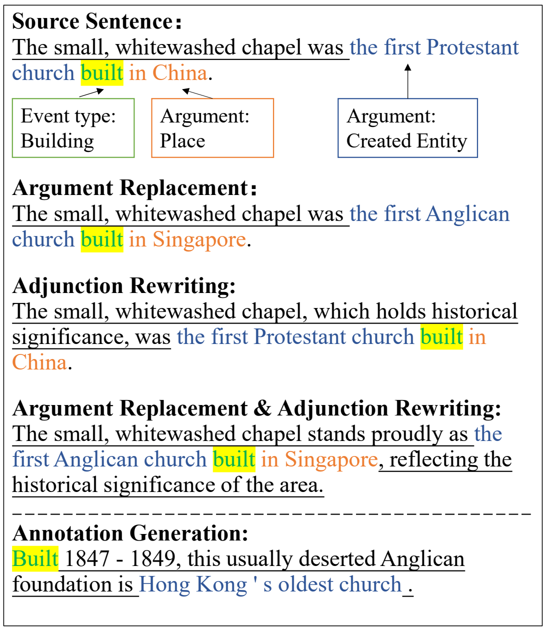
Boundary-Aware LLM Augmentation for Low-Resource Event Argument Extraction

Zhaoyue Sun[1,2], **Gabriele Pergola**[1], and **Yulan He**[2,3]
[1]Department of Computer Science, University of Warwick
[2]Department of Informatics, King's College London
[3]The Alan Turing Institute
{Zhaoyue.Sun, Gabriele.Pergola.1}@warwick.ac.uk
yulan.he@kcl.ac.uk

Abstract

Event argument extraction (EAE) is a crucial task in information extraction. However, its performance heavily depends on expensive annotated data, making data scarcity a persistent challenge. Data augmentation serves as an effective approach to improving model performance in low-resource settings, yet research on applying LLMs for EAE augmentation remains preliminary. In this study, we pay attention to the boundary sensitivity of EAE and investigate four LLM-based augmentation strategies: argument replacement, adjunction rewriting, their combination, and annotation generation. We conduct comprehensive experiments across four benchmark datasets, employing GPT-4o-Mini and DeepSeek-R1-7B as data generators. Our results show that boundary-aware augmentation consistently leads to greater performance improvements over boundary-agnostic methods. In addition to performance gains, we provide a detailed analysis of augmentation quality from multiple perspectives, including uncertainty reduction, error types, data quality, and data scale. This work offers both empirical evidence and practical guidance for leveraging LLMs to enhance event argument extraction under low-resource conditions.



Source Sentence:
The small, whitewashed chapel was the first Protestant church built in China.

Argument Replacement:
The small, whitewashed chapel was the first Anglican church built in Singapore.

Adjunction Rewriting:
The small, whitewashed chapel, which holds historical significance, was the first Protestant church built in China.

Argument Replacement & Adjunction Rewriting:
The small, whitewashed chapel stands proudly as the first Anglican church built in Singapore, reflecting the historical significance of the area.

Annotation Generation:
Built 1847 - 1849, this usually deserted Anglican foundation is Hong Kong ' s oldest church .

Figure 1: Examples of different augmentation methods.

1 Introduction

Event Argument Extraction (EAE) is a core event extraction subtask that focuses on identifying and classifying participants involved in an event (Pouran Ben Veyseh et al., 2020; Parekh et al., 2023). As a complex NLP task, EAE demands fine-grained semantic understanding and faces significant challenges, including the diversity and imbalance of argument roles, as well as the flexibility of argument boundaries. Although recent LLMs have shown strong performance in NLP, they still underperform fine-tuned models on EAE (Parekh et al., 2023; Ma et al., 2023; Sun et al., 2024). However, fine-tuning EAE models relies heavily on annotated data, which is expensive to obtain due to the complexity of event annotation, particularly in specialised domains such as healthcare. Consequently, data scarcity remains a major challenge in developing effective EAE models, especially in low-resource settings.

Data augmentation is an effective way to address data scarcity. However, boundary sensitivity is a critical consideration when generating data for EAE. Prior work has primarily addressed this by preserving argument positions while either replacing argument spans (Hong et al., 2022; Wang and Huang, 2024) or rewriting the surrounding context (Yang et al., 2019; Gao et al., 2022). Yet, most studies rely on knowledge-base matching for replacement or small language models for rewriting, limiting scalability and flexibility. For example, argument replacement is often limited to predefined entity types, whereas real-world arguments vary in span length rather than appearing as fixed entities.

LLMs, with their extensive knowledge and

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 4937–4953
March 24-29, 2026 ©2026 Association for Computational Linguistics

strong text generation capabilities, offer a promising solution for data augmentation in EAE. However, research on LLM-based augmentation for EAE is sparse and largely neglects its inherent boundary sensitivity (Sun et al., 2024; Meng et al., 2024). We argue that using LLMs for argument relabeling is limited by their extraction performance, which risks introducing noise that undermines augmentation effectiveness, as demonstrated by Sun et al. (2024).

This study explores different ways to leverage LLMs for EAE data augmentation in low-resource settings, providing a comprehensive evaluation from multiple perspectives. Specifically, we compare four LLM-based augmentation strategies: *argument replacement*, *adjunction rewriting*, *their combination*, and *annotation generation* (see examples in Figure 1). Among these, *argument replacement*, *adjunction rewriting*, and *their combination* are boundary-aware, as they preserve original argument positions when generating new samples. By contrast, *annotation generation* investigates the impact of LLM-generated labels, which are widely used in prior work, on augmentation effectiveness. Furthermore, *argument replacement* evaluates the LLM's capability to enhance argument diversity, whereas *adjunction rewriting* assesses its ability to improve sentence representation diversity, which are two distinct yet essential directions for EAE data augmentation.

We conduct extensive experiments on four benchmark datasets spanning general and domain-specific contexts, leveraging recent high-performing LLMs (GPT-4o-Mini and DeepSeek-R1-7B) as data generators. Our multi-faceted analysis offers several key findings: (i) **LLM-based boundary-aware augmentation outperforms non-boundary-aware approaches.** Methods preserving argument boundaries, such as *argument replacement* and *adjunction rewriting*, generally outperform annotation-based approaches across both Micro_F1 and Macro_F1 metrics. They effectively reduce spurious and partial argument errors by enforcing argument boundary precision, thereby improving both dominant and rare argument types. (ii) **Method effectiveness is context-sensitive.** Among LLM-based methods, *Argument replacement* excels on datasets with clear argument definitions and domain-specific semantics, whereas *adjunction rewriting* better suits datasets with vague roles by enhancing sentence-level diversity. We also show that this difference may be linked to the role of augmented samples in reducing uncertainty within the target domain. In addition, when compared to non-LLM boundary-aware baselines (e.g., *Mask-then-Fill*), LLM-based methods are better suited to the *argument replacement* strategy, likely due to their stronger semantic modelling capacity under explicit role constraints. (iii) **Data scaling and augmentation ratio offer complementary benefits.** Increasing the augmentation ratio yields stable improvements by mitigating noise and expanding semantic coverage, although gains eventually plateau due to sample homogeneity. Moreover, augmented data continues to provide benefits as the training set grows, highlighting the complementary relationship between data scale and augmentation. **These findings highlight the importance of boundary-aware augmentation and offer practical insights for advancing low-resource EAE.**

2 Related Work

Event Argument Extraction Event argument extraction (EAE) is a subtask of event extraction (EE) that typically follows event detection. Unlike event detection, EAE requires finer-grained semantic understanding and faces additional challenges from the diversity and imbalance of argument roles, as well as the variability of argument boundaries. Early EAE approaches were mainly **classification-based**, involving the selection of candidate argument spans followed by argument role assignment (Pouran Ben Veyseh et al., 2020; Ma et al., 2022b; He et al., 2023). However, classification-based methods struggled with overlapping arguments and were outperformed by **generation-based** approaches in recent years. The latter reframe EE as a sequence generation task, either by filling manually constructed natural language templates with arguments (Paolini et al., 2021; Hsu et al., 2022) or by transforming extraction targets into structured language representations that are then linearised (Lu et al., 2021, 2022). More recently, several studies further reformulated EAE as a **question answering (QA) task**, wherein argument role definitions are translated into questions and the model generates answers corresponding to argument spans (Li et al., 2020; Du and Cardie, 2020; Sun et al., 2022). QA-based EAE models can be further categorised into **extractive** and **generative** variants, depending on whether they utilise encoder-only or encoder–decoder architectures. Based on our empirical observations, **generative QA models** outperform extractive QA models and structured generation methods in low-resource settings. With the advancement of LLMs, some studies also explored **LLM-based prompting and in-context learning** for EAE (He et al., 2024; Sun et al., 2024; Sainz et al., 2024). Nonetheless, owing to the intrinsic complexity of EAE, LLMs still fall short of the performance achieved by task-specific fine-tuned models.

Data Augmentation for EAE EAE frequently suffers from limited training data due to the cost and complexity of annotation, making data augmentation a practical remedy for low-resource scenarios. General text augmentation techniques, such as text paraphrasing (Wei and Zou, 2019) and back translation (Shleifer, 2019), can inadvertently alter argument positions within a sentence, thereby complicating label alignment and introducing noise. Effective EAE data augmentation should preserve boundary accuracy, while existing methods fall into two main directions: (i) **Enhancing argument diversity**: This approach exploits existing datasets (e.g., ACE (Doddington et al., 2004)) or knowledge bases (e.g., Probase (Wu et al., 2012)) to retrieve entity types for each argument role and replace arguments with alternative instances of the same type. (Yang et al., 2019; Hong et al., 2022; Wang and Huang, 2024). However, it is constrained by fixed entity-type inventories and suffers from ambiguity in argument–entity alignment, often resulting in replacements that fail to match the sentence context. (ii) **Enhancing sentence diversity**: This approach rewrites adjunctions of the sentence while keeping arguments unchanged, typically through synonym replacement (Ma et al., 2022a) or mask-filling with a pre-trained language model (Yang et al., 2019; Gao et al., 2022). We argue that LLMs, with their strong reasoning abilities and extensive internal knowledge, are well-suited for EAE data augmentation and can more effectively address the limitations of prior methods. However, their application in this area remains limited. Although several recent works have employed LLMs for EAE augmentation, they often neglect boundary sensitivity and deliver only modest improvements. (Sun et al., 2024; Meng et al., 2024). In this study, we undertake a systematic investigation of boundary-aware LLM-based augmentation strategies for EAE and rigorously evaluate their efficacy.

3 Method

3.1 Task Formalisation

Event argument extraction is a subtask of event extraction, where an *event* is characterised by its *type*, *trigger*, and a set of *arguments*. *Event arguments* denote entities or non-entity spans providing event-related context, each defined by an *argument role*. EAE extracts the correct argument for each role given a sentence, an event type, and its trigger. We formalise EAE as a QA-style text generation task, building on the QA-based event extraction framework (Du and Cardie, 2020; Li et al., 2020). Specifically, given a sentence s containing multiple events $\{e_i\}$, we define arguments for each event e_i as $A_i = \{a_{i,j}\}$, where each argument $a_{i,j}$ is associated with a role $r_{i,j}$. For each event e_i and argument role $r_{i,j}$, we construct the following input:

```
Sentence: <SENTENCE>; Event: <EVT_TYPE>;
Trigger: <EVT_TRIGGER>; <ARG_ROLE>:
```

where `<SENTENCE>` is s, `<EVT_TYPE>` is the type of e_i, `<EVT_TRIGGER>` is its trigger word, and `<ARG_ROLE>` specifies the role $r_{i,j}$ to extract. The model is trained to output the text span corresponding to $a_{i,j}$. This formulation can be fine-tuned with any language model, making it broadly adaptable. We adopt Flan-T5 (Chung et al., 2024) as our backbone due to its efficiency in low-resource EAE, and denote this model as **Flan-T5 EEQA**.

3.2 LLM-based Data Augmentation

We examine four representative LLM-based data augmentation methods for event argument extraction: *argument replacement*, *adjunction rewriting*, *their combination*, and *annotation generation*. Illustrative outputs from these methods are presented in Figure 1.

Argument Replacement We employ LLMs for argument replacement by prompting them to generate new arguments consistent with role definitions and compatible with the sentence context, while leaving the remainder of the sentence unchanged. This leverages LLMs' strong language understanding and broad knowledge to produce diverse arguments, thereby improving the fine-tuned model's ability to learn argument semantics. Unlike traditional knowledge bases, LLMs are not restricted to predefined entities. Moreover, keeping the surrounding context intact preserves boundary accuracy, which is crucial for EAE.

To ensure that the LLM generates valid and easily parsable samples, we standardise both input and output in JSON format and include a complete input-output example in the prompt to guide the model's adherence to the expected structure. Specifically, the input consists of *a sentence* from the training set, its annotated *event type*, *event trigger*, and *arguments*, together with *definitions of the event type and argument roles*. The output includes the *generated sentence*, *event type*, *event trigger*, and *corresponding new arguments*. Although the event type and trigger remain fixed, we retain them in the output to enforce this constraint. However, the LLM occasionally deviates from instructions, generating invalid samples. To ensure data quality, we discard instances where the new argument or trigger word is missing from the generated sentence. The instruction prompt and input-output examples are provided in Appendix A.

Adjunction Rewriting Using LLMs for adjunction rewriting involves rewriting non-argument spans (i.e., adjunctions) while keeping arguments unchanged. This increases sentence diversity while preserving argument boundary accuracy, thereby improving the fine-tuned model's generalisation without loss of precision. To ensure consistency and quality, we adopt the same prompt and JSON input–output structure as in *Argument Replacement*, modifying only the instruction and example. Invalid outputs are filtered using the same rules.

Argument Replacement & Adjunction Rewriting We combine argument replacement and adjunction rewriting to progressively increase sample diversity by first generating new arguments and then rewriting the remaining sentence. To reduce costs, adjunction rewriting is applied to the outputs of argument replacement, efficiently yielding additional augmented data.

Annotation Generation Using LLMs for annotation generation leverages their predictive capabilities to create weakly supervised labels for unlabeled source texts. This approach benefits from unrestricted access to raw source data, enabling extensive sampling from domain-specific texts to improve authenticity and diversity. However, despite LLMs' strong reasoning ability, event extraction depends on complex annotation rules and precise boundary identification. Consequently, LLMs often produce noisy labels under limited in-context demonstrations, which can degrade the accuracy of fine-tuned models.

In our low-resource setting, we augment data using samples from the full training set excluded from the low-resource subset, treating LLM predictions as weak supervision labels. For annotation generation, we follow Sun et al. (2024)'s approach, retrieving the five most similar samples for each unlabeled instance using the BM25 (Trotman et al., 2014) algorithm. We use their inputs and annotations as in-context examples to prompt the LLM. Given the limited training data and the need for repeated trials in low-resource settings, we adopt a cost-efficient strategy: we sample validation instances equal in size to the low-resource training set as the retrieval corpus and generate augmented labels for all training samples in a single pass. During training, we filter out augmented samples that duplicate training instances to preserve data integrity.

4 Experimental Setup

We evaluate the effectiveness of the data augmentation methods using two models: GPT-4o-Mini (OpenAI, 2024) and DeepSeek-R1-Distill-Qwen-7B (Guo et al., 2025). Key aspects of the experimental setup are outlined below, with additional details provided in Appendix B.

Datasets We evaluate our methods on four datasets: two general-domain, **GENEVA** (Parekh et al., 2023) and **ACE** (Doddington et al., 2004), and two domain-specific, **PHEE** (Sun et al., 2022) and **CASIE** (Satyapanich et al., 2020). Detailed dataset descriptions, statistics, and examples are provided in Appendix C.

Low-resource Training We simulate low-resource conditions by randomly sampling n event mentions for training (Parekh et al., 2023), while keeping validation and test sets unchanged. Unlike few-shot training, which selects k samples per event type, this approach preserves the natural distribution of events and arguments, making it more representative of real-world scenarios. We therefore adopt it as our primary research setting. We conduct experiments across different resource levels, ranging from low (n = *25*) to moderate (n = *400*). For data augmentation, we generate additional samples at {1×, 2×, 4×} the size of the original training data per event mention.

Evaluation Metrics As argument spans can be long, making exact matching difficult, we follow Sun et al. (2022) and evaluate models using both exact match (EM) and token-level match. EM_F1 measures the F1 score of predicted spans that exactly match the ground truth, while Token_F1 computes the average token-overlap F1 score, allowing partial matches. We further report Micro and Macro variants. Micro_F1 aggregates true positives (TP) over all arguments before computing F1. For Macro_F1, to reflect differences in dataset characteristics—some being argument-dense, others event-dense—we report two metrics: Arg_Macro_F1, averaging F1 over all *argument types*, and Evt_Macro_F1, averaging over all *event types*. Overall, we evaluate performance using six metrics: {Micro_EM_F1, Micro_Token_F1, Arg_Macro_EM_F1, Arg_Macro_Token_F1, Evt_Macro_EM_F1, Evt_Macro_Token_F1}.

Baselines We evaluate our methods against the following *event argument extraction* baselines: (i) **GPT-4o-Mini** (OpenAI, 2024), an in-context learning baseline following Sun et al. (2024), which retrieves the five most similar training examples as demonstrations; (ii) **EEQA** (Du and Cardie, 2020), a QA-based extraction model that formulates argument extraction as a question-answering task using label semantics. However, unlike our encoder-decoder framework (e.g., Flan-T5), EEQA uses an encoder-only backbone (e.g., BERT); (iii) **UIE** (Lu et al., 2022), a structured text generation model pretrained for unified information extraction within a seq2seq framework. To adapt the above frameworks for the EAE task, we provide the event type and trigger word as part of the input.

For the *data augmentation baselines*, since most previous EAE augmentation methods lack open-source implementations, we reproduced the generalisable *Mask-then-Fill* approach (Gao et al., 2022) as a representative non-LLM method and adopted Sun et al. (2024)'s *Synthesize-and-Label* as the LLM-based baseline. The *Mask-then-Fill* approach constructs new synthetic samples by randomly masking text spans and fine-tuning a T5 model to fill the masked segments. While the original work applied masking only to adjunction phrases to generate new data, we further implemented a variant that masks argument spans to enable a fairer comparison with our LLM-based augmentation methods. The *Synthesize-and-Label* method prompts GPT-3.5 with an input sample and its annotated event to generate new sentences that share a similar event structure and subsequently extracts events from the generated text. For a fair comparison, we reproduced this method using the same LLM base and filtering strategy as in this study.

5 Results and Analysis

5.1 Overall Performance

Micro_F1 Results Table 1 presents the Micro_F1 results of various baseline models and data augmentation strategies across four datasets under low-resource conditions (n=200). Due to space limitations, this section primarily reports the data augmentation results for the best-performing base model, Flan-T5 EEQA, while the augmented results for the UIE model are provided in Appendix D for comparison.

Overall, the four proposed methods reveal that LLM-based boundary-aware data augmentation tends to outperform non-boundary-aware approaches. Specifically, *argument replacement* and *adjunction rewriting* yield notable gains, even when employing a comparatively weaker synthesiser such as DeepSeek-R1-Distill-Qwen-7B (hereafter referred to as DeepSeek-R1-7B). *Their combination* increases data diversity but degrades performance, likely due to error accumulation. Consequently, although it yields positive effects when a stronger model like GPT-4o-Mini is employed, it proves detrimental when applied with weaker models like DeepSeek-R1-7B. However, all methods markedly surpass *annotation generation*, whose inconsistent label quality can substantially degrade target model performance. Interestingly, the *synthesize-and-label* method by Sun et al. (2024), despite lacking in-context demonstrations, outperforms annotation generation with 5-shot examples, suggesting that prompting the model to imitate rather than directly generate labels may be more effective. Our boundary-aware augmentation shares this trait but imposes stricter boundary constraints, thereby delivering superior performance and underscoring the critical role of boundary precision in EAE data augmentation.

While *argument replacement* and *adjunction rewriting* achieve the best overall results, their relative effectiveness differs across datasets. *Argument replacement* performs best on GENEVA, likely because the dataset's broad coverage and clear argument definitions make it well-suited to argument augmentation. It also performs well on PHEE and CASIE when paired with GPT-4o-Mini, whereas DeepSeek-R1-7B yields weaker outcomes, possibly due to limited domain knowledge. On ACE, *argument replacement* is less effective, whereas

	GENEVA		ACE		PHEE		CASIE	
	Micro_EM_F1	Micro_Token_F1	Micro_EM_F1	Micro_Token_F1	Micro_EM_F1	Micro_Token_F1	Micro_EM_F1	Micro_Token_F1
GPT-4o-Mini	38.54	57.41	37.89	46.85	64.12	75.92	46.96	55.94
EEQA	25.26 ± 2.97	26.95 ± 1.93	22.13 ± 3.03	12.02 ± 1.71	45.20 ± 0.77	40.56 ± 0.82	29.13 ± 1.18	23.49 ± 2.36
UIE	44.22 ± 1.78	57.07 ± 2.22	45.83 ± 1.35	48.25 ± 1.75	67.53 ± 0.60	75.39 ± 0.55	46.10 ± 0.57	48.60 ± 1.01
Flan-T5 EEQA	**50.15 ± 2.80**	**58.87 ± 3.98**	**48.47 ± 1.72**	**51.45 ± 1.12**	**69.78 ± 0.97**	**77.57 ± 1.22**	**46.50 ± 3.68**	**49.67 ± 3.76**
Flan-T5 EEQA + Mask-then-Fill								
Argument Augmentation	50.89 ± 1.91	62.57 ± 1.91	36.41 ± 4.03	33.75 ± 7.42	66.45 ± 0.92	71.84 ± 1.48	47.58 ± 0.71	51.48 ± 0.84
Adjunction Augmentation	55.29 ± 1.36	64.36 ± 1.75	51.71 ± 1.52	54.62 ± 1.39	69.53 ± 0.92	77.70 ± 0.90	51.00 ± 1.74	54.85 ± 1.83
Flan-T5 EEQA + LLM Augmentation (GPT-4o-Mini)								
Synthesize-and-Label	54.33 ± 1.19	64.88 ± 1.26	49.55 ± 2.36	53.58 ± 2.16	69.23 ± 1.16	77.41 ± 1.50	47.22 ± 2.65	51.15 ± 3.52
Argument Replacement (ours)	**58.39 ± 1.30**	**67.68 ± 1.38**	47.39 ± 2.51	53.35 ± 2.49	**70.99 ± 0.42**	**79.17 ± 0.76**	**52.94 ± 1.37**	**56.95 ± 1.39**
Adjunction Rewriting (ours)	55.33 ± 2.82	65.31 ± 3.15	**52.81 ± 1.24**	**55.43 ± 0.86**	69.80 ± 1.38	78.08 ± 1.62	51.18 ± 1.19	55.51 ± 1.62
Argument + Adjunction (ours)	54.71 ± 2.21	65.08 ± 2.93	49.89 ± 2.51	54.01 ± 1.97	70.38 ± 0.53	78.66 ± 0.67	51.08 ± 1.02	55.33 ± 1.23
Annotation Generation (ours)	44.92 ± 1.25	61.21 ± 2.36	40.91 ± 2.73	45.58 ± 2.84	62.63 ± 1.50	72.23 ± 1.36	44.30 ± 1.34	49.09 ± 1.81
Flan-T5 EEQA + LLM Augmentation (DeepSeek-R1-Distill-Qwen-7B)								
Synthesize-and-Label	51.20 ± 1.49	61.36 ± 2.26	44.11 ± 3.29	48.84 ± 2.93	68.91 ± 1.58	77.14 ± 1.36	46.87 ± 2.73	51.25 ± 3.62
Argument Replacement (ours)	**52.85 ± 2.07**	**64.22 ± 2.00**	43.13 ± 1.83	49.37 ± 1.95	70.84 ± 0.66	**79.41 ± 0.71**	49.47 ± 1.18	54.17 ± 0.99
Adjunction Rewriting (ours)	52.74 ± 1.85	63.18 ± 2.11	**50.70 ± 1.00**	**54.01 ± 0.94**	**71.02 ± 0.58**	78.76 ± 0.46	**50.36 ± 1.07**	**54.59 ± 1.89**
Argument + Adjunction (ours)	46.69 ± 2.34	59.87 ± 1.40	40.76 ± 1.66	46.73 ± 1.28	69.68 ± 0.93	78.22 ± 0.90	47.61 ± 0.53	51.95 ± 0.93
Annotation Generation (ours)	28.89 ± 3.42	42.36 ± 6.32	33.44 ± 1.97	37.09 ± 2.66	61.44 ± 1.39	69.68 ± 1.31	39.69 ± 2.42	43.43 ± 3.01

Table 1: Micro_F1 performance (n=200, 4x augmentation). The GPT-4o-Mini baseline is evaluated on a single subset due to cost constraints. For others, the mean ± standard deviation over five runs is reported.

adjunction rewriting performs better. This is likely because ACE arguments are mostly generic entities, with event roles defined rather vaguely, making it difficult for the model to follow them during generation. Moreover, the brevity of ACE arguments also limits the semantic diversity achievable through argument replacement, making adjunction rewriting more beneficial. Notably, although DeepSeek-R1-7B performs inconsistently with *argument replacement*, it consistently benefits from *adjunction rewriting*, indicating that the latter places lower demands on the model's semantic understanding.

Moreover, a comparison between our proposed methods and *Mask-then-Fill* shows that, although *Mask-then-Fill* is also a boundary-aware approach, its ability to increase argument diversity is limited by the restricted knowledge capacity of smaller models and can even negatively affect performance in some cases. In contrast, LLMs are able to generate more appropriate arguments under explicit schema instructions, which explains their stronger performance in argument replacement. Although DeepSeek-R1-7B does not match the performance of GPT-4o-Mini in this respect due to its relative smaller model size, it still outperforms *Mask-then-Fill*. However, the overall gains of DeepSeek-R1-7B remain lower than those achieved by directly applying *adjunction rewriting* with *Mask-then-Fill*. This performance gap is likely attributable to the fact that *Mask-then-Fill* is fine-tuned for sentence reconstruction, which helps it avoid some of the structure-control issues that DeepSeek-R1-7B encounters when handling longer or more structurally complex sentences. This observation is helpful for making more informed decisions about model selection and augmentation strategy combinations in future data augmentation research.

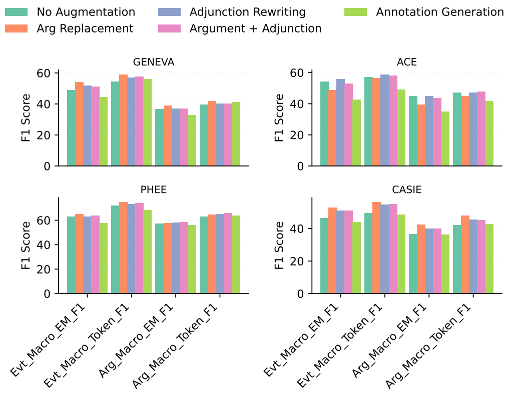


Figure 2: Macro_F1 Performance (n=200, GPT-4o-Mini, 4x augmentation). Bars represent five-run averages.

Macro_F1 Results The EAE task frequently suffers from imbalances between event and argument types. We employ macro-averaged F1 scores to evaluate the overall performance of different augmentation methods across all event and argument types. Figure 2 reports the results for GPT-4o-Mini, with corresponding results for DeepSeek-R1-7B presented in Appendix E.

In summary, boundary-aware augmentation methods generally yield improvements in both event and argument Macro_F1 scores, demonstrating their effectiveness in enhancing not only dominant arguments but also the rarer ones. By contrast, *annotation generation* often has a negative impact on Macro_F1. However, its effect on Arg_Macro_F1 is less severe than on Micro_F1, and in several cases it remains neutral or even produces slight gains. This suggests that the strong generalisation capability of LLMs can still facilitate the learning of rare arguments, and when combined with boundary-aware methods, it can further improve performance. Additionally, the patterns of different methods across datasets are largely consistent with those in Table 1, and are therefore not discussed further here.

5.2 Evaluating Augmentations Through Uncertainty Reduction

Drawing inspiration from Hübotter et al. (2025), who employed uncertainty estimation in test-time fine-tuning, we adopt a similar perspective to compare augmentation methods by quantifying their impact on test-sample uncertainty. Intuitively, if augmented data are well distributed in the embedding space, they should better span the regions occupied by test samples, thereby lowering uncertainty. Specifically, for each test sample x, we compute $\sigma_{D_{aug}}(x)$, representing its response uncertainty after fine-tuning on a given augmented dataset D_{aug}, and then average over all test samples. The metric $\sigma_{D_{aug}}$ is derived from kernel similarities in the model's embedding space, capturing how well augmented samples cover the representation regions where test points lie. Lower σ values indicate that the augmented data more span the embedding space around the test examples, resulting in reduced uncertainty. Table 2 reports uncertainty results across different augmentation methods.

	GENEVA	ACE	PHEE	CASIE
Argument Replacement	0.1545	0.1328	0.0362	0.0867
Adjunction Rewriting	0.1621	0.1313	0.0572	0.1082
Argument + Adjunction	0.1659	0.1138	0.0400	0.0868
Annotation Generation	0.0595	0.0624	0.0257	0.0098

Table 2: Uncertainty measure (↓) across different datasets and augmentation strategies (GPT-4o-Mini).

As illustrated, *adjunction rewriting* yields lower uncertainty than *argument replacement* for ACE, whereas the opposite pattern emerges for other datasets, mirroring earlier reported performance trends. This suggests that, given controlled boundary accuracy, *adjunction rewriting* enriches the space more effectively for ACE, whereas *argument replacement* is better suited for other datasets. This aligns with our earlier analysis of dataset characteristics: ACE arguments are shorter and context-dependent, whereas arguments in the other datasets are semantically richer and often dominate sentence. Notably, *annotation generation* attains the lowest $\sigma_{D_{aug}}(x)$, implying better embedding coverage, yet performs worst overall, as its outputs, though diverse, inherit label noise from LLM generation. This highlights the challenge of balancing data quality and diversity in data augmentation.

5.3 Argument Extraction Error Analysis

To analyse the impact of different data augmentation methods on EAE, we implemented an automated script to classify extraction errors in models trained on the respective augmented datasets. Figure 3 reports error statistics for GPT-4o-Mini, while results for DeepSeek-R1-7B and definitions of error types are presented in Appendix F.

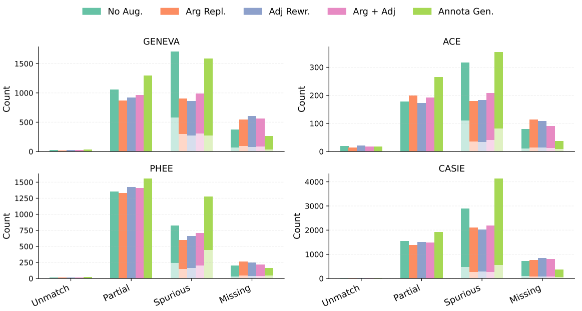


Figure 3: Error analysis of argument extraction (n=200, GPT-4o-Mini, 4× augmentation). Bars show five-run averages; lighter segments denote "role error" subtype.

The results show that partial matches and spurious arguments are the most frequent error types, whereas missing arguments are less common and unmatched arguments are rare. Among the four augmentation methods, *argument replacement*, *adjunction rewriting*, and *their combination* generally mitigate partial matches and substantially reduce spurious arguments, demonstrating the effectiveness of boundary-aware approaches in preserving argument boundaries and facilitating semantic learning. Although these methods slightly increase the incidence of missing arguments, this reflects an inherent trade-off between coverage and precision. In contrast, *annotation generation* provides modest gains in reducing missing arguments but significantly increases partial and spurious errors, suggesting that limitations of the LLM in capturing argument boundaries and semantics under few-shot conditions may propagate into the fine-tuned

Dataset	Unmatched	Partial Match	Spurious	Missing
GENEVA	53 (37)	110 (53)	988 (86)	101 (25)
ACE	16 (-1)	13 (-9)	209 (30)	17 (-16)
PHEE	24 (-2)	229 (58)	708 (109)	41 (-23)
CASIE	51 (26)	231 (106)	2186 (74)	134 (33)

Table 3: Changes in error types when moving from argument replacement to the combined augmentation method. Numbers in parentheses indicate net increases after subtracting reverse conversions.

models. Additionally, some spurious and missing argument errors arise from role confusion, where the model misclassifies argument types within the same event. These errors may be influenced by co-occurrence patterns in the training data but remain relatively infrequent overall.

In addition, to provide a more fine-grained analysis of the error accumulation in the combined method discussed in Section 5.1, we quantify how many originally correct arguments are converted into each error type when moving from argument replacement to the combined approach. As shown in Table 3, the most pronounced increases occur in partial-match and spurious errors. From our manual inspection, partial-match errors mainly fall into two categories: (i) the predicted argument overlaps only partially with the gold span, usually due to added or removed modifiers—an effect that can arise and accumulate when argument replacement or adjunction rewriting alters descriptive content in long or modifier-rich arguments; and (ii) multi-span predictions where some spans are incorrectly included or come from another argument role, which closely relates to spurious errors. These, as well as spurious errors, often result either from extracting spans that are lexically similar but semantically misaligned with the target role, or from confusion with another argument's meaning. The amplification of such errors through accumulation is not entirely clear, but one plausible explanation lies in the complex structure of events. For instance, when two arguments are nested or tightly intertwined in the original sentence, replacement and rewriting may preserve one argument faithfully while altering or deleting the other, but the annotation does not maintain alignment afterwards, thereby introducing noise. This quantitative and qualitative evidence elaborates on the possible sources of the propagation chain and highlights the inherent challenges of EAE data augmentation that warrant further study.

5.4 Quality of Augmented Data

We manually inspected the quality of augmented data generated by different augmentation strategies and generative models. Although the models do not strictly follow instructions, most errors are easily detected using rule-based heuristics. We categorise these errors and report statistics for different methods across the four datasets in Appendix G. These errors are subsequently filtered out from the generated data. Among them, the two most frequent categories are *invalid triggers* and *invalid arguments*, referring to cases where event annotations contain triggers or arguments that cannot be aligned with the original sentence. *Invalid triggers* mainly arise in boundary-aware augmentation methods, particularly when the trigger forms part of a replaced argument. In such cases, preserving the original event trigger is challenging. Moreover, triggers can also be lost during adjunction rewriting. For GPT-4o-Mini, *invalid arguments* primarily stem from tokenisation mismatches during preprocessing. By contrast, DeepSeek-R1-7B is more prone to hallucinations, often producing arguments that deviate substantially from the synthetic sentence.

Overall, GPT-4o-Mini retains significantly more of its generated data compared to DeepSeek-R1-7B. Among all datasets, ACE exhibits the highest data retention rate, likely because its arguments are generally shorter and thus less prone to triggering the filtering mechanisms. Among the augmentation methods, *argument replacement* results in the highest retention rate, particularly when GPT-4o-Mini is used on domain-specific datasets such as PHEE and CASIE. In these cases, *argument replacement* seldom causes data to be filtered out, likely because domain-specific argument types are more semantically coherent and narrowly defined. This advantage is less evident for DeepSeek-R1-7B, which may lack sufficient domain knowledge.

We further inspected the augmented data after filtering. At this stage, most samples are semantically accurate, syntactically fluent, and their arguments generally align well with both the sentence and schema. However, some subtle errors persist, occurring more often in general-domain datasets with broader argument semantics (examples are shown in the Appendix). Specifically, in *argument replacement*, GPT-4o-Mini often replaces only a subset of arguments when multiple occur in an event. This may reduce synthetic data diversity but does not markedly affect model accuracy. By contrast, DeepSeek-R1-7B often alters non-argument portions of the sentence. In extreme cases, it retains all arguments but heavily rewrites the sentence, entirely violating the instruction. DeepSeek-R1-7B is also more prone to spelling errors and occasionally mixes words from other languages. In *adjunction rewriting*, GPT-4o-Mini tends to paraphrase while preserving the original meaning, whereas DeepSeek-R1-7B often generates more flexible sentences. Although this increases variation, it sometimes results in arguments no longer fulfilling their original roles in the new sentence, thereby introducing errors. Combining the two methods may compound these issues, increasing argument-sentence mismatches. The quality of *annotation generation* depends entirely on the model's few-shot EAE ability and is not analysed further here.

5.5 Impact of Data Scale

To evaluate the effectiveness of data augmentation under varying resource conditions, we assess model performance across different data sizes and augmentation ratios (see Appendix H for results). Analysing performance across augmentation ratios, we observe: (i) **Data augmentation consistently improves performance, and higher augmentation ratios yield more stable gains**. Increasing the augmentation ratio helps mitigate noise in augmented data, thereby supporting more robust learning of argument semantics. However, due to the inherent homogeneity of augmented samples, further increasing the ratio cannot overcome the ceiling of achievable gains. Thus, users should select an appropriate augmentation ratio based on available resources and task needs. This also underscores the challenge of balancing sample quality and diversity, highlighting a key direction for future research. (ii) **As original training data increases, augmented data continues to provide clear gains**. In GENEVA, these gains remain stable across data sizes, whereas on ACE, they are initially modest but become more pronounced with more original data. This suggests that argument learning in ACE benefits more from diverse contextual cues, which may also explain why *adjunction rewriting* is particularly effective for this dataset.

6 Conclusion

This work presents boundary-aware LLM-based augmentation strategies that substantially improve low-resource event argument extraction. Our experiments demonstrate that boundary-aware LLM-based methods outperform non-boundary-aware approaches, where *argument replacement* proves most effective on datasets with well-defined arguments, while *adjunction rewriting* is better suited for context-dependent arguments or when synthesis models exhibit limited capability. In addition, LLM-based methods are better suited to the *argument replacement* strategy when compared to non-LLM boundary-aware baselines, benefiting from stronger semantic modelling under explicit role constraints. Moreover, our systematic analysis from the perspectives of uncertainty reduction, extraction error analysis, augmented data quality, and data scaling provides actionable insights for advancing low-resource event argument extraction and broader information extraction tasks. Our study highlights that balancing data diversity and label accuracy remains a central challenge for EAE augmentation, and future research should explore ways to further enhance argument coverage while maintaining boundary precision.

Limitations

Due to the lack of publicly available implementations of existing data augmentation methods and the absence of detailed data in prior work on event argument extraction under low-resource settings that would allow for direct comparison, this study reproduces only a limited number of representative approaches for reference. The focus of this paper is on systematically analysing different LLM-based augmentation methods from a boundary-aware perspective, rather than conducting an exhaustive performance comparison across various augmentation strategies.

Building upon this, the present study primarily focuses on directly applying LLM-generated data for augmentation, complemented by a simple filtering strategy to ensure data validity. While more advanced filtering techniques or noise-tolerant training approaches could potentially further improve the effectiveness of certain augmentation methods, exploring these directions lies beyond the scope of this paper and is left for future work.

Acknowledgments

This work was supported in part by the UK Engineering and Physical Sciences Research Council (EPSRC) (grant no. EP/V020579/1, EP/V020579/2, UKRI566).

References

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

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

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

Jun Gao, Changlong Yu, Wei Wang, Huan Zhao, and Ruifeng Xu. 2022. Mask-then-fill: A flexible and effective data augmentation framework for event extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4537–4544, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Shiming He, Yu Hong, Shuai Yang, Jianmin Yao, and Guodong Zhou. 2024. Demonstration retrieval-augmented generative event argument extraction. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4617–4625, Torino, Italia. ELRA and ICCL.

Yuxin He, Jingyue Hu, and Buzhou Tang. 2023. Revisiting event argument extraction: Can EAE models learn better when being aware of event co-occurrences? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12542–12556, Toronto, Canada. Association for Computational Linguistics.

Ruixin Hong, Hongming Zhang, Xintong Yu, and Changshui Zhang. 2022. Learning event extraction from a few guideline examples. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:2955–2967.

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

Jonas Hübotter, Sascha Bongni, Ido Hakimi, and Andreas Krause. 2025. Efficiently learning at test-time: Active fine-tuning of llms. In *The Thirteenth International Conference on Learning Representations*.

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

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

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

Manfu Ma, Xiaoxue Li, Yong Li, Xinyu Zhao, Xia Wang, and Hai Jia. 2022a. Small sample medical event extraction based on data augmentation. In *International Conference on Biomedical and Intelligent Systems (IC-BIS 2022)*, volume 12458, pages 823–833. SPIE.

Yubo Ma, Yixin Cao, Yong Hong, and Aixin Sun. 2023. Large language model is not a good few-shot information extractor, but a good reranker for hard samples! In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10572–10601, Singapore. Association for Computational Linguistics.

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

Zihao Meng, Tao Liu, Heng Zhang, Kai Feng, and Peng Zhao. 2024. CEAN: Contrastive event aggregation network with LLM-based augmentation for event extraction. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 321–333, St. Julian's, Malta. Association for Computational Linguistics.

OpenAI. 2024. Gpt-4o mini: Advancing cost-efficient intelligence.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Ma Jie, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, Stefano Soatto, et al. 2021. Structured prediction as translation between augmented natural languages. In *ICLR 2021-9th International Conference on Learning Representations*, pages 1–26. International Conference on Learning Representations, ICLR.

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

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 the Association for Computational Linguistics: EMNLP 2020*, pages 3651–3661, Online. Association for Computational Linguistics.

Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. 2024. Gollie: Annotation guidelines improve zero-shot information-extraction. In *The Twelfth International Conference on Learning Representations*.

Taneeya Satyapanich, Francis Ferraro, and Tim Finin. 2020. Casie: Extracting cybersecurity event information from text. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8749–8757.

Sam Shleifer. 2019. Low resource text classification with ulmfit and backtranslation. *arXiv preprint arXiv:1903.09244*.

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

Zhaoyue Sun, Gabriele Pergola, Byron Wallace, and Yulan He. 2024. Leveraging ChatGPT in pharmacovigilance event extraction: An empirical study. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 344–357, St. Julian's, Malta. Association for Computational Linguistics.

Andrew Trotman, Antti Puurula, and Blake Burgess. 2014. Improvements to bm25 and language models examined. In *Proceedings of the 2014 Australasian Document Computing Symposium*, pages 58–65.

Sijia Wang and Lifu Huang. 2024. Targeted augmentation for low-resource event extraction. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 4414–4428, Mexico City, Mexico. Association for Computational Linguistics.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q Zhu. 2012. Probase: A probabilistic taxonomy for text understanding. In *Proceedings of the 2012 ACM SIGMOD international conference on management of data*, pages 481–492.

Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan, and Dongsheng Li. 2019. Exploring pre-trained language models for event extraction and generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5284–5294, Florence, Italy. Association for Computational Linguistics.

A Prompt Examples

Table A1 presents the instruction prompt and input-output examples for the *argument replacement* augmentation method. In practice, the LLM receives both the instruction and a demonstration example adhering to the specified input-output format. *Adjunction rewriting* shares the same input-output format, but with instructions that require the model to rewrite the sentence while preserving the event trigger and arguments exactly as they are. For *annotation generation*, five demonstrations are provided in the same JSON format, and the model is instructed to generate annotations for a new sentence.

Instruction:
You are an AI assistant tasked with generating augmented data for an event argument extraction task.
Task Details:
1. Input: You will be given:
- A sentence with a labeled event and its arguments.
- The schema definition of the event, describing the roles and expected types of its arguments.
2. Your Task:
- Replace the event's arguments with new ones while keeping the rest of the sentence unchanged.
- Ensure that the new arguments conform to the schema's definition and are contextually appropriate within the sentence.
- Any part of the sentence except the arguments should remain unchanged.
- The event trigger is the word in the sentence indicating the occurence of the event, which should also be unchanged and displayed in the sentence.
3. Output Requirements:
- Generate exactly 5 augmented samples for each input sentence-event pair.
- Return the results in JSON format as shown in the example.
- Represent discontinuous arguments in lists.

Input Example:

```
{
    "sentence": "The biosecurity ...",
    "event": {
      "event_type": "scrutiny",
      "trigger": "looked",
      "arguments": {
        "cognizer": ["The biosecurity
            workshop" ...],
        "ground": ["at threats ..."]
      }
    },
    "schema": {
      "event_type": "scrutiny",
      "event_description": "...",
      "arguments": {
        "cognizer": "The Cognizer ...",
        "ground": "The Cognizer ..."
      }
    }
  }
```

Output Example:

```
  {
    "augmented_sentence": "The research
        ...",
    "event_type": "scrutiny",
    "trigger": "looked",
    "arguments": {
      "cognizer": ["The research
          committee", ...],
      "ground": ["at challenges ..."]
    }
  }
```

Table A1: Instruction prompt and input-output examples for *argument replacement*.

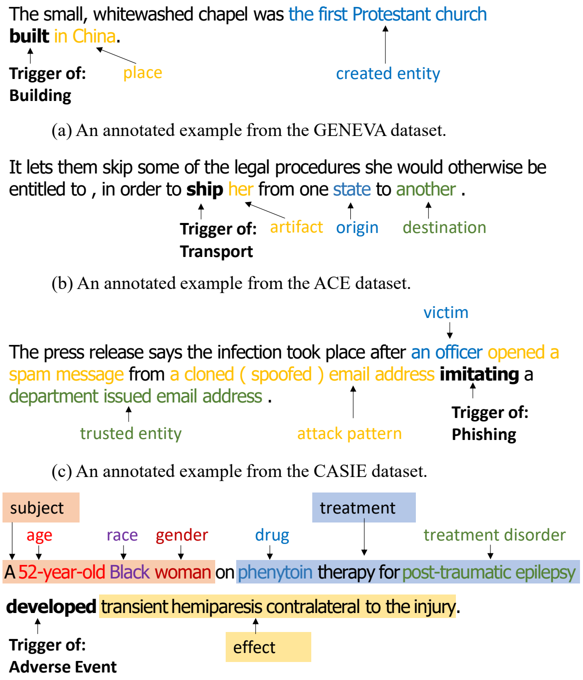
B Experimental Details

Data Generation: When generating *argument replacement* and *adjunction rewriting* augmented data, we generate five samples for each event mention in the training set. For *argument replacement & adjunction rewriting* augmentation, we apply *adjunction rewriting* to each *argument replacement* sample, generating two additional samples. *Annotation generation* produces annotations for all samples in the training set. For all augmented data, we first filter out the error types defined in Appendix F and then sample training data at different augmentation ratios.

Model Training: When training Flan-T5 EEQA, we sample empty arguments for each event with a probability of 0.2 and train the model to generate "None", enabling it to recognise empty arguments during inference. We use Flan-T5-base for Flan-T5 EEQA, UIE-base for UIE, and Bert-base for EEQA. For both Flan-T5 EEQA and UIE training, we use a batch size of 16, a learning rate of 1×10^{-4}, and apply early stopping if no improvement is observed on the validation set for 4 consecutive epochs. During inference, we perform beam search with a beam size of 2. EEQA training uses a batch size of 64 and a learning rate of 5×10^{-5}. All hyperparameters are selected based on preliminary experiments on the validation set. All experiments are conducted on a single NVIDIA A100 GPU.

C Supplementary Dataset Information

We evaluate our methods on four datasets, including two general-domain and two domain-specific event extraction datasets : (1) **GENEVA** (Parekh et al., 2023), a recent general-domain dataset with


The small, whitewashed chapel was the first Protestant church built in China.
Trigger of: Building
place
created entity

(a) An annotated example from the GENEVA dataset.

It lets them skip some of the legal procedures she would otherwise be entitled to , in order to ship her from one state to another .
Trigger of: Transport
artifact
origin
destination

(b) An annotated example from the ACE dataset.

victim
The press release says the infection took place after an officer opened a spam message from a cloned (spoofed) email address imitating a department issued email address .
trusted entity
attack pattern
Trigger of: Phishing

(c) An annotated example from the CASIE dataset.

subject
treatment
age
race
gender
drug
treatment disorder
A 52-year-old Black woman on phenytoin therapy for post-traumatic epilepsy
developed transient hemiparesis contralateral to the injury.
Trigger of: Adverse Event
effect

(d) An annotated example from the PHEE dataset.

Figure A1: Illustration of event annotations in four datasets. Triggers are shown in bold, with arguments indicated by colour. The PHEE dataset features hierarchical annotation, where *main arguments* are highlighted with a coloured background, and *sub-arguments* are indicated with coloured text.

	# Event Types	# Argument Types	# Sent.	# Event Mentions	# Argument Mentions
Train	115	412	1,968	4,170	6,777
Dev	115	346	783	1,442	2,383
Test	115	389	993	1,893	3,109

Table A2: Statistics of the GENEVA dataset.

	# Event Types	# Argument Types	# Sent.	# Event Mentions	# Argument Mentions
Train	33	102	19,216	4,419	6,607
Dev	22	65	901	468	759
Test	31	79	676	424	689

Table A3: Statistics of the ACE dataset.

	# Event Types	# Argument Types	# Sent.	# Event Mentions	# Argument Mentions
Train	2	32	2,898	3,004	16,081
Dev	2	32	961	1,003	5,509
Test	2	32	968	1,010	5,494

Table A4: Statistics of the PHEE dataset.

115 event types and 220 argument roles, offering broader coverage than ACE. (2) **ACE** (Doddington et al., 2004), a widely used general-domain dataset with 33 event types and 22 argument roles, where arguments are primarily general entities (e.g., *PERSON*, *PLACE*) with short spans. (3) **PHEE** (Sun et al., 2022), a medical-domain dataset containing two event types—*adverse event* and *potential therapeutic event* — each with 16 argument roles related to subject, treatment, and effect. While the dataset employs a hierarchical annotation scheme, we flatten all arguments to maintain consistency. (4) **CASIE** (Satyapanich et al., 2020), a cybersecurity-domain dataset containing 5 event types and 26 argument roles. In contrast to general-domain event extraction datasets, domain-specific schemas are informed by expert knowledge and often comprise fewer event types, yet exhibit more densely annotated arguments. Figure A1 presents annotated examples from the four datasets. Table A2 - A5 provides statistical information.

D UIE Model Results under Data Augmentation

Table A6 presents the results of enhancing the UIE model, and the relative performance of different augmentation strategies across datasets is largely consistent with the results observed in Flan-T5 EEQA. This suggests that the main findings of this paper are applicable to different event argument extraction model architectures. However, since the UIE model generates all event arguments in a single pass, its performance can be constrained under limited data conditions due to increased learning difficulty. Moreover, the noise among different arguments tends to interfere with each other, making the benefits of data augmentation less pronounced than those observed in Flan-T5 EEQA. Under low-resource settings, Flan-T5 EEQA is a more suitable choice for EAE, as it not only exhibits stronger baseline performance but also achieves more substantial gains from data augmentation.

E Macro F1 Performance for DeepSeek-R1-7B

Figure A2 presents the event and argument Macro-F1 scores obtained using DeepSeek-R1-Distill-Qwen-7B for data augmentation.

F Extraction Error Type Definitions and Statistics

We categorise the following error types for evaluating argument extraction:

- **Unmatch**: The model extracts an argument with the same role as the ground truth but with entirely different spans.

	# Event Types	# Argument Types	# Sent.	# Event Mentions	# Argument Mentions
Train	5	48	11,189	5,235	13,498
Dev	5	47	1,778	1,115	2,669
Test	5	48	3,208	2,121	5,699

Table A5: Statistics of the CASIE dataset.

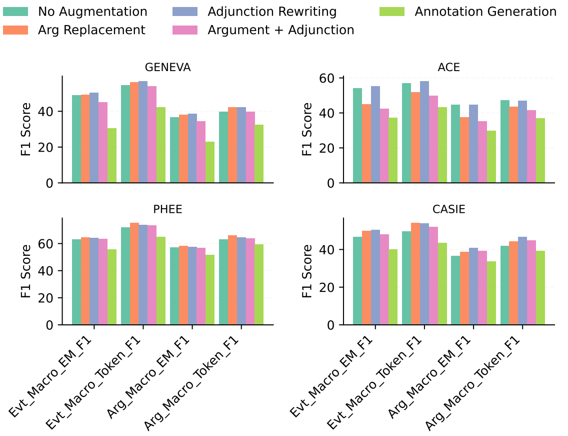


Figure A2: Macro_F1 Performance (n=200, DeepSeek-R1-7B, 4x augmentation). Bars represent five-run averages.

- **Partial Match**: The extracted argument partially overlaps with the ground truth, including cases where at least one span of a multi-span argument fully or partially matches the ground truth.

- **Spurious Argument**: The extracted argument is assigned a role that has no corresponding annotation in the ground truth. Specifically, we define a **role error** subclass, where a ground truth argument shares the same span as this predicted argument but is assigned a different role, indicating a potential misclassification by the model.

- **Argument Missing**: The model fails to extract an argument for a specific role present in the ground truth. Within this category, we also define a **role error** subclass, where a predicted argument shares the same span as this ground truth argument but is assigned a different role, suggesting a probable misclassification that led to its omission.

Figure A3 presents the error type analysis with DeepSeek-R1-Distill-Qwen-7B augmentation.

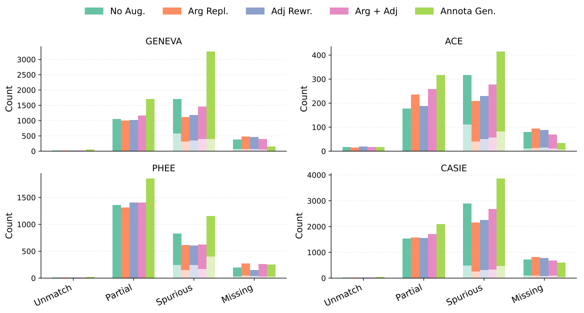


Figure A3: Error analysis of argument extraction (n=200, DeepSeek-R1-7B, 4× augmentation). Bars represent five-run averages; lighter segments denote the “role error” subtype.

G Supplementary of Augmented Data Quality

We categorised several easily detectable errors present in the synthetic data and applied filtering prior to its use. These errors include:

- **Broken JSON**: The responses generated by the data augmentation model fail to conform to the JSON format and therefore cannot be parsed. Once this occurs, none of the augmented samples associated with the same input instance can be parsed.

- **Format Violation**: While the generated responses are valid JSON, the internal dictionary structure differs from the predefined format or is partially missing, leading to incomplete information being extracted during parsing.

- **Invalid Trigger**: The trigger annotation generated by the model cannot be located within the corresponding augmented sentence, which prevents the example from being used in constructing input for the downstream argument extraction task.

- **Invalid Arguments**: The model-generated arguments are not present in the corresponding augmented sentence, which violates the definition of the argument extraction task and results in a noisy sample.

Tables A7–A10 present the statistics of these errors across different datasets. Table A11 shows representative examples of erroneous data augmentation.

	GENEVA		ACE		PHEE		CASIE	
	Micro_EM_F1	Micro_Token_F1	Micro_EM_F1	Micro_Token_F1	Micro_EM_F1	Micro_Token_F1	Micro_EM_F1	Micro_Token_F1
UIE	44.22 ± 1.78	57.07 ± 2.22	45.83 ± 1.35	48.25 ± 1.75	67.53 ± 0.60	75.39 ± 0.55	46.10 ± 0.57	48.60 ± 1.01
UIE + Mask-then-Fill								
Argument Augmentation	43.89 ± 1.61	58.15 ± 1.14	39.41 ± 2.51	38.64 ± 2.79	66.63 ± 0.23	74.57 ± 0.23	42.91 ± 0.71	45.80 ± 1.02
Adjunction Augmentation	45.95 ± 1.66	56.99 ± 1.68	46.55 ± 2.86	48.53 ± 2.63	67.94 ± 0.47	76.37 ± 0.35	45.75 ± 0.96	48.81 ± 1.33
UIE + LLM Augmentation (GPT-4o-Mini)								
Synthesize-and-Label	47.80 ± 0.90	61.24 ± 1.18	**47.55 ± 1.19**	**50.53 ± 1.58**	68.24 ± 0.96	76.42 ± 1.31	47.00 ± 1.33	50.60 ± 1.36
Argument Replacement (ours)	**49.32 ± 1.27**	61.31 ± 1.57	43.92 ± 1.69	48.86 ± 2.15	**69.62 ± 0.72**	**78.12 ± 0.61**	**48.19 ± 0.17**	**51.21 ± 0.32**
Adjunction Rewriting (ours)	45.02 ± 1.72	57.36 ± 1.14	47.07 ± 2.88	49.00 ± 2.49	68.00 ± 0.46	75.79 ± 0.65	46.17 ± 1.02	48.96 ± 1.39
Argument + Adjunction (ours)	45.37 ± 2.36	58.41 ± 1.76	44.43 ± 2.09	49.08 ± 1.97	68.88 ± 0.84	77.18 ± 0.52	47.45 ± 0.94	50.36 ± 0.96
Annotation Generation (ours)	44.62 ± 0.67	**62.45 ± 0.76**	40.11 ± 0.70	46.34 ± 0.56	63.67 ± 0.55	73.18 ± 0.58	45.04 ± 0.41	50.47 ± 0.99
UIE + LLM Augmentation (DeepSeek-R1-Distill-Qwen-7B)								
Synthesize-and-Label	43.93 ± 1.39	57.69 ± 1.66	41.66 ± 3.32	45.46 ± 2.33	67.09 ± 0.57	75.70 ± 0.32	43.25 ± 1.19	46.70 ± 1.83
Argument Replacement (ours)	**44.08 ± 1.13**	**58.67 ± 1.60**	40.99 ± 2.40	46.18 ± 1.92	**68.63 ± 1.29**	**77.26 ± 0.92**	44.58 ± 1.51	**48.19 ± 2.22**
Adjunction Rewriting (ours)	43.95 ± 1.95	57.42 ± 1.52	**45.30 ± 2.25**	**47.94 ± 2.42**	68.36 ± 0.73	76.15 ± 0.83	**45.57 ± 0.83**	48.15 ± 1.15
Argument + Adjunction (ours)	39.73 ± 1.24	55.09 ± 1.05	41.65 ± 0.88	46.45 ± 1.57	67.74 ± 1.62	75.83 ± 2.03	44.42 ± 1.44	48.07 ± 1.59
Annotation Generation (ours)	30.03 ± 0.65	49.77 ± 1.19	34.02 ± 1.28	38.91 ± 1.16	62.62 ± 0.58	72.87 ± 0.77	37.00 ± 0.85	43.05 ± 1.31

Table A6: LLM augmentation performance with UIE (n=200, 4x augmentation). The mean ± standard deviation over five runs is reported.

	Broken JSON		Format Violation		Invalid Trigger		Invalid Arguments		Total Filter Rate	
	GPT	DeepSeek	GPT	DeepSeek	GPT	DeepSeek	GPT	DeepSeek	GPT	DeepSeek
Argument Replacement	6	231	0	1187	5063	3267	2004	3702	34%	45%
Adjunction Rewriting	11	359	0	391	2636	5886	3722	4872	31%	62%
Argument + Adjunction	13	292	0	230	4224	6447	4087	3529	30%	47%
Annotation Generation	2	72	0	207	7	265	592	2168	14%	65%

Table A7: Filtering statistics for the augmented GENEVA dataset. “GPT” denotes GPT-4o-Mini, while “DeepSeek” denotes DeepSeek-R1-Distill-Qwen-7B.

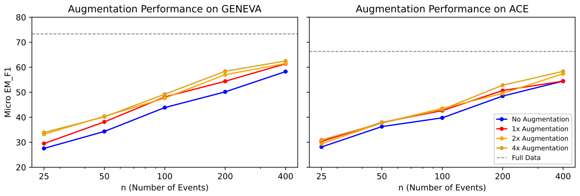


Figure A4: Micro_EM_F1 scores for models trained with varying data sizes and augmentation ratios. GENEVA is augmented using *argument replacement*, while ACE employs *adjunction rewriting*. All scores are averaged over five runs.

H Supplementary Data Scale Results

Figure A4 presents the results on GENEVA and ACE with varying data scales and augmentation ratios. Considering training costs, we conducted data scaling experiments using the best-performing augmentation method previously identified for each dataset.

I Ethics Statement

While our experiments indicate that leveraging LLMs for data augmentation can improve event argument extraction, it is important to note that the automatically generated data may occasionally contain factually incorrect or fabricated information. Such inaccuracies could pose risks if applied in safety-critical domains, such as healthcare contexts. Therefore, we strongly recommend that any deployment of our approach in these domains be accompanied by rigorous human verification and domain-specific safeguards.

J License For Artifacts

The Flan-T5 model used in this study is licensed under Apache-2.0. The UIE model is under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International Public License for Non-commercial use. The Deepseek-R1 model is under the MIT License. The GENEVA dataset is licensed under Creative Commons Attribution 3.0 Unported; the PHEE and CASIE datasets are under MIT License; The ACE dataset is under LDC License for non-commercial use. Our use of previous models and data adheres to their intended purposes. Additionally, we use data generated by GPT-4o-Mini solely for research purposes, in compliance with OpenAI’s Terms of Use and Usage Policies.

	Broken JSON		Format Violation		Invalid Trigger		Invalid Arguments		Total Filter Rate	
	GPT	DeepSeek	GPT	DeepSeek	GPT	DeepSeek	GPT	DeepSeek	GPT	DeepSeek
Argument Replacement	9	139	0	925	1345	1203	235	2439	9%	29%
Adjunction Rewriting	15	329	0	311	441	4621	416	1557	5%	44%
Argument + Adjunction	23	164	0	234	634	5681	231	1486	3%	30%
Annotation Generation	14	204	0	35	0	25	702	1061	16%	31%

Table A8: Filtering statistics for the augmented ACE dataset. “GPT” denotes GPT-4o-Mini, while “DeepSeek” denotes DeepSeek-R1-Distill-Qwen-7B.

	Broken JSON		Format Violation		Invalid Trigger		Invalid Arguments		Total Filter Rate	
	GPT	DeepSeek	GPT	DeepSeek	GPT	DeepSeek	GPT	DeepSeek	GPT	DeepSeek
Argument Replacement	5	119	0	515	50	707	139	3525	1%	36%
Adjunction Rewriting	2	130	0	109	2113	5909	3058	2389	34%	60%
Argument + Adjunction	11	127	0	195	3377	6397	4891	2378	28%	48%
Annotation Generation	4	62	0	4	0	12	545	1232	18%	44%

Table A9: Filtering statistics for the augmented PHEE dataset. “GPT” denotes GPT-4o-Mini, while “DeepSeek” denotes DeepSeek-R1-Distill-Qwen-7B.

	Broken JSON		Format Violation		Invalid Trigger		Invalid Arguments		Total Filter Rate	
	GPT	DeepSeek	GPT	DeepSeek	GPT	DeepSeek	GPT	DeepSeek	GPT	DeepSeek
Argument Replacement	1	203	0	1476	995	2036	137	4043	5%	36%
Adjunction Rewriting	2	386	0	430	2509	9533	4064	3475	28%	65%
Argument + Adjunction	8	196	0	231	4628	11697	7573	3261	27%	52%
Annotation Generation	0	130	0	27	7	107	822	2571	16%	56%

Table A10: Filtering statistics for the augmented CASIE dataset. “GPT” denotes GPT-4o-Mini, while “DeepSeek” denotes DeepSeek-R1-Distill-Qwen-7B.

Argument Relacement

Example 1
Sentence (from GENEVA):
Holmes[manipulator] smiled and *rubbed*[hold] **his hands .**[entity]
GPT Augmentation:
Holmes[manipulator] smiled and *rubbed*[hold] **his face .**[entity]
DeepSeek Augmentation:
Holmes[manipulator] nodded and *Rub*[hold] **his hands .**[entity]

Example 2
Sentence (from ACE):
later , **he**[entity] ' s supposed to helicopter to **jericho**[place] on the west bank to *meet*[meet] with the palestinian prime minister **mahmoud abbas .**[entity]
later , **he**[artifact] 's supposed to *helicopter*[transport] to **jericho**[destination] on the west bank to meet with the palestinian prime minister mahmoud abbas.
GPT Augmentation:
later , **he**[entity] ' s supposed to helicopter to **ramallah**[place] on the west bank to *meet*[meet] with the israeli prime minister **yair lapid .**[entity]
later , **he**[artifact] 's supposed to *helicopter*[transport] to **haifa**[destination] on the west bank to meet with the local council head yossi benayoun .
DeepSeek Augmentation:
While **he**[entity] was on vacation, he was supposed to fly to **jericho**[place] on the west bank to *meet*[meet] with the palestinian prime minister, **Mahmoud Abbas.**[entity]
later, **a delegation**[artifact] 's supposed to *helicopter*[transport] to **hebron**[destination] on the west bank to meet with the palestinian prime minister mahmoud abbas.

Adjunction Rewriting

Example 3
Sentence (from ACE):
We renewed our vows on Carnival Conquest on 10 / 31 / 2004 – **you**[entity] need to *contact*[phone write] Carnival wedding department and ask if they have any wedding / renewal packages available .
GPT Augmentation:
We renewed our vows on Carnival Conquest on 10 / 31 / 2004 – it is essential that **you**[entity] *contact*[phone write] Carnival wedding department and inquire if they have any wedding / renewal packages available.
DeepSeek Augmentation:
We *contacted*[phone write] the Carnival Conquest on 10 / 31 / 2004 to ask **you**[entity] about the availability of wedding or renewal packages through the Carnival wedding department.

Argument Replacement + Adjunction Rewriting

Example 4
Sentence (from ACE):
The recent rumors about **WalMart**[buyer] 's plans to *buy out*[transfer ownership] a shipping **company**[artifact] have given rise to the fear that the company may go the conglomerate route.
GPT Augmentation:
Amid the recent rumors about **eBay**[buyer]'s plans to *buy out*[transfer ownership] a **tech startup**[artifact], analysts are concerned about the implications for market competition.
DeepSeek Augmentation:
Due to rumors circulating about an **organization**[company]'s intentions, there is concern that the organization might consider a *buy out*[transfer ownership] of **WalMart**[buyer].

Example 5
Sentence (from CASIE):
All it takes is for **one black hat**[discoverer] to *find*[discover vulnerability] **a new zero - day networking software flaw**[vulnerability] and **wide - spread ransomworm**[capabilities] becomes a real possibility.
GPT Augmentation:
When **one security researcher**[discoverer] *finds*[discover vulnerability] **a new zero - day exploit in the system**[vulnerability], the potential for a **wide - spread phishing attack**[capabilities] becomes a significant concern.
DeepSeek Augmentation:
An unknown cybersecurity researcher[discoverer] only needs to *find*[discover vulnerability] **a zero-day attack in the aerospace industry**[vulnerability] and **a sophisticated hacking tool used in industrial control systems**[capabilities] to make it a real possibility.

Table A11: Representative examples of erroneous data augmentation. "GPT" denotes GPT-4o-Mini, while "DeepSeek" denotes DeepSeek-R1-Distill-Qwen-7B.