Revisiting Low-Resource Event Argument Extraction: Exploring Effective Use of LLMs for Data Augmentation

Anonymous ACL submission

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. Our experiments highlight the significance and effectiveness of enhancing argument diversity in low-resource EAE, with argument replacement demonstrating the best performance among all augmentation methods and surpassing the previous LLM-based approach. Additionally, we conduct a comprehensive evaluation from multiple perspectives, including task characteristics and data scale, providing valuable insights for the practical application of EAE in low-resource scenarios[1].

1 Introduction

Event Argument Extraction (EAE) is a key subtask of Event Extraction (EE) 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 requires a fine-grained semantic understanding of arguments and faces significant challenges, including the diversity and imbalance of argument roles, as well as the flexibility of argument boundaries. Although recent advancements in LLMs have demonstrated strong capabilities across various NLP tasks, their performance on EAE remains inferior to that of fine-tuned models (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.

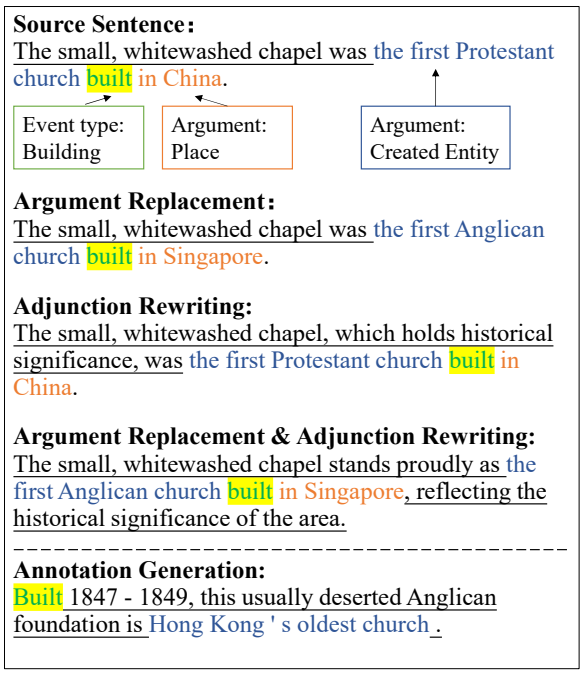


Figure 1: Examples of different augmentation methods.

Data augmentation is an effective approach to mitigating data scarcity. However, boundary sensitivity is a critical consideration when generating data for EAE. Prior studies, which have been shown to be effective in EAE, have 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). However, most prior research relies on knowledge base matching for argument replacement or small language models for adjunction rewriting, which intro-

[1]The code will be publicly available on GitHub.

duces certain limitations. For instance, argument replacement is typically restricted to predefined entity types, whereas in real-world tasks, arguments often appear as spans of varying lengths rather than fixed entities.

LLMs, with their extensive knowledge and strong text generation capabilities, offer a promising solution for data augmentation in EAE. However, research on LLM-based augmentation for EAE remains limited, with most studies overlooking the task's inherent boundary sensitivity (Sun et al., 2024; Meng et al., 2024). We argue that using LLMs for argument relabeling is inherently constrained by their extraction performance, introducing additional noise that may undermine 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** methods, as they preserve the positions of original arguments when generating new samples. In contrast, *annotation generation* investigates the impact of LLM-generated labels, which are widely used in previous methods, 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 experiments on two datasets with distinct characteristics: GENEVA (Parekh et al., 2023), a general-domain dataset covering hundreds of event and argument types, and PHEE (Sun et al., 2022), a medical-domain dataset with dense argument annotations but a limited set of types. **Our experiments reveal the following findings**: (i) Boundary-aware data augmentation methods all effectively enhance EAE performance, with *argument replacement* yielding the most significant improvement. On the argument-diverse general–domain dataset GENEVA, it increases Micro F1 by 8%. However, combining *argument replacement* with *adjunction rewriting* does not yield additional benefits, while using LLM for *annotation generation* can lead to performance degradation. (ii) The impact of data scaling is substantially greater on GENEVA than on PHEE, and augmentation proves more effective on GENEVA. This suggests that in low-resource EAE, the primary challenge lies in learning diverse argument semantics rather than handling complex argument structures or domain knowledge. Notably, LLM-based argument augmentation effectively addresses this challenge. (iii) Boundary-aware data augmentation improves both Micro F1 and Macro F1, though the relative gain in Micro F1 is higher. This indicates that the proposed augmentation methods help mitigate argument imbalance but still leave room for further improvement, warranting future research. Additionally, we provide in-depth analysis from the perspectives of **extraction errors**, **augmentation quality**, and **data scale**, offering valuable insights for EAE in low-resource scenarios.

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 more fine-grained semantic understanding and faces additional challenges due to the diversity and imbalance of argument roles, as well as the flexibility of argument boundaries. Early EAE methods were predominantly **classification-based**, involving the selection of candidate argument spans followed by the assignment of argument roles (Pouran Ben Veyseh et al., 2020; Ma et al., 2022b; He et al., 2023). However, classification-based methods struggled with overlapping arguments and have been surpassed by **generation-based** approaches in recent years. Generation-based methods 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). Recently, some studies have further reformulated the EAE task into a **Question Answering (QA) paradigm**, where argument role definitions are converted into questions, and the model generates answers (i.e., argument extractions) (Li et al., 2020; Du and Cardie, 2020; Sun et al., 2022). QA-based EAE models can be categorised into **extractive** and **generative** types

based on their base models, i.e., 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 scenarios. With the advancement of LLMs, some studies have also explored **LLM-based prompting and in-context learning approaches** for EAE (He et al., 2024; Sun et al., 2024; Sainz et al., 2024). However, due to the complexity of the EAE task, LLMs still fall short of achieving the performance of fine-tuned models in this domain.

Data Augmentation for EAE The EAE task often suffers from limited training resources due to annotation complexity, with data augmentation serving as a practical approach to alleviating low-resource challenges. General text augmentation techniques, such as text paraphrasing (Wei and Zou, 2019) and back translation (Shleifer, 2019), may alter the positions of arguments within a sentence, complicating label generation 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 leverages 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 other instances of the same type (Yang et al., 2019; Hong et al., 2022; Wang and Huang, 2024). However, it is constrained by predefined entity types, which limit the scope of replacements, and suffers from ambiguity in argument-entity matching, which can result in substitutions that do not fully align with 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 existing challenges. However, their application in this area remains limited. While some studies have explored LLMs for EAE data augmentation, they have largely overlooked boundary sensitivity and shown only marginal improvements (Sun et al., 2024; Meng et al., 2024). In this work, we conduct a broader investigation into LLM-based augmentation strategies specifically tailored for EAE and assess their effectiveness.

3 Method

In this section, we first define and formalise the event argument extraction task and then present four LLM-based data augmentation strategies explored in this work.

3.1 Task Formalisation

Event argument extraction is a subtask of event extraction, where an *event* is typically characterised by its *type*, *trigger*, and a set of *arguments*. *Event arguments* represent specific pieces of information related to the event, which can be either entities or non-entity spans that provide contextual details. The semantic scope of each argument is defined by its *role*. The task of EAE is to extract the appropriate argument for each role based on a given sentence, an event type, and its trigger.

We formalise EAE as a QA-style text generation task, derived from 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 the set of 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 each of its corresponding argument roles $r_{i,j}$, we construct the following input:

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

where `<SENTENCE>` represents the sentence s, `<EVT_TYPE>` denotes the event type of e_i, `<EVT_TRIGGER>` refers to the trigger word of e_i, and `<ARG_ROLE>` specifies the role $r_{i,j}$ of the argument to be extracted. The model's expected output is the text span corresponding to the argument $a_{i,j}$. In principle, this framework can be fine-tuned based on any language model, making it highly adaptable. We use Flan-T5 (Chung et al., 2024) as our primary experimental backbone due to its efficient performance in low-resource EAE. We refer to this base model as **Flan-T5 EEQA**.

3.2 LLM-based Data Augmentation

We investigate four LLM-based data augmentation methods for event argument extraction: *argument replacement*, *adjunction rewriting*, *their combination*, and *annotation generation*. Figure 1 presents example instances generated by these methods. We employ GPT-4o Mini (OpenAI, 2024) to generate the augmented data.

Argument Replacement Using LLMs for argument replacement involves prompting the model to generate new arguments that align with the event schema's role definitions and fit the sentence context while keeping the rest of the sentence unchanged. This approach leverages LLMs' strong language understanding and extensive knowledge to generate diverse arguments, thereby enhancing the fine-tuned model's ability to learn argument semantics. Unlike traditional knowledge bases, LLMs are not limited to predefined entities when generating new arguments. Moreover, maintaining the rest of the sentence unchanged helps preserve boundary accuracy, which is critical 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 LLM in adhering to the expected output structure. Specifically, the input consists of *a sentence* from the training set, the annotated *event type*, *event trigger*, and *arguments*, along 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 should remain unchanged, we explicitly retain them in the output to reinforce the LLM's adherence to 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 the rest of the sentence—i.e., adjunctions—while keeping the arguments unchanged. This approach aims to increase sentence diversity while ensuring argument boundary accuracy, thereby enhancing the fine-tuned model's generalisation ability without compromising precision. To ensure consistency and quality in the generated samples, we adopt the same prompt and input-output structure as in *Argument Replacement*, modifying only the instruction and example. We also apply filtering rules to remove invalid outputs.

Argument Replacement & Adjunction Rewriting The combination of argument replacement and adjunction rewriting progressively enhances sample diversity by first generating new arguments and then rewriting the rest of the sentence. To reduce costs in our experiments, we apply adjunction rewriting to the outputs of argument replacement, efficiently generating additional augmented data.

Annotation Generation Using LLMs for annotation generation leverages their predictive capabilities to create weakly supervised labels for unlabeled source texts. The key advantage of this approach is the unrestricted availability of source data, allowing extensive sampling from domain-specific texts to ensure authenticity and diversity. However, despite LLMs' strong reasoning abilities, event extraction tasks often rely on complex annotation rules and require precise boundary identification. As a result, LLMs struggle to generate high-quality annotations within limited in-context demonstrations, leading to significant label noise that can ultimately degrade the accuracy of fine-tuned models.

In our low-resource experimental setup, we augment data using samples from the full training set that are excluded from the low-resource subset, with LLM-generated predictions serving 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. These retrieved samples, along with their inputs and annotations, serve as in-context demonstration examples to prompt the LLM for argument extraction. 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 model training, we filter out augmented samples that duplicate the training instances to maintain data integrity.

4 Experimental Setup

This section provides fundamental information on the experimental setup, with more details available in Appendix B.

Datasets We conduct experiments on two datasets: PHEE (Sun et al., 2022) and GENEVA (Parekh et al., 2023). PHEE is a medical-domain event extraction dataset annotated with two event types—adverse event and potential therapeutic event—each with 16 argument roles related to sub-

	GENEVA (n=200)		GENEVA (Full)		PHEE (n=200)		PHEE (Full)	
	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	45.02	62.83	64.12	75.92	68.37	78.87
EEQA	25.26 ± 2.97	26.95 ± 1.93	57.72	55.06	45.20 ± 0.77	40.56 ± 0.82	53.57	46.95
UIE	45.08 ± 1.21	55.98 ± 1.58	**75.19**	**82.60**	67.91 ± 0.99	75.72 ± 1.12	**76.60**	82.97
Flan-T5 EEQA (ours)	50.15 ± 2.80	58.87 ± 3.98	73.33	80.13	69.78 ± 0.97	77.57 ± 1.22	75.02	82.09
Flan-T5 EEQA + Synthesize-and-Label	54.33 ± 1.19	64.88 ± 1.26	72.49	80.37	69.23 ± 1.16	77.41 ± 1.50	71.57	79.29
Flan-T5 EEQA + Argument Replacement (ours)	**58.39 ± 1.30**	**67.68 ± 1.38**	74.32	81.35	**70.99 ± 0.42**	**79.17 ± 0.76**	76.45	**84.24**

Table 1: Overall performance. For low-resource training, the mean ± standard deviation over five runs is reported for fine-tuning methods, while GPT-4o Mini is evaluated on a single subset due to cost constraints.

ject, treatment, and effect. Although PHEE follows a hierarchical argument annotation scheme, we treat all arguments as flat for consistency. GENEVA, in contrast, is a general-domain event extraction dataset containing 115 event types and 220 argument roles, making it broader in scope than previous general-domain EE datasets such as ACE (Doddington et al., 2004). We choose these datasets for their complementary characteristics: PHEE represents a domain-specific dataset with a small number of event types but dense argument annotations, while GENEVA is a large-scale dataset with diverse event types but fewer arguments per event (averaging four arguments per event). These differences allow us to evaluate model performance across varying event and argument distributions. Appendix C provides dataset statistics and annotation examples.

Low-resource Training Low-resource training involves randomly sampling n event mentions to construct the training dataset (Parekh et al., 2023), while keeping the validation and test sets unchanged. Unlike few-shot training, which selects k samples per event type, low-resource training preserves the natural distribution of events and arguments, making it more representative of real-world scenarios. Therefore, we adopt it as the primary research setting in this study. We conduct experiments across different resource levels, ranging from low ($n = 25$) to moderate ($n = 400$), and compare the results with *fully supervised training*. For data augmentation, we generate additional samples at $\{1\times, 2\times, 4\times\}$ the size of the original training data per event mention.

Evaluation Metrics Considering that arguments may consist of long spans, making exact matching difficult, we follow previous work (Sun et al., 2022) to evaluate 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 for evaluation of partial matches. In addition, we also report Micro_F1 and Macro_F1. **Micro_F1** is computed over all arguments by accumulating true positives (TP) before computing F1. For **Macro_F1**, we account for different dataset characteristics—some being argument-dense and others event-dense—by separately computing **Arg_Macro_F1**, which is the average F1 score across *argument types*, and **Evt_Macro_F1**, which is the average F1 score across *event types*. Therefore, we evaluate model 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 select the following methods as *baselines for event argument extraction*: (i) **GPT-4o Mini**: We reproduce the method proposed by Sun et al. (2024) and use GPT-4o Mini (OpenAI, 2024) as the base model to establish the LLM in-context learning (ICL) baseline. This approach retrieves the five most similar training samples for each test instance and uses them as demonstrations to prompt the LLM for event extraction. Specifically, for EAE, we include the event type and trigger word in the input. In addition, when sufficient retrieval samples are available, we prioritise demonstrations with the same event type. (ii) **EEQA**: We adopt the method proposed by Du and Cardie (2020) as a representative extractive QA-based approach for EE. Its core idea aligns with our framework (subsection 3.1), leveraging label semantics as questions and employing a question-answering objective to extract arguments. However, EEQA uses an encoder-only backbone (e.g., BERT), whereas our method adopts an encoder-decoder architecture (e.g., Flan-T5). We

also experimented with a decoder-only model (e.g., Llama3) but found that causal language models perform poorly when fine-tuned via teacher-forcing in low-resource settings, often failing to generate reasonable answers. Consequently, we exclude causal language model results from our experiments. (iii) **UIE**: The UIE model (Lu et al., 2022) is a representative structured text generation model for information extraction that linearises event structures and trains within a seq-to-seq framework. Pre-trained on a large-scale structured information extraction dataset, UIE has demonstrated strong few-shot generalisation in prior studies. To adapt it for EAE, we incorporate event type and trigger word information into the input.

For the *data augmentation baseline*, since most previous EAE data augmentation methods were based on small models and lacked open-source code (Yang et al., 2019; Gao et al., 2022; Meng et al., 2024), we compare against the approach by Sun et al. (2024), aligning with our focus on LLM-based augmentation. This approach inputs a sample and its annotated event into GPT-3.5, prompting it to generate a new sentence with a similar event structure and extract events from the generated text. To ensure a fair comparison, we implement their method using GPT-4o Mini, ensuring consistency with our strategies. Additionally, their original work applies a filtering strategy based on perplexity estimation from a fine-tuned model, but this results in extremely low data retention, discarding over two-thirds of the generated samples. For evaluation, we instead apply the same filtering rules as used in our proposed methods. We denote this data augmentation method as **Synthesize-and-Label**.

5 Results and Analysis

5.1 Overall Performance

Table 1 compares *argument replacement*, the best-performing EAE data augmentation method among the four proposed in this study (Section 5.2), against other baselines. Comprehensive results for all metrics are provided in Appendix D.

Overall, our base model, Flan-T5 EEQA, outperforms all baselines, achieving the highest performance under low-resource conditions. **The proposed *argument replacement* data augmentation method further enhances the base model, surpassing the compared data augmentation baseline and demonstrating its effectiveness for low-resource EAE.**

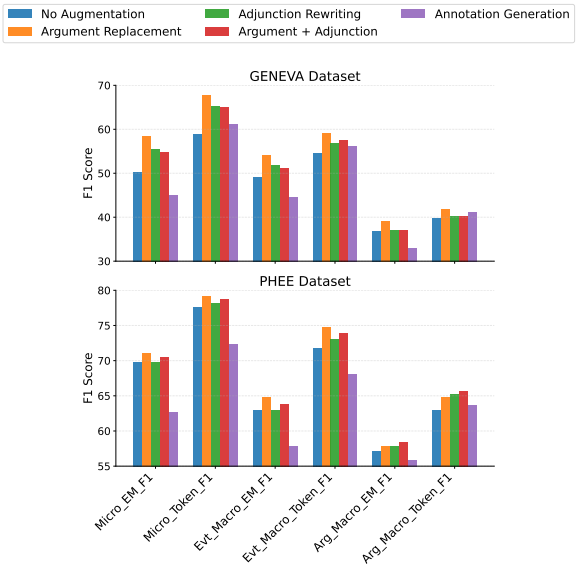


Figure 2: Comparison of data augmentation methods in the low-resource setting (n=200), with scores averaged over 5-fold experiments. All data augmentation methods shown use 4× augmented data.

Specifically, the performance gains from data augmentation are more pronounced on the GENEVA dataset, likely due to its broader set of argument types, making it more data-hungry. Our method consistently outperforms *Synthesize-and-Label* (Sun et al., 2024) across all metrics, achieving over an 8% performance gain under the low-resource setting in both Micro_EM_F1 and Micro_Token_F1 after augmentation.

In contrast, the impact of data augmentation on the PHEE dataset is more limited, yielding smaller performance gains. However, the performance gap between low-resource (n=200) and full-data (n=3000) training on PHEE is only 6%, compared to 23% on GENEVA (n=4163 for full fine-tuning), indicating lower data scarcity. Unlike GENEVA, PHEE contains densely annotated arguments per sentence but covers fewer event and argument types overall. This suggests that when argument semantics are more concentrated, fewer training resources are needed to achieve competitive performance, making data augmentation particularly beneficial in extremely low-resource settings.

5.2 Comparison of Data Augmentation Approaches

In this subsection, we analyse the differences among the four LLM-based EAE data augmentation methods proposed in this study from various perspectives.

Extraction Performance Figure 2 presents the performance of different data augmentation strategies across multiple metrics.

First, we observe that all boundary-aware augmentation methods, i.e., *argument replacement*, *adjunction rewriting*, and *their combination*, improve performance across all metrics, while using LLM-based *annotation generation* for data augmentation degrades performance across most metrics. It is expected that the limited EAE extraction capability of LLMs inevitably introduces noise into generated labels, undermining the effectiveness of data augmentation. However, boundary-aware methods mitigate this issue, allowing the generated samples to yield improvements even in exact matching evaluation. This highlights **the crucial role of preserving boundary accuracy in effective EAE data augmentation**.

Second, both *argument replacement* and *adjunction rewriting* prove effective for data augmentation, with *argument replacement* yielding the highest average performance across both datasets. This suggests that **both argument and semantic diversity are important for low-resource EAE, with argument semantics being more critical and benefiting more from data augmentation**. One possible explanation is that even small language models acquire some degree of text representation generalisation through pre-training, allowing them to present reasonable ability with limited training data. However, argument semantics are often task-specific and not sufficiently learned during pre-training, making them more dependent on data augmentation. Additionally, combining *argument replacement* with *adjunction rewriting* does not yield additional gains, likely because **semantic diversity is already enhanced as a byproduct of argument replacement**, making the extra adjunction rewriting step an unnecessary overhead without further benefits.

Third, the impact of these augmentation methods varies across different evaluation metrics. Performance gains are more pronounced in Micro_F1 and Evt_Macro_F1, while improvements in Arg_Macro_F1 remain relatively marginal. This indicates that argument imbalance is more severe than event imbalance in low-resource EAE and **current augmentation strategies enhance the extraction of rare arguments to some extent but do not fully resolve the data imbalance problem**.

	Unmatch	Partial Match	Spurious Argument (Role Error)	Argument Missing (Role Error)
No Augmentation	25	1052	1701 (579)	375 (74)
Argument Replacement	15	863	902 (299)	543 (95)
Adjunction Rewriting	19	918	864 (276)	604 (81)
Argument & Adjunction	19	960	988 (306)	557 (87)
Annotation Generation	29	1290	1587 (281)	264 (36)

Table 2: Argument extraction error analysis on the GENEVA (n=200) test set for models trained with different data augmentation (4x) methods. Values represent averages over five runs.

Argument Extraction Error Analysis To analyse the impact of different data augmentation methods on EAE, we developed an automated script to classify extraction errors in models trained with different augmented datasets. Table 2 reports error statistics for GENEVA. Results for PHEE and error type definitions are provided in Appendix E.

The results indicate that *partial match* and *spurious arguments* are the most frequent errors, whereas *argument missing* occurs less often, and fully *unmatched* spans are rare. Among the four data augmentation methods, *argument replacement*, *adjunction rewriting*, and *their combination* substantially reduce *partial match* errors, whereas LLM-based *annotation generation* increases them in fine-tuned models, demonstrating the effectiveness of boundary-aware methods in preserving boundary accuracy of arguments. Additionally, these boundary-aware strategies also significantly reduce *spurious arguments* albeit slightly increase *argument missing*, suggesting improved argument semantic learning. In contrast, LLM-based *annotation generation* fails to reduce *spurious arguments* effectively. Notably, on the PHEE dataset, it even increases *spurious argument* errors, underscoring the risk of misalignment between the LLM's internal knowledge and task-specific requirements. Furthermore, some *spurious argument* and *argument missing* errors result from role confusion, where the model misclassifies argument types within the same event. This is likely influenced by argument co-occurrence patterns in the training data, but this type of error remains relatively infrequent.

Quality of Augmented Data Our manual inspection of the augmented data reveals that LLM-generated samples are generally semantically accu-

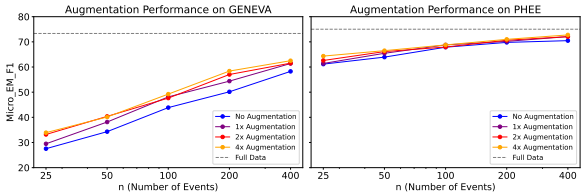


Figure 3: Micro_EM_F1 scores for models trained with varying data sizes and augmentation ratios using '*argument replacement*'. Scores are averaged over five runs.

rate and syntactically fluent. The newly generated arguments align well with the sentence context and definitions, while adjunction rewriting enhances sentence details beyond simple paraphrasing while preserving arguments.

However, we still observed cases where the LLM deviates from instructions. First, it struggles to keep the event trigger unchanged when the trigger is part of a replaced argument and may omit it during adjunction rewriting. Another common issue is that the model sometimes generates arguments without an exact match in the sentence, mostly due to tokenization mismatches in preprocessing. For example, the input provided to the LLM may include tokenized text such as '*Russia ' s*', while the model generates '*Russia 's*' as an argument span. Additionally, we observed occasional hallucinations when LLM generates data for the PHEE dataset, where for certain two-word argument types, such as '*time elapsed*', the LLM sometimes generates arguments incorrectly labelled as '*time_elapsed*'. While these issues do not impact sentence-level semantics, they introduce noise in argument extraction. Optimisation to the prompt or additional preprocessing steps may mitigate these errors, but given their low occurrence rate, we simply filtered these cases. Appendix F provides statistical details on different error types.

Specifically, for *argument replacement*, we observed that when an event contains multiple arguments, the LLM sometimes replaces only a subset (see Table A9 for examples). This likely occurs when certain arguments are semantically ambiguous or closely tied to the sentence context. While this does not generate incorrect samples, it may limit the effectiveness of data augmentation, worth further investigation in future work.

5.3 Impact of Data Scale

To assess the effectiveness and efficiency of data augmentation under varying resource conditions, we evaluate model performance across different data sizes and augmentation ratios, as illustrated in Figure 3. Analysing model performance across different augmentation ratios, we observe the following: (i) The performance variation due to different augmentation amounts is smaller than the difference between using and not using data augmentation. Additionally, adding an equivalent amount of augmented data yields lower gains than adding the same amount of original data, suggesting that **augmented data exhibits a degree of homogeneity, impacting augmentation efficiency**. However, this trade-off is necessary to maintain annotation accuracy, and balancing accuracy with efficiency remains a challenge for future research. (ii) **Even as the amount of original training data increases, augmented data continues to provide noticeable improvements**. On the GENEVA dataset, this improvement remains consistent, whereas on PHEE, it shows a declining trend. This suggests that for argument extraction with extensive roles, while increasing training data helps the model capture a broader range of argument types, further improvements in learning argument semantics can be achieved through augmentation with richer contexts. In contrast, for PHEE, where argument diversity is lower, this need diminishes as training data increases. (iii) **Higher augmentation ratios stably improve performance**. However, lower ratios sometimes achieve comparable results, making them a cost-effective alternative when computational resources are constrained.

6 Conclusion

This study explores multiple LLM-based data augmentation strategies for low-resource EAE. Our findings show that boundary-aware augmentation are more effective, with LLM-based argument replacement achieving the greatest improvements. This underscores the importance of preserving boundary accuracy and enhancing argument diversity in data augmentation for EAE. However, the augmented data generated by argument replacement exhibits a degree of homogeneity, potentially limiting its effectiveness. Moreover, existing methods provide only marginal improvements in extracting rare arguments, highlighting the need for further research to mitigate data imbalance and enhance augmentation efficiency.

Limitations

This study primarily focuses on directly applying LLM-generated data for augmentation, using a simple filtering strategy to ensure data validity. More advanced filtering techniques or noise-tolerant training approaches may further enhance the effectiveness of certain augmentation methods. However, due to space constraints, we leave these explorations for future work.

Given computational limitations, we evaluate data augmentation strategies using only the best-performing base model under the low-resource setting. Nevertheless, as our proposed augmentation methods are model-agnostic, the conclusions drawn in this study should remain broadly applicable. Similarly, we assess data scaling only for *argument replacement*, as it is the most effective among our proposed methods, making a deeper investigation into its resource requirements particularly meaningful.

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.

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.

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*.

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.

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 E 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

Figure A1 presents annotated examples from the GENEVA and PHEE datasets. Table A2 provides statistical information for the GENEVA dataset, while Table A3 summarises the statistics for the PHEE dataset.

D Supplementary Performance Tables

Table A4 reports the low-resource performance across all metrics, while Table A5 presents the full fine-tuning results.

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.

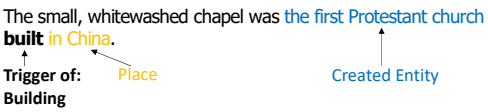
Input Example:

```
{
    "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 ..."
      }
    }
  }
```

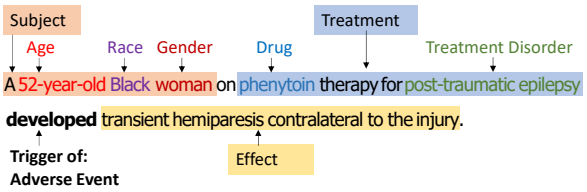
Output Example:

```
{
  "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*.



(a) An annotated example from the GENEVA dataset.



(b) An annotated example from the PHEE dataset.

Figure A1: Illustration of event annotations in the GENEVA and PHEE datasets. 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.

E Extraction Error Type Definitions and Statistics

We categorize 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.
- **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.

	# 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 A3: Statistics of the PHEE dataset.

Table A6 presents the error type analysis for models trained with different data augmentation methods on the PHEE dataset.

F Supplementary of Augmented Data Quality

Table A7 and Table A8 present the error type statistics for different data augmentation methods on the GENEVA and PHEE datasets. The error types are defined as follows: **Broken JSON** indicates that ChatGPT generated an unparseable JSON format; **Invalid Trigger** refers to cases where the event trigger in the augmented sample does not appear in the corresponding sentence; **Invalid Role** occurs when ChatGPT generates an argument type that is not defined in the schema; and **Invalid Argument** signifies that an argument in the augmented sample cannot be matched to the corresponding sentence.

Table A9 shows a generated sample for argument replacement, where only a subset of arguments is replaced.

G Potential Risks

Although our experiments demonstrate that leveraging LLMs for data augmentation can enhance event argument extraction, the generated data may introduce factually incorrect hallucinations, posing potential risks when applied to safety-critical domains such as healthcare.

H License For Artifacts

The Flan-T5 model used in this study is licensed under Apache-2.0. The GENEVA dataset is licensed under Creative Commons Attribution 3.0 Unported, and the PHEE dataset is under MIT License. Our use of previous models and data adheres to their intended purposes. Additionally, we use data generated by ChatGPT solely for research purposes, in compliance with OpenAI's Terms of Use and Usage Policies.

	GENEVA					
	Micro_EM_F1	Micro_Token_F1	Arg_Macro_EM_F1	Arg_Macro_Token_F1	Evt_Macro_EM_F1	Evt_Macro_Token_F1
GPT-4o Mini	38.54	57.41	35.58	50.63	17.78	26.04
EEQA	25.26 ± 2.97	26.95 ± 1.93	23.93 ± 2.82	23.59 ± 2.32	16.78 ± 2.01	16.11 ± 1.74
UIE	45.08 ± 1.21	55.98 ± 1.58	40.87 ± 1.59	46.43 ± 1.78	25.04 ± 2.21	27.43 ± 2.73
Flan-T5 EEQA (ours)	50.15 ± 2.80	58.87 ± 3.98	48.98 ± 2.44	54.48 ± 2.91	38.09 ± 1.96	41.13 ± 2.33
Flan-T5 EEQA + Synthesize-and-Label	54.33 ± 1.19	64.88 ± 1.26	51.44 ± 1.20	57.50 ± 1.12	39.09 ± 2.41	43.03 ± 2.30
Flan-T5 EEQA + Argument Replacement (ours)	**58.39 ± 1.30**	**67.68 ± 1.38**	**54.13 ± 1.28**	**58.97 ± 1.12**	**41.25 ± 2.66**	**44.22 ± 2.87**
	PHEE					
GPT-4o Mini	64.12	75.92	57.29	71.36	48.30	61.07
EEQA	45.20 ± 0.77	40.56 ± 0.82	39.44 ± 0.65	36.95 ± 1.07	37.45 ± 2.89	34.50 ± 2.90
UIE	67.91 ± 0.99	75.72 ± 1.12	61.09 ± 1.16	70.46 ± 1.24	48.38 ± 0.93	55.04 ± 0.93
Flan-T5 EEQA (ours)	69.78 ± 0.97	77.57 ± 1.22	62.89 ± 1.01	71.74 ± 1.52	57.11 ± 1.69	62.98 ± 1.83
Flan-T5 EEQA + Synthesize-and-Label	69.23 ± 1.16	77.41 ± 1.50	63.07 ± 1.26	72.65 ± 2.00	**58.60 ± 1.87**	**65.77 ± 2.67**
Flan-T5 EEQA + Argument Replacement (ours)	**70.99 ± 0.42**	**79.17 ± 0.76**	**64.81 ± 0.52**	**74.78 ± 0.77**	57.76 ± 1.82	64.74 ± 2.57

Table A4: Low-resource (n=200) performance across all metrics. The mean ± standard deviation over five runs is reported for fine-tuning methods, while GPT-4o Mini is evaluated on a single subset due to cost constraints.

	GENEVA					
	Micro_EM_F1	Micro_Token_F1	Arg_Macro_EM_F1	Arg_Macro_Token_F1	Evt_Macro_EM_F1	Evt_Macro_Token_F1
GPT-4o Mini	45.02	62.83	42.04	55.41	25.18	34.47
EEQA	57.72	55.06	53.28	48.07	43.40	40.38
UIE	**75.19**	**82.60**	**72.63**	**77.37**	58.72	60.43
Flan-T5 EEQA (ours)	73.33	80.13	72.13	76.86	59.53	62.16
Flan-T5 EEQA + Synthesize-and-Label	72.49	80.37	71.31	76.05	58.36	61.33
Flan-T5 EEQA + Argument Replacement (ours)	74.32	81.35	71.99	76.41	**60.58**	**62.93**
	PHEE					
GPT-4o Mini	68.37	78.87	63.60	75.87	56.47	67.56
EEQA	53.57	46.95	48.65	44.17	48.30	43.15
UIE	**76.60**	82.97	71.13	79.32	65.01	70.04
Flan-T5 EEQA (ours)	75.02	82.09	68.76	77.84	65.19	72.60
Flan-T5 EEQA + Synthesize-and-Label	71.57	79.29	65.72	74.55	62.41	68.26
Flan-T5 EEQA + Argument Replacement (ours)	76.45	**84.24**	**71.29**	**81.17**	**66.29**	**73.40**

Table A5: Full fine-tuning performance across all metrics.

	Unmatch	Partial Match	Spurious Argument (Role Error)	Argument Missing (Role Error)
No Augmentation	11	1353	825 (243)	200 (33)
Argument Replacement	13	1329	599 (152)	261 (44)
Adjunction Rewriting	14	1425	662 (165)	248 (40)
Argument & Adjunction	12	1409	709 (204)	217 (39)
Annotation Generation	16	1553	1276 (445)	163 (50)

Table A6: Argument extraction error analysis on the PHEE (n=200) test set for models trained with different data augmentation (4x) methods. Values represent averages over five runs.

	Broken Json	Invalid Trigger (#. Events)	Invalid Role	Invalid Argument (#. Args)
Argument Replacement	6	5,042 (20,820)	0	2,588 (32,310)
Adjunction Rewriting	11	2,454 (20,795)	0	5,057 (32,275)
Argument & Adjunction	13	4,224 (27,480)	0	6,919 (45,404)
Annotation Generation	2	7 (4,163)	17	629 (7,732)

Table A7: Error statistics of augmented data for GENEVA.

	Broken Json	Invalid Trigger (#. Events)	Invalid Role	Invalid Argument (#. Args)
Argument Replacement	5	50 (14,995)	1,000	194 (71,370)
Adjunction Rewriting	2	1,661 (15,010)	892	4,577 (71,410)
Argument & Adjunction	11	3,377 (29,590)	1,972	7,668 (140,556)
Annotation Generation	4	0 (3,000)	0	767 (14,080)

Table A8: Error statistics of augmented data for PHEE.

Input:


```
{
    "sentence": "Surrounded by acres of
        farmland hewn from the hard
        desert , the adobe fort became a
         focal point for the development
         of Las Vegas for the next fifty
         years .",
    "event_type": "becoming",
    "trigger": "became",
    "arguments": {
        "entity": [
            "the adobe fort"
        ],
        "final category": [
            "a focal point for the
                development of Las Vegas
                 for the next fifty
                years"
        ]
    }
}
```


Augmented Sample:


```
{
    "augmented_sentence": "Surrounded by
         acres of farmland hewn from the
         hard desert , the historic
        mansion became a focal point for
         the growth of Las Vegas for the
         next fifty years .",
    "event_type": "becoming",
    "trigger": "became",
    "arguments": {
        "entity": [
            "the historic mansion"
        ],
        "final category": [
            "a focal point for the
                growth of Las Vegas for
                the next fifty years"
        ]
    }
}
```


Table A9: Example of a generated sample for *argument replacement*, where only a subset of arguments is replaced.