OmniEvent: A Comprehensive, Fair, and Easy-to-Use Toolkit for Event Understanding

Hao Peng[1*], **Xiaozhi Wang**[1*], **Feng Yao**[3], **Zimu Wang**[4], **Chuzhao Zhu**[1], **Kaisheng Zeng**[1], **Lei Hou**[1,2†], **Juanzi Li**[1,2]
[1]Department of Computer Science and Technology, BNRist;
[2]KIRC, Institute for Artificial Intelligence,
Tsinghua University, Beijing, 100084, China
[3]University of California, San Diego [4]Xi'an Jiaotong-Liverpool University
{peng-h21, wangxz20}@mails.tsinghua.edu.cn

Abstract

Event understanding aims at understanding the content and relationship of events within texts, which covers multiple complicated information extraction tasks: event detection, event argument extraction, and event relation extraction. To facilitate related research and application, we present an event understanding toolkit OmniEvent, which features three desiderata: (1) **Comprehensive.** OmniEvent supports mainstream modeling paradigms of all the event understanding tasks and the processing of 15 widely-used English and Chinese datasets. (2) **Fair.** OmniEvent carefully handles the inconspicuous evaluation pitfalls reported in Peng et al. (2023), which ensures fair comparisons between different models. (3) **Easy-to-use**. OmniEvent is designed to be easily used by users with varying needs. We provide off-the-shelf models that can be directly deployed as web services. The modular framework also enables users to easily implement and evaluate new event understanding models with OmniEvent. The toolkit[1] is publicly released along with the demonstration website and video[2].

1 Introduction

Correctly understanding events is fundamental for humans to understand the world. Event understanding requires identifying real-world events mentioned in texts and analyzing their relationships, which naturally benefits various downstream applications, such as stock prediction (Ding et al., 2015), adverse drug event detection (Wunnava et al., 2019), narrative event prediction (Wang et al., 2021a), and legal case analysis (Yao et al., 2022).

As illustrated in Figure 1, event understanding covers three complicated information extraction tasks: (1) event detection (ED), which is to detect the event triggers (keywords or phrases evoking events in texts) and classify their event types, (2) event argument extraction (EAE), which is to extract the event arguments for each trigger and classify their argument roles, and (3) event relation extraction (ERE), which is to identify the complex relationships between events, typically including temporal, causal, coreference, and subevent relations. ED and EAE together constitute the conventional event extraction (EE) task.

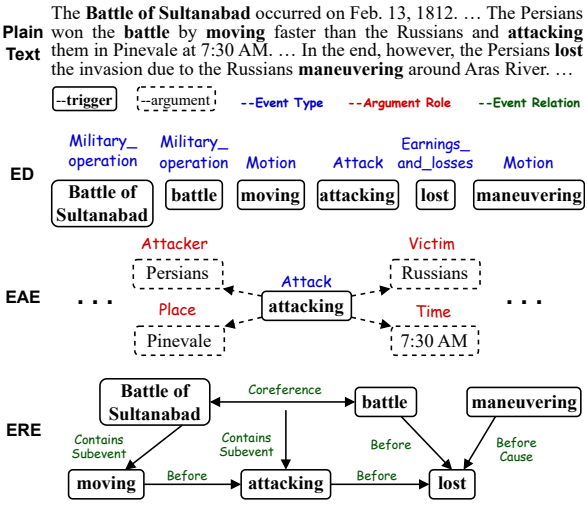

Figure 1: An illustration for the event understanding tasks, including event detection (ED), event argument extraction (EAE), and event relation extraction (ERE).

In recent years, event understanding research has grown rapidly (Ma et al., 2022; Wang et al., 2022; Yue et al., 2023; Huang et al., 2023), and multiple practical systems (Wadden et al., 2019; Lin et al., 2020; Zhang et al., 2020; Du et al., 2022; Zhang et al., 2022) have been developed. However, as shown in Table 1, existing systems exhibit several non-negligible issues: (1) **Incomprehensive Tasks**. Existing systems mainly focus on the two EE subtasks and rarely cover the whole event understanding pipeline with ERE tasks. The notable exception EventPlus (Ma et al., 2021) merely cov-

[*] Equal contribution.
[†] Corresponding author: L.Hou
[1] https://github.com/THU-KEG/OmniEvent
[2] https://omnievent.xlore.cn/

Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 508–517
December 6-10, 2023

ers the temporal relations. (2) **Limited Support for Redevelopment and Evaluation**. Most of the existing event understanding systems are highly integrated and not extensible, which means users cannot easily develop new models within their frameworks. Especially considering the recent rise of large language models (LLMs)[3], adequate support for LLMs is urgent but often missing. Moreover, the complicated data processing and evaluation details often lead to inconsistent and unfair evaluation results (Peng et al., 2023), but existing systems do not pay much attention to evaluations.

To address these issues, we develop OmniEvent, a comprehensive, fair, and easy-to-use toolkit for event understanding, which has three main features: (1) **Comprehensive Support for Task, Model, and Dataset.** OmniEvent supports end-to-end event understanding from plain texts, i.e., all the ED, EAE, and ERE tasks. For ED and EAE, we classify the mainstream methods into four paradigms, including classification, sequence labeling, span prediction, and conditional generation. We implement various representative methods for each paradigm. For ERE, we provide a unified modeling framework and implement a basic pairwise classification method (Wang et al., 2022). We also cover the preprocessing of 15 widely-used English and Chinese datasets. (2) **Fair Evaluation.** As found in Peng et al. (2023), there are three major pitfalls hidden in EE evaluation, including data processing discrepancy, output space discrepancy, and absence of pipeline evaluation. OmniEvent implements all the proposed remedies to help users avoid them. Specifically, we implement unified pre-processing for all the datasets and a method to convert the predictions of different paradigms into a unified space. OmniEvent also provides unified prediction triggers of supported datasets for fair pipeline comparisons. (3) **Easy-to-Use for Various Needs.** We design a modular and extensible framework for OmniEvent, which appeals to users with various needs. We provide several off-the-shelf models that can be easily deployed and used by users interested in applications. Model developers and researchers can train implemented methods within several lines of code or customize their own models and evaluate them. By integrating Transformers (Wolf et al., 2020) and DeepSpeed (Rasley et al., 2020), OmniEvent also supports efficiently fine-tuning LLMs as backbones.

[3]The definition of LLM is vague. Here we use "LLM" to refer to models with more than 10 billion parameters.

System	EE	ERE	#Supported Models	#Supported Datasets	LLM Support
DYGIE	✓	✗	1	1	✗
OneIE	✓	✗	1	4	✗
OpenUE	✓	✗	1	2	✗
EventPlus	✓	✓	1	N/A	✗
FourIE	✓	✗	1	N/A	✗
RESIN-11	✓	✗	1	N/A	✗
DeepKE	✓	✗	2	1	✓
OmniEvent	✓	✓	>20	15	✓

Table 1: Comparisons between OmniEvent and other event understanding systems. The number of supported models and datasets only includes those of event understanding tasks. N/A denotes that the system is an integrated service and does not process benchmark datasets. For OmniEvent, the module combination enables many possible models and 20 is the number of models we have tested for usability.

To demonstrate the effectiveness of OmniEvent, we present the results of several implemented methods on widely-used benchmarks. We also conduct experiments with models at different scales and show that fine-tuning LLMs helps achieve better event understanding results. We hope OmniEvent could facilitate the research and applications of event understanding.

2 Related Work

With the advancement of research in NLP, various toolkits or systems for event understanding have been developed. They tend to focus on developing advanced EE systems to achieve improved results on public benchmarks (Wadden et al., 2019; Lin et al., 2020; Nguyen et al., 2021) or perform robustly in real-world scenarios (Vossen et al., 2016; Du et al., 2022). However, these toolkits or systems, designed based on a specific EE model, do not support comprehensive implementations of EE models and are inconvenient for secondary development. There is also some work that has meticulously designed user-friendly algorithmic frameworks (Zhang et al., 2020, 2022), which are convenient for usage and secondary development. However, they are not specifically designed for event understanding, hence the corresponding support is limited. EventPlus (Ma et al., 2021) is the only work supporting the entire event understanding pipeline but it only supports temporal relation extraction and does not provide comprehensive implementations of event understanding models. Moreover, existing work also neglects the discrepan-

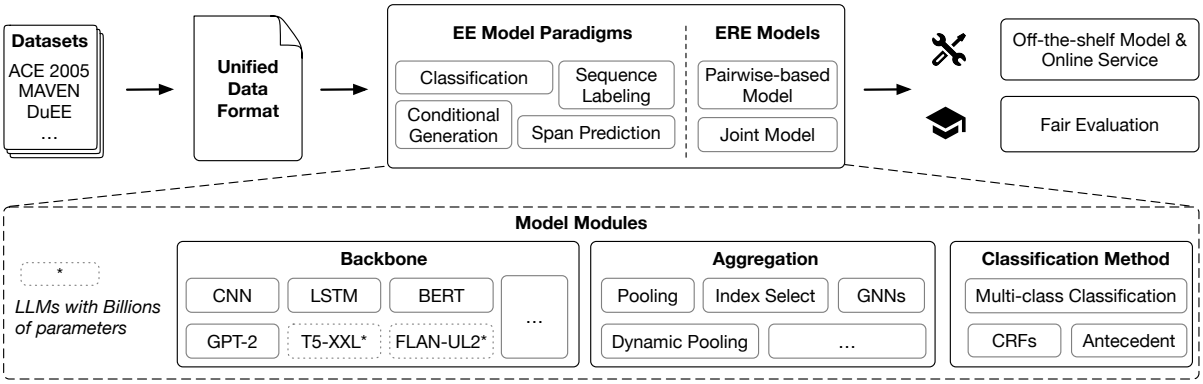

Figure 2: Overview of the OmniEvent toolkit. OmniEvent can serve as a system offering event understanding services to users, while also serving as a toolkit for researchers in model development and evaluation. OmniEvent provides pre-processing scripts for widely-used datasets and converts the datasets into a unified data format. OmniEvent provides modular components and users can easily develop a new model based on the components. OmniEvent also supports large language models (T5-XXL (Raffel et al., 2020) and FLAN-UL2 (Tay et al., 2023)).

cies in EE evaluation as mentioned in Peng et al. (2023), which may result in unfair comparison. Finally, in the era of LLMs, existing work (except for DeepKE) also lacks support for LLMs.

Considering the mentioned issues, we present OmniEvent, a comprehensive, fair, and easy-to-use toolkit for event understanding. Compared to other systems in Table 1, OmniEvent supports the entire event understanding pipeline and comprehensively implements various models. OmniEvent also supports efficient fine-tuning and inference of LLMs. Meanwhile, OmniEvent provides respective remedies for eliminating the discrepancies as mentioned in Peng et al. (2023). With a modular implementation and several released off-the-shelf models, OmniEvent is user-friendly and easy to use.

3 The OmniEvent Toolkit

We introduce the overview (§ 3.1) and main features of OmniEvent (§§ 3.2 to 3.4), as well as an online demonstration (§ 3.5) powered by OmniEvent.

3.1 Overview

The overall architecture of OmniEvent is illustrated in Figure 2. OmniEvent provides a data pre-processing module for unified pre-precessing. Users can either use the supported datasets or customize their own datasets. After pre-processing, OmniEvent provides a flexible modular framework for model implementation. OmniEvent abstracts and disassembles the mainstream models into three basic modules and implements the basic modules in a highly encapsulated way. By combining our provided modules or implementing their own modules, users can easily assemble a model. OmniEvent reproduces several widely-used models in this way. Finally, OmniEvent provides a fair evaluation protocol to convert predictions of different models into a unified and comparable output space.

3.2 Comprehensive Support

OmniEvent implements the entire event understanding pipeline, i.e., all the ED, EAE, and ERE tasks, and can serve as a one-stop event understanding platform. Furthermore, OmniEvent provides comprehensive coverage of models and datasets.

Models OmniEvent comprehensively implements representative models for ED, EAE, and ERE. For ED and EAE, OmniEvent covers four mainstream method paradigms, which contain: (1) classification methods, including DMCNN (Chen et al., 2015), DMBERT (Wang et al., 2019), and CLEVE (Wang et al., 2021b), which classify event or argument candidates into appropriate types, (2) sequence labeling methods, including BiLSTM+CRF (Wang et al., 2020b) and BERT+CRF (Wang et al., 2020b), which labels the sequences with the BIO format, (3) span prediction method, including EEQA (Du and Cardie, 2020), which predicts the boundaries of event and argument spans, (4) conditional generation method, including Text2Event (Lu et al., 2021), which directly generates the answers. Moreover, as shown in Figure 2, OmniEvent implements various basic modules and the users can easily combine different modules to build new models, e.g., combining GPT-2 (Radford et al., 2019) and CRF (Lafferty et al., 2001) (GPT-2+CRF). For event relation ex-

EE	ACE 2005 (Walker et al., 2006), TAC KBP (Ellis et al., 2014, 2015, 2016; Getman et al., 2017), RichERE (Song et al., 2015), MAVEN (Wang et al., 2020b), *ACE 2005 (zh)* (Walker et al., 2006), *LEVEN* (Yao et al., 2022), *DuEE* (Li et al., 2020), *FewFC* (Zhou et al., 2021)
ERE	MAVEN-ERE (Wang et al., 2022), ACE 2005 (Walker et al., 2006), TB-Dense (Chambers et al., 2014), MATRES (Ning et al., 2018b), TCR (Ning et al., 2018a), CausalTB (Mirza et al., 2014), EventStoryLine (Caselli and Vossen, 2017), HiEve (Glavaš et al., 2014)

Table 2: Currently supported datasets in OmniEvent. *Italics* represent Chinese datasets.

```python
from OmniEvent import convert_SL, convert_SP, convert_CG

text = "City A suffers a terrorist attack in 2021 ."
tokens = text.split()
events = [{
    "type": "attack",
    "trigger": "terrorist attack",
    "offset": [4, 6]
}]

# predictions generated by users
predictions_SL = [O, O, O, B-Attack, I-Attack, I-Attack, O,
      O, O]
# obtain comparable results
results = convert_SL([predictions_SL], [events], [tokens])

predictions_SP = [{"offset": [3, 6], "type": "attack"}]
results = convert_SP([predictions_SP], [events], [tokens])

# without offsets
predictions_CG = [{"trigger": "a terrorist attack", "type":
      "attack"}]
results = convert_CG([predictions_CG], [events], [tokens])
```

Code 1: Example for converting the sequence labeling, span prediction, and conditional generation predictions into a unified output space.

traction, OmniEvent implements a unified pairwise relation extraction framework. Especially for the event coreference resolution task, OmniEvent develops an antecedent ranking method. As extracting different relations (causal, temporal) may benefit each other (Wang et al., 2022), we develop a joint event relation extraction model in OmniEvent.

Datasets As shown in Table 2, OmniEvent includes various widely-used Chinese and English event understanding datasets, covering general, legal, and financial domains. For each included dataset, we provide a pre-processing script to convert the dataset into a unified format, as shown in appendix A. For datasets with different pre-processing scripts, e.g., ACE 2005, OmniEvent provides all the mainstream scripts for users.

3.3 Fair Evaluation

As discussed in Peng et al. (2023), there exist several pitfalls in EE evaluation that significantly influence the fair comparison of different models. They are in three aspects: data-preprocessing discrepancy, output space discrepancy, and absence of pipeline evaluation. OmniEvent proposes remedies for eliminating them.

Specify data pre-processing As the data pre-processing discrepancy mainly comes from using different processing options, OmniEvent provides all the widely-used data pre-processing scripts. Users only need to specify the pre-processing script for comparable results with previous studies.

Standardize output space As suggested in Peng et al. (2023), OmniEvent provides several easy-to-use functions to convert the predictions of different models into a unified output space. Code 1 shows the conversion codes of sequence labeling, span prediction, and conditional generation predictions for event detection. Users can easily utilize the functions to obtain fair and comparable results.

Pipeline evaluation The pipeline evaluation requires conducting EAE based on predicted triggers. Therefore, the results of EAE models are comparable only when using the same predicted triggers. OmniEvent provides a unified set of predicted triggers for widely-used datasets. Specifically, OmniEvent leverages CLEVE (Wang et al., 2021b), an advanced ED model, to predict triggers for widely-used EE datasets: ACE 2005, KBP 2016, KBP 2017, and RichERE.

3.4 Easy-to-Use

OmniEvent is designed to be user-friendly and easy to use. Specifically, OmniEvent incorporates the following designs.

Easy start with off-the-shelf models OmniEvent provides several off-the-shelf models for

```python
from OmniEvent.infer import infer
# input text
text = "U.S. and British troops were moving on the strategic
      southern port city of Basra Saturday after a massive
      aerial assault pounded Baghdad at dawn"
# event detection
ed_results = infer(text=text, task="ED")
# end-to-end event extraction
ee_results = infer(text=text, task="EE")
# end-to-end event understanding
# event extraction & relation extraction
all_results = infer(text=text, task="EE & ERE")
```

Code 2: Example of using inference interface and off-the-shelf models for event understanding.

event understanding. Specifically, we train a multilingual T5 (Xue et al., 2021) for ED and EAE on the collection of included EE datasets, respectively. And we train a joint ERE model based on RoBERTa (Liu et al., 2019) on the training set of MAVEN-ERE. As shown in Code 2, OmniEvent provides an interface for inference and users can easily use these models in their applications with a few lines of code.

Modular implementation As shown in Figure 2, OmniEvent abstracts and disassembles the mainstream models into basic modules. The backbone module implements various text encoders, such as CNN (Krizhevsky et al., 2012) and BERT (Devlin et al., 2019), to encode plain texts into low-dimension dense vectors. The backbone module also supports LLMs such as T5-XXL (Raffel et al., 2020) and FLAN-UL2 (Tay et al., 2023). The aggregation module includes various aggregation operations, which aggregate and convert the dense vectors into representations of events, arguments, and relations. The classification module projects the representations into distributions of classification candidates. With the highly modular implementation, users can easily combine the basic modular components to develop new models.

Efficient support for LLMs OmniEvent is built upon Huggingface's Transformers (Wolf et al., 2020) and DeepSpeed (Rasley et al., 2020), an efficient deep learning optimization library. With the built-in DeepSpeed support, OmniEvent can be used to train and infer LLMs efficiently with only modifications of the startup shell scripts.

3.5 Online Demonstration

Besides the OmniEvent toolkit, we also develop an online demonstration system[4] powered by OmniEvent. We train and deploy a multilingual T5_{BASE} model for EE and a $\text{RoBERTa}_{\text{BASE}}$ model for event relation extraction. The website example is shown in Figure 3. The online system supports EE based on various English and Chinese classification schemata and ERE based on the MAVEN-ERE schema. The website mainly contains three parts. The input part includes a text entry field and several options. Users can choose the language, task, and ontology (i.e., classification schema) for event understanding. The results of EE are shown in the output field with extracted triggers and arguments highlighted. The results of ERE are shown as an event knowledge graph, where a node is an event and an edge is an identified relation between events. The example in Figure 3 shows the results of end-to-end event understanding (ED, EAE, and ERE) from the input plain text.

[4] https://omnievent.xlore.cn/

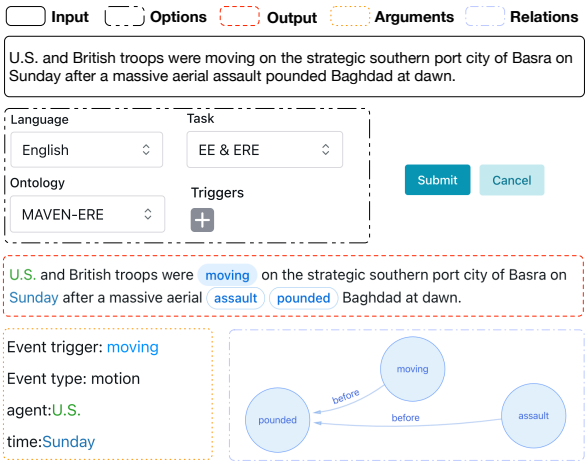

Figure 3: Example of the online demonstration. We re-arrange the layout of the website for a compact presentation. Better visualization in color.

Task	Dataset	CLS	SL	SP	CG
ED	ACE 2005	68.6	68.6	71.0	66.0
	RichERE	51.4	50.1	50.4	51.4
	MAVEN	68.6	68.6	68.1	61.9
	ACE 2005 (ZH)	75.8	75.9	73.5	71.6
	LEVEN	85.2	84.7	84.3	81.4
	FewFC	67.2	62.3	59.0	71.3
EAE	ACE 2005	58.7	49.4	40.1	45.7
	RichERE	68.3	59.7	24.3	24.9
	ACE 2005 (ZH)	73.1	67.9	35.4	49.0
	FewFC	68.7	59.8	46.7	53.7

Table 3: Experimental results (F1,%) of implemented EE models in OmniEvent on various EE datasets. CLS: Classification; SL: Sequence labeling; SP: Span prediction; CG: Conditional generation. We evaluate the representative models: DMBERT, BERT+CRF, EEQA, and Text2Event for CLS, SL, SP, and CG, respectively.

4 Evaluation

In this section, we conduct empirical experiments to evaluate the effectiveness of the OmniEvent toolkit on widely-used datasets.

4.1 Event Extraction

We evaluate the performance of representative EE models implemented in OmniEvent on various widely-used datasets. All the models are evaluated using the unified evaluation protocol, i.e., the

Relation Type	Dataset	P	R	F1
Coreference	ACE 2005	94.5	81.7	87.7
	MAVEN-ERE	97.9	98.5	98.2
Temporal	TB-Dense	67.9	54.0	60.2
	MATRES	87.2	93.8	90.4
	TCR	78.3	78.3	78.3
	MAVEN-ERE	53.3	61.4	57.1
Causal	CausalTB	100.0	50.0	66.7
	EventStoryLine	19.5	25.8	22.2
	MAVEN-ERE	36.0	26.4	30.5
Subevent	HiEve	21.4	13.4	16.5
	MAVEN-ERE	30.8	24.3	27.1

Table 4: Experimental results (%) of the implemented pairwise-based ERE model in OmniEvent on various ERE datasets. The backbone is $\text{RoBERTa}_{\text{BASE}}$. The evaluation metric for coreference is B-cubed (Bagga and Baldwin, 1998).

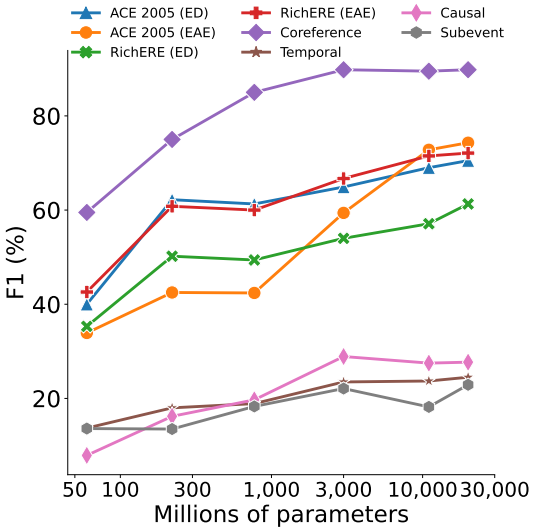

Figure 4: Experimental results of models at different scales on all event understanding tasks.

output space is standardized and the results of EAE are from pipeline evaluation. The pre-processing script for ACE 2005 is the same as in Wadden et al. (2019). For EEQA, we utilize the same prompts as in the original paper for ACE 2005 and manually curate prompts for all the other datasets. The results of event detection and event argument extraction are shown in Table 3. The results demonstrate the effectiveness of OmniEvent, which achieves similar performance compared to their original implementations. OmniEvent provides all the experimental configuration files in the YAML format, which records all the hyper-parameters. Users can easily reproduce the results using the corresponding configuration files.

4.2 Event Relation Extraction

We also conduct empirical experiments to evaluate the performance of ERE models developed in OmniEvent on various widely-used datasets. As shown in Table 4, the results are on par or slightly better than the originally reported results in Wang et al. (2022), which demonstrates the validity of ERE models in OmniEvent. We also provide configuration files containing all the hyper-parameter settings for reproduction.

4.3 Experiments using LLMs

OmniEvent supports efficient fine-tuning and inference for LLMs. To examine the effectiveness and validity of LLMs support in OmniEvent and investigate the performance of models at different scales, we train a series of models on several datasets. Specifically, for ED and EAE, we fine-tune FLAN-T5 (Wei et al., 2022) (from Small to XXL) and FLAN-UL2 (Tay et al., 2023), an LLM with 20 billion parameters on ACE 2005 and RichERE. For ERE, due to the lack of encoder-only LLMs, we use the same models as ED and EAE. We convert the ERE task into a sequence generation task. All the experiments are run on Nvidia A100 GPUs. Fine-tuning FLAN-UL2 on ACE 2005 consumes only about 25 GPU hours, which demonstrates the efficiency of LLMs support in OmniEvent. The results are shown in Figure 4. We can observe that larger models perform better and FLAN-UL2 achieves remarkable performance on ACE 2005 and RichERE datasets, which demonstrates the validity of LLMs support in OmniEvent. We can also notice that the results of ERE are much worse than the results in Table 4, which may be due to the extremely long contexts and complex output space of the ERE task. We hope the findings based on OmniEvent can inspire future research on how to better leverage LLMs for event understanding.

5 Conclusion and Future Work

In the paper, we present OmniEvent, a comprehensive, fair, and easy-to-use toolkit for event understanding. With the comprehensive and modular implementation, OmniEvent can help researchers and developers conveniently develop and deploy models. OmniEvent also releases several off-the-shelf models and deploys an online system for enhancing the applications of event understanding models. In the future, we will continually maintain OmniEvent to support more models and datasets.

Limitations

The major limitations of OmniEvent are threefold: (1) OmniEvent currently does not support document-level event extraction models and datasets, such as RAMS (Ebner et al., 2020) and WikiEvents (Li et al., 2021). OmniEvent also lacks support for a wider range of ERE models, such as constrained loss (Wang et al., 2020a) and ILP inference (Han et al., 2019). In the future, we will continue to maintain OmniEvent to support a broader range of models and datasets. (2) OmniEvent currently only supports two languages, Chinese and English, and does not yet support event relation extraction in Chinese. This might constrain the widespread usage of the OmniEvent toolkit. In the future, OmniEvent will support more languages. (3) Due to the limitations of training data and used models, the performance of our released model in practical applications is limited, especially in schemata and domains outside of the training data, such as the biomedical field. In the future, we will collect more training data and utilize advanced methods to develop more powerful and general models for event understanding.

Ethical Considerations

We will discuss the ethical considerations and broader impact of this work here: (1) **Intellectual property.** OmniEvent is open-sourced and released under MIT license[5]. We adhere to the original licenses for all datasets and models used. Regarding the issue of data copyright, we do not provide the original data and we only provide processing scripts for the original data. (2) **Environmental Impact.** The experiments are conducted on the Nvidia A100 GPUs and consume approximately 350 GPU hours. This results in a substantial amount of carbon emissions, which incurs a negative influence on our environment (Strubell et al., 2019). (3) **Intended Use.** OmniEvent can be utilized to provide event understanding services for users, and it can also serve as a toolkit to assist researchers in developing and evaluating models. (4) **Misuse risks.** OmniEvent **should not** be utilized for processing and analyzing sensitive or uncopyrighted data. The output of OmniEvent is determined by the input text and **should not** be used to support financial or political claims.

[5] https://opensource.org/license/mit

Acknowledgements

This work is supported by a grant from the Institute for Guo Qiang, Tsinghua University (2019GQB0003) and the NSFC Youth Project (62006136). The authors thank all the anonymous reviewers for their detailed and valuable comments and suggestions. The authors also thank all the valuable issues on our GitHub repository.

References

Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of COLING-ACL*, pages 79–85. Morgan Kaufmann Publishers / ACL.

Tommaso Caselli and Piek Vossen. 2017. The event StoryLine corpus: A new benchmark for causal and temporal relation extraction. In *Proceedings of the Events and Stories in the News Workshop*, pages 77–86.

Nathanael Chambers, Taylor Cassidy, Bill McDowell, and Steven Bethard. 2014. Dense event ordering with a multi-pass architecture. *Transactions of the Association for Computational Linguistics*, 2:273–284.

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.

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.

Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2015. Deep learning for event-driven stock prediction. In *Proceedings of IJCAI*.

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

Xinya Du, Zixuan Zhang, Sha Li, Pengfei Yu, Hongwei Wang, Tuan Lai, Xudong Lin, Ziqi Wang, Iris Liu, Ben Zhou, et al. 2022. Resin-11: Schema-guided event prediction for 11 newsworthy scenarios. In *Proceedings of NAACL-HLT: System Demonstrations*, pages 54–63.

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*.

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*.

Goran Glavaš, Jan Šnajder, Marie-Francine Moens, and Parisa Kordjamshidi. 2014. HiEve: A corpus for extracting event hierarchies from news stories. In *Proceedings of LREC*, pages 3678–3683.

Rujun Han, Qiang Ning, and Nanyun Peng. 2019. Joint event and temporal relation extraction with shared representations and structured prediction. In *Proceedings of EMNLP-IJCNLP*, pages 434–444.

Quzhe Huang, Yutong Hu, Shengqi Zhu, Yansong Feng, Chang Liu, and Dongyan Zhao. 2023. More than classification: A unified framework for event temporal relation extraction. In *Proceedings of ACL*.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Proceedings of NeurIPs*, pages 1106–1114.

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.

Sha Li, Heng Ji, and Jiawei Han. 2021. 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. 2020. DuEE: A large-scale dataset for chinese event extraction in real-world scenarios. In *Proceedings of NLPCC*, pages 534–545.

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.

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. *CoRR*, abs/1907.11692.

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.

Mingyu Derek Ma, Jiao Sun, Mu Yang, Kung-Hsiang Huang, Nuan Wen, Shikhar Singh, Rujun Han, and Nanyun Peng. 2021. Eventplus: A temporal event understanding pipeline. In *Proceedings of NAACL-HLT*, pages 56–65.

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*.

Paramita Mirza, Rachele Sprugnoli, Sara Tonelli, and Manuela Speranza. 2014. Annotating causality in the TempEval-3 corpus. In *Proceedings of the EACL 2014 Workshop on Computational Approaches to Causality in Language (CAtoCL)*, pages 10–19.

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 NAACL-HLT*, pages 27–38.

Qiang Ning, Zhili Feng, Hao Wu, and Dan Roth. 2018a. Joint reasoning for temporal and causal relations. In *Proceedings of ACL*, pages 2278–2288.

Qiang Ning, Hao Wu, and Dan Roth. 2018b. A multi-axis annotation scheme for event temporal relations. In *Proceedings of ACL*, pages 1318–1328.

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners.

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. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of ACM SIGKDD*, KDD '20, page 3505–3506.

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

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of ACL*, pages 3645–3650.

Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Dara Bahri, Tal Schuster, Huaixiu Steven Zheng, Denny Zhou, Neil Houlsby, and Donald Metzler. 2023. UL2: Unifying language learning paradigms. In *Proceedings of ICLR*.

Piek Vossen, Rodrigo Agerri, Itziar Aldabe, Agata Cybulska, Marieke van Erp, Antske Fokkens, Egoitz Laparra, Anne-Lyse Minard, Alessio Palmero Aprosio, German Rigau, Marco Rospocher, and Roxane Segers. 2016. Newsreader: Using knowledge resources in a cross-lingual reading machine to generate more knowledge from massive streams of news. *Knowl. Based Syst.*, 110:60–85.

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.

Haoyu Wang, Muhao Chen, Hongming Zhang, and Dan Roth. 2020a. Joint constrained learning for event-event relation extraction. In *Proceedings of EMNLP*, pages 696–706.

Shichao Wang, Xiangrui Cai, Hongbin Wang, and Xiaojie Yuan. 2021a. Incorporating circumstances into narrative event prediction. In *Findings of EMNLP*, pages 4840–4849.

Xiaozhi Wang, Yulin Chen, Ning Ding, Hao Peng, Zimu Wang, Yankai Lin, Xu Han, Lei Hou, Juanzi Li, Zhiyuan Liu, et al. 2022. MAVEN-ERE: A unified large-scale dataset for event coreference, temporal, causal, and subevent relation extraction. In *Proceedings of EMNLP*, pages 926–941.

Xiaozhi Wang, Xu Han, Zhiyuan Liu, Maosong Sun, and Peng Li. 2019. 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. 2020b. MAVEN: A Massive General Domain Event Detection Dataset. In *Proceedings of EMNLP*, pages 1652–1671.

Ziqi Wang, Xiaozhi Wang, Xu Han, Yankai Lin, Lei Hou, Zhiyuan Liu, Peng Li, Juanzi Li, and Jie Zhou. 2021b. CLEVE: Contrastive Pre-training for Event Extraction. In *Proceedings of ACL-IJCNLP*, pages 6283–6297.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. In *Proceedings of ICLR*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of EMNLP: System Demonstrations*, pages 38–45.

Susmitha Wunnava, Xiao Qin, Tabassum Kakar, Cansu Sen, Elke A Rundensteiner, and Xiangnan Kong. 2019. Adverse drug event detection from electronic health records using hierarchical recurrent neural networks with dual-level embedding. *Drug safety*, 42:113–122.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of NAACL-HLT*, pages 483–498.

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.

Zhenrui Yue, Huimin Zeng, Mengfei Lan, Heng Ji, and Dong Wang. 2023. Zero-and few-shot event detection via prompt-based meta learning. In *Proceedings of ACL*.

Ningyu Zhang, Shumin Deng, Zhen Bi, Haiyang Yu, Jiacheng Yang, Mosha Chen, Fei Huang, Wei Zhang, and Huajun Chen. 2020. Openue: An open toolkit of universal extraction from text. In *Proceedings of EMNLP: system demonstrations*, pages 1–8.

Ningyu Zhang, Xin Xu, Liankuan Tao, Haiyang Yu, Hongbin Ye, Shuofei Qiao, Xin Xie, Xiang Chen, Zhoubo Li, and Lei Li. 2022. DeepKE: A deep learning based knowledge extraction toolkit for knowledge base population. In *Proceedings of EMNLP: System Demonstrations*, pages 98–108.

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*.

Appendices

A Unified Data Format

An instance converted to the unified data format is shown in Code 3. The data format comprehensively records all the event-related information: triggers, arguments, and coreference, temporal, causal, and subevent relations.

```
{ # one instance
    "id": "instance.001.01",
    "text": "U.S. and British troops were moving on the strategic southern port city of Basra Saturday on Sunday after a massive
      aerial assault pounded Baghdad at dawn .",
    "events": [
        {
            "type": "attack",
            "triggers": { # triggers that have a coreference relation with each other
                "id": "trigger1",
                "trigger_word": "assault",
                "offset": [22, 23],
                "arguments": [
                    {"mention": "U.S.", "offset": [0, 1], "role": "attacker"},
                    {"mention": "British", "offset": [1 ,2], "role": "attacker"},
                    {"mention": "dawn", "offset": [26, 27], "role": "time"}
                ]
            }
        },
        {
            "type": "motion",
            "triggers": {
                "id": "trigger2",
                "trigger_word": "moving",
                "offset": [5, 6],
                "arguments": [
                    {"mention": "Sunday", "offset": [17, 18], "role": "time"},
                ]
            }
        },
        # .....
    ],
    "event-relations": {
        "temporal": [
            ["trigger1", "before", "trigger2"]
        ],
        "causal": [],
        "subevent": []
    }
}
```

Code 3: An instance with the unified data format. The triggers that recorded in an item of "events" have a coreference relation with each other.