Zero-Shot On-the-Fly Event Schema Induction

Anonymous EMNLP submission

Abstract

What are the events involved in a pandemic outbreak? What steps should be taken when planning a wedding? The answers to these questions can be found by collecting many documents on the complex event of interest, extracting relevant information, and analyzing it. We present a new approach[1] in which large language models are utilized to generate source documents that allow predicting, given a high-level event definition, the specific events, arguments, and relations between them to construct a schema that describes the complex event in its entirety. Using our model, complete schemas on any topic can be generated on-the-fly without any data collection needed, i.e., in a zero-shot manner. Moreover, we develop efficient methods to extract pertinent information from texts and demonstrate, in a series of experiments, that these schemas are considered to be more complete than human-curated ones in the majority of examined scenarios. Finally, we show that this framework is comparable in performance with previous supervised schema induction methods that rely on collecting real texts, while being more general and flexible by avoiding the need to use a predefined ontology.

1 Introduction

Event processing refers to tracking, analyzing, and drawing conclusions from streams of information about events. This event analysis aims at identifying meaningful events (such as opportunities or threats) in real-time situations and responding appropriately. Event processing can also be utilized to gain a deep understanding of the specific steps, arguments, and relations between them that are involved in a complex event. The information above can be consolidated into a graphical representation called an *event schema* (Li et al., 2021).

Consider the example schema of kidnapping presented in Fig. 1. This representation of events and participants assists in gaining an understanding of the complex event of kidnapping and could help composing a reaction plan if needed.

The NLP community has devoted much effort to understanding events that are described in a document or in a collection of documents for this purpose. These efforts include identifying event triggers (Lu and Roth, 2012; Huang et al., 2018; Wadden et al., 2019; Han et al., 2019), extracting event arguments (Punyakanok et al., 2008; Peng et al., 2016; Lin et al., 2020; Zhang et al., 2021a), and predicting the relations between events, e.g., temporal, coreference, causal or hierarchical relations (Do et al., 2012; Lee et al., 2012; Glavaš et al., 2014; Caselli and Vossen, 2017; Ning et al., 2018; Wang et al., 2020; Zhang et al., 2020a).

Previous works on event schema induction relied on the information extracted from collected documents to build the schema graph. For instance, Li et al. (2020) learn an auto-regressive language model (LM) over paths in the instance graphs depicting events, arguments and relations of instances of the complex events, and later on construct a schema graph by merging the top k ranked paths. However, their approach requires access to many documents on each topic of interest, which can be extremely laborious and time consuming to obtain.

Our goal, on the other hand, is to allow creating schemas on-the-fly by taking as input only the name of the complex event of interest (like a "pandemic outbreak" or an "armed robbery"). To avoid collecting many documents on the topic of the schema, we utilize pre-trained auto-regressive text generation models, specifically GPT-3 (Brown et al., 2020), to generate texts on the desired topic (examples presented in Fig. 2). These documents are then processed to extract pertinent information, from which a schema is constructed. The fact that we do not collect any data makes our learning

[1] Our code and data will be made publicly available upon acceptance.

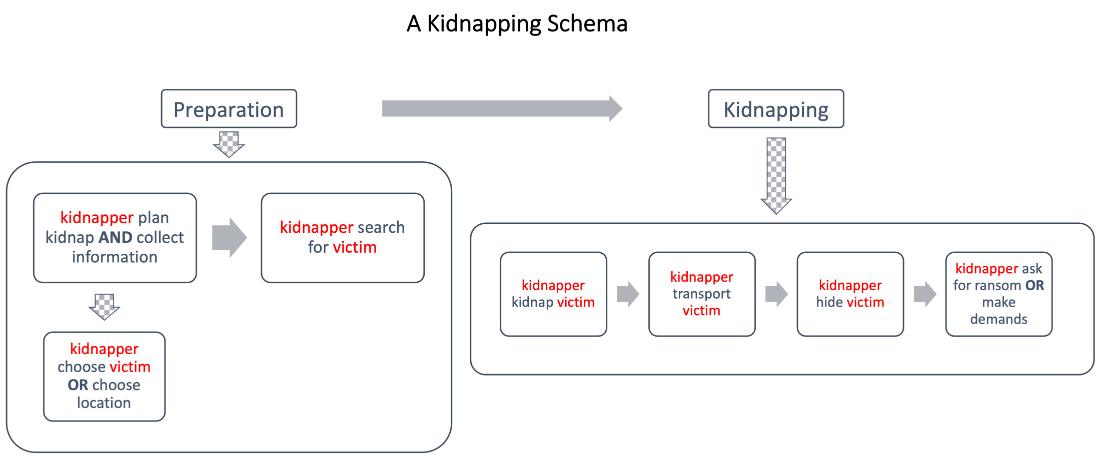


Figure 1: An example schema for the event of "Kidnapping". The gray arrows present temporal relations and the checkered arrows present hierarchical relations (PARENT-CHILD).

framework zero-shot since we do not rely on any human-collected articles or example schemas.

In addition to making the induction faster by eliminating the need to collect data, we also made the information extraction process faster by implementing new and efficient methods for identifying temporal and hierarchical relations between events mentioned in the text. These two steps are the most time consuming in the process of schema induction and could take up to 2 hours each. Accepting the whole text as input instead of two sentences at each time, the proposed model shortens the inference time significantly to several minutes without enduring a major loss in performance.

The process of generating texts is explained in Section §3, and the process of extracting relevant and salient information is described in Section §4, then we introduce the construction of the schema graph in Section §5. To evaluate our zero-shot schema generator we conduct experiments on a benchmark dataset for schema induction (LDC2020E25) and provide a new dataset for further evaluation called Schema-11. Additionally, we design a subject-matter expert Turing test, a.k.a. Feigenbaum test (Feigenbaum, 2003), to determine whether our algorithm could mimic experts' response towards several common complex event scenarios. The experiments and results are presented in Section §6.

The contributions of our work include:

1. Predicting an entire schema given the name of a complex event without collecting data.

2. Implementing a novel and efficient One-Pass approach for identifying temporal and hierarchical relations between events.

3. Presenting a method for automatically inducing logical relations between events based on temporal relations.

4. Offering a Feigenbaum test for evaluation on a new schema dataset, Schema-11.

2 Related Work

Schema induction: Early schema induction efforts focused on identifying the triggers and participants of atomic events without considering relations between atomic events that comprise complex schemas (Chambers, 2013; Cheung et al., 2013; Nguyen et al., 2015; Sha et al., 2016; Yuan et al., 2018). More recent work focuses on inducing schemas for pairs of events (Li et al., 2020) and multiple events (Zhang et al., 2021b; Li et al., 2021), but they require access to large corpora for the induction process. In this work, we induce schemas on-the-fly in a zero-shot manner. As is standard in state-of-the-art (SOTA) works (Li et al., 2020, 2021; Wen et al., 2021), we output all the information about relations between events and arguments extracted from the text, in addition to logical and hierarchical relations not studied previously.

Script learning: Early script learning works concentrated on chains of events with a single protagonist (Chambers and Jurafsky, 2008, 2009; Jans et al., 2012; Rudinger et al., 2015; Granroth-Wilding and Clark, 2016) and later extended to multiple protagonists (Pichotta and Mooney, 2014; Peng and Roth, 2016; Pichotta and Mooney, 2016; Modi, 2016; Weber et al., 2018, 2020; Zhang et al., 2020b). All of these works assume there exists a single line of events that describes all occurrences within a complex event. This work does not limit itself to generating single-chained schemas. We also consider more complex graphs as schema outputs. In addition, none of these works deals with zero-shot scenarios that do not require training data.

Pre-trained generation models: Large-scale pre-trained text generation models such as GPT-3 (Brown et al., 2020), BART (Lewis et al., 2020), T5 (Raffel et al., 2020), i.a. have been used in NLP to solve many tasks. These models are often seen as few-shot learners (Brown et al., 2020) and therefore used as inference methods. However, these text generation models are not explicitly trained to perform inference, rather they are trained to produce the most likely sequence of words to proceed a certain prompt, similar to language models (which is also how researchers refer to them).

In a recently published paper, Wang et al. (2021b) used GPT-3 to generate training data in a few-shot inference paradigm by querying the model with a prompt and a few examples in order to create additional examples with a desired label. Later they used the generated data to fine-tune standard pre-trained models to perform inference. Our work, however, uses these large pre-trained LMs only as text generators. We generate documents on a particular topic and use it as a corpus for extracting the topic's schema. We rely on the intuition that the generated text will include salient and stereotypical information that is expected to be mentioned in the context of the topic (e.g., for a topic of "planning a wedding," we assume most documents will include the event "order catering").

3 Data Generation

The schema induction process begins with generating texts using large LMs as text generation models. These texts are joined to form a knowledge base for the schema, including all of the potential information that the schema may present. One could, of course, create this knowledge base by crawling the web for real news articles or Wikipedia entries related to a certain topic.

We argue, however, that in addition to the obvious advantages of not having to rely on the availability of data online and not having to crawl the entire web for relevant documents on each topic, the generated data from these large generative models is more efficient in reporting salient events than random events described in the news, i.e., generated texts are more likely to mention important information than real documents do.

	Generated Text	Real Text
# events / # tokens	0.1252	0.0631
# arguments / # tokens	0.0545	0.0301

Table 1: The ratio of relevant events and relevant argument roles identified in generated text and real text for the scenario of IED attack.

Our analysis shows that the generated stories contain a higher percentage of relevant tokens than existing real news articles that are used for schema induction. To demonstrate this phenomenon, we compare manually gathered documents with those that are automatically generated for the event of Improvised Explosive Device (IED) attack (Li et al., 2021). To identify salient events and arguments concerning IED attacks, we adopt the DARPA KAIROS Phase 1 (v3.0) ontology[2], a fine-grained ontology for schema learning, with 24 entity types, 67 event types, and 85 argument roles.

We calculated the number of relevant event triggers and arguments identified in the text, where a relevant mention is one whose type appears in the ontology. The results shown in Table 1 demonstrate that the quality of the generated texts in terms of conciseness and appearance of important details is higher than that of real texts. For example, the ratio of relevant events per token is more than twice as high in generated texts as it is in real texts. Hence we are able to not only generate a schema for every given topic without putting any effort in searching the web, but the information we generate is also better suited for our end task of depicting all of the important aspects of a complex event.

Given a topic for which we want to create a schema, we generate multiple texts that discuss the topic event using the OpenAI GPT-3 API[3] with the Davinci-instruct-beta-v3 model. We use three prompting methods to generate documents of diverse genres as follows:

News articles: We begin by generating a headline using the prompt: "Write a news headline about *topic*." The output from this prompt, which appears between quotation marks, is then used in the following prompt: "Write a news story titled *headline*."

[2] The full ontology definition can be accessed at this link: https://bit.ly/3mIWJoN.

[3] https://openai.com/blog/openai-api/.

Prompt: What are the steps involved in holding a summit meeting?
1.
Output:

1. Plan the agenda and objectives of the meeting.
2. Select a venue that is appropriate for the meeting's purpose and size.
3. Send out invitations and ensure all participants have the appropriate information.
4. Prepare for the meeting by creating materials and briefing the participants.
5. Hold the meeting and follow up on any action items that are generated.

(a) Example of generation of steps involved in the event of holding a summit meeting.

First prompt: Write a news headline about filing a patent.
Output: New Patent Filing Could Speed Up Drug Development.
Second prompt: Write a news story titled "New Patent Filing Could Speed Up Drug Development".
Output: The United States Patent and Trademark Office has just published a new patent that could speed up the drug development process. The patent, filed by a team of researchers from the DPharm Institute, covers a new drug screening process that could help identify potential new drugs much more quickly. The new process, which is based on a computer algorithm, is said to be up to 10 times faster than traditional methods. It can identify potential new drugs by screening through large databases of existing drugs and drug candidates. The DPharm Institute is already in the process of licensing the patent to several pharmaceutical companies, and is planning to commercialize the technology over the next few years.

(b) Example of generation of news-like text for the event of filing a patent.

Prompt: Describe how to 3D print an object?
Output: To 3D print an object, you need to first design the object in a 3D modeling program. Once the object is designed, you need to export it as a 3D printable file format. Next, you need to choose a 3D printer and print settings. Finally, you need to print the object.

(c) Example of generation of how-to text for the event of 3D printing.

Figure 2: Examples of generated texts using different prompting methods. The highlighted text is relevant events that will be extracted in the information extraction step.

The output from the second prompt is added to the pool of generated texts. The process is repeated 30 times. See example in Fig. 2b.

How-to articles: For this genre type, we use the prompt: "Describe how to *topic*." The process is repeated 30 times and all the generated texts are added to the pool. See example in Fig. 2c.

Direct step-by-step schema: Here we use the prompt: "What are the steps involved in *topic*? 1."[4] to allow GPT-3 to generate a schema directly. We run this process once. See example in Fig. 2a.

Generating documents in various genres enables our model to induce comprehensive schemas on any given topic. Considering that some events are more likely to be in the news (e.g., elections, pandemic outbreaks) while others are more technical in nature and are hence less newsworthy (such as earning a Ph.D. degree or planning a wedding), we generate diverse texts and then use a ranking model to choose the most relevant documents. The ranking process includes embedding the texts and the topic with the model of Reimers and Gurevych (2019), then cosine similarity is calculated between each text and the topic embeddings, and only the 30 texts closest to the topic are selected together with the output from the direct step-by-step schema.

The following section describes the next step in generating a schema: extracting all the relevant information from the selected texts.

4 Information Extraction

For each document, we extract event triggers, arguments and relations between the events that are important and relevant to the schema topic. We do not work with a predefined ontology that defines in advance what events and arguments are essential, so we extract all the information and later filter it down to include just the most frequent items. Here are the steps involved in extracting the information:

1. **Semantic Role Labeling (SRL)**: We use the SOTA SRL system[5] trained on CoNLL12 (Pradhan et al., 2012) and Nombank dataset (Meyers et al., 2004) to extract both verb and nominal event triggers and arguments.

2. **Named Entity Recognition (NER)**: We employ the SOTA NER model to extract and map entities (potential arguments of events)

[4]The "1." in the prompt is for GPT-3 to automatically complete the steps.

[5]Details for the SRL and NER systems were removed for anonymity and will be published upon acceptance.

into entity types defined in the CoNLL 2002 dataset (Tjong Kim Sang, 2002) and the LORELEI project (Strassel and Tracey, 2016).

3. **Constituency Parsing**: Since the arguments extracted by SRL can be clauses and long phrasal nouns, we employ the constituency parsing model from AllenNLP[6] for argument head word extraction. For example, in this sentence "The first passengers rescued from a helicopter that ditched in the North Sea have arrived at hospital," the ARGM-LOC for "ditched" is "in the North Sea." However, the NER model can only extract "North Sea" instead of "in the North Sea," and thus we use the parser to match the argument to its type.

4. **Coreference Resolution**: We use the SOTA model (Yu et al., 2020) for event and entity coreference resolution to identify within-document coreferential relations.

5. **Temporal Relation Extraction**: We first try to use SOTA models (Ning et al., 2019; Zhou et al., 2021) to predict the temporal relations[7] between all possible pairs of extracted events but since the SOTA models accept two sentences containing events as input, the inference time[8] for an n-event document is $\mathcal{O}(n^2)$, making the schema induction process several hours long. We develop a One-Pass model[9] that takes the document as input and uses the contextual representation of events to predict relations between them. As shown in Table 2, the inference time is shortened 63-186 times on average, while the performance of the One-Pass model is comparable to SOTA models.

6. **Hierarchical Relation Extraction**: The extremely long inference time of SOTA models for predicting hierarchical relations (PARENT-CHILD, CHILD-PARENT, COREF, NOREL) (Zhou et al., 2020; Wang et al., 2021a) also impairs the efficiency of our schema induction system. Thus we use the same One-Pass methodology to extract hierarchical relations.

Corpus	Model	Metrics F_1 score	Speed	GPU Memory
HiEve	Zhou et al. (2020)	0.489	-	-
	Wang et al. (2021a)	0.522	41.68s	4515MiB
	One-Pass model	0.472	0.65s	2941MiB
MATRES	Ning et al. (2019)	0.767	30.12s	4187MiB
	Zhou et al. (2021)	0.821	89.36s	9311MiB
	One-Pass model	0.768	0.48s	2419MiB

Table 2: Performance comparison between our One-Pass model and SOTA models for event temporal and hierarchical relation extraction. We report F_1 scores on benchmark datasets (HiEve for hierarchical relations, MATRES for temporal relations), speed (average inference time for 100 relations), and required GPU memory during inference. The One-Pass models are 63-186 times faster than SOTA models and take up only 26%-65% of the GPU memory required by SOTA models, while being comparable in performance.

We observe that the inference time is greatly shortened, and the One-Pass model achieves comparable results to previous models, and it takes up less GPU memory (see Table 2).

After processing the data using the procedure described above, we get a list of events, their arguments, and relations between the events. We concentrate on events and relations that frequently appear in the generated texts since we assume those are the most important to add to the schema (without having any other source of information that could identify what is salient). The next section describes the process of building a schema.

5 Schema Induction

To consolidate the information extracted from the previous step, we build a schema as follows:

Make a list of events and relations: To compare similar event mentions in different texts, we compare the event trigger itself (whether they are the same verb or coreferential verbs[10]) and the NER types of its arguments. For example, the trigger "(take) precautions" appeared in 5 documents generated for the topic of Pandemic Outbreak. In two documents the subject of the verb phrase "take precautions" was "residents", in another two it was "people" and in the last one, it was "public". Nevertheless, the NER type is identical in all cases (PER), and thus we set the frequency of "(take) precautions" to 5. Similarly, we calculate the frequency of the temporal and hierarchical relations.

[6] https://demo.allennlp.org/constituency-parsing.

[7] The possible temporal relations (start-time comparison) are: BEFORE, AFTER, EQUAL and VAGUE.

[8] The inference time is mostly spent on obtaining the contextual representation of events using large fine-tuned LMs.

[9] We take advantage of the recently developed BigBird (Zaheer et al., 2020) that handles long sequences with sparse attention mechanism.

[10] We only consider coreferential relations if they appeared in more than 2 documents.

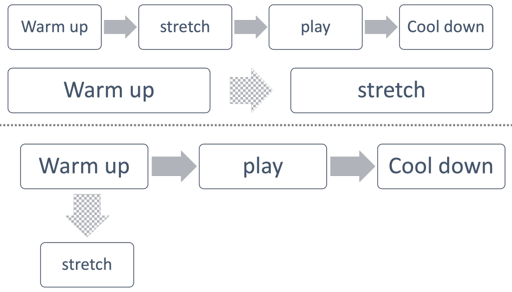


Figure 3: Example of amending a timeline in the schema of "Sports Games". The timeline at the top includes events of two different levels ("warm up" is the parent of "stretch"), hence it is rectified to include only events of the same level like the timeline at the bottom. Gray arrows mark temporal relations, and checkered arrows denote PARENT-CHILD.

We consider only the top-30 most frequent events and relations for the schema and continue to the next step.

Construct timelines: We construct the longest timelines from the list of temporal relations. This list is a list of tuples (A, B), indicating that event A happened before event B. To construct a timeline, we search recursively for the longest chains of the following form $(A, B), (B, C), (A, C)$ and so on.

Fix timelines according to hierarchical relations: We build a hierarchy of the events using the hierarchical relation list[11] and change the timelines so that they will only include events that appear in the same level of hierarchy (see example in Fig. 3).

Add logical relations: The final step is to combine the timelines and hierarchies into a single graph using logical relations (AND/OR). When observing two timelines with discrepancies between the order of events, we place a logical AND between them since we interpret this discrepancy as both events occurring at the same time or there is no significance to the order of those two events. For example, in Fig. 4, the events "call demands" and "clash officers" appear in different orders in different documents, hence we conclude that they occur simultaneously or interleaved. We use a logical OR to mark different outcomes or events that can happen simultaneously but not necessarily. For example, in Fig. 4, the events "police disperse crowd" and "government urge to exercise restraint" may both occur or either one of them occurs.

The final output is a schema graph that contains all the events, arguments, and the temporal, hierarchical and logical relations between the events. This schema generating model can also be used to extend the scope of existing schemas by further querying the model on more specific topics. For example, the schema in Fig. 1 does not cover the consequences of kidnapping, probably because the LM did not attend to this aspect. Hence an analyst can input another topic (e.g, consequences of kidnapping) to further develop the schema. Similarly, analysts can generate schemas for very specific events (e.g., kidnapping in a political setting).

Next, we provide an in-depth experimentation for the proposed schema induction framework.

6 Experiments

6.1 Data

We conduct experiments on a dataset for general schema learning released by LDC (LDC2020E25). The corpus includes 84 types of complex events, such as Cyber Attack, Farming and Recycling. This dataset includes ground-truth schemas created by LDC annotators.

In addition to the LDC dataset, we also collected human generated schemas for 11 complex events (denoted henceforth as the Schema-11 dataset)[12]. These schemas were generated by four human experts[13] that were instructed to write a schema on each topic based on their commonsense knowledge that includes a list of events, relations[14], arguments and their NER types[15].

6.2 Evaluation

We follow Li et al. (2021) to use instance coverage and last event prediction to evaluate our method on the LDC dataset; for the Schema-11 dataset, we ask human testers to assess the completeness and soundness of both human- and automatically-generated schemas.

Coverage and Prediction A common evaluation method in schema induction and script prediction is to calculate the recall of events and relations

[11]We only consider PARENT-CHILD and CHILD-PARENT relations that appear in more than 2 documents.

[12]The topics are: Bombing Attack, Business Change, Civil Unrest, Disaster and Rescue, Elections, International Conflict, Kidnapping, Mass Shooting, Pandemic Outbreak, Sports Games, and Terrorism Attack.

[13]Graduate students who are familiar with the research topic of schema induction and are not the authors of this paper.

[14]No restrictions were placed for the annotators. For example, in one case, an annotator mentioned causal relations that are not covered in our framework.

[15]The annotators are familiar with SRL annotations (e.g., ARG0, ARG1, etc.) and NER types (e.g., PER, ORG, etc.).

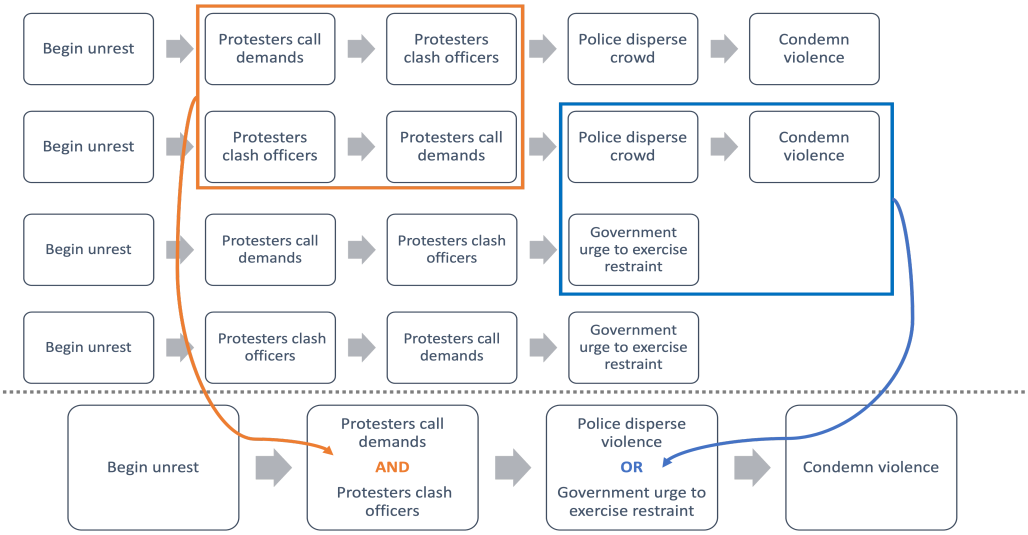


Figure 4: An example of integrating timelines and logical relations in the schema of Civil Unrest. The four upper timelines are the ones extracted from the generated texts and the lower one is their merger into a single timeline with logical relations.

predicted by the model, assuming the human annotators' results are gold labels (coverage) and to calculate the accuracy in predicting the final outcome of a scenario (prediction). Li et al. (2021), for example, calculated the accuracy of predicting the last event type of the LDC schemas. Here we present the results of predicting the last events using event triggers, instead of event types.

Feigenbaum Test We show human testers two schemas on each topic in the Schema-11 dataset (see example in §A). One schema was automatically generated by our model, and the other was randomly sampled from the Schema-11 corpus[16].

We ask the testers to determine which events and relations are valid to appear in the schema (soundness) and the following questions: which schema is more complete in the sense of including all the events needed to describe the topic, and which schema, in their opinion, was generated by a human expert (as opposed to a machine).

6.3 Results

Coverage We calculate the intersection between events in the generated schemas and the gold schemas in two ways: (a) match event triggers, and (b) match event triggers and synonyms of the events in the gold schemas (synonym coverage)[17]. We believe that calculating synonym coverage is a better evaluation methodology to avoid errors such as considering different verbs describing the same action as different (e.g., "buy" and "acquire") than using a predefined ontology of event types such as the one used in Li et al. (2021). The reason is twofold: firstly, any predefined ontology is limited to certain scenarios and it may impair the variety of events extracted; and secondly the typing mechanism may also inflict errors to the schema.

From the results in Table 3, we can observe that despite the difficulty of exact matching, our model can cover 23.73% events in the gold annotations, showing that the generated text has a good coverage of events required in the schemas. And if we use synonym coverage as our metric, we achieve a promising coverage of 36.35% while the state-of-the-art supervised event graph model (Li et al., 2021) covers 54.84% using limited event types. Furthermore, with the high quality event representations obtained from the One-Pass model and the proposed logical relation induction algorithm, our method covers 14.09% of all the relations annotated in the gold schemas, whereas the best performance achieved by the event graph model is 44.44%. The high coverage of the SOTA method can be attributed to the joint modeling of multiple relations using graph neural networks, which is impracticable in our zero-shot settings.

Prediction In the prediction task, our schemas are able to predict the final outcome in 46.42% of the cases for the LDC schemas (see Tab. 4). This

[16] In some cases we combine two randomly sampled schemas because the length of the human schemas tend to be shorter than the automatically generated ones.

[17] Implemented using the NLTK WordNet Python package.

	Ours			(Li et al., 2021)
	Coverage	Coverage (Synonym)	Total	Coverage
Event Match	23.73	36.35	36.35	54.84
Temporal Relations	1.99	5.80		
Hierarchical Relations	0.14	0.91	14.09	44.44
Logical Relations	4.56	7.38		

Table 3: Coverage results for the LDC dataset. The first row presents the percentage of events that appeared in both the LDC schemas and the automatically generated schemas (out of events in LDC schemas), and the three bottom rows calculates the same metric for relations of different types. Total is the sum of all three types of relations.

Model	Accuracy
Event Language Model	49.7
Sequential Pattern Mining	47.8
Human Schema	20.5
Event Graph Model	52.0
Zero-Shot Schema	28.5
Zero-Shot Schema Synonym	46.4

Table 4: Experimental results for last event prediction in the LDC dataset. The top 4 results are from (Li et al., 2021), and the metric is HITS@1 where the events are typed based on a predefined ontology.

	S1	**S2**	**S3**	**S4**	S5	S6	S7	S8	**S9**	S10	S11
Human	4	0	1	1	1	2	1	1	0	3	1
Automatic	2	3	4	2	1	1	1	1	4	0	1

Table 5: Distribution of votes for which is the more complete schema for Schema-11 dataset.

result is extremely impressive when it is compared with Li et al. (2021) since they predict event types instead of verbs, which is a much easier task due to the fact that the set of possible answers is limited.

Feigenbaum test In the soundness experiments, where the testers are asked to decide which events and relations are valid to appear in the schema, it turns out that human generated schemas contain 7.14% invalid events and 15.4% invalid relations on average. For the automatically generated schemas, 6.06% of the events and 22.9% of the relations are considered to be invalid on average. We observe that the average percentage of valid events is higher in the automated schemas, yet the soundness of induced relations is relative inferior.

For the completeness results, in 4 cases the testers agreed that the automatically generated schemas are more complete; in 3 cases they claimed that the human schemas are more complete; and the result is a tie in the remaining 4 cases. The distribution of votes for completeness is presented in Tab. 5. Hence our automatically generated schemas are of comparable quality to human generated ones in the sense of completeness.

Finally, in the Feigenbaum test, where testers are asked to decide whether a schema is generated by a human or a machine, 8 out of 11 times they correctly identify the human-generated schema, 1 incorrectly, and 2 ties. Some of the testers who succeeded in their guesses mentioned that it was easy to determine which schema was automatically generated since it tends to be longer and more complete. Although in the test, the machine-generated schemas fail to deceive the testers into misidentifying them as human generated ones, the experiments shed light on future directions, e.g., keeping the most salient events in the schema, improving the accuracy of temporal and hierarchical relation extraction, developing reliable approaches for causal relation extraction, and so forth. The full results from the Feigenbaum test appear in §B.

7 Conclusion

We propose a method to generate schemas given the sole input of a topic. We use GPT-3 to generate texts of diverse genres and a pipeline of information extraction tools to obtain relevant information before inducing logical relations and integrating the events and relations into a schema graph. To improve the efficiency of the pipeline, we implement One-Pass models for event temporal and hierarchical relations that achieve comparable performances with SOTA models but require far less inference time and GPU memory space. To evaluate our framework, we conduct experiments on the benchmark LDC dataset to show that our schemas cover a decent amount of pertinent information and display comparable ability for event prediction with supervised approaches. Although our proposed method fails the Feigenbaum test on Schema-11, we observe a very high percentage of valid events and relations and the testers endorsed the completeness of our machine-generated schemas.

8 Ethical Consideration

The proposed schema induction method does not present any direct societal implications. As is observed in Abid et al. (2021), the text generated by GPT-3 might include undesired social bias. Extracting events and relations from text with such social bias might potentially propagate the bias to the induced schemas. Besides, there are risks of malicious or unintended harmful uses of the generated schemas, for instance, the system might be used to inquire about making a bomb or contriving a terrorist attacks. Yet we believe that the proposed method can benefit various downstream NLP/NLU tasks like event prediction, task-oriented dialogue agents (Andreas et al., 2020) and risk detection (Pohl et al., 2012).

References

Abubakar Abid, Maheen Farooqi, and James Zou. 2021. Large language models associate muslims with violence. *Nature Machine Intelligence*, 3(6):461–463.

Jacob Andreas, John Bufe, David Burkett, Charles Chen, Josh Clausman, Jean Crawford, Kate Crim, Jordan DeLoach, Leah Dorner, Jason Eisner, Hao Fang, Alan Guo, David Hall, Kristin Hayes, Kellie Hill, Diana Ho, Wendy Iwaszuk, Smriti Jha, Dan Klein, Jayant Krishnamurthy, Theo Lanman, Percy Liang, Christopher H. Lin, Ilya Lintsbakh, Andy McGovern, Aleksandr Nisnevich, Adam Pauls, Dmitrij Petters, Brent Read, Dan Roth, Subhro Roy, Jesse Rusak, Beth Short, Div Slomin, Ben Snyder, Stephon Striplin, Yu Su, Zachary Tellman, Sam Thomson, Andrei Vorobev, Izabela Witoszko, Jason Wolfe, Abby Wray, Yuchen Zhang, and Alexander Zotov. 2020. Task-oriented dialogue as dataflow synthesis. *Transactions of the Association for Computational Linguistics*, 8:556–571.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Tommaso Caselli and Piek Vossen. 2017. The event StoryLine corpus: A new benchmark for causal and temporal relation extraction. In *Proceedings of the Events and Stories in the News Workshop*, pages 77–86, Vancouver, Canada. Association for Computational Linguistics.

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, Seattle, Washington, USA. Association for Computational Linguistics.

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. In *Proceedings of ACL-08: HLT*, pages 789–797, Columbus, Ohio. Association for Computational Linguistics.

Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 602–610, Suntec, Singapore. Association for Computational Linguistics.

Jackie Chi Kit Cheung, Hoifung Poon, and Lucy Vanderwende. 2013. Probabilistic frame induction. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 837–846, Atlanta, Georgia. Association for Computational Linguistics.

Quang Do, Wei Lu, and Dan Roth. 2012. Joint inference for event timeline construction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 677–687, Jeju Island, Korea. Association for Computational Linguistics.

Edward A Feigenbaum. 2003. Some challenges and grand challenges for computational intelligence. *Journal of the ACM (JACM)*, 50(1):32–40.

Goran Glavaš, Jan Šnajder, Marie-Francine Moens, and Parisa Kordjamshidi. 2014. HiEve: A corpus for extracting event hierarchies from news stories. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3678–3683, Reykjavik, Iceland. European Language Resources Association (ELRA).

Mark Granroth-Wilding and Stephen Clark. 2016. What happens next? event prediction using a compositional neural network model. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1).

Rujun Han, Qiang Ning, and Nanyun Peng. 2019. Joint event and temporal relation extraction with shared representations and structured prediction. 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 434–444, Hong Kong, China. Association for Computational Linguistics.

Lifu Huang, Heng Ji, Kyunghyun Cho, Ido Dagan, Sebastian Riedel, and Clare Voss. 2018. Zero-shot transfer learning for event extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2160–2170, Melbourne, Australia. Association for Computational Linguistics.

Bram Jans, Steven Bethard, Ivan Vulić, and Marie Francine Moens. 2012. Skip n-grams and ranking functions for predicting script events. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 336–344, Avignon, France. Association for Computational Linguistics.

Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. 2012. Joint entity and event coreference resolution across documents. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 489–500, Jeju Island, Korea. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Manling Li, Sha Li, Zhenhailong Wang, Lifu Huang, Kyunghyun Cho, Heng Ji, Jiawei Han, and Clare Voss. 2021. The future is not one-dimensional: Complex event schema induction by graph modeling for event prediction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5203–5215, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Manling Li, Qi Zeng, Ying Lin, Kyunghyun Cho, Heng Ji, Jonathan May, Nathanael Chambers, and Clare Voss. 2020. Connecting the dots: Event graph schema induction with path language modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 684–695, Online. Association for Computational Linguistics.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, Online. Association for Computational Linguistics.

Wei Lu and Dan Roth. 2012. Automatic event extraction with structured preference modeling. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 835–844, Jeju Island, Korea. Association for Computational Linguistics.

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004. Annotating noun argument structure for NomBank. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Ashutosh Modi. 2016. Event embeddings for semantic script modeling. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 75–83, Berlin, Germany. Association for Computational Linguistics.

Kiem-Hieu Nguyen, Xavier Tannier, Olivier Ferret, and Romaric Besançon. 2015. Generative event schema induction with entity disambiguation. 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 188–197, Beijing, China. Association for Computational Linguistics.

Qiang Ning, Zhili Feng, Hao Wu, and Dan Roth. 2018. Joint reasoning for temporal and causal relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2278–2288, Melbourne, Australia. Association for Computational Linguistics.

Qiang Ning, Sanjay Subramanian, and Dan Roth. 2019. An improved neural baseline for temporal relation extraction. 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 6203–6209, Hong Kong, China. Association for Computational Linguistics.

Haoruo Peng and Dan Roth. 2016. Two discourse driven language models for semantics. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 290–300, Berlin, Germany. Association for Computational Linguistics.

Haoruo Peng, Yangqiu Song, and Dan Roth. 2016. Event detection and co-reference with minimal supervision. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 392–402, Austin, Texas. Association for Computational Linguistics.

Karl Pichotta and Raymond Mooney. 2014. Statistical script learning with multi-argument events. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 220–229, Gothenburg, Sweden. Association for Computational Linguistics.

Karl Pichotta and Raymond Mooney. 2016. Learning statistical scripts with lstm recurrent neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1).

Daniela Pohl, Abdelhamid Bouchachia, and Hermann Hellwagner. 2012. Automatic sub-event detection in emergency management using social media. In *Proceedings of the 21st international conference on world wide web*, pages 683–686.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.

Vasin Punyakanok, Dan Roth, and Wen-tau Yih. 2008. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2):257–287.

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. 2019. 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.

Rachel Rudinger, Pushpendre Rastogi, Francis Ferraro, and Benjamin Van Durme. 2015. Script induction as language modeling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1681–1686, Lisbon, Portugal. Association for Computational Linguistics.

Lei Sha, Sujian Li, Baobao Chang, and Zhifang Sui. 2016. Joint learning templates and slots for event schema induction. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 428–434, San Diego, California. Association for Computational Linguistics.

Stephanie Strassel and Jennifer Tracey. 2016. LORELEI language packs: Data, tools, and resources for technology development in low resource languages. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3273–3280, Portorož, Slovenia. European Language Resources Association (ELRA).

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. 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 5784–5789, Hong Kong, China. Association for Computational Linguistics.

Haoyu Wang, Muhao Chen, Hongming Zhang, and Dan Roth. 2020. Joint constrained learning for event-event relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 696–706, Online. Association for Computational Linguistics.

Haoyu Wang, Hongming Zhang, Muhao Chen, and Dan Roth. 2021a. Learning constraints and descriptive segmentation for subevent detection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5216–5226, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zirui Wang, Adams Wei Yu, Orhan Firat, and Yuan Cao. 2021b. Towards zero-label language learning. *arXiv preprint arXiv:2109.09193*.

Noah Weber, Niranjan Balasubramanian, and Nathanael Chambers. 2018. Event representations with tensor-based compositions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Noah Weber, Rachel Rudinger, and Benjamin Van Durme. 2020. Causal inference of script knowledge. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7583–7596, Online. Association for Computational Linguistics.

Haoyang Wen, Ying Lin, Tuan Lai, Xiaoman Pan, Sha Li, Xudong Lin, Ben Zhou, Manling Li, Haoyu Wang, Hongming Zhang, Xiaodong Yu, Alexander Dong, Zhenhailong Wang, Yi Fung, Piyush Mishra, Qing Lyu, Dídac Surís, Brian Chen, Susan Windisch Brown, Martha Palmer, Chris Callison-Burch, Carl Vondrick, Jiawei Han, Dan Roth, Shih-Fu Chang, and Heng Ji. 2021. RESIN: A dockerized schema-guided cross-document cross-lingual cross-media information extraction and event tracking system. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*, pages 133–143, Online. Association for Computational Linguistics.

Xiaodong Yu, Wenpeng Yin, and Dan Roth. 2020. Paired representation learning for event and entity coreference. *arXiv preprint arXiv:2010.12808*.

Quan Yuan, Xiang Ren, Wenqi He, Chao Zhang, Xinhe Geng, Lifu Huang, Heng Ji, Chin-Yew Lin, and Jiawei Han. 2018. Open-schema event profiling for massive news corpora. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 587–596.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, volume 33, pages 17283–17297. Curran Associates, Inc.

Hongming Zhang, Muhao Chen, Haoyu Wang, Yangqiu Song, and Dan Roth. 2020a. Analogous process structure induction for sub-event sequence prediction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1541–1550, Online. Association for Computational Linguistics.

Hongming Zhang, Haoyu Wang, and Dan Roth. 2021a. Zero-shot Label-aware Event Trigger and Argument Classification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1331–1340, Online. Association for Computational Linguistics.

Li Zhang, Qing Lyu, and Chris Callison-Burch. 2020b. Reasoning about goals, steps, and temporal ordering with WikiHow. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4630–4639, Online. Association for Computational Linguistics.

Yi Zhang, Sujay Kumar Jauhar, Julia Kiseleva, Ryen White, and Dan Roth. 2021b. Learning to decompose and organize complex tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2726–2735, Online. Association for Computational Linguistics.

Ben Zhou, Qiang Ning, Daniel Khashabi, and Dan Roth. 2020. Temporal common sense acquisition with minimal supervision. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7579–7589, Online. Association for Computational Linguistics.

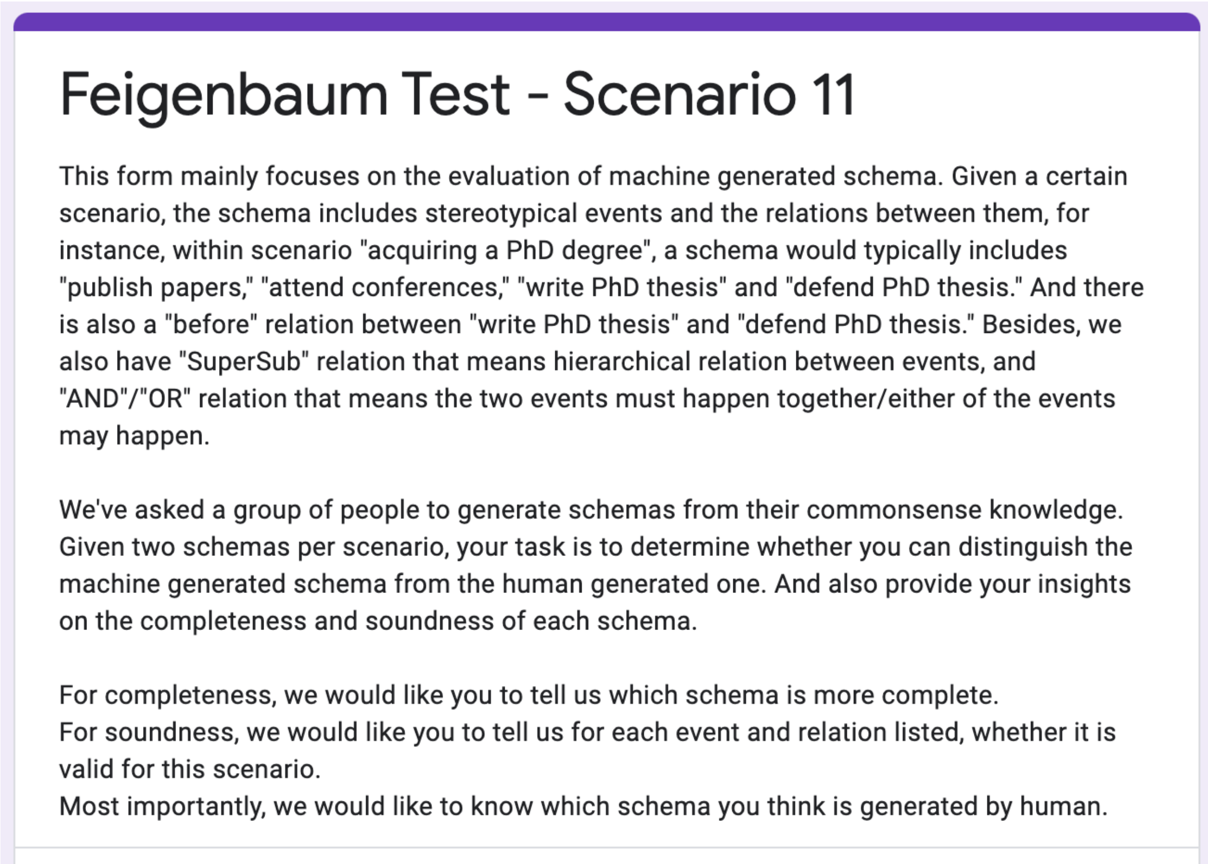
Ben Zhou, Kyle Richardson, Qiang Ning, Tushar Khot, Ashish Sabharwal, and Dan Roth. 2021. Temporal reasoning on implicit events from distant supervision. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1361–1371, Online. Association for Computational Linguistics.

A Feigenbaum Test Details

The experiment took place online through filling a Google Form and involved 11 annotators. Each annotator got 3-4 scenarios to annotate. The instructions for the survey appear in Figure 5. An example scenario and the questions of the survey are presented in Figures 6, 7, 8, and 9.

B Feigenbaum Test Results

In this section we present all the results from the experiments on the dataset Schema-11. Table 6 shows the distribution of answers for the question “which schema is more complete?” (same as depicted in Table 5), Table 7 presents the distribution of answers for the question "which schema was generated by a human?" together with the correct answer written in the bottom row, and Table 8 presents the percentage of invalid events and relations determined by the majority vote of the annotators in the automatic schema and the human schema.



Feigenbaum Test - Scenario 11

This form mainly focuses on the evaluation of machine generated schema. Given a certain scenario, the schema includes stereotypical events and the relations between them, for instance, within scenario "acquiring a PhD degree", a schema would typically includes "publish papers," "attend conferences," "write PhD thesis" and "defend PhD thesis." And there is also a "before" relation between "write PhD thesis" and "defend PhD thesis." Besides, we also have "SuperSub" relation that means hierarchical relation between events, and "AND"/"OR" relation that means the two events must happen together/either of the events may happen.

We've asked a group of people to generate schemas from their commonsense knowledge. Given two schemas per scenario, your task is to determine whether you can distinguish the machine generated schema from the human generated one. And also provide your insights on the completeness and soundness of each schema.

For completeness, we would like you to tell us which schema is more complete.
For soundness, we would like you to tell us for each event and relation listed, whether it is valid for this scenario.
Most importantly, we would like to know which schema you think is generated by human.

Figure 5: Instructions for the Feigenbaum test.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11
Human	4	0	1	1	1	2	1	1	0	3	1
Automatic	2	3	4	2	1	1	1	1	4	0	1

Table 6: Completeness results. The table presents the number of votes that were recorded for which schema is more complete - the human generated schema or the automatically generated schema. The majority vote is highlighted in yellow.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11
A	1	1	3	0	0	2	0	2	2	1	1
B	5	2	2	3	2	1	2	0	2	2	1
Correct Answer	B	B	B	B	B	A	B	A	A	B	B

Table 7: Feigenbaum test results. The annotators guesses which schema (A or B) was generated by humans. The number of votes for each option appear along with the correct answer in the bottom row. The correct majority guesses are marked with green and incorrect with red.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11
Invalid Events (Auto.)	0	0	0	0	0	8.33	0	7.69	0	14.28	0
Invalid Relations (Auto.)	46.15	16.66	25	25	0	23.52	0.4	11.76	12.5	22.22	46.15
Invalid Events (Human)	0	0	14.28	14.28	0	0	0	0	0	0	0
Invalid Relations (Human)	7.69	50	15.38	15.38	0	6.25	0	11.11	0	10	7.69

Table 8: Invalidity results. The table presents the percentage of invalid events and relations determined by the human annotators for each schema and scenario.

Scenario 11: Terrorism Attack (A)

Events:
1. event: kill, arg0: {PER, ORG, VEH, WEA}, arg1: PER
2. event: injure, arg0: {PER, ORG, VEH, WEA}, arg1: PER
3. event: detonate, arg0: PER, arg1: WEA
4. event: come, arg1: attack
5. event: open, arg0: {PER, ORG}, arg1: fire
6. event: wound, arg0: {PER, ORG, VEH, WEA}, arg1: PER
7. event: strike, arg0: {PER, ORG, WEA}
8. event: claim, arg0: ORG, arg1: responsibility
9. event: leave, arg0: {PER, VEH}
10. event: attack, arg0: {PER, ORG}
11. event: choose, arg0: {PER, ORG}, arg1: {PER, ORG, GPE}
12. event: select, arg0: {PER, ORG}, arg1: method
13. event: acquire, arg0: {PER, ORG}, arg1: WEA
14. event: carry out, arg0: PER

Relations:
1. before: 3->8
2. before: 3->5
3. before: 1->4
4. before: 1->9
5. before: 2->4
6. before: 2->9
7. before: 6->4
8. before: 6->9
9. before: 11->12->13->14->10
10. OR: 8,5
11. OR: 4,9
12. AND: 1,2,6
13. supersub: 10->7->1,2

Figure 6: An example schema in the topic of Terrorism Attack. This schema was generated automatically (information that was unknown to the annotators).

Scenario 11: Terrorism Attack (B)

Events:
1. event: find, arg0: PER, arg1: ORG, arg-loc:LOC
2. event: emerge, arg0: ORG, arg1: ORG
3. event: fade, arg0: ORG, arg-tmp: TMP
4. event: reemerge, arg0: ORG, arg-tmp: TMP, arg-loc: LOC
5. event: lead, arg0: ORG, arg1: losses
6. event: lost, arg0: ORG, arg1: LOC
7. event: declare, arg0: GPE, arg1: ORG
8. event: kill, arg0: GPE, arg1: PER
9. event: plan, arg0: PER
10. event: executes, arg0: PER
11. event: injures, arg0: the attack, arg1: PER
12. event: kills, arg0: the attack, arg1: PER
13. event: damages, arg0: the attack, arg1: infrastructure
14. event: calls, arg0: PER, arg1: PER
15. event: arrive, arg0: PER
16. event: treat, arg0: PER, arg1: PER
17. event: take, arg0: PER, arg1: PER
18. event: reports, arg0: PER
19. event: claims, arg0: the group, arg1: responsibility

Relations:
1. before: 9->10
2. before: 10->11
3. before: 10->12
4. before: 10->13
5. before: 10->14
6. before: 14->15->16->17
7. before: 10->18
8. before: 10->19
9. before: 1->3
10. before: 3->4
11. AND: 1->2
12. cause: 5->6
13. cause: 8->7

Figure 7: An example schema in the topic of Terrorism Attack. This schema was generated by a human (information that was unknown to the annotators).

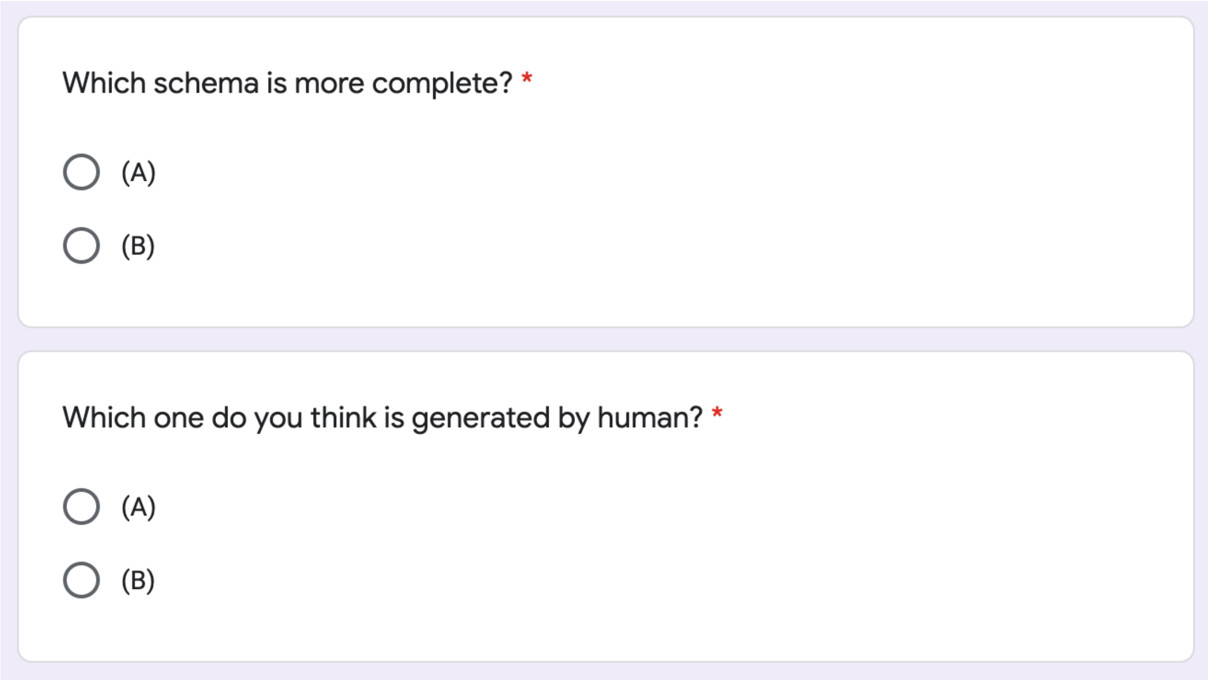


Figure 8: Questions that were asked about the completeness of the schemas and the generator of the schema.

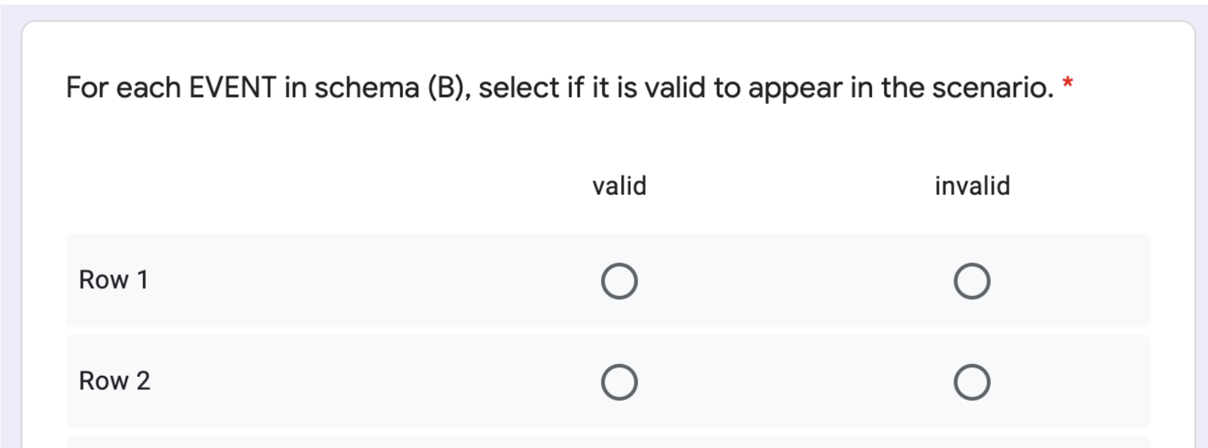


Figure 9: Questions about the validity of the events appearing in one of the schemas. This question was asked on both schemas and on the relations appearing in the schemas too.