Emancipating Event Extraction from the Constraints of Long-Tailed Distribution Data Utilizing Large Language Models

Zhigang Kan[1], **Liwen Peng**[1], **Fan Yang**[2], **Linbo Qiao**[1]*
[1]National University of Defense Technology, Changsha, Hunan, China
[2]Changsha Environmental Protection Vocational College, Changsha, Hunan, China
{kanzhigang13, pengliwen13, qiao.linbo}@nudt.edu.cn,yangfan0327@yeah.net

Abstract

Event **E**xtraction (EE) is a challenging task that aims to extract structural event-related information from unstructured text. Traditional methods for EE depend on manual annotations, which are both expensive and scarce. Furthermore, the existing datasets mostly follow the long-tail distribution, severely hindering the previous methods of modeling tail types. Two techniques can address this issue: transfer learning and data generation. However, the existing methods based on transfer learning still rely on pre-training with a large amount of labeled data in the source domain. Additionally, the quality of data generated by previous data generation methods is difficult to control. In this paper, leveraging **L**arge **L**anguage **M**odels (LLMs), we propose novel methods for event extraction and generation based on dialogues, overcoming the problems of relying on source domain data and maintaining data quality. Specifically, this paper innovatively transforms the EE task into multi-turn dialogues, guiding LLMs to learn event schemas from historical dialogue information and output structural events. Furthermore, we introduce a novel LLM-based method for generating high-quality data, significantly improving traditional models' performance with various paradigms and structures, especially on tail types. Adequate experiments on real-world datasets demonstrate the effectiveness of the proposed event extraction and data generation methods.

Keywords: Event extraction, Event generation, Long-tail distribution

1. Introduction

Events are defined as occurrences including specific participants or changes in status and are pervasive in media messages across multiple modalities (Doddington et al., 2004). Event extraction endeavors to automatically identify event information from unstructured textual messages and produce a structured output. It involves identifying event types contained within the given text, locating triggers that best reflect the occurrences of events, and extracting the participants and attributes of the events while assigning them their respective roles (Chen et al., 2015a). Figure 1 provides an intuitive depiction of the process of dialogue-based EE. Given the sentence in the figure, an event extractor should automatically and accurately identify events of two types: "Die" and "Attack" and output the triggers of both events, along with the corresponding event arguments for each of the roles involved.

Traditional EE approaches (Yang et al., 2019; Wadden et al., 2019; Liu et al., 2020a; Lin et al., 2020; Nguyen et al., 2021), which are based on neural networks, rely on manual high-quality annotations to learn features for triggers and event arguments, yielding impressive results. However, this line of work suffers from the long-tail distribution of current datasets. Figure 2 displays the statistics of samples in the widely used ACE2005 English dataset, comprising 33 sub-types and 4,419 events. Notably, the samples, especially triggers, roughly

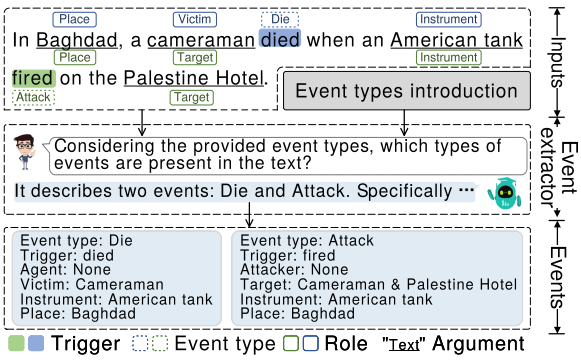

Figure 1: The diagram of the event extraction task.

follow power-law distributions, leading to the emergence of long-tail phenomena. The "Attack" type possesses the largest number of event samples, with about 1,600, while the "Pardon" type only has two events. The long-tail distribution of the dataset severely constrains the performance of traditional methods, especially on tail types such as "Pardon". In practice, due to the time-consuming and arduous nature of annotating events, it is highly expensive to manually annotate supplementary data to compensate for the shortcomings of the above datasets.

Intuitively, in order to alleviate this issue, there are two avenues that can be pursued: optimizing the model or enriching the dataset. On the model front, researchers can leverage transfer learning techniques (Huang et al., 2018; Zhang et al., 2021; Lyu

* Corresponding author

LREC-COLING 2024, pages 5644–5653

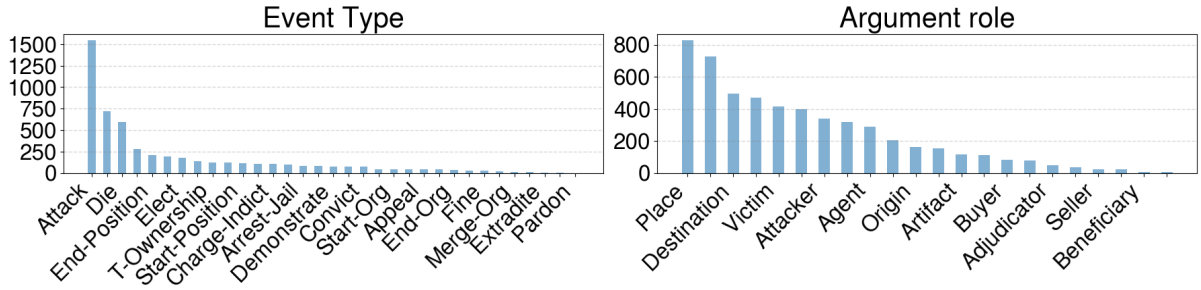

Figure 2: Statistics of ACE 2005 English Data.

et al., 2021) to train models on head-type events such as "attacks" and subsequently transfer them to tail-type events. However, traditional transfer learning methods still require pre-training on high-quality source domain data and, to some extent, still rely on manual annotations. From the data perspective, it is feasible to alleviate the problem of imbalanced sample distribution in datasets by automatically generating events of tail types. Yang et al. proposed a novel data generation approach based on argument replacement, which can generate arbitrary amounts of event data (Yang et al., 2019). Nevertheless, the adjunct token rewriting strategy they proposed to improve the smoothness of generated data may introduce a problem of role deviation. For instance, altering the adjunct token "from" in the sample "Katie drove from London" to produce the new sample "Katie drove to London" leads to a shift in the role of "London" from "origin" to "destination", introducing erroneous tags that undermine data quality.

The emergence of large language models, exemplified by ChatGPT[1], offers a potential solution to mitigate the above limitations in the text field. Based on the above intuitions, this paper explores the possibility of alleviating the limitations arising from the long-tail distribution commonly observed in current event datasets through utilizing LLMs. It includes directly leveraging the generative LLMs for EE in the zero-shot scenario, as well as utilizing the automatically generated data to enhance the performance of traditional EE models on tail types. Utilizing LLMs for EE faces two challenges: 1) formulating a means to effectively imbue the model with details of the event schemas, including the description of event types and their argument roles, and 2) guiding the model to output structural event information that programs can easily parse. Regarding data generation, it is crucial to emphasize improving the diversity and quality of provided data.

We present innovative strategies based on in-context learning to tackle the above challenges using large dialogue models. To be precise, we transform EE into dialogues and propose a **M**ulti-turn **D**ialogue-based **E**vent **E**xtraction (MDEE) method. During the dialog, our approach initially presents the LLM with instructions for delineating the scope of event types, explaining the definitions of types, and providing examples for event detection. Subsequently, based on the results of event identification, the proposed method dispatches argument extraction instructions to the LLM, sequentially extracting arguments playing specific roles. These meticulously designed instructions enable the LLM to acquire the event schema and output structural events with conventional format from the contextual environment. Moreover, unlike traditional in-context learning methods, our approach utilizes multiple rounds of historical dialogue information to delineate the relationship between triggers and event arguments during event extraction. For data generation, a three-stage approach involving decomposition, sampling and recombination is introduced to obtain a wealth of new events. Subsequently, we establish a dialogue with an advanced LLM to formulate coherent and logical sentences incorporating newly created structured events. Ultimately, we leverage the LLM to undertake a thorough rationality check and data filtration process, culminating in the enhancement of data quality.

In summary, this paper contributes as follows:

- We introduce an innovative EE method that caters to the zero-shot scenario. By adopting the multi-dialogue format, we enable LLMs to learn event schemas and produce well-organized events from historical chat records.
- We propose a novel approach for generating diverse and high-quality event data. Multiple experiments demonstrate the quality of the generated data in enhancing the performance of traditional models, particularly in extracting events with low-frequency types.

2. Related work

2.1. Event Extraction

Early EE approaches (Ahn, 2006; Liao and Grishman, 2010; Hong et al., 2011; McClosky et al.,

[1] https://openai.com/chatgpt

2011; Miwa et al., 2014) require manual feature engineering tailored for each event type and extract triggers and arguments through pattern matching. With the advent of deep neural networks, a category of methods for automatically capturing dense event features emerged (Chen et al., 2015b; Nguyen and Grishman, 2015; Nguyen et al., 2016; Sha et al., 2018). This line of work not only releases researchers from tedious feature engineering but also achieves inspiring performance with the help of high-quality annotations.

The emergence of Pre-trained Language Models (PLMs) (Liu et al., 2019; Raffel et al., 2020; Lewis et al., 2020) greatly propelled the development of event extraction techniques. Yang et al. first explored the possibility of utilizing PLMs for the EE task (Yang et al., 2019). Specifically, they regarded EE as a token classification task and constructed the model by attaching a multi-classification network to the tail of BERT(Devlin et al., 2019). Many subsequent methods followed the paradigm of attaching additional networks, which can be classified as "fine-tuning" based methods (Du and Cardie, 2020; Li et al., 2020; Chen et al., 2020; Liu et al., 2020b, 2022a). However, differences in task formats and network structures result in significant gaps between pre-training and fine-tuning, leading to the traditional fine-tuning-based EE methods having to rely heavily on large amounts of high-quality data to achieve high performance.

Many approaches based on the prompt paradigm have been proposed (Liu et al., 2023) to bridge the gaps. Such methods (Li et al., 2021; Liu et al., 2022b; Huang et al., 2022) reformulate EE into a cloze or text restoration task, which represents the pre-training tasks of PLMs while keeping the model architecture unchanged, in order to elicit the potential knowledge of the pre-trained model. Thanks to the effective utilization of the knowledge learned in the pre-training phase, prompt-based methods achieve impressive performance in data-scarce scenarios. In addition, Lu et al. proposed an end-to-end method for directly generating structured events (Lu et al., 2021). They converted EE to a sequence-to-sequence task and represented structured events as flattened textual sequences. Furthermore, unified information extraction frameworks represented by OneIE (Lin et al., 2020), FourIE (Nguyen et al., 2021), and UIE (Lu et al., 2022) jointly learn three information extraction tasks: named entity recognition, relation extraction, and EE, which benefit from potential correlations among entities, relationships, and events. CLEVE (Wang et al., 2021) and UIE are two pre-training-based event methods that first pre-learn EE-related knowledge on large-scale unlabelled data and then fine-tune their EE models on annotated data to achieve higher performance.

The conventional methods of EE rely heavily on annotated datasets, which are expensive and unevenly distributed. To tackle the challenge of sparsity in data, researchers proposed various methods targeting few-shot and zero-shot scenarios (Huang et al., 2018; Zhang et al., 2021; Lyu et al., 2021). These methods achieved promising results based on transfer learning, meta-learning and prompt-based paradigms. However, they still require high-quality annotations as supervision. Therefore, designing an event extraction method with a lower dependence on human-annotated datasets remains a worthwhile problem to explore.

2.2. Event Generation

Another technique route for alleviating problems caused by sparse data is data generation. Earlier methods (Bollacker et al., 2008; Liu et al., 2016) focused on mining additional events from knowledge bases such as FrameNet[2], which possess event-specific attributes. Subsequently, some methodologies are predicated on the assumption of remote supervision, whereby “if two entities share a relationship within a knowledge base, then all statements referencing these entities shall inherently imply their relational connection.” (Chen et al., 2017; Wang et al., 2019). They utilize knowledge bases such as FrameNet and WordNet[3] to annotate unlabeled data from publicly available sources automatically. However, in reality, this assumption introduces considerable noise into the generated samples, as not all co-occurring entities exhibit the expected relationship. Yang et al. obtained new events by replacing arguments and improving sentence smoothness through adjunct token rewriting (Yang et al., 2019). Nevertheless, despite the introduction of a scoring mechanism in their method, it still faces the issue of role deviation.

3. Dialogue-based Event Extraction

3.1. Task Formalization

The goal of event extraction consists of two parts: event detection and argument extraction, which are dedicated to automatically recognizing events and extracting corresponding event arguments from given texts, respectively. To elaborate, consider a sentence $S = \{s_1, s_2, \cdots, s_{|S|}\}$ and the event schema $\mathscr{S}$, where s_i signifies the i-th token in the sentence, $|S|$ represents the sentence's length, and $\mathscr{S}$ describes all the event types $\{T_1, T_2, \cdots\}$ to be extracted and the argument roles $\{role_1, role_2, \cdots\}$ that each type contains. Event detection aims to recognize events

[2]https://framenet.icsi.berkeley.edu
[3]https://wordnet.princeton.edu

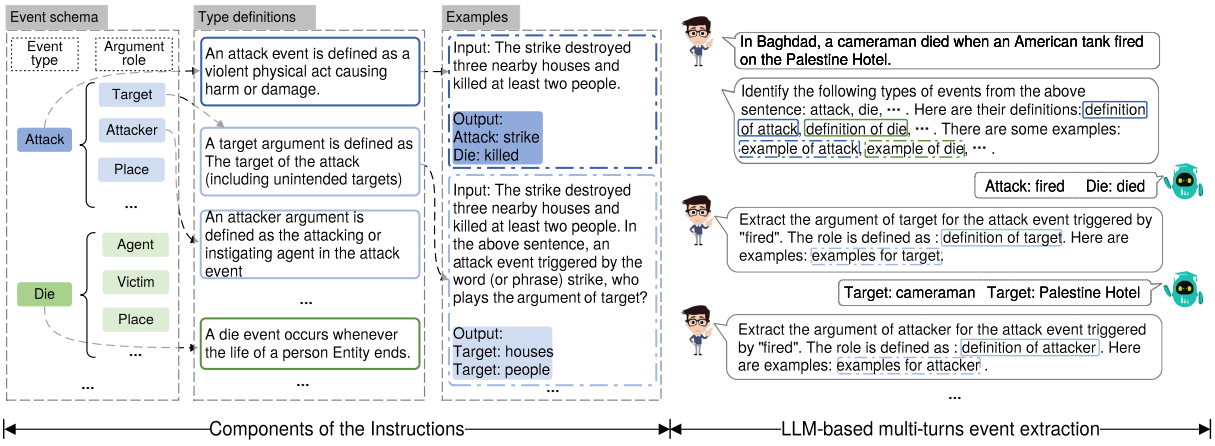

Figure 3: The overview of the proposed multi-turn dialogue-based event extraction method.

$\mathscr{E} = \{e_1, e_2, \cdots, e_{|\mathscr{E}|}\}$, where $|\mathscr{E}|$ refers to the count of events within the sentence S, with types predefined in $\mathscr{S}$. This process includes locating triggers $\text{Trigs} = \{\text{trig}_1, \text{trig}_2, \cdots, \text{trig}_{|\mathscr{E}|}\}$ that effectively symbolize the occurrence of events and classifying their respective event types. Here, Trig_i denotes the trigger for the i-th event. Concurrently, argument extraction is dedicated to identifying the arguments $\text{Args}^{e_j} = \{\text{arg}_1, \text{arg}_2, \cdots, \text{arg}_{|A|}\}$ of each event detected and attributing them with appropriate argument roles. $\text{arg}_i^{e_j}$ indicates the i-th argument participates in the j-th event and $|A|$ is the number of arguments.

3.2. EE based on Multi-turn dialogue

As depicted in Figure 3, the multi-turn dialogue-based event extraction method proposed in this paper guides the LLM to output event information in several stages. In each stage, it simulates a user sending the request, which incorporates the knowledge of the event schema and examples of trigger (or argument) extraction, to an LLM to extract triggers or event arguments based on in-context learning. Upon receiving feedback from the LLM, it automatically parses triggers or event arguments from the response. Details regarding the instruction design are introduced in Subsection 3.3.

Considering the prevalent close correlation between triggers and event arguments, as well as within arguments, the event information in historical dialogues significantly influences the generation of event information in new rounds of conversation. To effectively utilize the historical information, the event extraction method proposed in this paper incorporates multi-round historical dialogue information into the in-context learning paradigm. As shown in Algorithm 1, this method first constructs instruction I_T for trigger detection based on the given event schema. Subsequently, it obtains feedback F_T after sending it to the LLM and parses the feedback to obtain all event triggers Trigs contained in the sentence. Lastly, for each event parsed, the MDEE further extracts the event arguments corresponding to that event on a role-by-role basis.

In the argument extraction phase, MDEE iteratively extracts the event arguments for each predicted event $(\text{type}_i, \text{trig}_i)$. During the initialization stage, it sets the instruction for argument extraction as the trigger detection instruction I_T and initializes the historical feedback to the feedback from the trigger extraction phase F_T. Subsequently, the method obtains the argument roles $Roles$ corresponding to the type type_i from the event schema

Algorithm 1 EE based on Multi-turn dialogue

Input: Sentence S, Event schema $\mathscr{S}$
Output: Structural events list $\mathcal{E}$

Initiate $\mathcal{E} \leftarrow \varnothing$
Construct the event detection instruction I_T
Send I_T to LLMs and get feedback F_T
Parse event types and triggers $(Tpyes, Trigs)$ from F_T
for $(type_i, trig_i)$ in zip$(Tpyes, Trigs)$ **do**
    Initiate $Args \leftarrow \varnothing$, $R_A \leftarrow I_T$, $F_A \leftarrow F_T$
    for $role_j$ in $Roles$ **do**
        Construct the argument extraction instruction I_A for the argument role $role_j$
        $R_A \leftarrow R_A \bigoplus F_A \bigoplus I_A$
        Send R_A to LLMs and get feedback F_A
        Parse arguments $args$ from F_A
        for arg_k in $args$ **do**
            Append $\{role_j : arg_k\}$ to $Args$
        end for
    end for
    Append $(type_i, trig_i, Args)$ to $\mathcal{E}$
end for

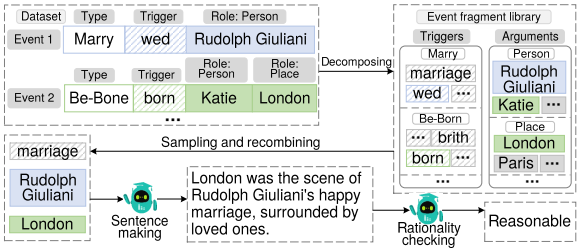

Figure 4: The overview of dialogue-based event generation.

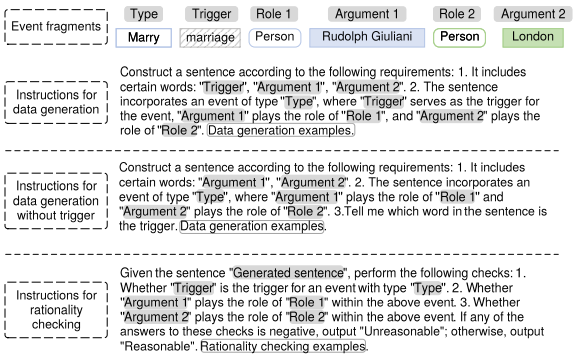

Figure 5: Instructions templates utilized in the process of event generation.

$\mathscr{S}$. For each argument role role_j, MDEE constructs a request R_A that includes historical conversation information. Specifically, the request for each round consists of the request from the previous round, the argument extraction feedback F_A from the previous round, and the instruction I_A constructed for the current argument role role_i. Our method simulates a user sending an updated request R_A with historical information to LLM and gathers event arguments from the feedback F_A. Ultimately, MDEE integrates event information from all the conversations to form the final structural events $\mathscr{E}$.

3.3. Instructions Construction

Figure 3 illustrates MDEE's integration of type (or role) definitions and examples, aiming to communicate both the event schema information and knowledge about the EE task to the LLM. Specifically, for every event type and its associated argument roles within the event schema, our instructions draw from definitions and relevant examples as laid out in annotation guidelines. The detailed contents of each component of the instructions are depicted in Figure 3. To ensure a consistent output from the LLM, an output requirement—"*Output the result directly, without any interpretation, following the output format in the examples.*"—is appended to the tail of each LLM request in both event detection and argument extraction.

4. Dialogue-based Event Generation

In addition to the zero-shot event extraction method based on multi-turn dialogue, we mitigate the long-tail limitation of existing datasets by automatically generating events with tail types from a data perspective. Figure 4 illustrates the workflow of this data generation method. Broadly speaking, our event generation approach consists of two stages: event fragment library construction and new data production. In the first stage, the proposed method first decomposes events in the dataset, yielding various types of triggers and event arguments. Subsequently, it aggregates triggers of the same type to form a trigger library. For event arguments, the method merges argument roles with similar attributes into a category, such as consolidating the roles "victim" and "person".

In the process of data generation, our method first randomly samples a trigger and several event arguments from the event fragment library based on the type of event that needs to be generated. Then, leveraging generative LLMs, we generate corresponding text based on In-context learning. Finally, we use LLMs again to validate the generated data for rationality, confirming that the sampled trigger initiates an event with the given type and event arguments play specified roles in the sentence. Additionally, in the real dataset, the quantity of certain types of events is exceedingly scarce. For these categories, besides generating databases via the above process, we also allow for the absence of triggers in sampling and utilize LLM for pinpointing the words in the generated text that best represent occurring events. Figure 5 displays the templates of the instructions employed in this section. In actual application, the characters with a grey background in the templates are replaced with corresponding contents. The examples within the templates are derived from the real dataset.

5. Experiments

5.1. Experimental Settings

5.1.1. Implementation Details

In this paper, we selected ChatGPT as the LLM for dialogue-based event extraction and generation methods. More specifically, the version of ChatGPT we used in experiments is "GPT-3.5-Turbo-16k-0613". All experiments performed in this paper were conducted on an NVIDIA RTX A6000 graphics card. For event generation, we generated 200 samples for each event type for training. In the experiment evaluating the effect of event generation

methods on alleviating the long-tail distribution limitation of the dataset, we generate 100 instances of each event type for testing. Three graduate students with outstanding skills in natural language processing were enlisted to review the generated test data, ensuring its quality.

5.1.2. Dataset

We select ACE2005, the most widely used EE dataset with a very pronounced long-tail limitation, to serve as the evaluation corpus. It contains 599 documents covering 33 event types and 22 argument roles. Currently, there exist two variants of this dataset (Lin et al., 2020): ACE-05E and ACE-05E+, with the former filtering out events whose triggers consist of multiple tokens based on the latter. In our experiments, we employ the more comprehensive ACE-05E+ version. In the zero-shot scenario, we adopt the settings established in prior works (Huang et al., 2018; Zhang et al., 2021): utilizing the events from the top 10 categories with the highest number of samples as the training set and evaluating models on the remaining instances. For data generation, we employ the same data split as mainstream approaches (Hsu et al., 2022; Lu et al., 2021). We parse the ACE2005 dataset using pre-processing procedures provided by previous researchers (Hsu et al., 2022).

5.1.3. Evaluation Metric

We follow the same evaluation metric as previous studies (Du and Cardie, 2020; Hsu et al., 2022; Lu et al., 2021): a trigger is correctly extracted if its offset and event type are the same as the ground truth. Concerning event arguments, they are considered accurate only if their event types, triggers, offsets, and argument roles align with the actual values. Consistent with prior works, we use **P**recision (P), **R**ecall (R), and **F1** Score (F1) as evaluation metrics.

5.1.4. Baseline

For zero-shot event extraction, we select four strong baseline methods for comparison: (1) "**Transfer**" (Huang et al., 2018) offers a shared semantic representation space for all event types, enabling the model to handle unseen event types. (2) "**LR-ILP**" (Zhang et al., 2021) identifies events using established tools and then maps them to their semantically closest categories through label representation. (3) "**TE/QA**" (Lyu et al., 2021) formulates zero-shot event extraction as Textual Entailment (TE) and Question Answering (QA) queries. (4) "**DEGREE**" (Hsu et al., 2022) is a prompt-based event extraction method demonstrating excellent performance in data-scarce scenarios.

Table 1: Results (%) of event extraction in the zero-shot scenario.

Model	Event Detection		
	P	R	F1
Transfer	75.5	36.3	49.1
ILP	54.1	53.1	53.6
TE/QA	-	-	41.7
DEGREE†	52.4	53.7	53.1
ChatGPT-joint	51.7	53.4	52.5
MDEE	52.0	53.8	52.8
Model	**Argument Extraction**		
	P	R	F1
Transfer	16.1	15.6	15.8
ILP	4.6	10.0	6.3
TE/QA	-	-	16.8
DEGREE†	45.1	15.3	22.8
ChatGPT-joint	15.9	16.4	16.1
MDEE	23.3	24.1	23.6

We evaluate the event generation method proposed in this paper based on three representative event extraction methods: (1) "**OneIE**" (Lin et al., 2020) is a fine-tuning-based method that applies encoder-only PLMs as its backbone, demonstrating substantial effectiveness in scenarios replete with data. (2) "**Text2event**" (Lu et al., 2021) is a generative method utilizing an encoder-to-decoder structure PLM to directly generate structural event information. (3) "**DEGREE**" (Hsu et al., 2022) is based on a decoder-only pre-trained language model. Furthermore, we also select an event generation method (Yang et al., 2019) proposed by Yang et al. as a competitive baseline.

5.2. Zero-shot Event Extraction

Table 1 shows the performance of our proposed multi-turn dialogue-based event extraction method against baseline approaches in the zero-shot scenario. Herein, "ChatGPT-joint" indicates instructing the LLM to output all information, including triggers and event arguments, in one step, which would ordinarily require multiple steps in MDEE. Compared to the baseline methods, the approach proposed in this paper achieves competitive performance in event detection and attains significant improvement in argument extraction. This demonstrates the effectiveness of the dialogue-based event extraction method proposed under the zero-shot scenario. In comparison to "ChatGPT-joint", though MDEE performs equivalent in event detection, it shows remarkable improvement in argument extraction. This attests to the efficacy of our proposed multi-turn dialogue technology in enabling the LLM to capture event arguments. Additionally, we observed noteworthy room for improvement in both event

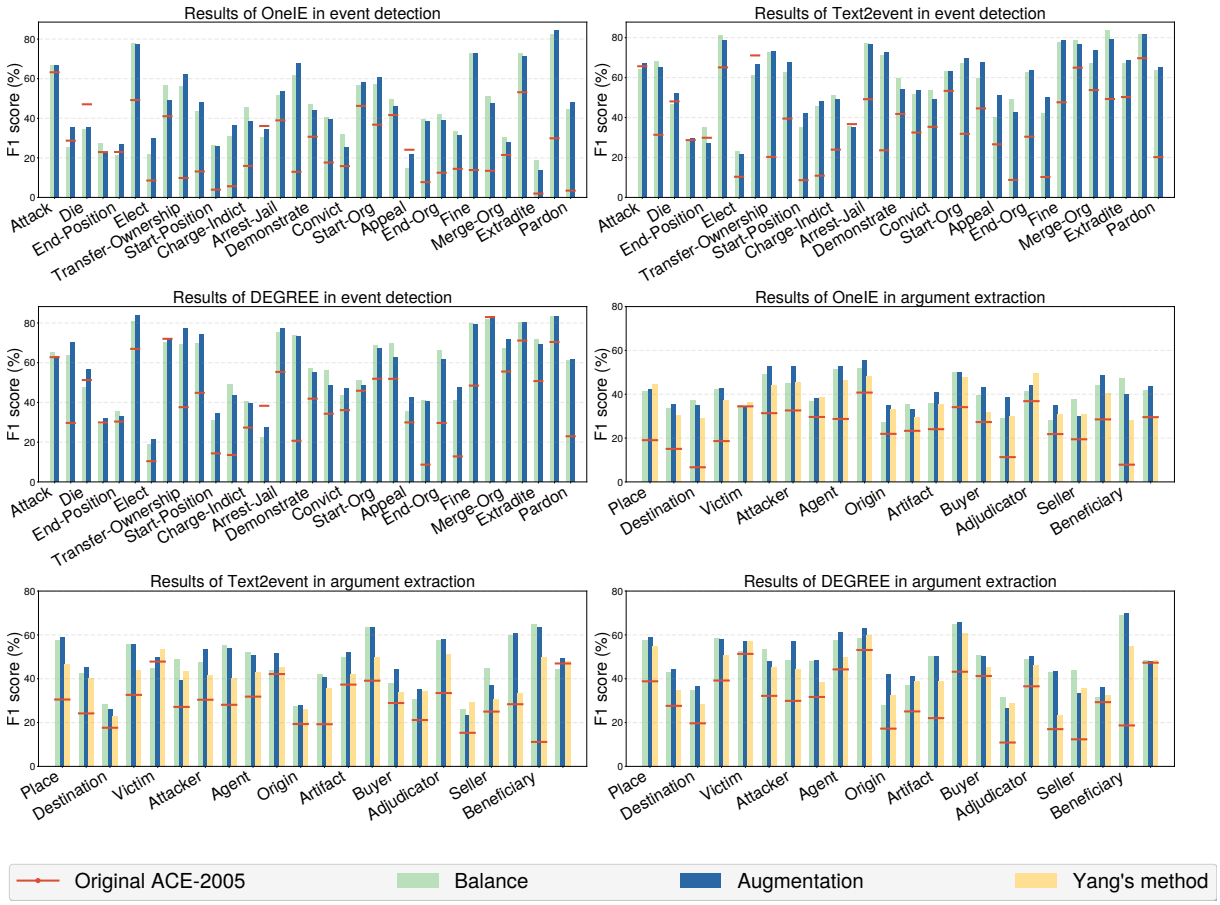

Figure 6: Comparisons of F1 scores for each category in event detection and argument extraction after improving the training set using different methods.

extraction and argument extraction for MDEE, the causes of which are analyzed in Subsection 5.4.

5.3. Event Generation

To vividly demonstrate the effectiveness of the proposed event generation method in alleviating the constraints of long-tail distribution in current datasets, we conduct comparative experiments on three representative event extraction methods, each employing different paradigms and structures. "Original ACE-2005" indicates training EE models on the initial training set of the ACE-2005 dataset. We leverage the event generation method introduced in this study to enhance the training set's data distribution via two strategies: "Balance" and "Augmentation". The "Balance" strategy ensures a constant sample size of 200 for each event type within the training set. We randomly sample 200 instances from the initial training set for categories with over 200 instances and enhance those with fewer instances using generated data. "Augmentation" involves enriching the original training set with generated data. "Yang's method" implies that we devise 200 instances for each event type using their approach to boost the argument samples in the ACE-2005 dataset.

5.3.1. Category-specific Results

Figure 6 presents comparisons of F1 scores across different categories achieved by three EE methods, assisted by different data generation methods. (1) The event extraction performances of the three methods under consideration are sensitive to the distribution of raw data and exhibit significant variability in the extraction of disparate event types. The disparity is more noticeable in methods using the fine-tuning paradigm than the prompt-based approach. (2) Data generated using the method proposed in this paper considerably mitigates the original dataset's disadvantageous long-tail distribution. Evident in Figure 6, the introduction of "Balance" or "Augmentation" significantly enhances the F1 score of each category, irrespective of the EE method employed. This improvement is markedly noticeable for event types with extremely few samples. (3) Compared to Yang's method, using our generated data brings substantial performance improvement in various categories of event arguments.

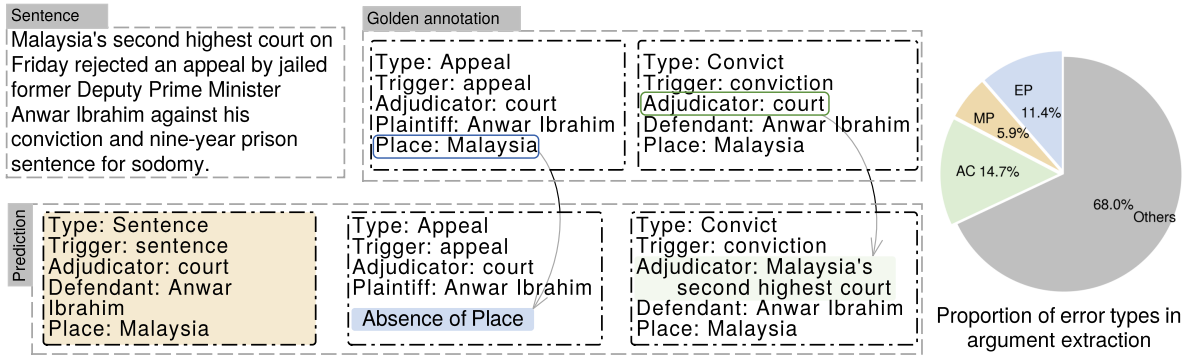

Figure 7: Schematic diagram of typical “misjudgments” in the zero-shot scenario.

Table 2: F1 scores (%) on ACE-2005 with different training data.

Data	Event Detection		
	OneIE	Text2event	DEGREE
Original ACE	37.2	37.9	40.1
Balance	68.1	60.3	60.9
Augmentation	68.6	61.8	61.8
Model	**Argument Extraction**		
	OneIE	Text2event	DEGREE
Original ACE	23.0	26.7	34.4
Yang’s method	48.2	50.4	50.7
Balance	51.8	53.3	54.0
Augmentation	58.6	52.9	52.2

5.3.2. Overall Results

Table 2 illustrates the overall performance of the three existing EE methods on balanced test sets for various data augmentation strategies. It can be observed that models trained on the original training set perform poorly on the balanced test set. The introduction of additional generated events significantly improves the performance of the existing event extraction methods. Compared to the method proposed by Yang et al., the event data generated in this study bring about more remarkable improvement. This confirms the effectiveness of the event generation method proposed in this study.

5.4. Error Analysis in Zero-shot EE

We analyzed MDEE’s output in the zero-shot scenario, discovering some “misperceptions” that, strictly speaking, cannot be categorized as errors. We classified them into the following three categories: Extra Prediction (EP), Missing Prediction (MP), and Argument Co-reference (AC), which are vividly illustrated in Figure 7 using a compelling example. Specifically, “extra prediction” refers to the instances when the MDEE outputs an event or argument that appears in the sentence but is not annotated. For instance, in the sentence in Figure 7, the “Sentence” event is not included in the “Golden annotation”, yet the remote large language model still captures it due to its powerful generative understanding. “Missing prediction” indicates an absence of elements in our method’s output that are present in the actual annotation, with no direct evidence for its appearance. For example, in the sample sentence, there is no direct evidence to suggest that “Anwar Ibrahim” appealed in “Malaysia”. “Element co-reference” refers to the incongruity between the text representation of the predicted arguments by the MDEE and their actual values in golden annotations but they essentially refer to the same entity or attribute value. The pie chart in Figure 7 shows that these three types of "misjudgments" make up a significant proportion of all errors in argument extraction. Therefore, MDEE actually possesses more substantial event extraction capabilities than reported in Table 1.

6. Conclusion

In response to the issue of event extraction being constrained by the long-tail distribution of current datasets, this paper proposes solutions from both methodological and data perspectives. Firstly, leveraging large language models, we propose a zero-shot event extraction approach based on multi-turn dialogues, which extracts triggers and arguments of each type by preserving historical dialogue information for In-context learning. Secondly, from the data perspective, a three-step approach of decomposition, sampling, and reconstruction for event generation is proposed. Compared to traditional methods, it enables the generation of semantically coherent and variously formatted event data. Comprehensive numerical experiments validate the effectiveness of our proposed methods for event extraction and generation.

7. Acknowledgements

We extend our heartfelt thanks to the reviewers, AC, SAC, and PC, for the considerable amount of time and energy they have selflessly invested into carefully reviewing our paper. Their meticulous evaluation process has been invaluable in improving our work. We would also like to express our profound appreciation to the distinguished scholars from the National University of Defense Technology. Their invaluable insight, enlightening discussions, and supportive feedback have enhanced both the quality of our research and our understanding of the topic.

8. Bibliographical References

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

Kurt D. Bollacker, Colin Evans, Praveen K. Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of SIGMOD*, pages 1247–1250.

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. 2015a. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of ACL*, pages 167–176.

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

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

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*, pages 4171–4186.

George R. Doddington, Alexis Mitchell, Mark A. Przybocki, Lance A. Ramshaw, Stephanie M. Strassel, and Ralph M. Weischedel. 2004. The automatic content extraction (ACE) program - tasks, data, and evaluation. In *Proceedings of LREC*, pages 1–4.

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

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

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*, 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.

Lifu Huang, Heng Ji, Kyunghyun Cho, Ido Dagan, Sebastian Riedel, and Clare R. Voss. 2018. Zero-shot transfer learning for event extraction. In *Proceedings of ACL*, pages 2160–2170.

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.

Fayuan Li, Weihua Peng, Yuguang Chen, Quan Wang, Lu Pan, Yajuan Lyu, and Yong Zhu. 2020. Event extraction as multi-turn question answering. In *Proceedings of EMNLP*, pages 829–838.

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

Shasha Liao and Ralph Grishman. 2010. Filtered ranking for bootstrapping in event extraction. In *Proceedings of COLING*, page 680–688.

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.

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, Wei Bi, and Xiaojiang Liu. 2020b. Event extraction as machine reading comprehension. In *Proceedings of EMNLP*, pages 1641–1651.

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

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9):1–35.

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.

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

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*, pages 2795–2806.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information extraction. In *Proceedings of ACL*, pages 5755–5772.

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 (Volume 2: Short Papers)*, pages 322–332.

David McClosky, Mihai Surdeanu, and Christopher D. Manning. 2011. Event extraction as dependency parsing for bionlp 2011. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 41–45.

Makoto Miwa, Paul Thompson, Ioannis Korkontzelos, and Sophia Ananiadou. 2014. Comparable study of event extraction in newswire and biomedical domains. In *Proceedings of ICCL*, pages 2270–2279.

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*, pages 27–38.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *Proceedings of NAACL*, 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.

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.*, 21:1–67.

Lei Sha, Feng Qian, Baobao Chang, and Zhifang Sui. 2018. Jointly extracting event triggers and arguments by dependency-bridge RNN and tensor-based argument interaction. In *Proceedings of AAAI*, pages 5916–5923.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. In *Proceedings of EMNLP*, pages 5783–5788.

Xiaozhi Wang, Xu Han, Zhiyuan Liu, Maosong Sun, and Peng Li. 2019. Adversarial training for weakly supervised event detection. In *Proceedings of NAACL*, pages 998–1008.

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.

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.

Hongming Zhang, Haoyu Wang, and Dan Roth. 2021. Zero-shot label-aware event trigger and argument classification. In *Proceedings of Findings of ACL/IJCNLP*, pages 1331–1340.