Schema-Based Data Augmentation for Event Extraction

Xiaomeng Jin, Heng Ji
University of Illinois Urbana-Champaign
{xjin17, hengji}@illinois.edu

Abstract

Event extraction is a crucial task for semantic understanding and structured knowledge construction. However, the expense of collecting and labeling data for training event extraction models is usually high. To address this issue, we propose a novel schema-based data augmentation method that utilizes event schemas to guide the data generation process. The event schemas depict the typical patterns of complex events and can be used to create new synthetic data for event extraction. Specifically, we sub-sample from the schema graph to obtain a subgraph, instantiate the schema subgraph, and then convert the instantiated subgraph to natural language texts. We conduct extensive experiments on event trigger detection, event trigger extraction, and event argument extraction tasks using two datasets (including five scenarios). The experimental results demonstrate that our proposed data-augmentation method produces high-quality generated data and significantly enhances the model performance, with up to 12% increase in F_1 score on event trigger detection task compared to baseline methods.

Keywords: Information Extraction, Data Augmentation, Event Extraction

1. Introduction

The goal of event extraction is to automatically extract events mentioned in natural language texts, which is essential for semantic understanding and structured knowledge construction. To train an effective event extraction model, a large annotated corpus with labeled event mentions is typically required. However, collecting and annotating such a corpus is notoriously difficult and expensive. One solution to this data insufficiency issue is data augmentation (Nishizaki, 2017; Liu et al., 2021; Jiang et al., 2021; Shorten et al., 2021; Ma et al., 2022), which creates new synthetic training data that can be used to train event extraction models.

Existing data augmentation methods mainly rely on language models (LMs) (Papanikolaou and Pierleoni, 2020; Zhang et al., 2021; Tang et al., 2022). For example, Zhang et al. (2021) propose a method of masking words in the original texts and utilizing GPT-2 (Radford et al., 2019) to make predictions as new data, and Gao et al. (2023) allow for more flexible manipulation by infilling a variable-length text span with BERT (Devlin et al., 2018). Recently, Large Language Models (LLMs) such as ChatGPT (Ouyang et al., 2022) and GPT4 (OpenAI, 2023b) have achieved great success in NLP tasks, which can also be used to generate augmented training data. However, relying solely on LMs or LLMs has two potential issues: (1) Simply masking text spans and filling in blanks do not provide sufficiently new contexts for the LMs/LLMs, resulting in limited diversity of the generated data. (2) Relying solely on LMs/LLMs to generate entire articles without proper constraints leads to the generation of unrelated and noisy events and contexts, which can harm the training of event extraction models.

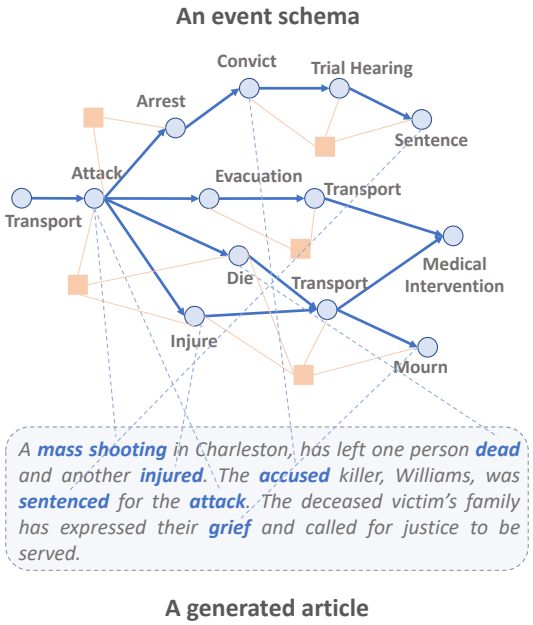

Figure 1: An example of schema-based data augmentation for event extraction. Event schemas act as a fundamental framework to direct language models in generating new articles as the augmented data for the event extraction task. By adopting this technique, the generated articles exhibit greater diversity and lower levels of noise compared to those directly generated by language models.

In this paper, we propose using event schemas to guide data augmentation. As illustrated in Figure 1, event schemas capture the common patterns of complex events in a specific domain, which are suitable for regulating the events and context gen-

LREC-COLING 2024, pages 14382–14392
20-25 May, 2024.

erated in new synthetic data. Furthermore, event schemas are typically large enough to provide sufficient diversity for the generated data. We utilize schema graphs as event skeletons for data augmentation. Specifically, we first sub-sample from a given schema graph and obtain a connected subgraph. Then, we instantiate the subgraph by replacing the abstract event/entity types with real events/entity mentions. The candidates of real event/entity mentions are collected from either the gold annotation of the news article datasets, or external commonsense knowledge bases. Finally, the instantiated subgraph is linearized into a sequence of edges according to their topological order, and converted to natural language texts using generative graph-to-text methods. These generated news articles can be used as augmented training data for event extraction models.

We conduct extensive experiments on five scenarios including Kidnapping, Ukraine Crisis, International Conflict, Disease Outbreak, and Mass Shooting. The experimental results on the event trigger detection, event trigger extraction, and event argument extraction tasks demonstrate that our proposed method achieves state-of-the-art performance, with up to 12% increase in F_1 score compared to baseline methods. Additionally, we demonstrate that the augmented training articles generated by our method exhibit significantly higher diversity compared to those directly generated by LLMs.

In summary, the key contributions of this paper are as follows:

- We propose a novel framework of data augmentation that utilizes event schemas to guide the data generation process. In contrast to previous approaches that primarily rely on word masking, substitution, or sentence rephrasing, our method enables the generation of diverse contexts and provides a larger volume of training data.
- The effectiveness of our framework is validated through experiments on event trigger detection, event trigger extraction, and event argument extraction tasks. In comparison to existing methods, our schema-based data augmentation approach produces high-quality data and demonstrates an improvement of up to 12% in F_1 score compared to baseline methods.
- Additionally, we demonstrate that relying solely on LLMs is insufficient for generating a large amount of training data with adequate diversity. In contrast, event schemas provide a robust supervision signal for LLMs and effectively guide them to generate more diverse augmented data.

2. Problem Formulation

Suppose we have a small set of news articles $\mathcal{A} = \{A_1, A_2, \cdots\}$, describing complex events in a specific domain, such as car bombing. These articles have been annotated by humans with all event mentions labeled. In addition, we have an event schema graph S available, which depicts the common pattern of these complex events. The nodes in the schema S can be either events or entities, and we use t_i to represent the type of node $i \in S$. For example, the type of an event node in S can be *Injure*, whose "victim" argument is an entity node with type *PER*.

Given that the gold annotated news article set is usually too small to train event extraction methods sufficiently, we aim to generate some new articles as augmented training data, under the guidance of the event schema. The generated data can be used to improve the performance of event extraction methods.

3. Approach

Our proposed approach consists of three steps: schema graph sub-sampling, schema subgraph instantiation, and graph-to-text generation.

3.1. Schema Graph Sub-Sampling

The event schema S usually includes all possible events and their evolution pattern in a specific domain, so it is a "superset" of new articles $\mathcal{A}$. It is important to note that a news article typically focuses on only a subset of possible events within a specific scenario. Thus, to generate a realistically plausible news article, we conduct a sub-sampling process on S. This involves extracting a subset S' from S to serve as the framework for generating the new article. For this sub-sampling, we employ a weighted sampling strategy that prioritizes event nodes. Specifically, when adding the first event into S', the probability of selecting event node i is defined as

$$p(i) = \frac{1}{2}\left(\frac{d(i)}{\Sigma_{j\in S} d(j)} + \frac{f(t_i)}{\Sigma_{j\in S} f(t_j)}\right), \quad (1)$$

where $d(i)$ is the degree of event node i in schema graph S, and $f(t_i)$ is the frequency of t_i (the type of node i) in news articles $\mathcal{A}$. In this way, the importance of an event node is measured in terms of both schema and annotated news articles: an event node is more likely to be sampled if it is a central node in S or its type appears frequently in $\mathcal{A}$.

When S' is not empty, our objective is to sample a next node that is connected to S', in order to maintain a connected graph. This is important as

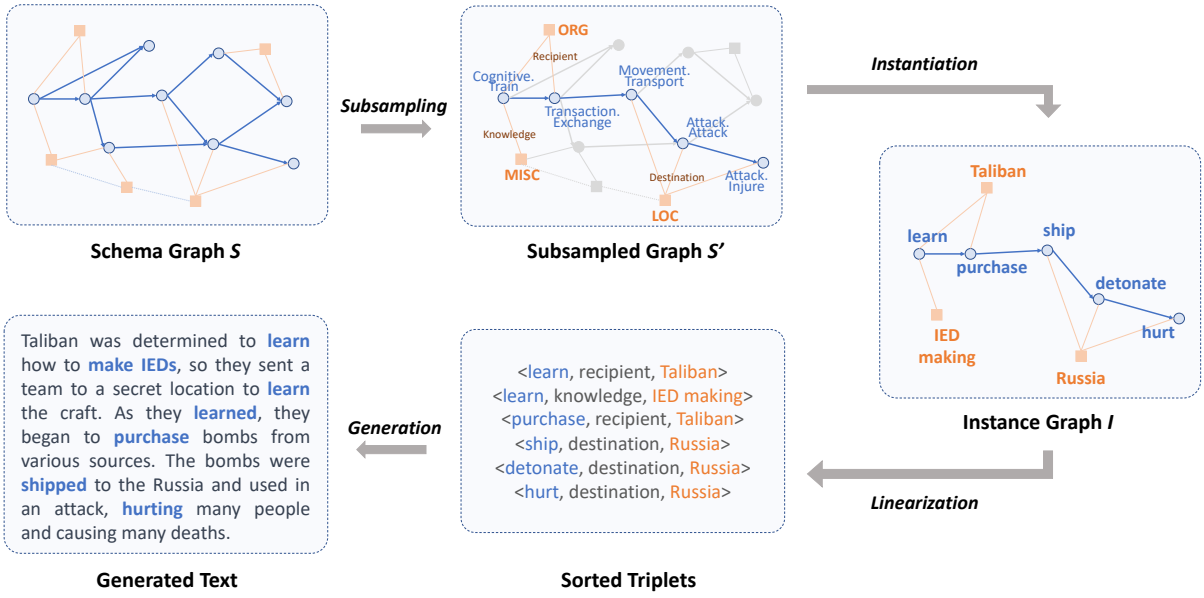

Figure 2: The overall framework of our data augmentation method. We first sample a subgraph S' from the event schema graph S. Then, we instantiate S' to I by replacing event/entity types with real events/entities. The instantiated graph I is then linearized to a list of triplets and finally, converted to texts using generative language models.

it guarantees the semantic coherence and fluency of the final news article, which is generated based on a connected schema subgraph. Therefore, the probability of adding event node i to S' when S' is not empty is defined as

$$p(i) = \begin{cases} \frac{1}{2}\left(\frac{d(i)}{\sum_{j\in\mathcal{N}_{S'}} d(j)} + \frac{f(t_i)}{\sum_{j\in\mathcal{N}_{S'}} f(t_j)}\right), & \text{if } i \in \mathcal{N}_{S'}, \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

where $\mathcal{N}_{S'} \triangleq \cup_{i\in S'}\mathcal{N}_i - S'$ and $\mathcal{N}_i$ is the set of immediate neighbors of node i in S.

We repeat the sampling process for n times and end up with a connected subgraph $S' \subseteq S$ with n event nodes, where n is a configurable hyperparameter. After obtaining S', we add all entity nodes that are directly connected with them to S'. We also add back all edges between these nodes to S', including event-event temporal relations, event-entity argument links, and entity-entity relations.

3.2. Schema Subgraph Instantiation

After sampling the subgraph S' from the schema graph S, we aim to instantiate S', since the nodes in S' only represent their abstract types rather than real event/entity information. As shown in Figure 2, the event node *Transaction.Exchange* is instantiated as *purchase*, and its "recipient" argument (whose type is *ORG*) is instantiated as *Taliban*. The instantiated subgraph can be used in the next step of graph-to-text generation, which will convert the abstract information into a natural language text.

Event Node Instantiation. To instantiate event nodes, we first establish a mapping $\mathcal{M}_{event}$ from event types to event mention candidates. The event mention candidates come from two sources:

The *internal* source of event mentions is the gold annotated news articles. For an event type t, we collect all event mentions in $\mathcal{A}$ whose types are annotated as t. For example, the event mentions *attack* and *detonate* are annotated as *Attack.Attack* in $\mathcal{A}$, so *attack* and *detonate* are added to the event mention candidates of *Attack.Attack*, and we now have $\mathcal{M}_{event}(Attack.Attack) = \{attack, detonate\}$.

The *external* source of event mentions comes from commonsense knowledge base. Specifically, we choose ConceptNet (Speer et al., 2017), a commonsense knowledge graph that connects words and phrases of natural language with labeled edges. For an event type t, we first link it to a node in ConceptNet that matches t. Then we find all nodes that have an outgoing edge with label "IsA" to this node, and use these nodes as the candidate event mentions for t. For example, ConceptNet has the following two triplets: ⟨bombing, IsA, attack⟩ and ⟨strafe, IsA, attack⟩. Therefore, *bombing* and *strafe* are added to $\mathcal{M}_{event}(Attack.Attack)$. The final mapping result of *Attack.Attack*

is therefore $\mathcal{M}_{event}(Attack.Attack) = \{attack, detonate, bombing, strafe\}$.

Entity Node Instantiation. Similar to event nodes, we build a mapping $\mathcal{M}_{entity}$ from entity types to entity mention candidates. Note that the schema S includes the following four entity types: *LOC*, *ORG*, *PER*, and *MISC*. Therefore, we first train a Named Entity Recognition (NER) model on the CoNLL-2003 (Sang and De Meulder, 2003) dataset based on the BERT-base pretrained model (Devlin et al., 2018), and then use the NER model to identify the named entities in news articles $\mathcal{A}$ with the above four entity types. Those identified named entities are taken as the entity mention candidates. For example, three location mentions are identified in $\mathcal{A}$: *Russia*, *Ukraine*, and *USA*, so we have $\mathcal{M}_{entity}(LOC) = \{Russia, Ukraine, USA\}$.

After obtaining the above two mappings, we can randomly sample one candidate mention from $\mathcal{M}_{event}(i)$ or $\mathcal{M}_{entity}(i)$ to instantiate an event or entity node i.

It is worth noting that after instantiating entity nodes, the instantiated events/entities may be unrelated or conflicting with each other. For instance, an event node is instantiated as *Attack.Attack* while its argument "victim" is instantiated as *Taliban*. Here, we propose two possible solutions: (1) Restricting the collection of candidate entity mentions to a single news article ensures that the mentions are related to each other; (2) We can prompt the graph-to-text language models (as detailed in the next subsection) to correct obvious factual errors during text generation.

Nevertheless, it is important to emphasize that the purpose of generating news articles is to provide augmented data for event extraction tasks. The primary goal is to include an adequate number of diverse event and entity mentions in the generated articles, thereby enhancing the performance of event extraction models, rather than strictly ensuring factual accuracy in the generated text. Our experimental results indicate that whether the generated news articles contain unrelated entities or factual errors, it does not significantly impact the performance of event extraction models.

3.3. Graph-to-Text Generation

After event and entity instantiation, the schema subgraph S' is converted into the instance graph I, which is used to generate natural language texts. The instance graph I is first linearized into a list of edges (triplets), which are either event-entity links or entity-entity links. Event-entity links are sorted according to the topological order of their event nodes,[1] while entity-entity links are randomly sorted since they do not have order information. Then we take the list of triplets as input for GPT 3.5 (OpenAI, 2023a), a generative language model, to generate a news article. The prompt for GPT 3.5 is "`write a news article about [scenario] using the following relations: [list of triplets]`".

We annotate the generated articles by labeling a word as an event mention if its original form matches any event node in the instance graph I. These annotated articles can be used as the augmented data to train event extraction methods and improve their performance.

4. Experiments

4.1. Datasets

To evaluate the empirical performance of our proposed method, we conduct experiments on the following datasets.

- *Kidnapping*, *Ukraine Crisis*, *Mass Shooting*, and *Disease Outbreak*. These four datasets consist of news articles gathered by us from the reference links provided on Wikipedia. Each dataset is dedicated to a specific type of complex event. The number of articles in these datasets are as follows: 54 for Kidnapping, 120 for Ukraine Crisis, 46 for Mass Shooting, and 75 for Disease Outbreak. Following that, we proceed to manually label the event triggers and their event types for each article within these datasets.
- *WCEP* (Ghalandari et al., 2020). The WCEP dataset was initially created for the purpose of multi-document summarization. It comprises news articles acquired from the Wikipedia Current Events Portal. For our training data, we select 70 news articles from WCEP, all centered around the topic of "International Conflict". Furthermore, we use a manually curated event schema that describes the complex event of international conflicts (Du et al., 2022), encompassing a total of 57 distinct event types. We manually annotate the event triggers and their corresponding event types, in accordance with the aforementioned event schema.

In our experiments, we utilize the following four available human-curated [2] event schema graphs:

[1] This is also the reason of excluding event-event temporal links from graph linearization, as they are already implied by the topological order of event-entity links.

[2] We also evaluate our proposed method with machine-generated event schemas, and present the results in ablation study.

Dataset	Kidnapping	Ukraine Crisis	WCEP	Mass Shooting	Disease Outbreak
# train/val/test documents	30/9/15	70/20/30	40/10/20	25/6/15	40/15/20
# avg. event types per doc	14.9	10.0	9.7	22.4	8.5
Corresponding schema name	**Kidnapping**	**International-conflict**		**Mass-shooting**	**Disease-outbreak**
# event/entity nodes	60/14	57/51		34/13	85/27
# event-event links	37	36		23	51
# event-entity links	57	86		29	88

Table 1: Statistics of the five datasets and their corresponding four event schemas.

Kidnapping, International-conflict, Mass-shooting, and Disease-outbreak (Du et al., 2022). The Kidnapping schema corresponds to the Kidnapping dataset, the International-conflict schema corresponds to the Ukraine Crisis and WCEP datasets, the Mass-shooting schema corresponds to the Mass Shooting dataset, while the Disease-outbreak schema corresponds to the Disease Outbreak dataset. Overall, these four schema graphs include a total number of 74, 108, 47, and 112 nodes, respectively. The detailed statistics of the datasets and schemas are presented in Table 1.

4.2. Baseline Methods

We compare our proposed method with four baseline methods to demonstrate its effectiveness. The first baseline method aims to highlight the advantages of our data augmentation technique over the absence of any data augmentation. The remaining three baseline methods serve to illustrate the improvement our method offers over other data augmentation approaches.

- *No-augmentation*. This method utilizes only the existing training data to train the event extraction model without incorporating any augmented data.
- *Mask-then-Fill* (Gao et al., 2023). In this approach, certain words in the sentences are randomly masked, and then the blanks are filled with text of variable length using a fine-tuned T5 model (Raffel et al., 2020).
- *InfoSurgeon* (Fung et al., 2021). This method involves parsing sentences from the training data into AMR (Abstract Meaning Representation) graphs. Subsequently, two event mentions are randomly selected, and their positions in the AMR graphs are altered. The modified AMR graphs are converted back into sentences and treated as augmented data.
- *Direct-generation*. This method employs GPT-3.5 (OpenAI, 2023a) to directly generate augmented data using few-shot learning. Initially, we present GPT-3.5 with a set of example news articles and request it to generate additional articles on the same topic. The prompt template we use is: *Here are some news articles examples. Article 1: XXX (Example 1). Article 2: XXX (Example 2). Article 3: XXX (Example 3). Now following a similar writing style as in the previous articles, write a news article about XXX (the scenario).* The output articles generated by GPT-3.5 are utilized as augmented training data.

4.3. Experimental Setup

Evaluation Tasks To assess the effectiveness of data augmentation methods, we generate augmented data based on the original training data, and then compare the evaluation metrics on the test set with and without using the augmented data. We employ three evaluation tasks: event trigger detection, event trigger extraction, and event argument extraction.

- *Event trigger detection.* This task aims to identify event trigger words within the given texts. It involves a binary classification where each word is labeled as either an event trigger or not.
- *Event trigger extraction.* In this task, we identify each event trigger word within the texts and classify it into one of the event types specified in the predefined ontology.
- *Event argument extraction.* This task aims to identify the entities as the event arguments and predict the argument roles they play in an event.

Foundation Models For event trigger detection and event trigger extraction task, we use DMBERT (Wang et al., 2019) as the foundation model. DMBERT (Dynamic Multi-pooling BERT) first utilizes BERT (Devlin et al., 2018) to calculate token embeddings within an input sentence. Subsequently, for each token, it calculates the max pooling results of its left and right token sequences, respectively, then concatenate the two results as the final embedding for this token. The final token embeddings

Method	Kidnapping			Ukraine Crisis			Mass Shooting		
	$Precision$	$Recall$	F_1	$Precision$	$Recall$	F_1	$Precision$	$Recall$	F_1
No-augmentation	0.657	0.881	0.753	0.779	0.778	0.778	0.807	0.856	0.831
Mask-then-Fill	0.695	0.874	0.774	0.779	0.864	0.819	0.817	0.854	0.835
InfoSurgeon	0.719	0.857	0.782	0.786	0.849	0.816	0.810	0.862	0.836
SchemaAug (ours)	**0.753**	**0.859**	**0.803**	**0.797**	**0.882**	**0.837**	**0.826**	**0.882**	**0.853**

Table 2: Results of $Precision$, $Recall$, and F_1 on Kidnapping, Ukraine Crisis, and Mass Shooting datasets for event trigger detection task. The best results are highlighted in bold.

Method	WCEP			Disease Outbreak		
	$Precision$	$Recall$	F_1	$Precision$	$Recall$	F_1
No-augmentation	0.774	0.327	0.460	0.692	0.395	0.503
Mask-then-Fill	0.789	0.477	0.595	0.735	0.565	0.639
Direct-generation	0.667	0.652	0.659	0.740	0.643	0.689
SchemaAug (ours)	**0.795**	**0.750**	**0.772**	**0.846**	**0.786**	**0.815**

Table 3: Results of $Precision$, $Recall$, and F_1 on WCEP and Disease Outbreak datasets for event trigger extraction task. The best results are highlighted in bold.

are fed into a classification head to facilitate event type classification. We use the base uncased version of BERT in experiments.

For event argument extraction task, we use DEGREE (Hsu et al., 2022) as the foundation model, which is an efficient generation-based event extraction method on low-resource data. Specifically, given a passage and a manually designed prompt, DEGREE learns to summarize the events mentioned in the passage into a natural sentence that follows a predefined pattern. The final event trigger and argument predictions are then extracted from the generated sentence with a deterministic algorithm.

Evaluation Metrics The tasks of event trigger detection, event trigger extraction, and event argument extraction can be viewed as binary or multi-class classification tasks performed on each token within the input text. As a result, we utilize $Precision$, $Recall$, and F_1 as evaluation metrics on the test set to assess the accuracy of the predicted outcomes.

Hyper-parameter Settings In all data augmentation methods, the augmented data size is twice that of the original training data. Our model is optimized using Adam with a learning rate of 5×10^{-5} and a training batch size of 16. We conduct training experiments with three different random seeds and report the average results.

4.4. Results

Comparison with Baseline Methods The results of $Precision$, $Recall$, and F_1 for event trigger detection task, event trigger extraction, and event argument extraction tasks are presented in Tables 2, 3, and 4, respectively. We utilize the Kidnapping, Ukraine Crisis, and Mass Shooting datasets for the event trigger detection task, and the WCEP and Disease Outbreak datasets for the event trigger extraction and event argument extraction task. While all four data augmentation methods demonstrate improvement in model performance, our method achieves the best results in all five scenarios across all three evaluation metrics, highlighting the effectiveness of our approach. Specifically, our method outperforms the best baseline method by 2.1%, 1.8%, and 1.7% in terms of F1 score for the event trigger detection task, by 11.3% and 12.6% for the event trigger extraction task, and by 10.2% and 7.7% for the event argument extraction task.

It is worth noting that our method exhibits a significant improvement in recall compared to the baseline methods. This can be attributed to our utilization of event schemas to generate augmented data, which includes a wider range of event types and provides greater diversity in the generated event mentions.

Diversity of the Augmented Data To assess the diversity of the generated augmented data, we define the following two types of similarity:

- *External similarity*. This refers to the average similarity between an article from the augmented data and an article from the training data. Low external similarity suggests that the augmented data significantly differ from the original training data. This distinction can yield more valuable signals for training the event extraction model.

- *Internal similarity*. This entails the average pairwise similarity among articles within the augmented data. A low internal similarity in-

Method	WCEP			Disease Outbreak		
	$Precision$	$Recall$	F_1	$Precision$	$Recall$	F_1
No-augmentation	0.237	0.178	0.203	0.436	0.203	0.277
Mask-then-Fill	0.265	0.210	0.234	0.494	0.223	0.307
SchemaAug (ours)	**0.373**	**0.306**	**0.336**	**0.552**	**0.294**	**0.384**

Table 4: Results of $Precision$, $Recall$, and F_1 on WCEP and Disease Outbreak datasets for event argument extraction task. The best results are highlighted in bold.

Dataset	Method	Similarity	
		External	Internal
WCEP	Mask-then-Fill	0.947	0.935
	Direct-generation	0.392	0.720
	SchemaAug (ours)	0.458	0.609
Disease Outbreak	Mask-then-Fill	0.945	0.947
	Direct-generation	0.501	0.812
	SchemaAug (ours)	0.510	0.689

Table 5: The diversity of the augmented data generated by our method and two baseline methods on WCEP and Disease Outbreak datasets. External similarity refers to the similarity between the augmented data and the original training data, whereas internal similarity refers to the similarity among the augmented data. A lower similarity indicates greater diversity.

dicates that the augmented data exhibit sufficient diversity and minimize repetition.

Given a pair of articles, we first use USE (Universal Sentence Encoder) (Cer et al., 2018) to encode articles into embeddings, then calculate their cosine similarity.

The diversity of the augmented data generated by our method and two baseline methods on WCEP and Disease Outbreak datasets is presented in Table 5. It is evident that the Mask-then-Fill approach exhibits the highest external and internal similarity, indicating a significantly lower diversity in the augmented data it generates. In the Mask-then-Fill method, words in articles are first masked and then filled using T5. This simplistic approach of merely masking and replacing words fails to generate sufficiently diverse contents, as the underlying contexts remain unchanged. In contrast, both Direct-generation and our SchemaAug methods achieve comparable low external similarity. Furthermore, our method demonstrates significantly lower internal similarity compared to Direct-generation. This suggests that while the data generated by GPT-3.5 differs from the original training data, it still exhibits a considerable amount of repetition. In contrast, our method leverages subsampling from event schemas, which enables the generation of diverse combinations of event sequences, thus generating articles that is more distinct with each other.

4.5. Ablation Study

Impact of the Size of the Generated Data To investigate the impact of the amount of the augmented data on the performance of the event extraction method, we conduct experiments with varying amount of augmented data relative to the size of the original training data, ranging from 0% to 300%. The results, presented in Table 6, demonstrate that our method consistently enhances the performance of the event trigger extraction method across all scenarios. However, as the size of the augmented data continues to increase, the improvements become marginal or even show a slight decline. This finding suggests that the event extraction model has effectively leveraged the generated data when its size reaches approximately 100% to 200% of the original training data size.

Impact of Event Schemas In our experiments, we utilize human-curated event schema graphs. To assess the influence of event schemas on the performance of our data augmentation method, we employ INCSCHEMA (Li et al., 2023) to automatically induce event schema graphs for the disease outbreak scenario. With this machine-induced event schema, we conduct data augmentation for the event trigger extraction task using the same settings as introduced before. The obtained F1 score is 0.759. When compared to the F1 score achieved using human-curated schema (0.815), this outcome demonstrates the correlation between the quality of event schemas and the performance of our method. However, it is worth noting that our result still significantly outperforms other baseline methods (0.503, 0.639, and 0.689).

Impact of Factual Errors As mentioned in Section 3.2, the factual accuracy is not the primary goal when we instantiate the event and entity nodes, we still perform experiments to investigate how many factual error there are in the augmented data and how they affect our model. Specifically, we manually checked 192 generated event-argument relations and examined their logical errors. It turns out that the percentage of incorrect pairs is quite low (14/192=7.3%). We further manually corrected these factual errors and trained the IE model us-

Dataset	Size of the augmented data			
	0%	100%	200%	300%
Kidnapping	0.753	**0.805**	0.803	0.803
Ukraine Crisis	0.778	0.820	**0.837**	0.836
Mass Shooting	0.831	0.844	**0.853**	0.850

Table 6: The impact of the size of the generated data on the performance of the event trigger extraction method. The numbers are F_1 scores. The best results are highlighted in bold.

ing the "gold" data. The F1 scores on WCEP and disease outbreak datasets for the event argument extraction task only increased 0.37% and 0.64%, respectively. These results demonstrate that the factual error rate is quite low and has subtle influence to the models.

4.6. Case Study

We conduct a case study to demonstrate the quality of the augmented data generated by our method compared to the baseline methods. To do this, we randomly select an event called `evacuation` and present the original training data, as well as the augmented data generated by Mask-then-Fill method and our method.

Original training data: `Some of the villages have already been evacuated and there are no obvious infrastructure targets in the areas that have been shelled.`

Augmented data generated by Mask-then-Fill: `Some of the villages have already been evacuated and there are no obvious infrastructure targets in the areas that are currently accessible.`

Augmented data generated by ours: `Efforts to evacuate individuals from the conflict zone have taken place, with Syria serving as an enclosure for those seeking safety.`

It is clear that Mask-then-Fill simply replaces the spans "`that have been shelled`" with "`that are currently accessible`" from the original sentence, demonstrating its insufficient ability to generate new contexts for event mentions. In contrast, our method is able to produce a new story for the event `evacuate`, thus providing a greater variety of training data.

5. Related Work

Event Schemas Our proposed method utilizes schema graphs to create new training data. Event schemas are induced from complex events and describe their common pattern. Researchers have proposed many schema induction methods that can automatically generate event schemas (Chambers, 2013; Li et al., 2021; Jin et al., 2022). The induced schemas can be applied in many NLP tasks (Li et al., 2019; Yao et al., 2022). For example, Wang et al. (2022) predict missing event nodes for event graphs through mapping the event instance graphs to schema graphs, then decide whether a candidate event node should be added to the instantiated event graph. Li et al. (2019) extract frames that express event information from FrameNet to construct event schema, then utilize the hierachical structure of generated schemas for event extraction tasks. Dror et al. (2023) propose to utilize large language models to generate event schemas without any manual data collection. In this paper, we utilize event schemas for a new task of data augmentation.

Data Augmentation Given the high cost of NLP data collection and annotation, data augmentation methods serve as effective means to generate synthetic datasets for numerous NLP tasks (Feng et al., 2021). The data augmentation methods in NLP field can be classified into three categories: rule-based methods, example interpolation methods, and model-based methods. Rule-based methods use easy-to-compute and predefined rules to generate new data, including random operations on word tokens (Şahin and Steedman, 2019; Wei and Zou, 2019). Example interpolation methods augment the data by interpolating the inputs between multiple real examples (Zhang et al., 2017; Verma et al., 2019; Faramarzi et al., 2020). As language models have achieved promising improvement in NLP, model-based methods use LMs for data augmentation (Sennrich et al., 2015; Yang et al., 2020; Ng et al., 2020). For example, Xie et al. (2020) propose a back-translation method that translates an existing example from one language to another, then translates back to obtain an augmented example; Yang et al. (2019) sample a batch of sentences and mask tokens using BERT language model (Devlin et al., 2018) for training data augmentation; Similarly, Gao et al. (2023) propose a Mask-then-Fill process that randomly masks some adjunct text spans and fills with variable length of words using a finetuned T5 model (Raffel et al., 2020). However, the aforementioned frameworks generate new data that closely resemble the original training data, which fail to provide sufficient new data to train the model.

Large Language Models Recently, large language models have achieved the SOTA performance in many downstream tasks. For example, Wang et al. (2023) utilize various prompts to guide LLMs perform zero-shot cross-lingual sum-

marization; (Tan et al., 2023) propose a framework McL-KBQA that incorporates in-context learning of LLMs into KBQA method that improves the effectiveness of QA tasks. While LLMs demonstrate great potential in various tasks, relying solely on them for data augmentation may introduce excessive unrelated content that adds noise to the training process. With the assistance of event schemas, LLMs are able to generate higher quality data, thereby enhancing the model performance.

6. Conclusion and Future Work

In this paper, we propose a novel data augmentation method that utilizes schema graphs to generate more synthetic training data for event extraction. Our method significantly improves the performance of event extraction models, without the need of high expense for collecting annotated data. Experimental results demonstrate that our method outperforms the existing data augmentation baselines by generating more diverse data.

In the future, we aim to integrate a schema curation step within the framework. This will enable the utilization of event schemas with varying degrees of quality as inputs. By implementing this enhancement, our framework will gain increased adaptability to a wide range of schema induction methods. In addition, applying the data augmentation method to more IE-related tasks, such as event temporal ordering, is also a promising direction.

7. Ethical Consideration

We acknowledge that our word is aligned with the *ACL Code of the Ethics* (Gotterbarn et al., 2018) and will not raise ethical concerns. We do not use sensitive datasets/models that may cause any potential issues.

8. Limitations

Our proposed method demonstrates effectiveness only in training event detection models. However, to enhance the practical utility of our model, it would be advantageous to expand the application of our method to include additional Information Extraction (IE) tasks, including relation extraction and event temporal ordering. In addition, the size of the corpora we used in the experiments is small, and we aim to expand to a larger-scaled dataset.

Furthermore, as discussed earlier, the efficacy of our method is influenced by the quality of the predefined event schemas. To address the challenge of lower-quality event schemas, it would greatly empower our framework if we incorporate schema refinement methods into our approach.

Acknowledgement

We thank the anonymous reviewers helpful suggestions. This research is based upon work supported by U.S. DARPA KAIROS Program No. FA8750-19-2-1004 and SemaFor by U.S. DARPA SemaFor Program No. HR001120C0123. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

9. Bibliographical References

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Nathanael Chambers. 2013. Event schema induction with a probabilistic entity-driven model. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 167–176.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Rotem Dror, Haoyu Wang, and Dan Roth. 2023. Zero-shot on-the-fly event schema induction.

Xinya Du, Zixuan Zhang, Sha Li, Pengfei Yu, Hongwei Wang, Tuan Lai, Xudong Lin, Ziqi Wang, Iris Liu, Ben Zhou, et al. 2022. Resin-11: Schema-guided event prediction for 11 newsworthy scenarios. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: System Demonstrations*, pages 54–63.

Arslan Erdengasileng, Qing Han, Tingting Zhao, Shubo Tian, Xin Sui, Keqiao Li, Wanjing Wang,

Jian Wang, Ting Hu, Feng Pan, et al. 2022. Pretrained models, data augmentation, and ensemble learning for biomedical information extraction and document classification. *Database*, 2022.

Mojtaba Faramarzi, Mohammad Amini, Akilesh Badrinaaraayanan, Vikas Verma, and Sarath Chandar. 2020. Patchup: A regularization technique for convolutional neural networks. *arXiv preprint arXiv:2006.07794*.

Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for nlp. *arXiv preprint arXiv:2105.03075*.

Yi Fung, Christopher Thomas, Revanth Gangi Reddy, Sandeep Polisetty, Heng Ji, Shih-Fu Chang, Kathleen McKeown, Mohit Bansal, and Avirup Sil. 2021. Infosurgeon: Cross-media fine-grained information consistency checking for fake news detection. 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 1683–1698.

Jun Gao, Changlong Yu, Wei Wang, Huan Zhao, and Ruifeng Xu. 2023. Mask-then-fill: A flexible and effective data augmentation framework for event extraction. *arXiv preprint arXiv:2301.02427*.

Demian Gholipour Ghalandari, Chris Hokamp, Nghia The Pham, John Glover, and Georgiana Ifrim. 2020. A large-scale multi-document summarization dataset from the wikipedia current events portal. *arXiv preprint arXiv:2005.10070*.

DW Gotterbarn, Bo Brinkman, Catherine Flick, Michael S Kirkpatrick, Keith Miller, Kate Vazansky, and Marty J Wolf. 2018. Acm code of ethics and professional conduct.

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.

Zhengbao Jiang, Jialong Han, Bunyamin Sisman, and Xin Luna Dong. 2021. Cori: Collective relation integration with data augmentation for open information extraction. *arXiv preprint arXiv:2106.00793*.

Xiaomeng Jin, Manling Li, and Heng Ji. 2022. Event schema induction with double graph autoencoders. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2013–2025.

Manling Li, Sha Li, Zhenhailong Wang, Lifu Huang, Kyunghyun Cho, Heng Ji, Jiawei Han, and Clare Voss. 2021. Future is not one-dimensional: Graph modeling based complex event schema induction for event prediction. *arXiv preprint arXiv:2104.06344*.

Sha Li, Ruining Zhao, Manling Li, Heng Ji, Chris Callison-Burch, and Jiawei Han. 2023. Open-domain hierarchical event schema induction by incremental prompting and verification. In *Proc. The 61st Annual Meeting of the Association for Computational Linguistics (ACL2023)*.

Wei Li, Dezhi Cheng, Lei He, Yuanzhuo Wang, and Xiaolong Jin. 2019. Joint event extraction based on hierarchical event schemas from framenet. *IEEE Access*, 7:25001–25015.

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.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Manfu Ma, Xiaoxue Li, Yong Li, Xinyu Zhao, Xia Wang, and Hai Jia. 2022. 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.

Nathan Ng, Kyunghyun Cho, and Marzyeh Ghassemi. 2020. Ssmba: Self-supervised manifold based data augmentation for improving out-of-domain robustness. *arXiv preprint arXiv:2009.10195*.

Hiromitsu Nishizaki. 2017. Data augmentation and feature extraction using variational autoencoder for acoustic modeling. In *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1222–1227. IEEE.

OpenAI. 2023a. Chatgpt: Optimizing language models for dialogue.

OpenAI. 2023b. Gpt-4 technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language

models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Yannis Papanikolaou and Andrea Pierleoni. 2020. Dare: Data augmented relation extraction with gpt-2. *arXiv preprint arXiv:2004.13845*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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.

Gözde Gül Şahin and Mark Steedman. 2019. Data augmentation via dependency tree morphing for low-resource languages. *arXiv preprint arXiv:1903.09460*.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Connor Shorten, Taghi M Khoshgoftaar, and Borko Furht. 2021. Text data augmentation for deep learning. *Journal of big Data*, 8(1):1–34.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*.

Chuanyuan Tan, Yuehe Chen, Wenbiao Shao, and Wenliang Chen. 2023. Make a choice! knowledge base question answering with in-context learning.

Changhao Tang, Kun Ma, Benkuan Cui, Ke Ji, and Ajith Abraham. 2022. Long text feature extraction network with data augmentation. *Applied Intelligence*, pages 1–16.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. 2019. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pages 6438–6447. PMLR.

Hongwei Wang, Zixuan Zhang, Sha Li, Jiawei Han, Yizhou Sun, Hanghang Tong, Joseph P Olive, and Heng Ji. 2022. Schema-guided event graph completion. *arXiv preprint arXiv:2206.02921*.

Jiaan Wang, Yunlong Liang, Fandong Meng, Beiqi Zou, Zhixu Li, Jianfeng Qu, and Jie Zhou. 2023. Zero-shot cross-lingual summarization via large language models.

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

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33:6256–6268.

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.

Yiben Yang, Chaitanya Malaviya, Jared Fernandez, Swabha Swayamdipta, Ronan Le Bras, Ji-Ping Wang, Chandra Bhagavatula, Yejin Choi, and Doug Downey. 2020. Generative data augmentation for commonsense reasoning. *arXiv preprint arXiv:2004.11546*.

Yunzhi Yao, Shengyu Mao, Xiang Chen, Ningyu Zhang, Shumin Deng, and Huajun Chen. 2022. Schema-aware reference as prompt improves data-efficient relational triple and event extraction. *arXiv preprint arXiv:2210.10709*.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Meng Zhang, Zhiwen Xie, and Jin Liu. 2021. Data augmentation based on pre-trained language model for event detection. In *China Conference on Knowledge Graph and Semantic Computing*, pages 59–68. Springer.