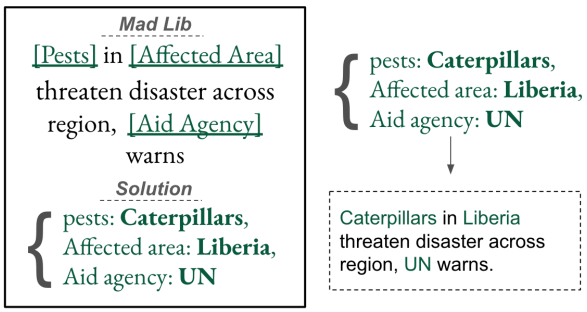
Document-Level Event-Argument Data Augmentation for Challenging Role Types

Joseph Gatto, Omar Sharif, Parker Seegmiller, Sarah M. Preum
Department of Computer Science, Dartmouth College
{joseph.m.gatto.gr}@dartmouth.edu

Abstract

Event Argument Extraction (EAE) is a daunting information extraction problem — with significant limitations in few-shot cross-domain (FSCD) settings. A common solution to FSCD modeling is data augmentation. Unfortunately, existing augmentation methods are not well-suited to a variety of real-world EAE contexts, including (i) modeling long documents (documents with over 10 sentences), and (ii) modeling challenging role types (i.e., event roles with little to no training data and semantically outlying roles). We introduce two novel LLM-powered data augmentation methods for generating extractive document-level EAE samples using *zero in-domain training data*. We validate the generalizability of our approach on four datasets — showing significant performance increases in low-resource settings. Our highest performing models provide a 13-pt increase in F1 score on zero-shot role extraction in FSCD evaluation.



(a) Mad Lib Generation and Solving **(b) Struct2Text Generation**

Figure 1: In (a) we leverage LLM's prior knowledge of Mad Libs to generate document-level extractive EAE data. In (b) we generate synthetic event structures and use them to produce event-centric documents. We employ semantic n-gram matching to align the structure with spans in the generated document for EAE data creation. Our approach generates DocEAE samples using *zero in-domain training data*, making it generalizable to new domains.

1 Introduction

In recent years, there has been considerable progress in the domain of Event Argument Extraction (EAE), as advances in question answering (Du and Cardie, 2020), prompt tuning (Ma et al., 2022), and semantic graph modeling (Yang et al., 2023b) have led to state-of-the-art results on EAE benchmarks. However, some simplifying assumptions made in prior work limit their applicability in a more complex real-world setting, including using only sentence-level EAE annotations (Doddington et al., 2004; Song et al., 2015; Parekh et al., 2023), reliance on large-scale annotation (Ebner et al., 2020; Sun et al., 2022), or that training and testing data come from the same domain (Ebner et al., 2020; Doddington et al., 2004; Sun et al., 2022; Sharif et al., 2024).

In many real-world applications, EAE systems need to be prepared to extract information from *long documents* (10+ sentences) given limited training data in a new domain. For example Tong et al. (2022); Yang et al. (2023a) demonstrate the challenge of EAE on long documents in novel domains with limited training data. In recent years, curation of event-centric datasets focusing on niche domains has become increasingly popular. For example, event extraction is now studied in the context of social media discourse (Sharif et al., 2024), clinical case reports (Ma et al., 2023a), pharmacovigilance (Sun et al., 2022), and cyber security (Satyapanich et al., 2020), among others. When considering the high-cost of large-scale document-level span annotation from domain-experts, event extraction in a low-resource, cross-domain setting becomes increasingly important.

One solution to this problem is to strategically generate training samples for event modeling in a given domain. However, many related works in EAE data augmentation such as (Gao et al., 2022;

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 25109–25131
July 27 - August 1, 2025

Ma et al., 2023b; Wang et al., 2023) only focus on *sentence-level* tasks. The few prior works in DocEAE augmentation do not generate new data samples, but rather augment existing data (Liu et al., 2022) or use a pre-trained model to weakly label unannotated corpora (Liu et al., 2021). To the best of our knowledge, no prior work provides a solution that can help resolve the following challenges in DocEAE data generation: (i) generating novel, event-specific long documents for argument span extraction, where arguments for a given event can occur **anywhere throughout the document**, (ii) generating data for **challenging role types**, including those with little to no training data, and those which are semantically outlying compared to other roles.

As a motivating example of this challenging generation task, consider the cross-domain EAE task introduced in DocEE (Tong et al., 2022) where the task is to transfer knowledge from a collection of source domain events (e.g., *Road Crash*, *Celebrity Death*, *Bank Robbery*) to a set of events in a new domain, Natural Disasters (e.g., *Droughts*, *Earthquakes*, *Tsunamis*). Each event in the target domain training set only has 5 documents available for training. Across all samples representing the ***Volcano Eruption*** event, for example, zero documents contain annotation for the role *The State of the Volcano (Dormant or Active)*. This makes this role challenging for many DocEAE models to extract. Similar challenges may arise as one tries to develop a document-level EAE model for a new domain or a new dataset, with limited labeled data, incomplete event schema, and/or challenging roles.

We address this issue by introducing two strategies for LLM-powered DocEAE data augmentation as shown in Figure 1: (i) **Mad Lib Generation (MLG)** is inspired by the insight that new DocEAE samples can be formulated as a Mad Lib, a popular game involving categorically templated documents. In MLG, we consider the "blanks" found in Mad Libs as roles in an event schema. This allows us to leverage LLMs' strong conceptual priors of Mad Libs for templated text generation. We use LLMs to both generate and solve Mad Libs in this framework. (ii) **Struct2Text (S2T):** an LLM-powered event-document generation pipeline, which can transform synthetic structured information (e.g., a dictionary of role-argument mappings) into event-centric documents. Note that both MLG and S2T are designed to generate DocEAE data using only an *event schema* (i.e., the event name and a list of valid roles per event). This makes MLG and S2T generalizable to datasets with and without in-depth ontologies.

Our contributions in this study are as follows:

1. We develop two new LLM-powered methods for cross-domain DocEAE data generation, MLG and S2T. [1] We explore our methods in the context of a diverse set of both propriety and open-source LLMs. Our top-performing methods provide a statistically significant increase in the F1-score on challenging DocEE sub-tasks such as 0-shot role extraction. We also demonstrate strong performance on task-specific metrics for performance assessment on challenging role types.

2. We demonstrate the generalizability of our proposed method via evaluation on additional EAE datasets namely RAMS (Ebner et al., 2020), DiscourseEE (Sharif et al., 2024), and PHEE (Sun et al., 2022). Experimental results show a 13.6-point average increase in F1-score on low-resource EAE tasks without relying on in-domain data for generation.

2 Related Work

2.1 Document-Level Event Argument Extraction

Early work studying Event Argument Extraction (EAE) focuses on sentence-level data, using the popular ACE05 (Doddington et al., 2004) dataset. More recently, some work has focused on document-level datasets such as RAMS (Ebner et al., 2020), WikiEvents (Li et al., 2021), DiscourseEE (Sharif et al., 2024), and DocEE (Tong et al., 2022). In this study, we primarily focus on DocEE as it is unique in providing a large dataset (20k+ samples), long documents (10+ sentences) and a pre-defined cross-domain evaluation task. For this reason, other recent works have also focused their attention on DocEE (Yang et al., 2023a). We provide secondary analysis on other relevant EAE tasks in Section 5 to demonstrate the generalizability of our methods.

2.2 Data Augmentation for EAE

To the best of our knowledge, there is no other work that aims to generate novel documents for DocEAE. However, there are adjacent works in the

[1] Paper resources can be found here https://tinyurl.com/Doc-EAE-Data-Aug

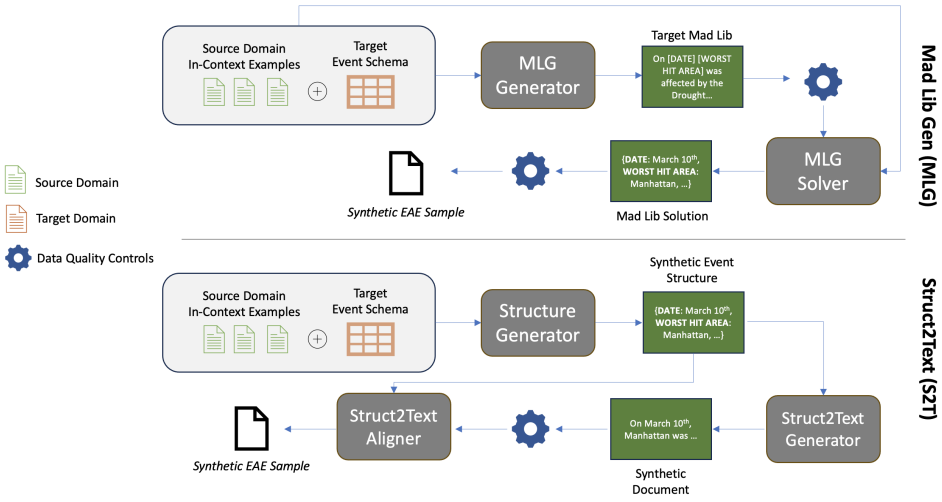


Figure 2: Visualizing our data generation pipeline for both MLG and S2T. In-context demonstrations are sampled from a source domain. Source domain demonstrations are used alongside semantics derived from the the target event schema for generation of new DocEAE samples.

realm of EAE data augmentation. Liu et al. (2021) use pre-trained EAE models to silver-label (i.e., use a pre-trained model to annotate) unannotated documents as additional data augmentation. Liu et al. (2022) use pre-trained language models to augment existing samples by masking annotated argument spans and then generating alternate ones. Gao et al. (2022) do the opposite by first masking unannotated spans, then using a pre-trained T5 model to replace such spans as data augmentation. Wang et al. (2023) perform a reinforcement-learning based solution to sentence-level EE data generation. Contrary to our work, none of the aforementioned methods generate fully novel documents and often rely on augmenting existing samples. Works that augment existing samples have limited control of role-type distribution and are not suited to model 0-shot roles, which is a focal point of our work. One recent work that leverages LLMs for sentence-level EAE data augmentation is Ma et al. (2023b) where they employ LLMs in a multi-step prompt and self-reflect structure-to-text generation pipeline. This method relies heavily on event ontology information not available for many datasets such as event definitions, role definitions, and valid entity types per role. Crucially, the list of datasets which do not have in-depth ontologies includes DocEE (Tong et al., 2022) as well as other datasets evaluated in this work, such as Sharif et al. (2024). Future works may explore how to adapt this method for a more diverse set of DocEAE tasks.

3 Methods

In Figure 2 we showcase our data generation pipeline for both Mad Lib Generation (MLG) and Struct2Text (S2T). We describe each method in detail in the following section.

3.1 Mad Lib Generation (MLG)

The goal of Mad Lib Generation is to create a templated text document with masked placeholders which each have a semantic category (See Figure 1). Mad Libs are related to information extraction tasks such as EAE as they assign categories to textual spans. In this work, we leverage LLMs to generate event-specific Mad Libs, where the masked document placeholder categories are sourced from an event schema. LLM-Generated solutions to Mad Libs create novel DocEAE samples, as masked placeholders represent argument spans in the generated document.

MLG Generator: To generate an event-driven Mad Lib, we first construct k in-context demonstrations from the source domain which map an event name and a corresponding event schema (i.e. a list of valid roles for the given event) to a Mad Lib-formatted document. Source domain documents are converted to Mad Libs by using role annotations to replace argument spans with masked placeholders (e.g. The event happened in New York $\rightarrow$ The event happened in [LOCATION]).

We use the in-context demonstrations to guide the LLM on how to map event information from the target domain into a templated document. Here,

in-context learning serves only to guide the LLM on output structure, while parametric knowledge is utilized to map target event semantics to novel documents. Note that for each generation, we aim to output a document that contains *every possible role* for a given event. Our strategy does not generate documents that selectively include/exclude certain event role subsets. We choose this strategy for the following reason. A core challenge of DocEAE data generation is the issue of *under-labeled documents*. That is, if the desired length of a document about a single event is large while the number of roles to be included is small, it is very difficult to prevent the LLM from accidentally generating an unintended role. We mitigate this issue by providing the LLM with all possible target roles as Mad Lib categories and then validate its efficacy extrinsically.

MLG Solver: Once we have generated a target domain Mad Lib, we use k in-context demonstrations to prompt the LLM to replace the templated components of the Mad Lib with contextually valid argument spans. Specifically, source-domain data is used to guide the LLM on generation of *{role : argument}* pairs that can be slotted into the templated document using string replacement (shown in Figure 1). The result is an event-centric document with annotation for argument extraction. Since each document aims to represent all roles in an event, MLG facilitates the generation of challenging role types.

Data Quality Controls: Given that we are generating long documents using a variety of LLMs of varying quality, generation errors may occur. Thus, upon detection of imperfections such as a large number of missing arguments or hallucinated roles, we perform a post-processing step to ensure data accuracy, which is discussed in detail in Appendix B.

For both the Mad Lib Generation and Solving tasks, we use k=3 samples from the source domain with moderate length and high role density to minimize API costs for use with proprietary LLMs. Generated data statistics are shown in Appendix A Table 3. Additionally, we provide our prompts and sample data in Appendix B and I, respectively.

3.2 Struct2Text (S2T)

We ground both our Structure Generator and Struct2Text Generator to an existing data generation framework from LangChain (LangChain, 2024). We describe the functionality of the framework and our modifications below.

Structure Generator: As shown in Figure 2, the goal of this module is to generate an event record (i.e. a dictionary of role $\rightarrow$ argument mappings). To perform generation, we first construct an empty event record using Pydantic (Colvin et al., 2024), which allows us to encode event information as Python classes. Pydantic is utilized in (LangChain, 2024) due to its strong integration with OpenAI models. During structure generation, an LLM is prompted to populate an empty event record for a given event, generating arguments for each role. Note that to encourage generalizability to EAE datasets with varying degrees of event ontology information, target domain event semantics are derived solely using event and role names in this setting.

Struct2Text Generator: The structure generator output is then fed to an LLM which is prompted to transform the event structure into a document containing the corresponding event information. We modify the prompt structure from (LangChain, 2024) to better control the generation by providing style guides and generation length requirements to match the target distribution.

Struct2Text Aligner: A key limitation of (LangChain, 2024) is its inability to reliably generate documents containing exact substrings required for extractive tasks such as EAE. In other words, while information from the structure may be present in the generated document, the LLM may have rephrased the event arguments in a way that makes identifying argument location difficult. Thus, to adapt this method for DocEAE data augmentation, we implement a Struct2Text Aligner to map event arguments output by the structure generator to spans in the S2T-generated document. Specifically, we perform semantic n-gram matching between the structure and document. Our matching algorithm compares the argument in the event structure with every possible n-gram (up to n=20) in the document and returns the n-gram with the highest cosine similarity using SBERT (Reimers and Gurevych, 2019a). Any arguments that have cosine similarity lower than an inclusion threshold α are considered to have not been generated and are discarded from the event record. Similar to MLG, S2T aims to generate role-dense training examples via generation of DocEAE data containing annotation for all roles in an event schema. This paradigm allows us to generate zero-shot roles without reliance on existing samples. We provide S2T sample data in Appendix I, generated data

statistics in Table 3, as well as additional S2T methodological details in Appendix C.

4 Experiments

This section presents the (i) datasets, (ii) baseline augmentation strategies, (iii) EAE models, and (iv) evaluation metrics used in our study.

4.1 Evaluation Datasets

Primary Evaluation: Our primary evaluation in this study examines MLG and S2T capacity for augmentation of challenging role types. We use the cross-domain split of the DocEE dataset for this evaluation. See Appendix A Table 6 for DocEE dataset statistics. The task is structured to allow pre-training on a large number of source domain events, followed by fine-tuning on a target domain for which there are only very few annotated documents. In this dataset, the target domain is comprised of 10 event types, which are all under the common theme of "Natural Disasters". This grouping of source and target event domains was chosen to reduce overlap between source and target role types (Tong et al., 2022). During few-shot cross-domain training, only 5 documents for each type of event are provided.

Secondary Evaluation: We additionally explore the capacity of our methods to improve performance in a classic augmentation setting on datasets with fewer event types and/or less role diversity when compared to DocEE. Specifically, we evaluate on RAMS (Ebner et al., 2020), DiscourseEE (Sharif et al., 2024), and PHEE (Sun et al., 2022). [2] These datasets span a variety of domains (i.e. news articles, social media discourse, and medical text) and text lengths (1-22 sentences). We maintain the cross-domain nature of the task by *only using data from DocEE* when sourcing in-context examples for data generation. This experiment demonstrates the capability of our proposed methods to generate cross-domain samples, and to generalize across multiple datasets and domains.

Note that we take various pre-processing steps to map each of RAMS, DiscourseEE, and PHEE to DocEE's data format to standardize task structure. Additionally, we provide a style-guide to the LLM to help match the target domain. For instance, in DiscourseEE, a DocEAE dataset sourced from Reddit, we augment the prompt with the style guide '*Reddit post from user looking for help with opioid use disorder*'. We sub-sample the training set for each task and evaluate the impact of augmentation when using only 10%, 50% and 100% of the original training set alongside 500 samples generated using MLG-GPT-4o. Please see Appendix F for more details on experimental setup and additional data preparation steps for each of these tasks.

[2]Note we only evaluate on DiscourseEE samples in the 'explicit' data subset, which contains span annotation.

4.2 Baseline Augmentation Methods

While this work is the first to explore document-level EAE data generation, we compare our approach to a relevant EAE augmentation method developed for a sentence-level task and can be adapted for DocEE as it uses no ontological information: **Mask-then-Fill (MTF)** (Gao et al., 2022). MTF augments DocEE documents by masking a single contiguous span of non-annotated text in each document and infilling the mask using a T5 (Raffel et al., 2020) model fine-tuned on the Gigaword corpus (Graff et al., 2003). Since MTF was developed for sentence-level tasks, we implement a document-level variant: Doc-MTF. **Doc-MTF** is similar to MTF, except the T5 model is trained to fill masks of larger contiguous spans of non-annotated text in documents from the MultiNews dataset (Fabbri et al., 2019), which is more similar in style and length to DocEE texts than the original Gigaword corpus. Additionally, Doc-MTF is trained to fill multiple masks in a single document, which enables more diverse augmentations than the original single-mask variant.

4.3 EAE Models

In this study, we evaluate the impact of EAE data augmentation using the following EAE models motivated by the DocEE Paper (Tong et al., 2022).

LongFormer(LF)-Seq: This model treats EAE as a token labeling task. Specifically, for a document D with tokens $\{t_0, \ldots, t_n\} \in D$, we label argument spans using the label space $\{r_0, \ldots, r_n\} \in R$, where a given r_i denotes an event role. Spans of contiguous tokens with the same label correspond to event arguments. Given that we are processing long documents, we use the LongFormer-base (149M parameters) (Beltagy et al., 2020) architecture, which can process up to 4096 tokens.

BERT-QA: This model treats EAE as an extractive question-answering task, inspired by (Du and Cardie, 2020). In this formulation, the question is the argument role, and the context is the document. The goal is to predict the span(s) $(start_i, end_i)$

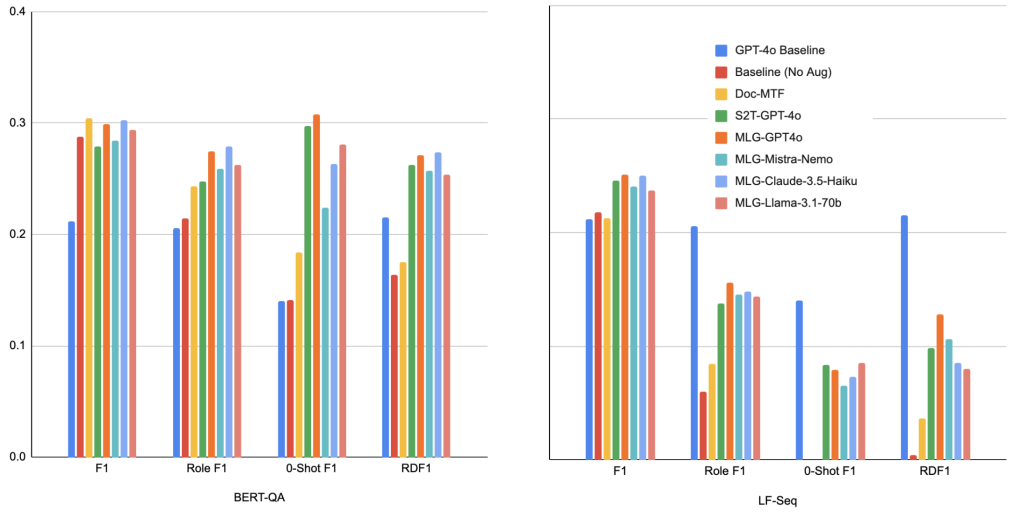


Figure 3: Results of training both BERT-QA and LF-Seq with MLG and S2T-augmented training sets at 4x augmentation. We compare to models trained with no augmentation (Baseline (No Aug)) and Doc-MTF augmentation. Additionally, we demonstrate the performance of GPT-4o on the EAE task for reference. We find that MLG and S2T-based augmentation provides a significant improvement over baseline models on 0-Shot F1 and RDF1 metrics. Full results tables with statistical significance tests are available in Appendix H.

corresponding to the argument of the role in the document. Note that for this baseline, we use a BERT-base (110M parameters) (Devlin et al., 2019) model as in the original DocEE paper.

We additionally benchmark the capacity of GPT-4o on DocEE. Please refer to Appendix D for details regarding the GPT-4o baseline, as well as all EAE model hyperparameters, compute resources, and training processes.

4.4 LLMs for Data Generation

Since MLG uses an in-context learning-based prompting strategy, we can explore a diverse set of LLMs such as (i) Llama 3.1-70b (Dubey et al., 2024), (ii) Mistral-Nemo (Mistral-AI, 2024) (iii) Claude-3.5-Haiku (Anthropic, 2024) (iv) GPT-3.5 (OpenAI, 2024) and (v) GPT-4o (Achiam et al., 2023). S2T, on the contrary, relies on a LangChain implementation which is optimized for OpenAI models, making it challenging to integrate with smaller open-source LLMs.[3] We thus only investigate S2T in the context of GPT-3.5 and GPT-4o.

4.5 Evaluation Metrics

- **F1**: We compute an F1 score as in prior work (Tong et al., 2022; Sharif et al., 2024) using the full set of predicted and ground truth triplets. [4]

- **Role-F1**: F1 is limited in its ability to demonstrate performance on challenging role types, as common role types contribute disproportionately to overall performance. Inspired by (Li et al., 2022), we can remove this bias by computing the Role-F1, which computes the F1 score for *each role independently*, with Role-F1 being the macro average of role-specific F1 scores in a given dataset. Each role thus contributes equally to the F1 score in Role-F1.

- **0-Shot F1**: This metric computes Role-F1 on all DocEE 0-shot roles. There are the 9 roles in the target domain test set with no training data.

- **Role Depth F1 (RDF1)**: We introduce a new metric, RDF1, for evaluating cross-domain DocEAE performance for semantically outlying roles. We use statistical depth (Seegmiller and Preum, 2023) to identify the top 25% of roles in the target domain that are most semantically outlying for roles in the source domain. RDF1 aims to emphasize performance on roles in the target domain that are highly specific to natural disasters, placing less emphasis on roles such as *date* or *location* which are not domain-specific. Statistical depth identifies 16 roles in DocEE as outliers which are used in this evaluation. The set

[3] https://www.langchain.com/

[4] Specifically, we use a tuple-based F1 implementation from Peng et al. (2023) to compute overall F1.

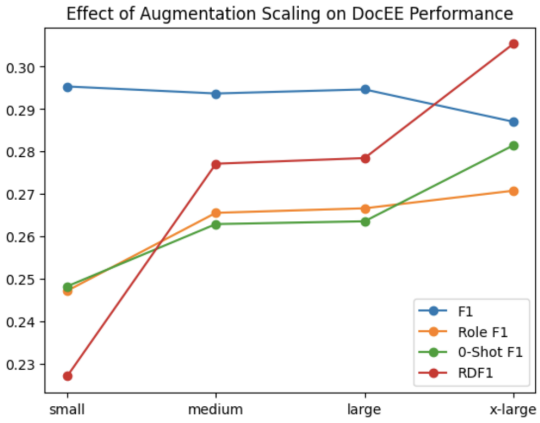


Figure 4: This figure captures how performance changes on BERT-QA with varying amounts of data augmentation. We explore 2x (small) 4x (medium) 5x (large) and 6x (x-large) augmentation scales. We find that there is a trade-off between optimizing for overall-F1 and F1 for challenging roles on DocEE.

	F1	Role F1
DiscourseEE (10%)	0.13	0.059
DiscourseEE (10%) + Aug	0.334†	0.342†
DiscourseEE (50%)	0.163	0.092
DiscourseEE (50%) + Aug	0.361†	0.357†
DiscourseEE (Full)	0.18	0.106
DiscourseEE (Full) + Aug	0.403†	0.396†
RAMS (10%)	0.134	0.128
RAMS (10%) + Aug	0.214†	0.186†
RAMS (50%)	0.323	0.298
RAMS (50%) + Aug	0.343†	0.33†
RAMS (Full)	0.388	0.38
RAMS (Full) + Aug	0.393	0.375
PHEE (10%)	0.42	0.303
PHEE (10%) + Aug	0.544†	0.53†
PHEE (50%)	0.596	0.581
PHEE (50%) + Aug	0.599	0.594
PHEE (Full)	0.621	0.618
PHEE (Full) + Aug	0.618	0.608

Table 1: Performance of BERT-QA on three EAE datasets using MLG-GPT-4o augmentation. We experiment utilizing 10%, 50%, and all of the real training sets for each dataset, alongside 500 synthetic samples. All synthetic data is generated using *zero in-domain exemplars*. All results are the mean performance over three experimental trials. Statistically significant performance increases ($p < 0.05$) are marked †.

of outlying roles in the target domain identified by the RDF1 metric align with human judgment 78% of the time. Please refer to Appendix E for additional details on RDF1, including our human-verification of this metric.

For each metric, we consider the exact match between a *set* of predicted and ground truth tuples consisting of (document id, role type, normalized_argument). Each ground truth tuple in our evaluation set contains an argument for a given role. We report the mean of three experimental trials with different random seeds for both the BERT-QA and LF-Seq models. For results including statistical significance testing using the Wilcoxon signed rank test, please refer to Tables 7 and 8 in the Appendix.

5 Results

Significant Improvement in F1 Score on Challenging Roles in DocEE: In Figure 3, our results show that both MLG and S2T have the capacity to significantly improve the breadth of roles which can be accurately predicted by a DocEAE model. On BERT-QA, both MLG and S2T provide significant increases in performance on 0-Shot and RDF1 (i.e. semantically outlying) role subsets. Additionally, MLG provides a statistically significant increase on Role-F1 when compared to all baseline models. We find that LF-Seq is less performative than BERT-QA across the board, causing it to under-perform the GPT-4o baseline on certain metrics. However, similar to BERT-QA, MLG and S2T-based LF-Seq models significantly outperform no augmentation as well as Doc-MTF augmentation on all metrics, highlighting the effectiveness of our approach.

We validate the efficacy of MLG on a diverse set of LLMs. [5] We find that closed-source models such as GPT-4o and Claude-3.5-Haiku outperform smaller models such as Mistral-Nemo and Llama-3.1-70b on most metrics. However, models such as Llama-3.1-70b still show strong performance and are a viable open-source option for DocEAE data augmentation.

Limitations of Classic F1 Score on Challenging Role Type Performance: Our results on DocEE in Figure 3 show that across all experiments for the highest performing model BERT-QA, there is no marked improvement being made by MLG/S2T-augmented models on overall F1 score when compared to baseline methods. This is because roles that have significant representation in both the training and testing set often do not benefit from augmentation. This is intuitive, as the model can overfit to high training frequency roles and improve

[5] Recall that S2T is grounded to an OpenAI-specific implementation of data generation.

overall F1 since *the test set is dominated by a small set of common roles*. For example, the role *Date*, which appears 28 times in the training set (3rd most frequently occurring training role), and 1841 times in the test set, is less likely to benefit from augmentation. A model can thus overfit to Date and gain a significant performance increase in overall F1 score. This is in contrast to the role *The State of the Volcano (Dormant or Active)* that occurs only 52 times in the test set, and never occurs in the training set. Here our methods are well-suited to significantly improve model performance as the model has no prior exposure to such a role type. Thus, MLG and S2T are useful to end-users who wish to increase the diversity of roles that can be extracted by DocEAE models. This phenomena is further evidenced via results in Figure 4, where we see additional augmentation scale has a positive effect on challenging role types, but a mild negative effect on overall F1 score. Future works should thus consider both overall F1 as well as role-specific F1 scores when evaluating EAE models, as breadth of extracted roles is an important characteristic of model capacity.

MLG Improves Low-Resource Performance on DiscourseEE, RAMS, and PHEE: In Table 1 we explore how our top-performing augmentation model, MLG GPT-4o, improves BERT-QA performance on DiscourseEE, RAMS, and PHEE. We find that across all experiments, MLG provides a significant performance increase in extremely low-resource settings (i.e. when only 10% of training data is available). Notably, we see that MLG consistently improves overall performance on DiscourseEE by $\approx$ 20 F1-points. This is an important result as DiscourseEE is naturally very low-resource, highly domain-specific, and contains highly complex argument types. Interestingly, as the amount of training data from the true distribution increases for RAMS and PHEE, we find a diminishing impact from MLG augmentation. This occurs for a number of reasons (i) RAMS and PHEE have thousands of annotated training documents, making the impact of 500 MLG-generated samples less important as the full training distribution is shown to the model (ii) datasets like RAMS do not utilize semantically meaningful role names, which are the crux of the MLG model which relies only on event schema information. This makes it difficult to model roles like "Instrument" which LLMs may assume refer to musical instruments when in fact RAMS uses these terms more abstractly, such as the instrument of destruction in a military attack (iii) LLMs can struggle with argument diversity for certain roles where the LLM's sampling distribution is narrowly focused on certain tokens. For example, 47% of the MLG-generated arguments for the role *pre-existing conditions* in PHEE are 'hypertension'. Future works may explore how to increase MLG and S2T argument diversity to further increase performance.

Should You Use MLG or S2T?: Throughout our experiments, we found that MLG and S2T often perform similarly on a variety of metrics. Thus, what are the use-cases for each approach?

The Case For MLG: MLG is easily compatible with open-source LLMs, making this more accessible and affordable to the wider research community. Additionally, S2T is reliant on string matching and semantic alignment to map synthetic arguments to document spans. This process may produce mild span annotation errors which can reduce S2T sample quality. MLG does not have this issue as it simply relies on in-filling of masked placeholders, thus providing higher-quality span annotations. S2T is also limited in its reliance on OpenAI models for data generation.

The Case For S2T: S2T's advantage over MLG is it's use of PyDantic objects for structure generation. PyDantic provides additional flexibility to encode additional event structure information in the form of class / variable descriptions. S2T thus provides easier integration of external event information when compared to MLG.

6 Conclusion

In this study, we introduce two novel document-level EAE data generation strategies, MLG and S2T. We demonstrate the effectiveness of our methods across a diverse set of both open-source and proprietary LLMs. Our methods produce high-quality DocEAE annotation and significantly improve performance on various metrics across four EAE datasets. On DocEE, we specifically show show strong performance increases on challenging tasks such as extracting 0-shot roles and our new metric for exploring semantically outlying roles. We show that MLG and S2T are impactful augmentation strategies for low-resource event modeling and require minimal human efforts for data generation.

7 Limitations

This work is limited in that we do not explore the capacity of our model to generate large-scale data augmentation with fine-grained control over the role distribution. This was deemed out of scope for this paper due to the cost of data generation and the focus on a few-shot task. Additionally, this is an extremely difficult sub-task that requires a more targeted solution. Future works will explore scaling our framework in this way, on both in-domain and cross-domain formatted tasks. Future works will also explore varying k in our in-context learning based experiments as well as varying the length of the generated samples.

A limitation of the S2T model is its reliance on LangChain and OpenAI models. Future works may wish to explore how to adapt this method to a open-source LLMs. Another general limitation of this study is that we do not thoroughly explore the capacity of these methods to generate multi-argument roles. For example, if there are two different causes to an event, having the MLG model generate templates for [Cause #1] and [Cause #2]. Such extensions will be the focus of future work. Additionally, a limiting assumption of this work is that each document discusses a single event. We explore EAE data augmentation through this lens as it is the nature of DocEE's data annotation strategy, but future works will explore how to adapt our methods to multi-event documents such as those provided in the WikiEvents dataset (Li et al., 2021).

Acknowledgments

This work was partially supported by the National Science Foundation Research Traineeship, Transformative Research and Graduate Education in Sensor Science, Technology and Innovation (DGE-2125733).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Anthropic. 2024. Introducing the next generation of claude. Accessed: 2024-12-07.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.

Samuel Colvin, Eric Jolibois, Hasan Ramezani, Adrian Garcia Badaracco, Terrence Dorsey, David Montague, Serge Matveenko, Marcelo Trylesinski, Sydney Runkle, David Hewitt, Alex Hall, and Victorien Plot. 2024. Pydantic. If you use this software, please cite it as above.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie M Strassel, and Ralph M Weischedel. 2004. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec*, volume 2, pages 837–840. Lisbon.

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.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2020. Multi-sentence argument linking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8057–8077, Online. Association for Computational Linguistics.

Alexander R. Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir R. Radev. 2019. Multi-news: a large-scale multi-document summarization dataset and abstractive hierarchical model.

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.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English gigaword. *Linguistic Data Consortium, Philadelphia*, 4(1):34.

LangChain. 2024. Data generation use cases in langchain. `https://python.langchain.com/docs/use_cases/data_generation/`. Accessed: 2024-12-09.

Rui Li, Wenlin Zhao, Cheng Yang, and Sen Su. 2022. A dual-expert framework for event argument extraction. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1110–1121.

Sha Li, Heng Ji, and Jiawei Han. 2021. Document-level event argument extraction by conditional generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 894–908, Online. Association for Computational Linguistics.

Jian Liu, Yufeng Chen, and Jinan Xu. 2021. Machine reading comprehension as data augmentation: A case study on implicit event argument extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2716–2725, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jian Liu, Chen Liang, and Jinan Xu. 2022. Document-level event argument extraction with self-augmentation and a cross-domain joint training mechanism. *Knowledge-Based Systems*, 257:109904.

Mingyu Derek Ma, Alexander Taylor, Wei Wang, and Nanyun Peng. 2023a. DICE: Data-efficient clinical event extraction with generative models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15898–15917, Toronto, Canada. Association for Computational Linguistics.

Mingyu Derek Ma, Xiaoxuan Wang, Po-Nien Kung, P. Jeffrey Brantingham, Nanyun Peng, and Wei Wang. 2023b. Star: Improving low-resource information extraction by structure-to-text data generation with large language models.

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

Mistral-AI. 2024. Mistral nemo. Accessed: 2024-12-07.

Hiroki Nakayama. 2018. seqeval: A python framework for sequence labeling evaluation. Software available from https://github.com/chakki-works/seqeval.

OpenAI. 2024. Gpt-3.5 turbo model. Accessed: 2024-12-08.

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.

Hao Peng, Xiaozhi Wang, Feng Yao, Kaisheng Zeng, Lei Hou, Juanzi Li, Zhiyuan Liu, and Weixing Shen. 2023. The devil is in the details: On the pitfalls of event extraction evaluation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9206–9227, Toronto, Canada. Association for Computational Linguistics.

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

Nils Reimers and Iryna Gurevych. 2019a. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019b. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Taneeya Satyapanich, Francis Ferraro, and Tim Finin. 2020. Casie: Extracting cybersecurity event information from text. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8749–8757.

Parker Seegmiller and Sarah Preum. 2023. Statistical depth for ranking and characterizing transformer-based text embeddings. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9600–9611, Singapore. Association for Computational Linguistics.

Omar Sharif, Joseph Gatto, Madhusudan Basak, and Sarah Masud Preum. 2024. Explicit, implicit, and scattered: Revisiting event extraction to capture complex arguments. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12061–12081, Miami, Florida, USA. Association for Computational Linguistics.

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

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.

MeiHan Tong, Bin Xu, Shuai Wang, Meihuan Han, Yixin Cao, Jiangqi Zhu, Siyu Chen, Lei Hou, and Juanzi Li. 2022. DocEE: A large-scale and fine-grained benchmark for document-level event extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3970–3982, Seattle, United States. Association for Computational Linguistics.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Bo Wang, Heyan Huang, Xiaochi Wei, Ge Shi, Xiao Liu, Chong Feng, Tong Zhou, Shuaiqiang Wang, and Dawei Yin. 2023. Boosting event extraction with denoised structure-to-text augmentation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11267–11281, Toronto, Canada. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Xianjun Yang, Yujie Lu, and Linda Petzold. 2023a. Few-shot document-level event argument extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8029–8046, Toronto, Canada. Association for Computational Linguistics.

Yuqing Yang, Qipeng Guo, Xiangkun Hu, Yue Zhang, Xipeng Qiu, and Zheng Zhang. 2023b. An AMR-based link prediction approach for document-level event argument extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12876–12889, Toronto, Canada. Association for Computational Linguistics.

A DocEE Dataset Statistics

In this section we describe the statistics of the DocEE dataset as well as the statistics of the generated data used to augment DocEE in Figure 3.

A.1 DocEE

Split	# Samples	# Events	# Argument Instances
Source Train	23,630	49	154,225
Target Train	50	10	308
Target Dev	1,850	10	11,094
Target Test	1,955	10	13,227

Table 2: DocEE data distribution. The source domain contains annotation for over 23k documents across 49 different event types. The target domain contains annotation for documents with 10 distinct event types, all under the common theme of "Natural Disasters".

A.2 Generated Data Statistics

	Num Sent	Num Words	Arg Mentions Per-Doc
MLG-GPT-3.5	9.84	224.91	11.96
MLG-GPT-4o	10.52	253.21	12.98
MLG-Mistral-Nemo	10.25	249.76	12.84
MLG-Llama-3.1-70b	10.34	237.7	13.96
MLG-Claude-3.5-Haiku	10.68	243.5	12.72
S2T-GPT-4o	13.4	344.45	15.14
S2T-GPT-3.5	11.52	275.87	19.23

Table 3: Generated data statistics from each model. Each column represents the average statistic for each category across all generated documents used in our results table.

B Additional Details — Mad Lib Generation

In this section, we provide additional details of the Mad Lib Generation augmentation module.

B.1 Hyperparameters

1. **In-Context Learning:** We use k=3 in-context examples when performing MLG-based generation.

2. **Temperature:** For OpenAI and Claude models, we use temperature = 1.0 to maxmize output diversity. For smaller models, we found lower temperatures were helpful in producing the correct output structure. Thus, for Mistral-Nemo and Llama-3.1-70b we use temperature = 0.35 and 0.3 respectively.

B.2 Mad Lib Generation Data Quality Controls

To ensure high-quality data, we post-process our generated Mad Libs. An error that can occur during this process is *hallucinated role types*. For example, if the LLM invents a new role not found in the event schema when generating the Mad Lib, we may handle this in two ways: (i) we can use SBERT (Reimers and Gurevych, 2019b), a popular textual embedding method, to encode the hallucinated category name and compute the cosine similarity between it and all possible role names for a given sample. We can often reliably match the generated category to the true category using this method for mild hallucinations (e.g. mapping "The place" back to "Location"). (ii) For smaller LLMs we find hallucinations can be more extreme, thus we simply remove the sentence containing the hallucinated role from the document. Since DocEE is largely comprised of news articles, simply removing the sentence is a non-destructive way to deal with hallucination as the documents are often comprised of a sequence of factual statements. In our experiments, we focus on method (ii) as it is less ambiguous and works with all models.

Additionally, the generation of a longer text will be particularly sensitive to the randomly drawn few-shot sample selection. Thus, there can be outlying generation attempts where Mad Libs are generated (i) with little to no role information (ii) with incorrectly used brackets (iii) with improperly formatted solutions. When this happens, we simply draw a new set of few-shot examples and try again. Note: this issue becomes increasingly rare as model quality increases.

B.3 Prompts

MLG Generator:

> "A MadLib is a templated text document with masked placeholders that users fill with specified word categories to complete the content.
>
> Write a madlib for a "{event_type}" event. Use the following categories for Mad Lib blanks. Make sure all categories appear in the MadLib. Do not generate any other categories. Write the MadLib in the style of a {style_guide}:
>
> Categories: {categories}
>
> Madlib:"

MLG Solver

> "A MadLib is a templated text document with masked placeholders that users fill with specified word categories to complete the content.
>
> Solve the MadLib. Fill-in missing information in the document. Return your answer as a bulleted list of the format "### CATEGORY: ANSWER".
>
> Make sure the following categories are in your solution: {categories}
>
> Madlib: {madlib}
>
> MadLib Solution:"

C Additional Details — Struct2Text

C.1 Hyperparameters

1. **In-Context Learning:** We use k=3 in-context examples when performing S2T-based generation. Note that ICL is only used for structure generation in S2T.

2. **Temperature:** We use temperature = 1.0 for all S2T experiments.

3. **S2T Aligner α:** The threshold used to determine a valid match between synthetic argument and document span is 0.6. We choose this threshold empirically after manual inspection of automated argument mappings.

C.2 Struct2Text Data Quality Controls

Much of the data quality control comes from the S2T Alignment module. However, other small data quality controls are implemented as part of S2T.

JSON Error Handling: S2T's reliance on outputting valid JSON can introduce random generation errors. Similar to MLG, we thus allow S2T multiple attempts at generation in the event of such errors.

Generated Data Processing: (i) We transform the data into lower-case to facilitate string matching (ii) We search for and filter empty event arguments which can come in various forms such as null values or the string "None".

C.3 Prompts

Structure2Text Generator We adapt the default prompt from (LangChain, 2024) for document-level tasks.

"Given the following fields, create a document about them. Make the document detailed and interesting. Use every given field. If any additional preferences are given, use them during document construction as well.

Fields: {fields} Preferences: {preferences} Document:"

We control the style and length of the output using the *preferences* field.

D Baseline Models

We train all models on Google Colab using an NVIDIA A100 GPU. Throughout the course of the project, we estimate an upper limit of 150 compute hours.

Details of our baseline models are found in the following sections.

D.1 BERT-QA

All of the BERT-QA experiments are run using the Huggingface (Wolf et al., 2019) Transformers library. We use a BERT-base model (110m parameters) [6] as our base model. Given that BERT has a max token length of 512, we process longer documents in chunks using a stride of 50 tokens. All models are fine-tuned for 3 epochs using the default Huggingface Trainer parameters. [7] When applicable (i.e., when a development set is available), validation is loss is used for model selection.

To help encourage extraction of multi-argument roles, we only include unique arguments as training examples for QA-based extraction. During inference, we take up to 2 predictions per-question. We perform inference using the Huggingface question-answering pipeline. We consider roles present in test set during evaluation. We use a batch size of 48 for all BERT-QA experiments and train our models on Google Colab use an NVIDIA A100 GPU.

D.2 LongFormer-Seq

All LongFormer-Seq experiments use the LongFormer-base (149M parameters) [8] and are run using the Huggingface Transformers library. In our cross-domain experiment, we fine-tune the model for 3-epochs on the source training set with default parameters. For cross-domain fine-tuning, we train the source model with a new classification head for the target events for up to 10 epochs using validation F1 score from SeqEval (Nakayama, 2018) to determine the best model. We use a batch size of 4 for all LongFormer experiments and train our models on Google Collab use an NVIDIA A100 GPU.

D.3 Evaluation Metrics

We use an implementation of F1 score for EAE from the OmniEvent library (Peng et al., 2023). When computing F1, since the same argument can be expressed in slightly different ways throughout a document, we normalize the argument string before evaluation using a function provided by the DocEE authors. This makes outputs lowercase, removes articles, white space, and punctuation. For example, If the prediction was (0, date, The 14th of May.), normalization would return (0, date, 14th of may). This is helpful since the same argument can be expressed in many slightly different ways throughout a long document.

D.4 LLM EAE Baseline Details

Our LLM-based EAE baselines use the following prompt to perform Event Argument Extraction.

```
##Instruction##

Concisely extract the arguments for following roles from the document. Make sure the extracted arguments are found directly in the text as substrings.

Use the provided role descriptions to extract arguments as a JSON. Return 'null' if any arugment is not present in the document. Concisely give the output as requested, no extra description is needed.

##Roles Descriptions##

{role_descriptions}

 ##Document##

{document}

Return "null" if any arugment is not present. Return arguments in JSON. Separate multiple arguments of a role values by semicolon (;).
```

[6] bert-base-uncased

[7] https://huggingface.co/docs/transformers/main_classes/trainer

[8] allenai/longformer-base-4096

If the LLM outputs a multiple arguments for a role separated by a semicolon (;), we split these into separate predicted tuples before evlauation.

E RDF1 Evaluation

E.1 Role-Depth-F1 (RDF1)

RDF1 is a performance metric which aims to evaluate how well cross-domain EAE models are predicting role types that are semantically different from those observed in the source domain. To compute RDF1, we propose using transformer-based text embeddings (TTE) depth (Seegmiller and Preum, 2023) to identify the degree to which target domain roles deviate semantically from source domain roles. Formally, given a set of source domain roles, $\{s_1, s_2, \ldots, s_n\} = S$, and a set of target domain roles, $\{t_1, t_2, \ldots, t_m\} = T$, TTE depth assigns a score to each target role $t_i \in T$ indicating how well t_i represents S.

We embed each role in each set using SBERT [9] (Reimers and Gurevych, 2019b) to get a set of source role embeddings $\{s'_1, \ldots, s'_n\} = S'$ and a set of target role embeddings $\{t'_1, \ldots, t'_m\} = T'$. TTE depth scores each target role embedding $t'_i \in T'$ according to

$$D_\delta(t'_i, S') := 2 - E_{S'}[\delta(t', H)] \tag{1}$$

where $H \sim S'$ is a random variable with uniform distribution over S', and δ is cosine distance.

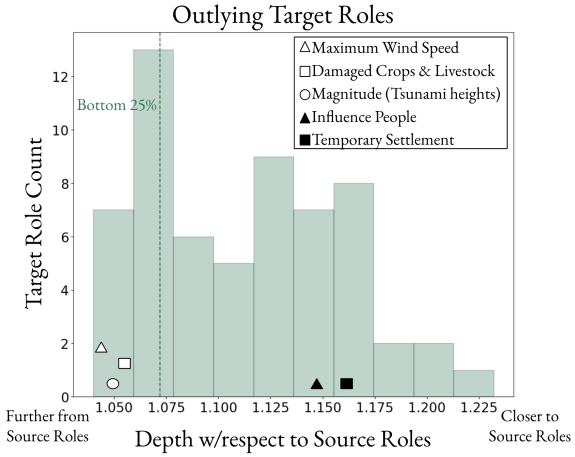


Figure 5: Visualization showing how TTE Depth ranks DocEE roles. We find that roles such as "Maximum Wind Speed", "Damaged Crops & Livestock" and "Magnitude (Tsunami Heights)" are outliers compared to roles such as "Influence People" and "Temporary Settlement".

[9] all-MiniLM-L6-v2

TTE depth thus assigns each target role $t_i \in T$ a score, with higher scores indicating that the target role is representative of the source roles (S), and lower scores indicating that the target role is a semantic outlier with respect to the source roles. We formalize RDF1 as follows. First, consider a set of predicted tuples P and ground truth tuples G, where each tuple $g_i \in G$ is formatted as $g_i = (\text{document-id}, \text{role}, \text{argument})$. We define Role F1, i.e. the F1 score for a single role, as

$$\text{Role F1}(t_i) = F_1(\hat{P}_{t_i}, \hat{G}_{t_i}) \tag{2}$$

Where $\hat{P}_{t_i}$ and $\hat{G}_{t_i}$ are the subset of all tuples in P and G respectively, which have arguments for role t_i. We then define RDF1 as the mean Role_{F_1} for only semantically outlying roles

$$RDF1 = \frac{1}{k}\sum_{i=1}^{m} \begin{cases} \text{Role } F_1(t_i) & \text{if } D_\delta(t'_i, S') < \tau \\ 0 & \text{if } D_\delta(t'_i, S') > \tau \end{cases} \tag{3}$$

Where m is the total number of roles in the test set, and k is the number of roles $t_i \in T$ which have $D_\delta(t'_i, S') < \tau$, and τ is the depth threshold chosen to extract the top 25% most semantically outlying roles. We choose 25% for this TTE depth threshold as it strikes a good balance between filtering for target roles that differ from source domain roles, while maintaining a good amount of roles for evaluation. By computing F1 only on these roles, i.e., RDF1, we measure the ability of an FSCD EAE model to perform domain transfer to roles that deviate semantically from the source domain. In Figure 5, we visualize how depth scores rank all roles in the target domain with respect to the source domain.

In DocEE, 16 roles comprise the set of semantic outliers evaluated in RDF1. We additionally note that the embedding strategy for creating S and T from R_S and R_T is flexible; event ontologies that include additional information, such as role descriptions or valid entity types may be used to enrich the role representations. Since DocEE provides no such information, opting instead for using semantically-rich role names, the role names themselves are used.

E.2 Full List of RDF1 Roles for DocEE

Maximum Wind Speed, Storm Movement Speed, Maximum Rainfall, Magnitude (Tsunami heights), Related Rivers

or Lakes, Influenced Crops and Livelihood, Water Level, Amount of Precipitation, Damaged Crops & Livestock, Number of Damaged Houses, Number of Rebuilding House, Fire Warning Level, Storm Formation Location, The State of the Volcano (Dormant or Active), Tsunami Warning Level, Number of Influenced People,

E.3 Human Evaluation of RDF1 Classes

To ensure that the cross-domain roles identified by RDF1 align with human judgment, we have two human annotators perform the following analysis.

1. Each annotator is familiar with the DocEE dataset, and reads through the full list of 60 roles in the cross-domain data split, whose theme is natural disasters.

2. Each annotator then is asked to select any role which does *not* appear to be specific to natural disasters. In other words, the annotators ask themselves "is it likely that this role pertains to something which is not natural disaster." An example of this could be the role "location".

3. The resulting annotation produces a human filtered role set, which RDF1 roles should *not* include in evaluation.

4. We then compare the human filtered role set to our automatically extracted RDF1 roles. The percentage of roles humans believe should have been filtered, which are found in the RDF1 roles, is determined to be the error rate of the model.

5. We report the 1 - (mean error rate) from both annotators as our assessment of human-RDF1 alignment.

Our results show that RDF1 roles align with human judgment 78.1% of the time. Common roles which were filtered by humans, but not RDF1, include Number of Influenced People and Water Level.

Annotator Details: The annotators who performed this analysis were two graduate students in Computer Science who live in the United States. The annotation was done as part of their normal research practice and took less than 30 minutes to complete.

F Additional Dataset Information

F.1 PHEE

F.1.1 Dataset Pre-Processing

We convert PHEE into DocEE format by taking the following steps:

- As in DocEE, we do not consider triggers and ignore them during pre-processing.
- PHEE has a hierarchical argument structure consisting of main and sub arguments. We concatenate the main and sub arguments into one large argument set to align evaluation across all tasks.
- Given that we have merged main and sub arguments, we manually correct an overlapping role name “disorder" which occurs both in the context of subjects and treatments. If we are in the context of a subject, we denote disorder as *pre_existing_condition* and if we are in the context of a treatment, we denote disorder as *disorder_treated*.
- The train, development, and test splits of PHEE remain unchanged and are sourced directly from https://github.com/ZhaoyueSun/phee-with-chatgpt/

F.1.2 Dataset Distribution

Split	# Samples	# Events	# Argument Instances
Train	2,898	2	14,536
Dev	961	2	4,894
Test	968	2	4,952

Table 4: PHEE Data Distribition

F.2 DiscourseEE

F.2.1 Dataset Pre-Processing

We convert DiscourseEE into DocEE format by taking the following steps:

- DiscourseEE is a trigger-free dataset with hierarchical annotation for event arguments. We use the provided train, development, and test splits from https://github.com/omar-sharif03/DiscourseEE.git.
- DiscourseEE includes both explicit arguments (those which are found as spans of a document) as well as implicit and scattered arguments (which can be inferred through context).

As DocEE is an extractive task, we consider only the explicit subset of DiscourseEE.

- As DiscourseEE only provides string annotations, we map the explicit arguments back to the text using substring/boundary matching approaches.

F.2.2 Data Distribution

Split	# Samples	# Events	# Argument Instances
Train	246	3	869
Dev	50	3	194
Test	100	3	392

Table 5: DiscourseEE Data Distribution. Note we only explore explicit arguments in DiscoursEE as Implicit and Scattered arguments are not extractive tasks.

F.3 RAMS

F.3.1 Dataset Pre-Processing

We convert RAMS into DocEE format by taking the following steps:

- We download the RAMS dataset from `https://nlp.jhu.edu/rams/RAMS_1.0c.tar.gz` and leverage a RAMS pre-processing script provided by OmniEvent (Peng et al., 2023). We use the provided train, development, and test splits.
- We ignore all triggers to map the problem onto DocEE format.
- As RAMS is trigger-centric and single-event, there can be many duplicate documents. We thus drop duplicates during pre-processing. This makes the dataset slightly smaller than the original RAMS dataset.

F.3.2 Data Distribution

Split	# Samples	# Events	# Argument Instances
Train	6322	139	14612
Dev	800	127	1894
Test	770	128	1771

Table 6: RAMS Data Distribution after pre-processing.

G Visualizing Document Similarity

In Figure 6 we visualize the similarity between real DocEE test documents and MLG-GPT-4o outputs using t-SNE (van der Maaten and Hinton, 2008).

H Additional Results

In this section, we provide full results tables for DocEE Evaluation in Figure 3. We note that the implementation of the LLM-baseline employed relies on LLMs outputting valid JSON, which made evaluation of smaller LLMs (e.g. LLama) challenging, i.e., they ended up yielding much lower f1-score as some outputs could not be properly evaluated due to JSON errors. We thus focused our baseline on GPT-4o. We note that GPT-3.5 was a suitable model, producing an F1, Role-F1, 0-Shot, and RDF1 of 0.177, 0.162, 0.122, and 0.171 respectively. However, as GPT-4o was much stronger, we highlight these results in the paper.

I Example Data

In this section, we provide various examples of outputs from MLG and S2T on our various evaluation tasks.

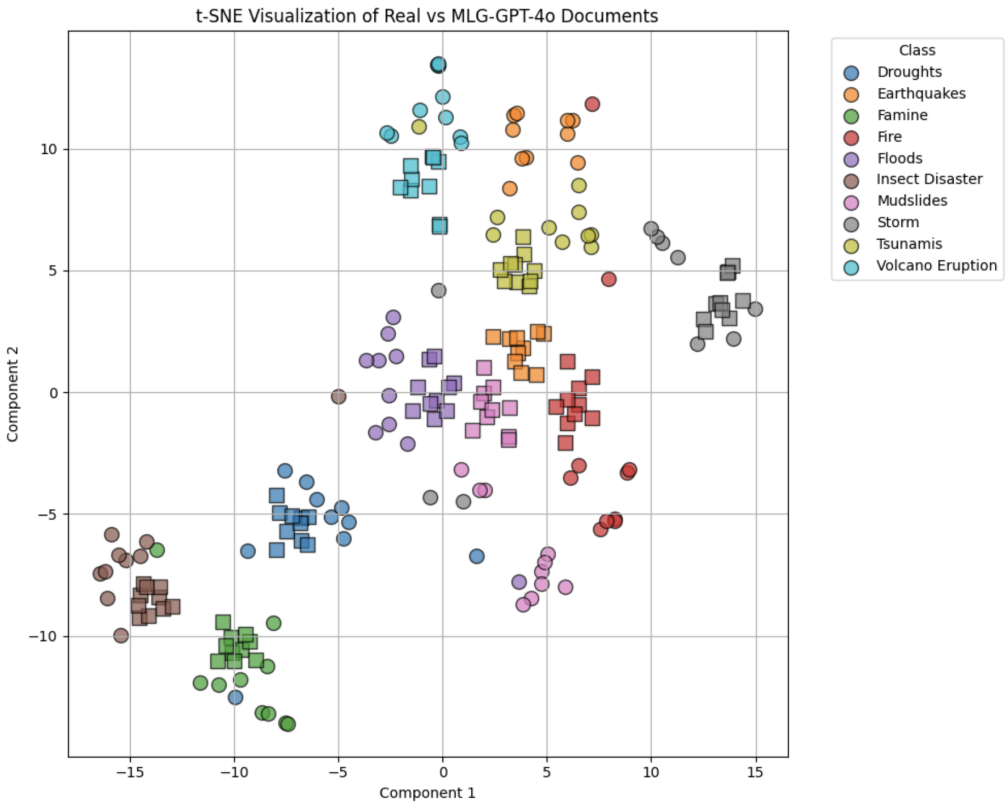


Figure 6: In this figure, we use SBERT to embed 100 documents generated by MLG-GPT-4o as well as 100 documents sampled from the DocEE target test set. **Circular markers** (○) denote real test samples, while **square markers** (□) denote MLG-generated data. We specifically sample 10 documents per-event type from each source. We find that for many classes, MLG data is highly similar to ground truth data.

	F1	Role F1	0-Shot F1	RDF1
Baselines				
Baseline (No Aug)	0.288	0.214	0.141	0.164
Doc-MTF	0.304	0.243	0.184	0.175
GPT-4o	0.212	0.206	0.14	0.215
Mad Lib Generation				
GPT-3.5	0.291	0.261	0.264*	0.259*
GPT-4o	0.299	0.275*	0.308*	0.271*
Mistral-Nemo	0.284	0.259	0.224	0.257*
Claude-3.5-Haiku	0.303	0.279*	0.263*	0.274*
Llama-3.1-70b	0.294	0.262	0.281*	0.254*
Struct2Text				
GPT-3.5	0.29	0.245	0.311*	0.251*
GPT-4o	0.279	0.248	0.297*	0.262*

Table 7: Results of BERT-QA on DocEE. Augmentation results which are statistically significant ($p < 0.05$) with respect to all 3 baselines are marked '*'.

	F1	Role F1	0-Shot F1	RDF1
Baselines				
Baseline (No Aug)	0.218	0.06	0.0	0.004
Doc-MTF	0.213	0.084	0.0	0.036
GPT-4o	0.212	0.206	0.14	0.215
Mad Lib Generation				
GPT-3.5	0.25*	$0.140^{\dagger}$	$0.063^{\dagger}$	$0.084^{\dagger}$
GPT-4o	0.251*	$0.156^{\dagger}$	$0.079^{\dagger}$	$0.128^{\dagger}$
Mistral-Nemo	0.241*	$0.145^{\dagger}$	$0.065^{\dagger}$	$0.106^{\dagger}$
Claude-3.5-Haiku	0.25*	$0.148^{\dagger}$	$0.073^{\dagger}$	$0.085^{\dagger}$
Llama-3.1-70b	0.237*	$0.144^{\dagger}$	$0.085^{\dagger}$	$0.080^{\dagger}$
Struct2Text				
GPT-3.5	0.232*	$0.124^{\dagger}$	$0.083^{\dagger}$	$0.080^{\dagger}$
GPT-4o	0.246*	$0.138^{\dagger}$	$0.083^{\dagger}$	$0.098^{\dagger}$

Table 8: Results of LF-Seq on DocEE. Augmentation results which are a statistically significant ($p < 0.05$) improvement with respect to only Doc-MTF and Baseline (No-Aug) are marked †. Results showing a statistically significant improvement compared to all 3 baselines are marked *.

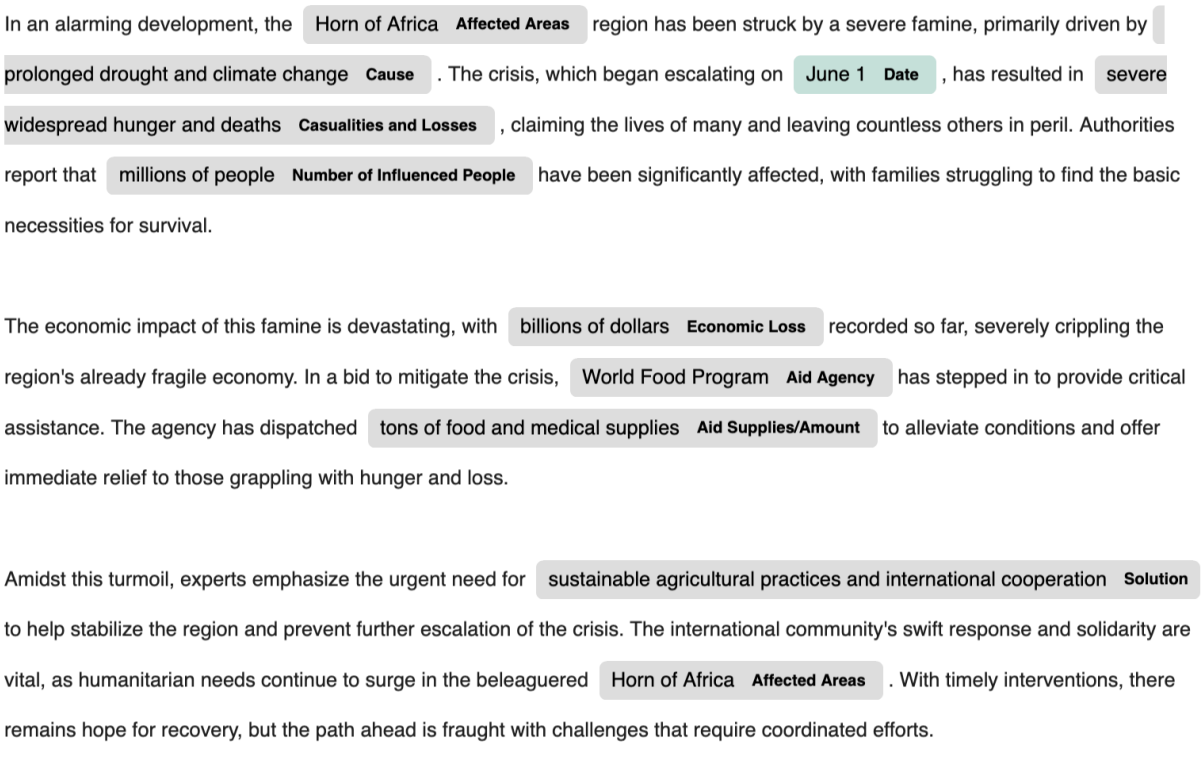


Figure 7: Example data from MLG-GPT-4o on DocEE.

hurricane zephyr **Storm Name** , an ominous category 4 **Storm Warning Level** storm, is on a determined northwest **Storm Direction** march toward the florida panhandle **Storm Hit Location** , promising a furious impact as it bears down on the united states. forming initially on october 4th, 2023 **Storm Formation Time** , in the churning waters of the eastern atlantic ocean **Storm Formation Location** , zephyr has quickly intensified, commanding the attention of meteorologists and residents alike. as reported by the national hurricane center **People/Organization who predicted the disaster** , the storm currently centers at coordinates 25.3°n, 75.1°w **Storm Center Location** , showcasing a hauntingly powerful maximum wind speed of 150 mph **Maximum Wind Speed** .

in anticipation of zephyr's arrival, evacuation orders have been issued for over 500,000 residents, **Influence People** emphasizing the storm's unprecedented strength and life-threatening potential. the florida panhandle **Storm Hit Location** is bracing for zephyr's impact, with initial landfall predicted to occur on october 9th, 2023 **Storm Hit Time** . residents in the storm's projected path are urged to take immediate precautions, as the hurricane is expected to bring a staggering 12 inches **Amount of Precipitation** of precipitation, leading to the potential for widespread flooding.

traveling at a steady pace of 15 mph **Storm Movement Speed** , zephyr is cutting a swath across the atlantic, signaling to forecasters and those on the ground the urgent need for vigilance and preparation. communities within the expected zone of impact are ramping up efforts, ensuring that all emergency protocols are in place. experts from the national hurricane center **People/Organization who predicted the disaster** have been closely tracking the storm's rapid development, stressing the importance of heeding evacuation notices and preparing for potentially devastating conditions.

the days leading up to hurricane zephyr **Storm Name** 's landfall are critical, with officials encouraging residents to finalize their emergency plans and seek safety far from the storm's projected path. as the florida panhandle **Storm Hit Location** waits anxiously, the air is a palpable mix of urgency and resolve, underscoring the community's collective experience with nature's might. the resilience of those facing hurricane zephyr **Storm Name** serves as a testament to human spirit in the face of mother nature's formidable power.

Figure 8: Example data from S2T-GPT-4o on DocEE.

Severe Drought Devastates Central California **Areas Affected** Region

On May 15th **Date** , meteorological experts confirmed a critical drought situation impacting San Joaquin Valley **The Worst-Hit Area** , with significant consequences for agricultural and economic stability. The drought, caused by prolonged climate change and reduced precipitation **Cause** , has dramatically reduced water levels in Sacramento River and San Joaquin River **Related Rivers or Lakes** , creating unprecedented challenges for local communities.

Approximately 750,000 **Influenced People** residents have been directly affected by the environmental crisis. Agricultural sectors have reported massive 35% agricultural output **Production Cuts** , with an estimated 500,000 acres of farmland and 25,000 cattle **Damaged Crops & Livestock** suffering complete loss.

Economic analysts project the total $1.2 billion **Economic Loss** could reach unprecedented levels, potentially destabilizing the regional economy for years to come. Local government officials have declared a state of emergency, requesting immediate federal assistance to mitigate the drought's catastrophic impact.

Water management authorities warn that without significant rainfall or intervention, the current conditions could persist, threatening food security and local livelihoods. Emergency water distribution programs have been initiated to support affected populations in San Joaquin Valley **The Worst-Hit Area** .

The drought represents one of the most severe environmental challenges the region has faced in recent decades, underscoring the urgent need for comprehensive climate adaptation strategies.

Figure 9: Example data from MLG-Claude-3.5-Haiku on DocEE.

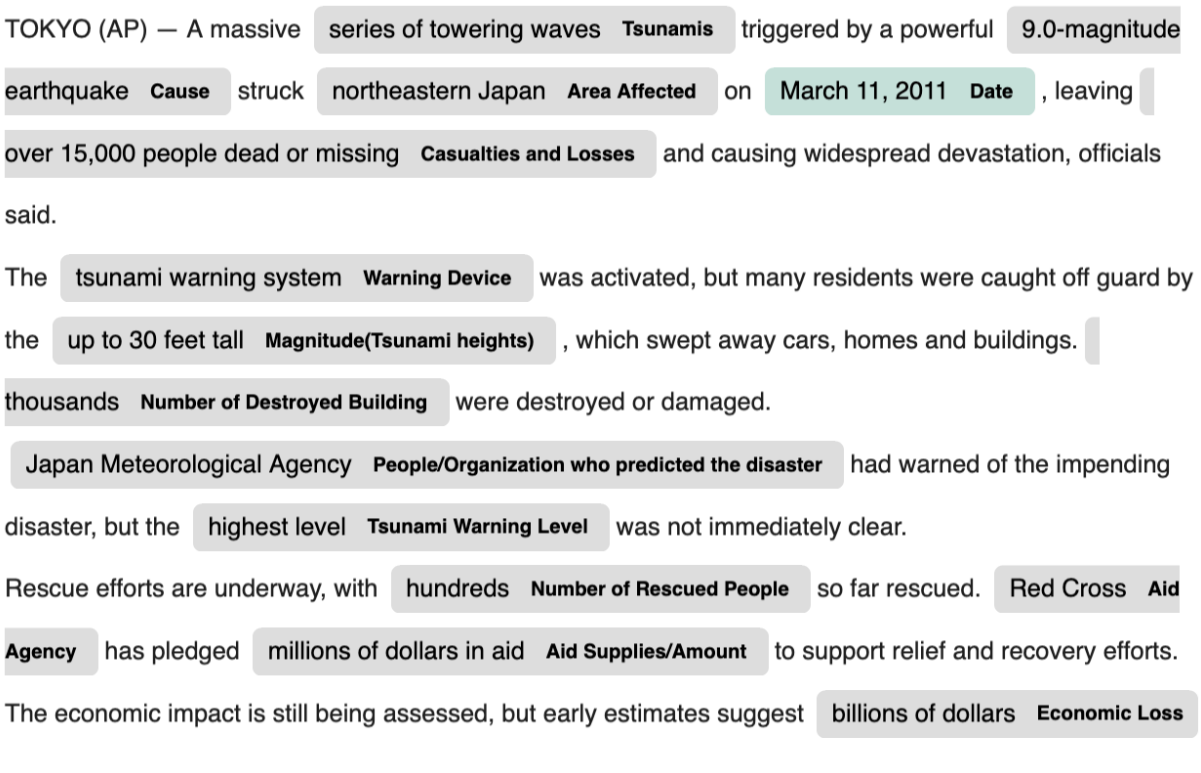


Figure 10: Example data from MLG-Mistral Nemo on DocEE.

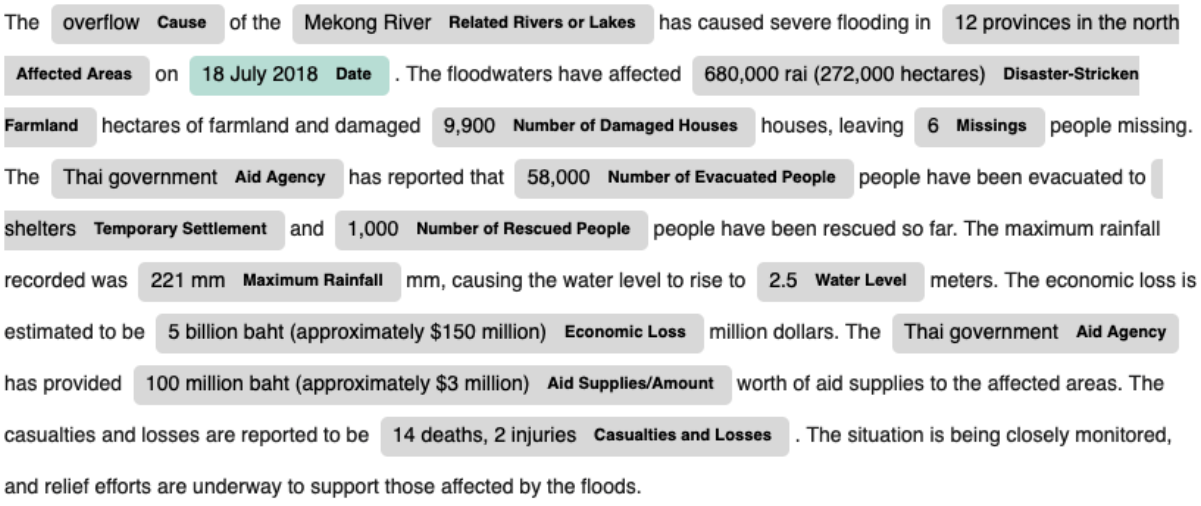


Figure 11: Example data from MLG-Llama-3.1-70b on DocEE.

Event Type: contact.threatencoerce.broadcast

In an alarming development this week, authorities in Cityville place reported an incident in which a threatening broadcast was made by an unknown group communicator . The communication was directed towards local residents recipient , sparking widespread concern among the local population. Sources indicate that the broadcast contained coercive language, raising tensions in the region. Officials are currently investigating the origins of the threat, seeking to identify and apprehend those responsible. This incident underscores the growing need for enhanced security measures to protect citizens against such unsettling forms of communication. Further updates on the situation are anticipated as the investigation progresses.

Figure 12: Example data from MLG-GPT-4o on RAMS.

Event Type: adverse_event

A 45 age -year-old female gender of Caucasian race descent with a history of chronic kidney disease pre_existing_condition presented with lactic acidosis effect following a three-month duration course of 500 mg twice daily dosage of metformin drug . The oral route administration was part of a diabetes management treatment regimen for the treatment of type 2 diabetes disorder_treated . This adverse event has been documented in rarely freq within the general population population.

Figure 13: Example data from MLG-GPT-4o on PHEE.

Event Type: tapering

Hey everyone, I've been on opioids taper_medications for chronic pain conditions for quite a while now, and I'm really considering tapering down. I started at 50mg initial_dosage and with everything that's been going on, recent personal events trigger has made me think it's time to change things up. I'm currently 34 years old age years old and dealing with chronic pain condition , and the side effects have been no joke. I'm talking moderate severity fatigue and dizziness side_effects that last for about several hours side_effect_duration .

I took my first step by starting to taper two weeks start_time ago, with the plan to reach my 10mg goal_dosage in three months target_duration . Right now, my 40mg current_dosage is still pretty high, but I'm determined to follow through. My biggest fear is that without the meds, my chronic pain condition might flare up again. Anyone have tips on how to deal with fatigue and dizziness side_effects while tapering?

Also, if you've found other lifestyle changes intervention helpful while trying to taper off, I'd love to hear your experiences. I'm really hoping to make this transition as smooth as possible and focus on a healthier me. Thanks for any advice you can share!

Figure 14: Example data from MLG-GPT-4o on DiscourseEE.

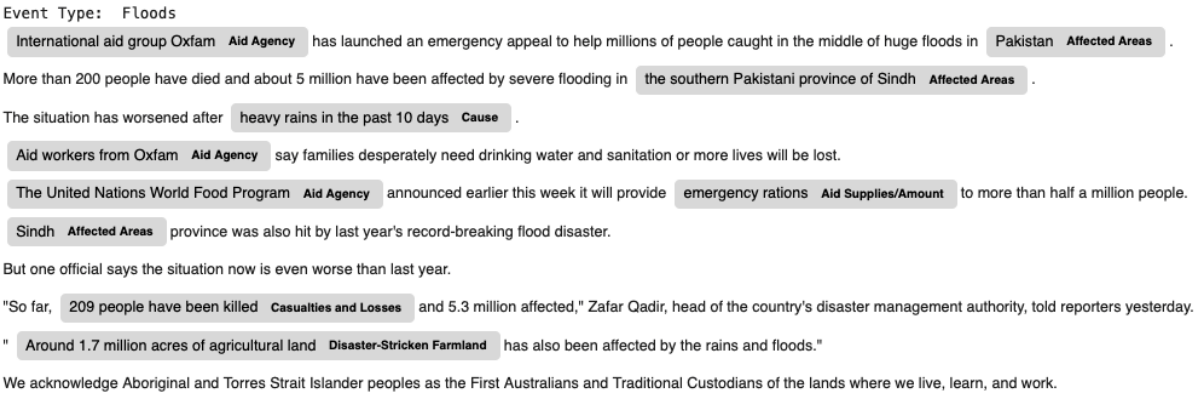


Figure 15: Example real sample from DocEE for the event “Floods". Can be compared to a synthetic sample for the same event in Figure 16.

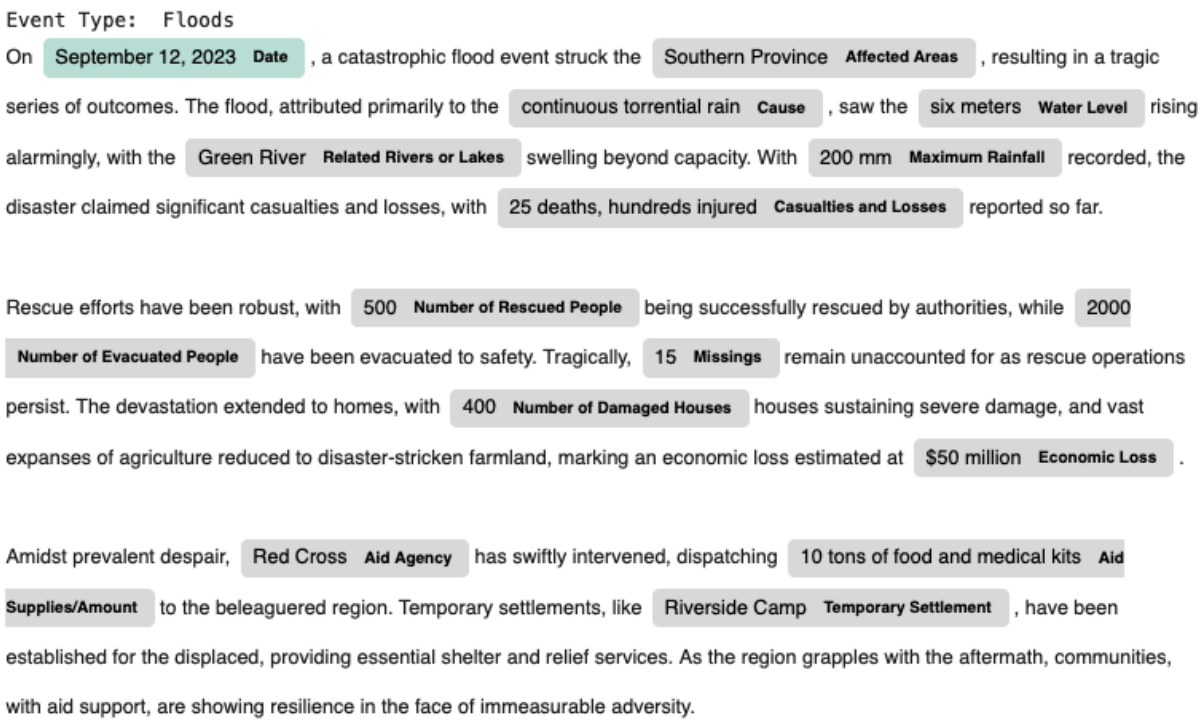


Figure 16: MLG-GPT-4o generation for the DocEE event “Floods". Can be compared with a real sample from this event in Figure 15.