Zero-Shot On-the-Fly Event Schema Induction

Rotem Dror∗, Haoyu Wang∗, and Dan Roth
Department of Computer and Information Science
University of Pennsylvania
{rtmdrr,why16gzl,danroth}@seas.upenn.edu

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 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 manual data collection, 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 and even reaching the best score in the prediction task.

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). For instance in Fig. 1, the graph 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, coreferential, causal or hierarchical relations (Do et al., 2012; Lee et al., 2012; Glavaš et al., 2014; Ning et al., 2018; Wang et al., 2020; Zhang et al., 2020a; Trong et al., 2022).

Previous works on event schema induction relied on the information extracted from manually 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 then construct a schema graph by merging the top $k$ ranked paths. Their approach, however, requires access to many documents on each topic of interest, which can be extremely laborious and time consuming to obtain.

In this paper, our goal 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 manually collecting many documents on the topic of the schema, we utilize pre-trained text generators, e.g., GPT-3 (Brown et al., 2020), to obtain documents of diverse genres 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 framework zero-shot since we do not rely on any human-collected articles or example schemas.

---

Footnotes:

∗Indicating equal contribution.

https://cogcomp.seas.upenn.edu/page/publication_view/995
In addition to 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 using state-of-the-art models proposed by Zhou et al. (2021); Wang et al. (2021). Sending the whole text as input instead of two sentences at each time, our 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 schema graphs 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. We also demonstrate that documents generated by GPT-3 are informative and useful for the task of schema induction. 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 essential information about relations between events and arguments extracted from the text, in addition to logical and hierarchical relations not studied previously in schema induction.

Script Learning: Early script learning work 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 deal with zero-shot scenarios that do not require training data.

**Pre-Trained Generation Models:** Large-scale pre-trained text generation models such as GPT-2 (Radford et al., 2019), GPT-3 (Brown et al., 2020), BART (Lewis et al., 2020), T5 (Raffel et al., 2020), i.a. have been used in many NLP 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, but to produce the most likely sequence of words to proceed a certain prompt, similar to language models. In our work, we use these large pre-trained LMs as text generators. The generated documents on a particular topic are leveraged as a corpus for extracting the schema of the given topic. 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 the topic of “planning a wedding,” we assume most documents will include “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.

Our analysis shows that the generated stories contain a higher percentage of relevant tokens than real news articles that are used for schema induction. To demonstrate this phenomenon, we compare manually collected documents with those that are automatically generated using GPT-3 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 — a fine-grained ontology for schema learning, with 24 entity types, 67 event types, and 85 argument roles.

We calculate 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 with the Davinci-instruct-beta-v3 model and we also experiment with the Hugging Face GPT-2 API. 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 is then used in the following prompt: “Write a news story titled headline.” 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:** We use the prompt: “Describe how to topic.” to generate wikiHow-like instruction

<table>
<thead>
<tr>
<th></th>
<th>Generated Text</th>
<th>Real Text</th>
</tr>
</thead>
<tbody>
<tr>
<td># events / # tokens</td>
<td>12.52%</td>
<td>6.31%</td>
</tr>
<tr>
<td># arguments / # tokens</td>
<td>5.45%</td>
<td>3.01%</td>
</tr>
</tbody>
</table>

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

---

1The full ontology definition can be accessed at this link: [https://bit.ly/3mIWJoN](https://bit.ly/3mIWJoN).

2https://openai.com/blog/openai-api/.

3https://huggingface.co/gpt2
Prompt: What are the steps involved in holding a summit meeting?

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 generated text of steps involved in the event of “holding a summit meeting”.

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.

(b) Example generation of How-To article for the event of “3D printing”.

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

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

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 what events and arguments are salient in advance because we allow generating a schema on any topic. Instead, we employ a statistical approach by extracting all the information and later filter it down to include just frequent items. Here are the steps involved in our information extraction pipeline:

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

Named Entity Recognition (NER): We employ the SOTA NER model (Guo and Roth, 2021) to extract and map entities (potential arguments of events) into entity types defined in the CoNLL 2002 dataset (Tjong Kim Sang, 2002) and the LORELEI project (Strassel and Tracey, 2016).

5 The “1.” in the prompt is for the LM to automatically complete the steps.
Constituency Parsing: The arguments extracted by SRL can be clauses and long phrasal nouns, hence we employ the AllenNLP constituency parsing model for argument head word extraction.

Coreference Resolution: We use the SOTA model (Yu et al., 2022) for event and entity coreference resolution to identify within-document coreferential relations.

Temporal Relation Extraction: We first try to use SOTA models (Ning et al., 2019; Zhou et al., 2021) to predict the temporal relations between all possible pairs of extracted events but since the SOTA models accept two sentences containing events as input, the inference time for an n-event document is \( O(n^2) \), making the schema induction process several hours long.

One-Pass Model: We develop a One-Pass model that takes the document as input and uses the contextual representation of events to predict relations between them. A document \( D \) is represented as a sequence of tokens \( D = [t_1, \ldots, e_1, \ldots, e_2, \ldots, t_n] \) where some of the tokens belong to the set of annotated event triggers, i.e., \( E_D = \{e_1, e_2, \ldots, e_k\} \), whereas the rest are other lexemes. We employ the transformer-based language model Big Bird (Zaheer et al., 2020) to encode a whole document and obtain the contextualized representations for all the event mentions. These representations are fed into a multi-layer perceptron in a pairwise fashion and the cross-entropy loss for each pair is calculated and accumulated for a batch of documents. As shown in Tab. 2, the inference time is shortened 63-186 times on average, while the performance of the One-Pass model is comparable to SOTA models.

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., 2021) also impairs the efficiency of our schema induction system. Thus we use the same One-Pass methodology to extract hierarchical relations. We observe that the inference time is greatly shortened, and the One-Pass model achieves comparable results to previous models while taking up less GPU memory (see Tab. 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 any other source of information that could identify what is salient). We describe the process of building a schema in the following section.

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) 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. We only consider relations and events that appeared in more than one document.

Construct timelines: We construct the longest timelines from the list of temporal relations. This

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Model</th>
<th>(F_1) score</th>
<th>Speed</th>
<th>GPU Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>HiEve</td>
<td>Zhou et al. (2020)</td>
<td>0.489</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Wang et al. (2021)</td>
<td>0.522</td>
<td>41.68s</td>
<td>4515MiB</td>
</tr>
<tr>
<td></td>
<td>One-Pass model</td>
<td>0.472</td>
<td>0.65s</td>
<td>2941MiB</td>
</tr>
<tr>
<td>MATRES</td>
<td>Ning et al. (2019)</td>
<td>0.767</td>
<td>30.12s</td>
<td>4185MiB</td>
</tr>
<tr>
<td></td>
<td>Zhou et al. (2021)</td>
<td>0.821</td>
<td>89.36s</td>
<td>9311MiB</td>
</tr>
<tr>
<td></td>
<td>One-Pass model</td>
<td>0.768</td>
<td>0.48s</td>
<td>2419MiB</td>
</tr>
</tbody>
</table>
both events occurring at the same time or there is no significance to the order between them. We use a logical OR to mark events that can occur simultaneously but not necessarily. See Fig. 3 for example of both logical relations.

The final output is a schema graph that contains all the events, arguments, and temporal, hierarchical and logical relations between the events. It is noteworthy that our proposed schema generation model can be easily 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, we also collected human generated schemas for 11 newsworthy sce-
The schemas were generated by four human experts who were instructed to write a schema on each topic based on their commonsense knowledge that includes a list of event triggers, event arguments and their NER types,12 and relations.13

6.2 Evaluation Protocols

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 predicted by the model, assuming the human annotations are gold labels (coverage), and to calculate the accuracy in predicting the final outcome of a scenario (prediction). For instance, the accuracy of predicting the last event type of the LDC schemas is reported in Li et al. (2021). Here we present the results of predicting the last events using event triggers instead of event types since our schemas do not use an ontology of event types.

Feigenbaum Test We show human testers two schemas on each topic in the Schema-11 dataset (see example in Appx. §A). One schema is automatically generated by our model, and the other is randomly sampled from the Schema-11 corpus.14 Then, we ask the testers to determine which events and relations are valid to appear in the schema (soundness), and answer 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) the matching of event triggers, and (b) the matching of event triggers and synonyms of the events in the gold schemas (synonym coverage).15 We believe that synonym coverage is a better evaluation metric 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. In the calculation of coverage of relations we only take into account relations \((a, b)\) where both events, \(a\) and \(b\), appear in the generated schema.

From the results in Table 3, we observe that despite the difficulty of exact matching, our model with GPT-3 covers 23.73% of the gold events, showing that generated texts are useful. If we use synonym coverage as our metric, we achieve a promising coverage of 37.84% while the SOTA supervised event graph model (Li et al., 2021) covers 54.84% using limited event types. In addition, we calculated an average number of 26.19 additional events that appeared in the generated schemas and not in the LDC schema, pointing to the potential of using generated documents for expanding existing schemas. With the high quality event representations obtained from the One-Pass model and the proposed logical relation induction algorithm, our method can successfully cover a high percentage of multiple types of relations.

Prediction In the prediction task, our schemas are able reach SOTA performance and predict the final outcome in 63.1% of the cases for the LDC schemas (see Tab. 4). This 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.

Schema-11 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-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, meaning that the average

---


12 The annotators are familiar with SRL annotations (e.g., ARG0, ARG1, etc.) and NER types (e.g., PER, ORG, etc.). See additional details in App. C.

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

14 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.

15 Implemented using the NLTK WordNet Python package.
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 present the same metric for relations of different types.

<table>
<thead>
<tr>
<th>Event Match</th>
<th>GPT2 Coverage</th>
<th>Coverage (Syn)</th>
<th>GPT3 Coverage</th>
<th>Coverage (Syn)</th>
<th>Li et al. (2021) Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event Match</td>
<td>14.88</td>
<td>29.55</td>
<td>23.73</td>
<td>37.84</td>
<td>54.84</td>
</tr>
<tr>
<td>Temporal Relations</td>
<td>10.80</td>
<td>33.31</td>
<td>31.07</td>
<td>49.99</td>
<td></td>
</tr>
<tr>
<td>Hierarchical Relations</td>
<td>33.33</td>
<td>33.33</td>
<td>11.11</td>
<td>13.88</td>
<td></td>
</tr>
<tr>
<td>Logical Relations</td>
<td>4.16</td>
<td>24.99</td>
<td>43.76</td>
<td>49.81</td>
<td></td>
</tr>
</tbody>
</table>

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.

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event Language Model</td>
<td>49.7</td>
</tr>
<tr>
<td>Sequential Pattern Mining</td>
<td>47.8</td>
</tr>
<tr>
<td>Human Schema</td>
<td>20.5</td>
</tr>
<tr>
<td>Event Graph Model</td>
<td>52.0</td>
</tr>
<tr>
<td>Zero-Shot Schema GPT2</td>
<td>25.0</td>
</tr>
<tr>
<td>Zero-Shot Schema Synonym GPT2</td>
<td>45.2</td>
</tr>
<tr>
<td>Zero-Shot Schema GPT3</td>
<td>35.7</td>
</tr>
<tr>
<td>Zero-Shot Schema Synonym GPT3</td>
<td>63.1</td>
</tr>
</tbody>
</table>

percentage of legitimate events is even higher in machine-generated schemas. For the completeness results presented in Tab. 5, 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 a tie in the remaining 4 cases. 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, eight out of eleven times they correctly identify the human-generated schema, one incorrectly, and two 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 comprehensive. The full results from the Feigenbaum test are shown in Appx. §B.

**Wizard of Oz Experiment** There seems to be a discrepancy between the low event coverage results and the quality of generated texts that were presented in Section §3. We, therefore, conducted another experiment to identify if the problem stems from the quality of the generated documents. In this experiment, one of the authors sampled 10 complex event names from the LDC dataset and generated, using GPT-3 text davinci-002 model, 3 texts for each scenario using the prompting methods presented in Section §3. Then, the author manually extracted all relevant events and relations from each document and built a schema based solely on those events and relations.

This experiment, in which the author pretends to be the IE and schema generator models, aims to demonstrate that if we had perfect IE and schema induction systems, then the texts generated by GPT-3 would be sufficient and even superior to other corpora collected manually. The macro-average coverage of events in this experiment is 68% and the micro-average is 74%. Furthermore, GPT-3 texts generated schemas that included, on average, 6.5 additional events not mentioned in LDC schemas but relevant to the scenario at hand. As a result, we conclude that the generated texts from GPT3 contain much of the necessary information to generate schemas in a variety of topics, and can even be used to enrich existing schemas generated by other models or humans. Two example scenarios and more details appear in Appx. §D.

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 identifying temporal and hierarchical relations that achieve comparable performances with SOTA models but require far less inference time and memory space. To evaluate our framework, we conduct experiments on a
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. We observe a high percentage of valid events and relations generated for the Schema-11 dataset and the testers endorsed the completeness of our machine-generated schemas.

8 Acknowledgments

The authors would like to thank the anonymous ACL ARR reviewers for their insightful feedback on our work. This work was supported by Contract FA8750-19-2-1004 with the US Defense Advanced Research Projects Agency (DARPA). Approved for Public Release, Distribution Unlimited. This research is also based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA Contract No. 2019-19051600006 under the BETTER Program. 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 ODNI, IARPA, the Department of Defense, 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 Limitations

The paper presents a method for building an event schema without manually collecting documents from sources such as news articles or Wikipedia. In order to generate diverse and informative documents on any topic, we rely on large pre-trained language models. Our model, which uses GPT-3, generates schemas that are comparable to those generated by manually searching the web for documents, however, when we use inferior LMs such as GPT-2, we see a decline in performance (see Tab. 3 and Tab. 4).

Our assumption is that the quality of the generated schema depends on the quality of the LM and the level of coverage of the selected topic in the LM training data. If, for instance, we were to ask our model to generate a schema for a unique topic such as "conducting an archaeological dig in an unexplored territory" we doubt that the results would be as useful to an archaeologist as if they were looking for information themselves due to the low coverage of this topic in the corpus the model was trained on.

Despite our model’s reliance on pre-trained LMs, we believe the generated schemas can always serve as a good basis for further development.

10 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


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.


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, Didac 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.


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.


Ben Zhou, Qiang Ning, Daniel Khashabi, and Dan Roth. 2020. Temporal common sense acquisition with minimal supervision. In Proceedings of the
A Feigenbaum Test Details

The experiment took place online through filling a Google Form and involved 11 volunteer 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 Fig. 6, Fig. 7, Fig. 8, and Fig. 9.

B Feigenbaum Test Results

In this section we present all the results from the experiments on the dataset Schema-11. Tab. 6 shows the distribution of answers for the question “which schema is more complete?” (same as depicted in Tab. 5). Tab. 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 Tab. 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.

C Details on Human Schema Curation

Here are the instructions that were given to the annotators that generated the human schemas for the Schema-11 dataset. All the annotators are graduate students that previously were involved in research projects that include schema induction, SRL, NER or other relevant tasks:

We are developing a system that generates schemas automatically given a topic. We want to compare our automatically-generated schema to schemas derived by people using their commonsense (without relying on texts). To do this, we need expert human annotators and would appreciate your assistance.

A schema is defined as a list of events with their argument types, and the relationships between the events. For example, here is a schema I wrote that describes the event of “armed robbery”:

List of events and arguments:

• intend: arg0 - perpetrator [PER], arg1 - commit a felony
• acquire: arg0 - perpetrator [PER], arg1 - weapon [WEA]
• arrive: arg0 - perpetrator [PER], arg-loc - crime scene [LOC]
• assault: arg0 - perpetrator [PER], arg1 - [PER]
• threaten: arg0 - perpetrator [PER], arg1 - [PER]
• get: arg0 - perpetrator [PER], arg1 - money or goods
• injure: arg0 - perpetrator [PER], arg1 - [PER]
• kill: arg0 - perpetrator [PER], arg1 - [PER]
• flee: arg0 - perpetrator [PER], arg-loc - crime scene [LOC]
• call: arg0 - [PER], arg1 - police [ORG]
• chase: arg0 - police [ORG], arg1 - perpetrator[PER]
• catch: arg0 - police [ORG], arg1 - perpetrator[PER]
• manage to escape: arg0 - perpetrator[PER]

Temporal and logical relations (in the form of a timeline):

• a perpetrator (PER) intent to commit a felony ->
• the perpetrator (PER) acquires weapon (WEA) ->
• the perpetrator (PER) arrives at the scene (LOC) ->
• perpetrator (PER) assault victim (PER) with weapon (WEA) at the scene (LOC) OR perpetrator (PER) threatens a person (PER) with the weapon (WEA) at the scene (LOC) ->
• perpetrator (PER) gets money or goods from the person (PER) OR victim injured OR victim killed ->
• perpetrator flees the scene of the crime (LOC) AND someone (PER) calls the police (ORG) ->
• the police (ORG) are chasing the criminal (PER) ->
• the police (ORG) catches the perpetrator (PER) XOR the criminal (PER) manages to escape.

The complex events we are interested in are the following: (1) Disease Outbreak (2) IED Bombing (3) Civil Unrest (4) International Invasion (5) Disaster and Rescue (6) Terrorism Attacks (7) Election (8) Kidnapping (9) Business Change (10) Mass Shooting.
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.

<table>
<thead>
<tr>
<th></th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
<th>S6</th>
<th>S7</th>
<th>S8</th>
<th>S9</th>
<th>S10</th>
<th>S11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Automatic</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th></th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
<th>S6</th>
<th>S7</th>
<th>S8</th>
<th>S9</th>
<th>S10</th>
<th>S11</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Correct Answer</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>A</td>
<td>B</td>
<td>A</td>
<td>B</td>
<td>A</td>
<td>B</td>
<td>B</td>
</tr>
</tbody>
</table>

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.

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

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-temp: TMP
4. event: reemerge, arg0: ORG, arg-temp: 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.

Which schema is more complete? *

- (A)
- (B)

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.

For each EVENT in schema (B), select if it is valid to appear in the scenario. *

<table>
<thead>
<tr>
<th></th>
<th>valid</th>
<th>invalid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row 1</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Row 2</td>
<td>☐</td>
<td>☐</td>
</tr>
</tbody>
</table>

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.
D Wizard of Oz Experiment

Two examples of the Wizard of Oz experiment for the events of invasion and remodel a facility. The examples show the three generated texts using the three prompting methods, the schema created by the author, and the LDC schema together with a matching between the events in the two schemas. The entire list of sampled topics is: print a 3D object, cyber attack, make a website, religious conversion, invasion, shipping and mailing, remodel a facility, obtain and provide medical treatment, build a drone, and nonviolent protest march.

D.1 Example 1: Invasion

Prompts and Outputs:

1. Prompt: describe how to perform an invasion.
Output: 1. Choose a target: Identify a target population that you want to invade. This target population can be a group of people, a city, or a country. 2. Prepare your troops: Assemble your troops and supplies in preparation for the invasion. 3. Launch your attack: Attack the target population using whatever means necessary to take control. 4. Consolidate your power: Once you have taken control of the target population, consolidate your power by establishing your authority and eliminating any resistance.

2. Prompt: what are the steps involved in an invasion?
Output: 1. The first step in an invasion is to gather intelligence about the enemy. This includes learning about their strengths, weaknesses, and their territory. 2. The next step is to plan the invasion. This includes deciding what resources will be needed and how they will be used. 3. The third step is to launch the invasion. This is when the actual fighting begins. 4. The fourth step is to consolidate the gains. This is when the invading force secures their position and prepares for the next phase of the operation.

3. Prompt: write a news headline about invasion.
Output: U.S. troops invade Canada in surprise attack.
Prompt: write a news article titled: “U.S. troops invade Canada in surprise attack.”
Output: In a stunning move, the United States troops came across the border early this morning, catching the Canadians by surprise. The Canadians are currently fighting back, but it is unknown how long they will be able to hold out against the might of the United States military. This is a developing story, and more details will be released as they become available.

Schema generated based on the three outputs

1. Choose target.
2. Gather intelligence on target: learn strengths and weaknesses and territory.
3. Plan invasion: decide on resources and how the resources will be used.
5. Launch attack: come across the border.
6. Fight or use whatever means possible.
7. Take control.
9. Prepare for the next phase of operation.

LDC Schema

   (a) Step 1.1: IdentifyTargets[1]: Leadership builds a list of entry points for the invasion and selects one or more as the target(s) of the initial invasion.
   (b) Step 1.2: DevelopPlan[3]: A team develops a master invasion plan.
   (c) Step 1.3: OrderDissemination [not matched]: Planners send out detailed instructions to military forces about the invasion plan.
2. Step 2: DeployTroops[4]: Amass and deploy troops for invasion.
3. Step 3: Invasion[5]: Military force launch the invasion and enter the target territory.
Results: The GPT3 schema covers 6 out of 7 events that are mentioned in the LDC schema. The matching between LDC events and schema events are marked in [] next to the event on the LDC schema. GPT3 schema has 7 additional events: gather intelligence, learn strengths, decide on resources, assemble troops and supplies, come across border, take control, consolidate power, establish authority, eliminate resistance, prepare for next phase.

D.2 Example 2: Remodel A Facility

Prompts and Outputs:

1. Prompt: describe how to remodel a facility.
   Output: There is no one-size-fits-all answer to this question, as the best way to remodel a facility depends on the specific needs and goals of the organization. However, some tips on how to remodel a facility include: 1. Define the scope of the project. 2. Create a budget and timeline. 3. Hire an experienced architect or designer. 4. Work with a reputable contractor. 5. Use high-quality materials. 6. Follow building codes and regulations. 7. Get input from stakeholders. 8. Test the new space before moving in.

2. Prompt: what are the steps involved in remodeling a facility?
   Output: 1. The first step in remodeling a facility is to develop a plan. This plan should include the goals of the remodel, the budget, the timeline, and the scope of work. 2. The next step is to obtain the necessary permits. Depending on the scope of work, this may require a building permit, a zoning permit, and/or a permit from the local health department. 3. Once the permits are obtained, the next step is to select a contractor. The contractor should be licensed and insured, and should have experience with the type of work that needs to be done. 4. The contractor will then begin the work, which may include demolition, construction, painting, and/or installation of new fixtures and appliances. 5. Once the work is completed, the facility will need to be inspected to ensure that it meets all applicable codes and regulations. 6. Finally, the facility can be reopened to the public.

3. Prompt: write a news headline about remodeling a facility.

Output: Facility Remodeling Underway

Prompt: write a news article titled: "Facility Remodeling Underway"

Output: In an effort to better serve the needs of the community, the local library is undergoing a remodel. The project is expected to take several months to complete, but the end result will be a more user-friendly space that meets the needs of today’s patrons. Some of the changes include a new layout for the adult fiction section, additional study rooms, and updated furniture throughout the facility. The children’s area will also be expanded to create a more inviting and interactive space. The library staff is excited about the changes and can’t wait to see the finished product. They are confident that the updates will make the library a more popular destination for both residents and visitors.

Schema generated based on the three outputs

1. Develop a plan: define scope and goals and create budget and timeline.

2. Hire an architect or designer and obtain necessary permits.

3. Select a contractor (preconditions: contractor licensed, insured, have experience).

4. Contractor works: use high quality materials AND follow building code and regulations and demolition or construction or painting or installation.

5. Get input from stakeholders.

6. Inspect facility.

7. Test space.

8. Reopen facility.

9. Facility is user-friendly and meets needs of patrons.

LDC Schema

1. Step 1: Acquisition [not matched]: Acquire facility.

2. Step 2: Planning[1]: Research and plan expected usage, budget, changes, legal issues, dependencies.
3. Step 3: Labor[2,3]: Arrange for skills, or laborers, or both.
   (a) Step 3.1: AcquireSkills [not matched]: Acquire skills or knowledge required for remodeling.
   (b) Step 3.2: HireLaborers[2,3]: Hire skilled person or organization to perform remodeling work.


5. Step 5: Remodel[4]: Facility is remodeled.
   (a) Step 5.1: Demolition[4.2]: Deconstruction or demolition of portions of building and/or equipment installations.
   (b) Step 5.2: DebrisRemoval [not matched]: Hauling away/dumping of debris.
   (c) Step 5.3: Modification[4.2]: Modification, addition, or installation of building or systems/equipment in building.

6. Step 6: Inspection[6,7]: Inspect and/or test new portions of facility and/or new systems of facility for functionality and compliance with laws and regulations.

Results: The GPT3 schema covers 8 out of 11 events that are mentioned in the LDC schema. The matching between LDC events and schema events are marked in [ ] next to the event on the LDC schema. GPT3 schema has 9 additional events: contractor works, follow building code and regulations, preconditions on contractor, painting, installation, construction.