
			ACE-DYGIE			
DMCNN	58.6 ± 2.28	67.0 ± 0.88	62.5 ± 1.08	38.6 ± 1.58	30.4 ± 0.99	34.0 ± 1.20
DMBERT	66.4 ± 0.69	70.2 ± 0.73	68.3 ± 0.43	45.6 ± 1.46	51.0 ± 0.87	48.1 ± 0.91
CLEVE	70.7 ± 0.87	75.3 ± 0.82	72.9 ± 0.53	52.2 ± 1.47	57.6 ± 1.40	54.8 ± 1.26
BiLSTM+CRF	68.5 ± 1.27	75.8 ± 2.28	72.0 ± 0.99	36.4 ± 1.21	36.1 ± 0.37	36.2 ± 0.55
BERT+CRF	64.0 ± 1.94	72.8 ± 1.57	68.1 ± 1.01	46.3 ± 1.35	49.5 ± 2.04	47.8 ± 0.70
EEQA	65.3 ± 3.46	74.5 ± 1.22	69.5 ± 1.41	49.0 ± 3.88	44.3 ± 1.30	46.4 ± 1.06
PAIE	N/A	N/A	N/A	56.5 ± 0.49	56.5 ± 1.28	56.5 ± 0.87
Text2Event	67.2 ± 0.82	72.4 ± 0.62	69.7 ± 0.72	48.5 ± 2.60	51.6 ± 1.04	50.0 ± 0.89
			ACE-OneIE			
DMCNN	61.5 ± 2.66	64.5 ± 2.86	62.8 ± 0.40	36.7 ± 2.48	34.1 ± 1.88	35.2 ± 0.22
DMBERT	64.4 ± 2.89	75.4 ± 3.21	69.4 ± 1.36	41.5 ± 1.84	54.7 ± 1.42	47.2 ± 0.99
CLEVE	72.3 ± 1.86	78.0 ± 0.91	75.0 ± 0.81	52.1 ± 1.99	57.6 ± 0.47	54.7 ± 1.31
BiLSTM+CRF	73.0 ± 1.55	71.8 ± 0.11	72.4 ± 0.82	37.0 ± 2.33	33.1 ± 1.01	34.9 ± 1.56
BERT+CRF	69.6 ± 4.08	69.2 ± 4.23	69.2 ± 1.18	48.9 ± 3.25	45.5 ± 2.75	47.1 ± 1.04
EEQA	66.7 ± 1.73	71.8 ± 2.51	69.1 ± 0.28	50.1 ± 1.73	41.0 ± 1.92	45.0 ± 0.70
PAIE	N/A	N/A	N/A	56.1 ± 0.30	57.4 ± 0.55	56.7 ± 0.29
Text2Event	71.4 ± 1.44	74.1 ± 1.77	72.7 ± 0.20	51.5 ± 1.46	51.6 ± 0.65	51.6 ± 0.99
			ACE-Full			
DMCNN	65.0 ± 3.33	69.7 ± 0.62	67.2 ± 1.53	45.3 ± 4.79	41.6 ± 1.93	43.2 ± 1.79
DMBERT	72.1 ± 0.80	77.1 ± 1.53	74.5 ± 0.85	50.5 ± 1.53	60.0 ± 1.82	54.8 ± 1.67
CLEVE	76.4 ± 2.49	80.4 ± 1.54	78.3 ± 2.03	56.9 ± 2.86	65.9 ± 2.06	61.0 ± 2.44
BiLSTM+CRF	74.2 ± 1.62	78.9 ± 0.45	76.5 ± 1.02	42.8 ± 1.20	32.4 ± 0.23	36.9 ± 0.60
BERT+CRF	72.4 ± 2.34	74.5 ± 1.23	73.4 ± 1.29	55.6 ± 1.51	43.2 ± 1.31	48.6 ± 0.96
EEQA	70.5 ± 2.93	77.3 ± 3.28	73.6 ± 0.38	65.8 ± 2.98	25.5 ± 4.68	36.4 ± 4.49
PAIE	N/A	N/A	N/A	61.4 ± 1.70	46.2 ± 0.64	52.7 ± 0.77
Text2Event	76.1 ± 0.25	74.5 ± 1.28	75.2 ± 0.68	59.6 ± 0.96	43.0 ± 1.49	50.0 ± 1.07

Table 11: Experimental results (%) under our consistent evaluation on ACE-DYGIE, ACE-OneIE, and ACE-Full. We report averages and standard deviations over three runs. All the results are under pipeline evaluation.

ACL
Chen et al. (2015), Bronstein et al. (2015), Nguyen and Grishman (2015) Sha et al. (2016), Huang et al. (2016) Ghaeini et al. (2016), Feng et al. (2016), Liu et al. (2016), Wei et al. (2017), Liu et al. (2017), Chen et al. (2017), Chan et al. (2019), Yang et al. (2019), Sims et al. (2019), Lu et al. (2019), Lin et al. (2019), Lin et al. (2020), Naik and Rose (2020), Tong et al. (2020), Du and Cardie (2020a), Zhang et al. (2020b), Zhang et al. (2021), Lyu et al. (2021), Ngo Trung et al. (2021), Pouran Ben Veyseh et al. (2021a), Lu et al. (2021), Deng et al. (2021), Lou et al. (2021), Cong et al. (2021), Xie et al. (2021), Wang et al. (2021), Sheng et al. (2021), Shen et al. (2021), Xi et al. (2021), Wei et al. (2021), Yang et al. (2021), Xu et al. (2021), Liu et al. (2022a), Wang et al. (2022), Liu et al. (2022c), Ma et al. (2022), Huang et al. (2022), Du et al. (2022), Pouran Ben Veyseh et al. (2022)
EMNLP
Wurzer et al. (2015), Lee et al. (2015), Nguyen and Grishman (2016), Peng et al. (2016), Lu and Nguyen (2018), Chen et al. (2018), Liu et al. (2018a), Orr et al. (2018), Liu et al. (2018b), Liu et al. (2019a), Ding et al. (2019), Yan et al. (2019), Wang et al. (2019b), Espinosa et al. (2019), Wadden et al. (2019), Zheng et al. (2019), Subburathinam et al. (2019), Du and Cardie (2020b), Huang and Ji (2020), Man Duc Trong et al. (2020), Cao et al. (2020), Liu et al. (2020b), Li et al. (2020a), Liu et al. (2020a), Lai et al. (2020), Cui et al. (2020), Huang et al. (2020a), Ramponi et al. (2020), Ma et al. (2020), Pouran Ben Veyseh et al. (2020), Li et al. (2021a), Liu et al. (2021a), Pouran Ben Veyseh et al. (2021b),Yu et al. (2021), Lai et al. (2021), Chen et al. (2021),Nguyen et al. (2021), Liu et al. (2021b), Huang and Jia (2021)
NAACL
Intxaurrondo et al. (2015), Jagannatha and Yu (2016), Yang and Mitchell (2016), Nguyen et al. (2016), Bekoulis et al. (2019), Liu et al. (2019b), Yagcioglu et al. (2019), Li et al. (2019), Wang et al. (2019a), Zhang and Ji (2021), Li et al. (2021b), Zhang et al. (2022), Nguyen et al. (2022), Hsu et al. (2022), Guzman-Nateras et al. (2022), Sainz et al. (2022), Zeng et al. (2022), Zhou and Mao (2022), Xu et al. (2022)
COLING
Ge et al. (2016), Judea and Strube (2016), Hsi et al. (2016), Araki and Mitamura (2018), Huang et al. (2020b), Shen et al. (2020), Li et al. (2022b), Ren et al. (2022), Wei et al. (2022), Liu et al. (2022b), Mi et al. (2022), Cao et al. (2022), Li et al. (2022a), Zhou et al. (2022)

Table 12: The complete list of papers for meta-analysis, categorized by venues and sorted by publication years.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
In Section "Limitations"

☑ A2. Did you discuss any potential risks of your work?
In Section "Ethical Considerations"

☑ A3. Do the abstract and introduction summarize the paper's main claims?
In Section "Abstract" and "Introduction"

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

In Section 3 and Section 4

☑ B1. Did you cite the creators of artifacts you used?
In Section 3 and appendix A

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
In Section "Ethical Considerations"

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
In Section "Ethical Considerations"

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
In Section "Ethical Considerations"

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
In Section 3 and Section "Ethical Considerations"

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
In Section 3

C ☑ Did you run computational experiments?

In Section 3, Section 4, and appendix B

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
In appendix A

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
In appendix A

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
In appendix A and appendix B

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
In appendix A

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.