AFRL-RI-RS-TR-2024-043

LEARNING EVENT SCHEMA TEMPORALLY AND TRANSMODALLY (LESTAT)

UNIVERSITY OF SOUTHER CALIFORNIA

MAY 2024

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This report is available to the general public, including foreign nations. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2024-043 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /
ROSANNE PELLI
Work Unit Manager

/ S /
MATTHEW KOCHAN
Technical Advisor
Intelligence Systems Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

1. REPORT DATE
MAY 2024

2. REPORT TYPE
FINAL TECHNICAL REPORT

3. DATES COVERED
START DATE AUGUST 2019
END DATE DECEMBER 2023

4. TITLE AND SUBTITLE
LEARNING EVENT SCHEMA TEMPORALLY AND TRANSMODALLY (LESTAT)

5a. CONTRACT NUMBER
FA8750-19-2-0500

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER
62303E

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
R2XE

6. AUTHOR(S)
Freedman, Marjorie; Hedges, Alex; Kankanampati, Yash; Lichtefeld, Jacob; May, Jon; Weischedel, Ralph

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Southern California Information Sciences Institute
890 Winter Street
Waltham MA 02451

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory/RIEA
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RI

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-RI-RS-TR-2024-043

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report summarizes the University of Southern California (USC) Information Sciences Institute (ISI) Learning Event Schema Temporally and Transmodally (LESTAT) team's effort on DARPA's Knowledge-directed Artificial Intelligence Reasoning Over Schemas (KAIROS) program. The goal was twofold: (1) to develop a human-machine collaborative process for building a library of schemas through leveraging natural language processing (NLP) techniques to augment human background knowledge and (2) to create a repository of schemas. Each script includes a natural language description of the steps in the complex event with links to the Wikidata ontology.

15. SUBJECT TERMS
Schema library of complex events; natural language processing; machine aided script curator; complex event steps; automating temporal ordering; curation of fully ontologized schemas

16. SECURITY CLASSIFICATION OF:
a. REPORT U
b. ABSTRACT U
C. THIS PAGE U

17. LIMITATION OF ABSTRACT
SAR

18. NUMBER OF PAGES
57

19a. NAME OF RESPONSIBLE PERSON
ROSANNE PELLI

19b. PHONE NUMBER *(Include area code)*
N/A

PREVIOUS EDITION IS OBSOLETE.

TABLE OF CONTENTS

List of Figures ii

List of Tables iv

1.0 SUMMARY 1

2.0 INTRODUCTION 2

3.0 METHODS, ASSUMPTIONS, AND PROCEDURES 3

3.1 Learning Schema Sequences by Induction from Text (ISI) 3

3.2 Automatically Generating New Ontology Concepts 4

3.3 Computer Vision for Processing Multi-Media Sources 5

3.4 Algorithms in the MASC Curation Tool 7
3.4.1 Suggesting Events in Text for a Schema 10
3.4.2 Suggest Wikidata qNodes for an Event 13
3.4.3 Assign Arguments to Events 15
3.4.4 Suggest Wikidata qNode for Each Argument 16
3.4.5 Suggest Temporal Ordering of Events 17
3.4.6 Argument Relations 18
3.4.7 Estimates of Importance of an Event 19
3.4.8 Structure and Hierarchy 20
3.4.9 Generating SDF (Schema Data Format) JSON 27

4.0 RESULTS AND DISCUSSION 28

4.1 Error Analysis 28
4.1.1 Errors Found 28
4.1.2 Desired Matches 30
4.1.3 Conclusions from Error Analysis 30

4.2 Schema Completeness and Machine Suggestion of Events Typical of a Complex Event 31
4.2.1 Machine Suggestion of Events while Creating an Initial Schema 31
4.2.2 Assessing the Challenge of "Completeness" of a Schema 31

5.0 CONCLUSIONS 36

6.0 References 37

APPENDIX A – Publications 39

APPENDIX B – System details 40

LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS 49

LIST OF FIGURES

Figure 1: Timeline leading to a shift in strategy on learning schemas. 4

Figure 2: An example of adding nodes to an ontology. 5

Figure 3: An illustration of the approach used by ASU 6

Figure 4: Two examples of the algorithm for video processing. 7

Figure 5: Initiating creation of a schema for a complex event 8

Figure 6: A screenshot of the user interface for reviewing and selecting among GPT suggestions. 12

Figure 7: Examples of Non-Named Concepts with Links in Wikipedia. From articles on Civil Unrest (upper) and Bubonic Plague (lower). 14

Figure 8: A screenshot of 6 alternative Wikidata qNodes for the event “A bus driver picks up passengers.” 14

Figure 9: Review pane in MASC to support review and correction of roles and their arguments for events in a schema 16

Figure 10: Pane for user to review automatically suggested Wikidata qNodes for arguments 17

Figure 11: The Argument Relations modal with the schema's arguments in two columns and the relations available in the center. At the bottom of the modal, the existing relations in the schema can be found, along with a delete function. 19

Figure 12: PIER example broken out by anchored stage on a ‘Disease Outbreak’ topic. The top-Incident stage contains an internal breakdown of PIER, demonstrating hierarchical reuse as a component of PIER layouts. The top-Response stage also demonstrates this hierarchical reuse utilizing only two of the PIER stages. 21

Figure 13: Above Figure 12 PIER’s stage examples converted into a simplified schema. See Table 2 for the alignment of the event or gate ID to sentence examples. MASC visualizes the two separately for readability, we reproduce this layout here. 22

Figure 14: RESIN's SDF Phase 2B Dryrun Output converted to display in MASC. 29

Figure 15: KGTK similarity metrics between the qNodes for writing and editing using various techniques. 33

Figure 16: A subset of batch event recommendations in MASC 34

Figure 17: Schema Index 41

Figure 18: Schema Description for Riot Schema 42

Figure 19: Suggestions of Steps for the Riot Complex Event 43

Figure 20: Suggested QNodes for the text "The Rioters Gather" 43

Figure 21: Adding Temporal Order without Logical Gates.. 44

Figure 22: Adding Temporal Order Incorporating Logical Gates... 44

Figure 23: Adding a Named Logical Gate... 44

Figure 24: Graph Structure that Illustrates a Temporal Relation Between an OR Gate............... 44

Figure 25: Tabular View of Graph Structure with Columns for Temporal Information (Comes Before, Comes After) and the Named Logical Gates (Container)....................................... 45

Figure 26: PIER example broken out by anchored stage on a ‘Disease Outbreak’ topic. The top-Incident stage contains an internal breakdown of PIER, demonstrating hierarchical reuse as a component of PIER layouts. The top-Response stage also demonstrates this hierarchical reuse utilizing only two of the PIER stages. 45

Figure 27: Arguments as Suggested 46

Figure 28: Selected Arguments.. 46

Figure 29: QNode Suggestions for the Selected Arguments 47

Figure 30: Argument Relation Suggestions.. 48

LIST OF TABLES

Table 1: Core Functionality contributed by algorithms. 10

Table 2: Example schema alignment from IDs to qNode label and example sentence. 23

Table 3: ChatGPT examples of self-organized schemas on Chemical Spills and Military Coups 24

Table 4: Examples of assessments of three events predicted by RESIN 30

Table 5: Script comparison by library from Ciosici et al. (2021) and from Zhang et al. (2023).. 32

Table 6: Subset of event steps, shown as qNodes, from the JOB script from Ciosici et al. (2021) and from Zhang et al. (2023). Steps are shown without respect to temporal order. 32

Table 7: Acceptance of event suggestions. Columns for Curators 1 and 2 are unique event suggestions added. 35

1.0 SUMMARY

55Events within a script are connected by (a) temporal order (e.g., negotiating the price of a car happens before buying the car) and by (b) shared arguments (e.g., the person buying a car is also the person who negotiated its price). An extrinsic evaluation of the quality of the repository was managed by the National Institute of Standards and Technology (NIST); and UIUC's RESIN engine predicted likely events given streaming input.

USC ISI designed the Machine-Aided Script Curator (MASC), a script-creation tool, to be used by non- experts in NLP. While approaches to script discovery from text suffer from the incompleteness of text, human attempts to write machine-interpretable scripts suffer from the writer's own tendency to omit steps and, where required, the challenge of mapping to a formal ontology. To assist the script creators, MASC makes the following types of suggestions:

(1) The ontological type for each event;
(2) Arguments for each event and their corresponding role in the event;
(3) A fine-grained ontological type for suggested arguments;
(4) Steps that the curator might have forgotten

USC ISI also explored automating temporal ordering. However, we found that human review for temporal order was critical, in part because for many particular sub-events with a complex event, there was no single prescribed order. In addition to organizing schemas via temporal-relations between sub-events, MASC supports the introduction of hierarchy to allow meaningfully groupings with complex event. With MASC, they explored different approaches to hierarchy, both incorporating human-interpretable names on logical gates that define non-temporally ordered relations between sub-events and allowing the creation of named, reusable sub-schema units that could be imported into larger schemas across the library.

USC ISI's accomplishments include the following:

- Development of a human-machine intelligent interface, MASC, for creation, review, and curation of fully ontologized schemas describing the event structure of a complex event.
- Creation of roughly 200 complex event schemas in the repository. Many were created by programmers; many by master's students from diverse backgrounds.
- Submission of a schema repository for formal evaluation. That repository included 52 schemas that focused on five classes of complex events: terrorist events, disease outbreak, chemical spills, riots, and coups.

2.0 INTRODUCTION

The DARPA KAIROS program is summarized at https://www.darpa.mil/program/knowledge-directed-artificial-intelligence-reasoning-over-schemas. In addition to USC ISI, the team consisted of University of Central Florida's (UCF) Center for Research in Computer Vision, Arizona State University's (ASU) Active Perception Group, and the University of California, Los Angeles (UCLA). KAIROS was divided into two technical areas:

- TA1, techniques to develop a library of schemas of complex events, divided into sub-events that typically occur in the complex events, such that all events and their arguments are ontologized and conform to a JSON specification usable by TA2 systems. Thus, TA1 covers a learning phase.
- TA2, techniques that given an input library of schemas and a set of multi-media documents, hypothesizes (1) what complex event (if any) may be underway and (2) what sub-events might have happened, though not (yet) seen.

The ISI team was funded for TA1 only. For phase 1, the LESTAT schema repository was used by the IBM team and the Johns Hopkins team. For the final phase 2 evaluations, our schema repository was used by the UIUC team with its RESIN prediction engine.

Research interest in schemas for complex events began over 40 years (Schank and Abelson, 1977) with manually crafted "scripts" that encoded procedural knowledge for understanding stories. In the form of checklists, recording procedural knowledge has revolutionized fields like medicine and aviation by encoding expert knowledge and best practices (Degani and Wiener, 1993; Gawande, 2010). In the last few years, researchers have turned their attention to automatic script discovery from text (Chambers, 2013; Weber et al., 2020, 2018). However, exclusively data-driven sub-event discovery methods face the challenge that narrative descriptions often omit common knowledge. Common knowledge might be missing from narrative descriptions due to the quantity and relevance maxims (Grice, 1975).

ISI's goal has been a process for building a library of scripts through human-machine collaboration, leveraging NLP techniques to augment human background knowledge. The resulting demonstration system serves two related purposes. First, it is a knowledge acquisition tool that supports the development of a repository of scripts for use by downstream applications. Second, it is an annotation tool that supports the creation of a library to aid our understanding of how people create scripts.

Such a library can inform and/or benchmark future schema discovery approaches. Each schema includes a natural language description of the steps in the complex event with links to an ontology. Events within a script are connected by (a) temporal order (e.g., negotiating the price of a car happens before buying the car) and (b) by shared argument (e.g., the person buying a car is also the person who negotiated its price). Machine-Aided Script Curator (MASC), the resulting script-creation tool, was designed to be used by non-NLP experts and was used by several students at the University of Southern California.

While approaches to schema discovery suffer from the incompleteness of text, human attempts to write machine-interpretable scripts suffer from the writer's own tendency to omit steps and, where required, the challenge of mapping to a formal ontology. To assist the script creators, MASC makes three types of suggestions: (1) the ontological type for each event; (2) a fine-grained ontological type for suggested arguments; and (3) steps that the curator might have forgotten.

Section **Error! Reference source not found.** describes **Error! Reference source not found.**; Section **Error! Reference source not found.** presents **Error! Reference source not found.**; Section **Error! Reference source not found.** concludes.

Throughout this final report, cited publications appear in the references section (**Error! Reference source not found.**). URLs for downloadable software and/or data, where available, are provided in footnotes.

3.0 METHODS, ASSUMPTIONS, AND PROCEDURES

Our team's approach shifted during the second year in part due to lessons learned in year 1 and in part due to DARPA's direction. Induction of schemas from newswire (Sections 3.1 and 3.2) was replaced by a mixed initiative approach (Section 3.4). During the second year, DARPA urged the LESTAT team to focus exclusively on text input; therefore, the work reported in Section 3.3 was stopped.

3.1 Learning Schema Sequences by Induction from Text (ISI)

One of our assumptions at the outset was that an algorithm could learn the constituent (the subevents) sequences of a complex schema from newswire articles. We tried two approaches.

One approach employed open-source tools to analyze text. The steps were

- Extract event words and temporal relations among them using the software reported in (Han, 2019). ISI's code used a deep neural network to detect event words and temporal relations among them plus integer linear programming to capture global constraints. For instance, in the brief following example: *It's unclear how the bombs **(e1:detonated)**. ... The bombs were **(e2:activated)** remotely.* Detonated *and* activated have been identified as events; the temporal relation between <e2, e1> is **before**.
- Semantic role labeling provides arguments (who, what, when, where) to events, using https://demo.allennlp.org/semantic-role-labeling. The output is *It's unclear how (ARG1:the bombs) **(e1:detonated)**. ... (ARG1:The bombs) were **(e2:activated)** remotely,* thus identifying the object of each of the events e1 and e2.
- Coreference of the arguments was computed by a pre-existing ISI coreference algorithm.

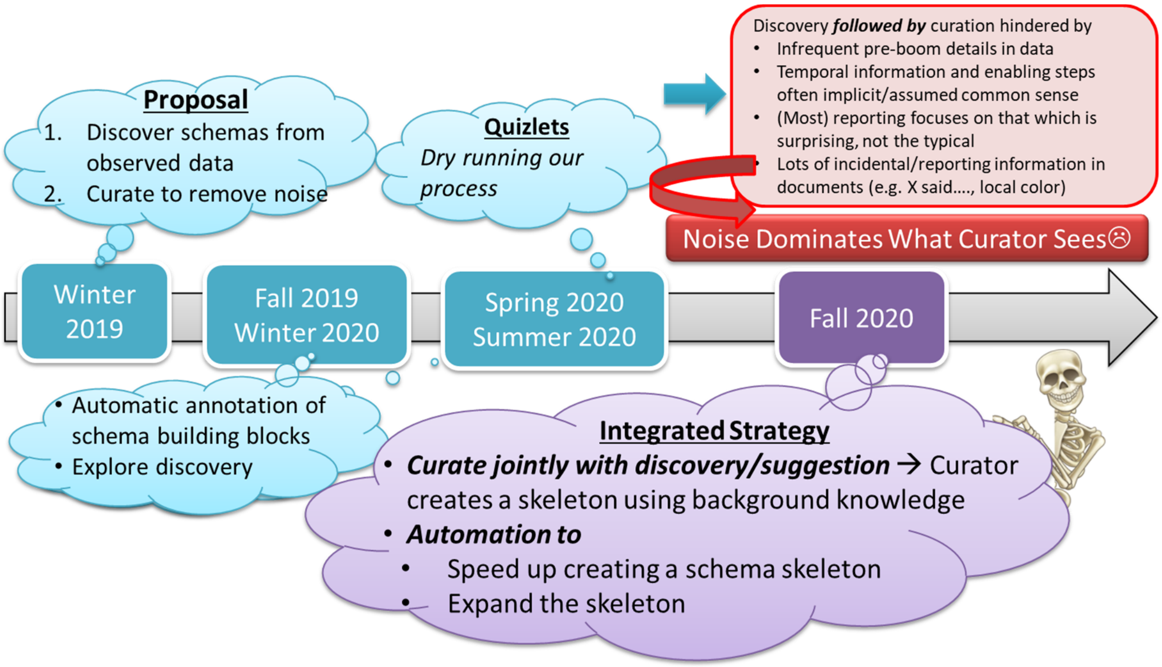
Together these components formed a system to detect temporally ordered sequences of events, with their arguments, and coreference connections across the event arguments. Each of the components exhibited state of the art results. We called the system TEAR, short for Temporal Events and Relations. We made it available to our partner ASU for processing text associated with a video.

Both that approach and the second approach, described next, suffered from two flaws in our assumptions:

(1) In news, many reported events are incidental only, potentially of interest to news readers, but not generic to the complex event. For instance, a few articles about a terrorist bombing attempt mentioned that the co-conspirators sat apart on a flight to their destination, an interesting detail, but not an essential element of planning a terrorist bombing.
(2) In many news articles, an event (e.g., a detonator was activated) is not mentioned because it is inferable but not newsworthy.

A *second approach* used a simpler representation used and tried to cope with item (1) above by modeling salience of events and discourse. Natural language text interleaves a narrative of concrete, salient events with background information, contextualization, opinion, and other elements that are important due to discourse and pragmatics acts but are not part of the principal chain of events being communicated. In (Zhang, et al., 2021), they summarized a technique that extracts a temporally ordered event chains, pre-trains a large language model (LLM) on those extracted chains, and predicts events. While a promising approach, it did not deal with item (2) above, this effort was superseded by direct predictions of event sequences using LLMs in the MASC curation tool.

A timeline for learning schemas by induction from text in LESTAT appears in Figure 1.



Proposal
1. Discover schemas from observed data
2. Curate to remove noise

Quizlets
Dry running our process

Discovery ***followed by*** curation hindered by
- Infrequent pre-boom details in data
- Temporal information and enabling steps often implicit/assumed common sense
- (Most) reporting focuses on that which is surprising, not the typical
- Lots of incidental/reporting information in documents (e.g. X said…., local color)

- Automatic annotation of schema building blocks
- Explore discovery

Integrated Strategy
- ***Curate jointly with discovery/suggestion*** → Curator creates a skeleton using background knowledge
- ***Automation to***
  - Speed up creating a schema skeleton
  - Expand the skeleton

Figure 1: Timeline leading to a shift in strategy on learning schemas.

3.2 Automatically Generating New Ontology Concepts

In both approaches in Section 3.1, the system must still map the words for events (*detonate*, *explode*, etc.) to an ontology. We had proposed automatic ontology induction from text. The input was:

- A tree-structured ***ontology***, where each node is a *concept* and each directed edge indicates a parent-child *relation*
- A set of ***new concepts*** to be added to the ontology.

The output was a candidate concept(s) for each new concept, thus anchoring new events to preexisting ontology nodes. See Figure 2; new concepts for *wear masks*, *medicate*, and *distribute* are automatically added to the ontology. By the end of year 1, the KAIROS program had standardized on a subset of Wikidata as its ontology; thus, this approach was not integrated.

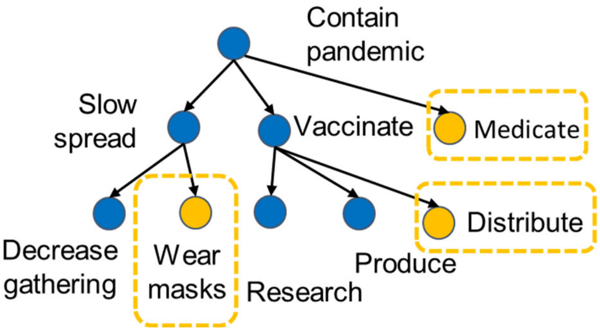


Figure 2: An example of adding nodes to an ontology.

3.3 Computer Vision for Processing Multi-Media Sources

Two approaches were pursued in the first part of LESTAT. *One approach* was developed by UCF. The key was to represent complex activities in videos in the form of graphs of entities, activities, and relations between them; any speech associated with the videos was automatically transcribed into text, which supported assigning semantics to graph nodes. To develop the algorithms in the first year, UCF began with videos from epic-kitchens[1].

We focused on a multimodal approach to learning sub-action interactions over time. We observe that the videos provided with KAIROS data release are highly audio based and this motivated us to take this multi-modal approach for sub-action discovery. We use a combination of attention mechanisms to learn attention between audio, video and text. We proposed a novel criss-cross attention mechanism that allows for better joint embedding between audio and video. We finally reverse map the attention values to the original words and generate a graphical representation that represents the complex activity the node interactions correspond to.

UCF started their modeling process by first extracting audio, visual and text features using pre-trained, frozen networks. For text they used Word2Vec[2]; for audio and video Convolutional Neural Networks. UCF generated a joint embedding using their novel criss-cross multi-modal attention mechanism that uses one modality to attend to the other and then combines the resulting values. This joint embedding is then used in a multi-modal attention mechanism to attend to text extracted from YouTube captions, either automatically generated or manually provided. The output of this attention mechanism is a set of nodes per time segment of the video and their respective attention values. The model learns interactions between nodes and time using separable-depthwise convolutions over the time and node dimensions respectively. The output of this process is a set of maximally attended nodes for each time segment with a final set of features. Using reverse mapping of the pooling operations, UCF maps the final set of nodes to the original words. These models are trained in an unsupervised and supervised way on labeled and unlabeled instructional videos extracted from YouTube.

---

[1] https://epic-kitchens.github.io/2023

[2] https://jalammar.github.io/illustrated-word2vec/

When comparing the unsupervised learned representations of videos to single-modality approaches on a downstream classification task on labeled instructional videos, their approach performs significantly better where the single-modality approach is unable to converge. Furthermore, when training a single-modal approach, the algorithm again performs significantly better with the single-modality approach when unable to converge. In addition to better learned representations of these noisy-instructional videos, the approach generates multi-partite graphical representations of the objects interacting with other objects based on the class of the respective word, for example noun or verb.

A second approach was pursued by ASU. ASU focused on developing a new scheme of structured video events - video descriptions in a modularized format that relates to different event participants, illustrated in Figure 3. A structured video event is anchored on a trigger action verb, followed by a sequence of descriptive arguments. A structured video event is effectively a truncated dependency parse tree. Each represents an aspect of a video, with relations to various arguments such as subject and object. Verbs are the anchoring trigger as inspired by the linguistic theory that a predicate serves as the root syntactic component in dependency parsing. Essentially, by conditioning on a trigger action verb and contextual visual features of a video, the task aims to identify and retrieve the proper participants of the given action out of the video.

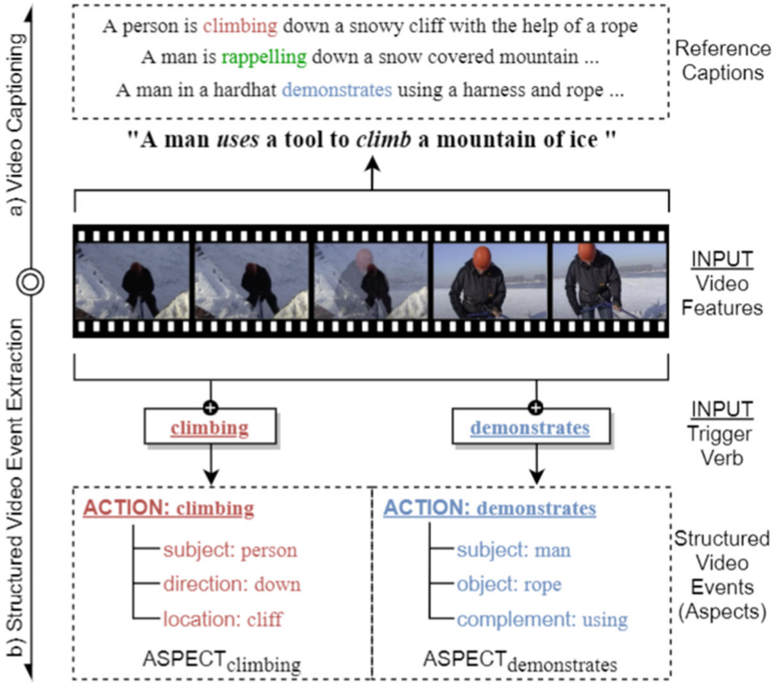
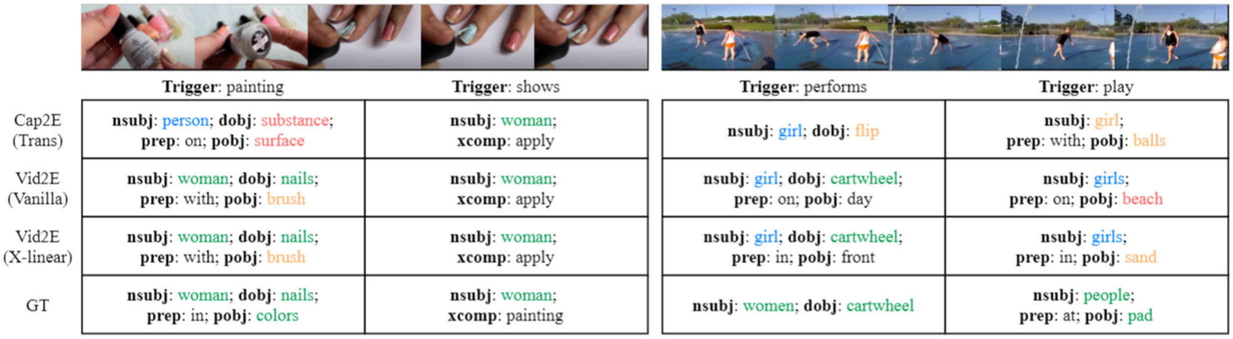

a) Video Captioning
b) Structured Video Event Extraction
A person is climbing down a snowy cliff with the help of a rope
A man is rappelling down a snow covered mountain ...
A man in a hardhat demonstrates using a harness and rope ...
Reference Captions
"A man uses a tool to climb a mountain of ice "
INPUT Video Features
climbing
demonstrates
INPUT Trigger Verb
ACTION: climbing
subject: person
direction: down
location: cliff
ASPECT climbing
ACTION: demonstrates
subject: man
object: rope
complement: using
ASPECT demonstrates
Structured Video Events (Aspects)

Figure 3: An illustration of the approach used by ASU

The task also calls for solid referential ground truth in form of the structured video event to supervise the training process. A major challenge was the lack of existing structured video annotations. ASU also experimented with an automatic video event construction pipeline to generate referential structured video events from ground truth video captions, taking advantage of the state-of-the-art dependency parsing and part-of-speech tagging techniques. ASU was able to extract multiple ground truth video events out of one video caption sentence with this pipeline. And such augmentation enables us to enrich the aspects of a given video, helping us diversify the prediction outputs and thus laying the foundation to the goal of the flexible video description.

ASU developed a video2event prototype model to show the capability of predicting such diversified video aspects with decent categorical accuracy. In our method, we explicitly take in a trigger verb token as part of the input to guide the prediction into a specific video aspect. Sequentially argument word tokens together with their respective argument labels are generated as the related entities involved in that particular video aspect; see as an illustration of qualitative results. The performance of Video2Event was compared against traditional video captioning approaches, and the results show that it can retrieve event argument tokens more accurately in general. All experiments were performed on top of the large-scale video description dataset VATEX[3], which is a highly diverse video dataset in terms of both video contents and natural language descriptions.



	Trigger: painting	**Trigger**: shows	**Trigger**: performs	**Trigger**: play
Cap2E (Trans)	**nsubj**: person; **dobj**: substance; **prep**: on; **pobj**: surface	**nsubj**: woman; **xcomp**: apply	**nsubj**: girl; **dobj**: flip	**nsubj**: girl; **prep**: with; **pobj**: balls
Vid2E (Vanilla)	**nsubj**: woman; **dobj**: nails; **prep**: with; **pobj**: brush	**nsubj**: woman; **xcomp**: apply	**nsubj**: girl; **dobj**: cartwheel; **prep**: on; **pobj**: day	**nsubj**: girls; **prep**: on; **pobj**: beach
Vid2E (X-linear)	**nsubj**: woman; **dobj**: nails; **prep**: with; **pobj**: brush	**nsubj**: woman; **xcomp**: apply	**nsubj**: girl; **dobj**: cartwheel; **prep**: in; **pobj**: front	**nsubj**: girls; **prep**: in; **pobj**: sand
GT	**nsubj**: woman; **dobj**: nails; **prep**: in; **pobj**: colors	**nsubj**: woman; **xcomp**: painting	**nsubj**: women; **dobj**: cartwheel	**nsubj**: people; **prep**: at; **pobj**: pad

Figure 4: Two examples of the algorithm for video processing.

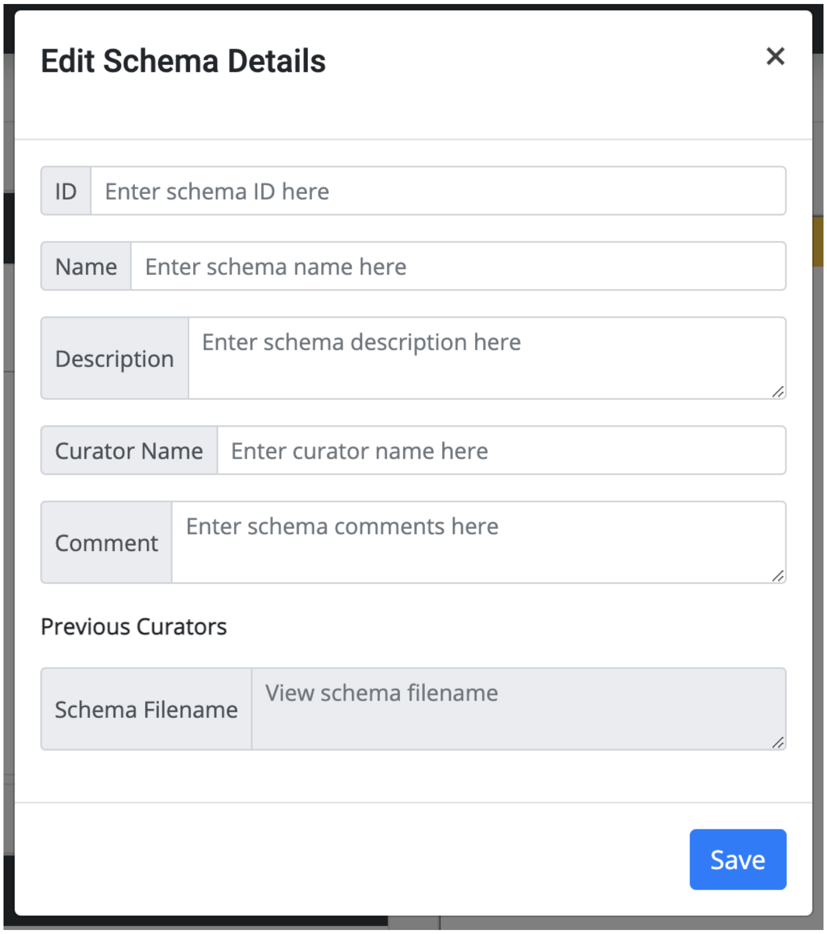
3.4 Algorithms in the MASC Curation Tool

From the time of drafting a proposal, it was clear that human review was essential to creating a schema repository. The results of the first year and of DARPA's de-emphasis of non-text sources in creating such a repository improved understanding of what needed to be in such a human-machine interface for curation. Beyond the requisite capabilities of browsing, searching, creating, reviewing, and modifying the formal schemas, several functionalities were required; algorithms would assist human curation. ISI developed a graphical user interface, first reported in (Ciosici et al., 2021), and continued to expand its capabilities and sophistication for the remainder of the effort. MASC was made accessible to the Government through a password-protected interface at https://lestat-curator.isi.edu/. While this server has been taken offline with the end of the program, corresponding source code for MASC has been shared with the government.

For purposes of illustration here (and in our publications), we typically chose complex events likely to be highly familiar to a reader, such as buying a car, rather than less common complex events of potential interest, like a terrorist attack, a disease outbreak, violent riots, a coup, and chemical spills. The human curator initiates script creation by providing a name and description for the script. See Figure 5 for a screen shot of the pane to initiate a schema in MASC.

---

[3] https://eric-xw.github.io/vatex-website/about.html#:~:text=VATEX%20is%20a%20new%20large,in%20both%20English%20and%20Chinese



Edit Schema Details

ID: Enter schema ID here

Name: Enter schema name here

Description: Enter schema description here

Curator Name: Enter curator name here

Comment: Enter schema comments here

Previous Curators

Schema Filename: View schema filename

Figure 5: Initiating creation of a schema for a complex event

Events are then entered, as text, whether suggested by a person or by algorithm. Step entry is free-form, but we have noticed a tendency for curators to enter short, imperative sentences around a central agent's actions (e.g., go to a car dealership, take a test drive). Schema creation, unlike traditional annotation for information extraction or classifying into categories, is not dependent on any particular document. In cases where the curator is not familiar with a topic, we have used external resources to provide context (e.g., a Wikihow page open in a different window). An ontology type must be assigned to the main event of each step (e.g., Movement for both *go to a car dealership* and *take a test drive*). The ontology is configurable and can be replaced. We include the project-specific ontology of Wikidata with MASC's source code. When saved, scripts include both the curators' description and the selected ontology type. This choice allows type decisions to be revisited if the ontology changes and limits the degree to which the small number of event types constrains the script's expressiveness. Downstream applications can choose whether to use the linguistic representation of the events or the normalized ontology types.

After all events have been entered, connections between events temporal order and shared arguments (e.g., the same person is the agent of both Movement events go to a car dealership and take a test drive). While this process is mostly manual, MASC uses the ontology’s constraints to limit the available label options. In addition to project-specific entity types, MASC suggests links to the much larger set of types available using Wikidata entities (e.g., suggesting Q786803 for car deal-ership). These links provide a connection to an extensive knowledge graph and can provide addi-tional information when the scripts are applied.

MASC involves several artificial intelligence algorithms (see Table 1). The challenges and ap-proach are summarized in Table 1. Subsections 3.4.1 - 3.4.8 describe the algorithm used in each row of that table.

Table 1: Core Functionality contributed by algorithms.

Capability	Machine Technique
Suggest events in text for schema	An LLM suggests an event(s) for a schema 1. Interactive mode. Given events e_1, … e_n, suggest the next event e_{n+1} 2. Batch mode. Suggest an ordered sequence of events for the schema
Suggest Wikidata qNode for event	1. Automated event detection through part-of-speech tagging and nominal identification. 2. Term expansion/rewriting with morphologically derived heuristics 3. qNode candidate retrieval through KGTK search 4. qNode candidate ranking via trained classifier
Assign arguments to events	Semantic role labeling (LLM fine-tuned for token classification) + lexical resource (Wikidata/XPO overlay) + human-in-the-loop confirmation & amendments
Suggest Wikidata qNode for each argument	1. qNode candidate retrieval through KGTK search 2. qNode candidate ranking via trained classifier
Suggest temporal ordering of events	Machine suggestions from LLM have temporal ordering; human provides ordering for human-suggested events
Argument relations	Natural language inference
Importance/likelihood estimates	Similarity metric of the event to the remaining part of the schema
Generating JSON	Algorithmic mapping from MASC internal form to JSON output form

3.4.1 Suggesting Events in Text for a Schema

A potential challenge for any human is knowing that all the events typical of a complex event have been added to a schema. It is always easier to spot an error in event representation in a schema than to identify an event that should be added. To our knowledge, ISI was the first team to use a large language model (LLM) to hypothesize missing events. This was first reported in (Ciosici et al., 2021); we used GPT-2 from OpenAI; it is a downloadable[4], free resource, available in 2020. From 2020 through 2022, MASC used GPT-2 with in-house algorithms to minimize poor output from GPT-2. In 2021 and 2022, OpenAI began selling LLM output as a service using much larger models, such as GPT-3, GPT-3.5, and GPT-4.

---

[4] https://huggingface.co/gpt2

3.4.1.1 Using GPT-2 for Suggestions

The human can ask that MASC suggest events based on the manually entered description and initial text of events, either in *interactive mode* (to suggest events the user may have overlooked) or in *batch mode* (to provide text descriptions for events of the schema from scratch). The suggestions can add intermediate steps that the curators may have missed, complete a script that was intentionally unfinished by the curator, or suggest alternative related paths (e.g., leasing instead of purchasing a car).

An example input to the LLM in interactive mode is as follows:

- title/name of the schema (e.g., *buying a car*),
- a description of the complex event (e.g., *Purchasing a car is a large investment that requires careful documentation and consideration of transportation requirements*.),
- a request (e.g., *Describe steps of buying a car*.), followed by the first few events of the script (e.g., *1. Identify your needs 2. Decide on your budget 3.*).

An output of six alternative, ranked, next events is produced and displayed

In batch mode, the input is only the title/name of the schema and the complex event’s description.

With GPT-2 predictions, MASC included automatic filters to improve its output. Those filters include:

- Undesirable strings, such as empty strings (Stahlberg and Byrne, 2019)
- Strings of less than two words and
- Those with sequences of non-alphabetic characters.
- Redundant output. GPT-2 often generated strings with a similar meaning, but lexically different, e.g., for a script on buying a car, it might generate *buy*, *buy the car*, and purchase the car. It is superfluous to show users all three suggestions. The filters eliminate redundant strings among the alternative outputs and redundance with the human-curated schema. To account for semantic duplicates, such as go to dealership and go to the car dealership, we used a variant of Gestalt Pattern Matching (Ratcliff and Metzener, 1988) through Python’s difflib. For usability, we suggest at most 12 sub-events per script.

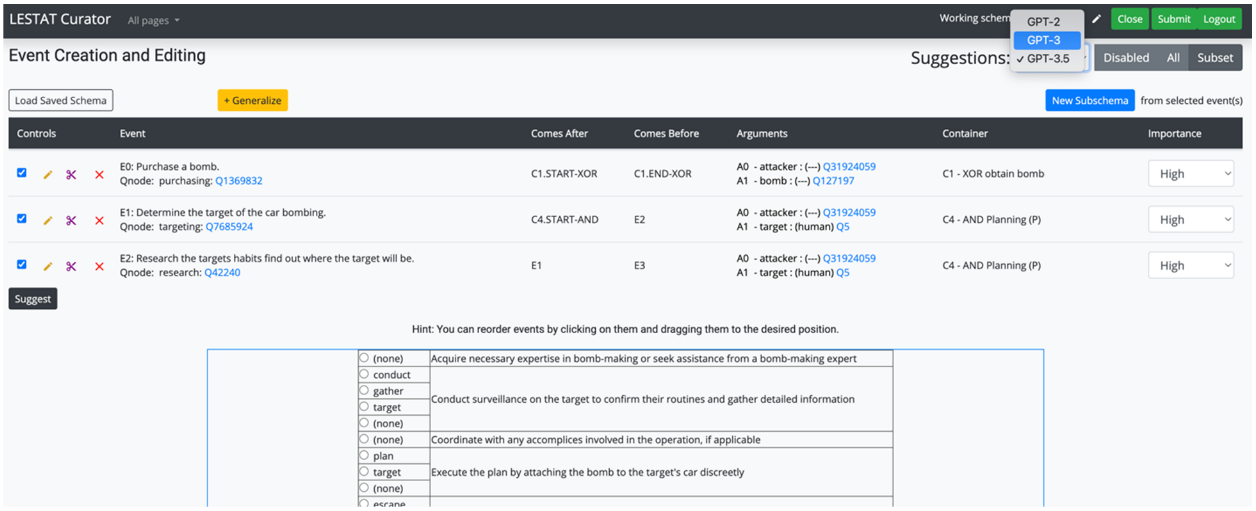
3.4.1.2 Performance on Case Studies

We measure the performance of GPT-2 recommendations in two ways. First, we generate recommendations for five scripts created by curators and ask the curators to accept relevant GPT-2 recommendations. We instruct curators to accept recommendations even if the recommended events represent alternative paths (or are semantically redundant). With these instructions, the curators accept 98% of GPT-2’s recommendations. The high acceptance rate indicates that even with our simple setup for event recommendation using a language model, the system suggests domain-relevant events. For the second evaluation, we instruct the curators to accept only those GPT-2 recommendations that add to their existing script. In other words, they only accept events that add details to the scripts or supply some missing information. We instruct curators to reject recommendations for alternative script scenarios. With these instructions, curators accept 23% of GPT-2’s recommendations. This result illustrates the feasibility of supplementing human knowledge with generations from language models. Since MASC uses GPT-2 after the human felt the script was complete, the machine identifies events previously overlooked by the human.

3.4.1.3 GPT-* Suggestions

After the release of ChatGPT by OpenAI, ISI experimented with OpenAI's paid services GPT-3, GPT-3.5, and GPT-4. The user must be signed-in to access GPT-3 and GPT-3.5-based completions, which use OpenAI's inference API. The prompt (input) to GPT-3 uses the same pre-processing system and prompting template as with GPT-2. However, because GPT-3.5 is a "chat" model rather than a "completion" model, our prompting process was updated. Both GPT-3 and GPT-3.5 required less post-processing than GPT-2 because both are generally better at creating properly formatted output. GPT-3.5 is asked to generate only one completion, in the form of a numbered list, which allows the model to generate a coherent group of suggestions, e.g., enabling consistency of granularity and diversity of step contents. Overall, in subjective assessments, the GPT-3.5 event suggestion pipeline produced the best generations, although no formal evaluation was done to quantify performance. In Section **Error! Reference source not found.**, experience with GPT-4 after the final evaluation is reported.

A screenshot for the review of automatic suggestions of next events appears in Figure 6. Note that the events already in the schema appear near the top of the screen; alternatives are presented near the bottom.



Event Creation and Editing

Suggestions: ✓ GPT-3.5

New Subschema from selected event(s)

Controls	Event	Comes After	Comes Before	Arguments	Container	Importance
☑	E0: Purchase a bomb. Qnode: purchasing: Q1369832	C1.START-XOR	C1.END-XOR	A0 - attacker : (---) Q31924059 A1 - bomb : (---) Q127197	C1 - XOR obtain bomb	High
☑	E1: Determine the target of the car bombing. Qnode: targeting: Q7685924	C4.START-AND	E2	A0 - attacker : (---) Q31924059 A1 - target : (human) Q5	C4 - AND Planning (P)	High
☑	E2: Research the targets habits find out where the target will be. Qnode: research: Q42240	E1	E3	A0 - attacker : (---) Q31924059 A1 - target : (human) Q5	C4 - AND Planning (P)	High

Hint: You can reorder events by clicking on them and dragging them to the desired position.

○ (none)	Acquire necessary expertise in bomb-making or seek assistance from a bomb-making expert
○ conduct ○ gather ○ target ○ (none)	Conduct surveillance on the target to confirm their routines and gather detailed information
○ (none)	Coordinate with any accomplices involved in the operation, if applicable
○ plan ○ target ○ (none)	Execute the plan by attaching the bomb to the target's car discreetly
○ escape	

Figure 6: A screenshot of the user interface for reviewing and selecting among GPT suggestions.

3.4.1.4 Ranking Machine-Suggested Events

Since LLMs can make many, many suggestions, a goal in MASC is to present only a few and rank them so that the user reviews as few as possible. Therefore, each machine-suggested event is compared against all the other events in the same schema. This comparison is done by passing a tuple of the suggested event, and a concatenation of the rest of the events in the same schema, through a BERT model. Next, the importance BERT assigns to each of these entities (calculated using an average of attention values of each of the heads in the last layer of BERT) is assigned as a score to the given tuple. Next all the suggestions are ranked based on this score and the top 5 (or less if that is the case) suggestions are shown to the user. This re-ranking of GPT suggestions using another masked language model like BERT shows improved quality of suggestions shown to the curator; ISI did not try to quantify the degree of improvement.

3.4.2 Suggest Wikidata qNodes for an Event

3.4.2.1 Detect Event Mentions

Each event is ontonlogized and mapped to a qNode in Wikidata. The first step in generating qNode suggestions is extracting event mentions from the provided event text. Candidate event mentions often occur as verbs, such as "attack", "review" and "obtain". Event mentions that occur this way are extracted from the event text using SpaCy’s part-of-speech tagger. To extract event mentions that do not occur as verbs, we start by collecting a set of desirable Nnode labels. Key qNodes representing high-level concepts of events such as “Action” and “Occurrence” are first identified. For each high-level concept, the descendent qNodes are identified using Wikidata"s "instance of" and “subclass of” attributes. Each token in the event text, generated using a sentence-transformer[5] as well as it’s lemma and present-tense inflection, are used as search queries to check if they are present in the set of desired qNodes. This results in a list of candidate event mentions that can be used to query for qNode suggestions.

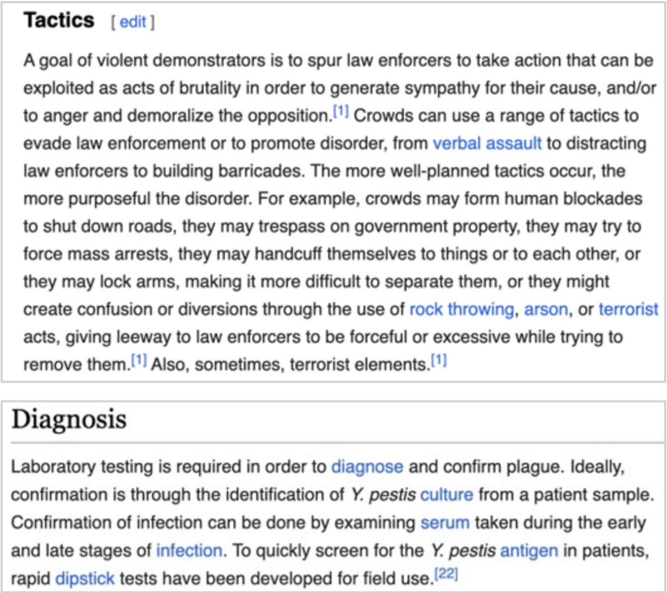
3.4.2.2 Find and Rank qNode Suggestions

An ElasticSearch API built on qNode labels and aliases is used to query and rank relevant qNodes. qNodes often include several aliases associated with them. However, the coverage of aliases is inconsistent. For example, “*chase” “pursue”*, “*chasing*”, *“pursuing”* are aliases for the qNode “*pursuit*”, but for the qNode “*driving*”, *“drive”* does not appear as an alias. To ensure that the candidate event mention are compared with both the qNode label or its aliases, we perform query expansion while querying for qNode suggestions. To perform, query expansion, a collection of English words is taken and stemmed. The words are organized into groups, indexed by their stems. When processing an input query, all terms that share the same stem as the input query are queried as well. The query process produces a list of appropriate qNodes and their corresponding retrieval scores. The retrieval scores are used to rank the retrieved qNodes and subsequently used as Wikidata suggestions for the event. MASC accesses Wikidata through the Knowledge Graph Toolkit[6] (KGTK) (Ilievski, et al., 2020) which provides an initial ranking.

---

[5] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

[6] https://github.com/usc-isi-i2/kgtk/ holds the main repository. https://github.com/usc-isi-i2/kgtk-browser is the browser. The search code is in https://github.com/usc-isi-i2/kgtk-search.



Tactics [edit]

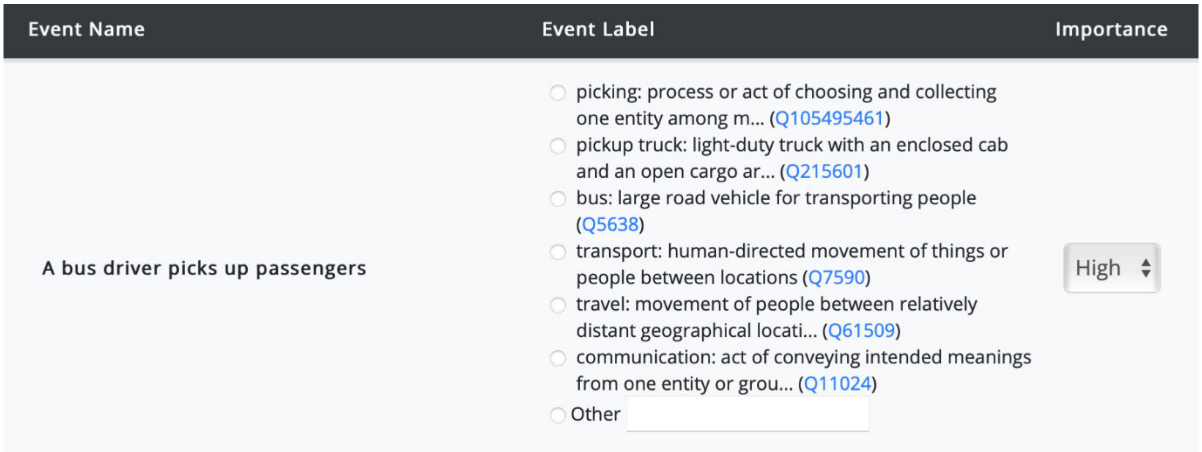
A goal of violent demonstrators is to spur law enforcers to take action that can be exploited as acts of brutality in order to generate sympathy for their cause, and/or to anger and demoralize the opposition.[1] Crowds can use a range of tactics to evade law enforcement or to promote disorder, from verbal assault to distracting law enforcers to building barricades. The more well-planned tactics occur, the more purposeful the disorder. For example, crowds may form human blockades to shut down roads, they may trespass on government property, they may try to force mass arrests, they may handcuff themselves to things or to each other, or they may lock arms, making it more difficult to separate them, or they might create confusion or diversions through the use of rock throwing, arson, or terrorist acts, giving leeway to law enforcers to be forceful or excessive while trying to remove them.[1] Also, sometimes, terrorist elements.[1]

Diagnosis

Laboratory testing is required in order to diagnose and confirm plague. Ideally, confirmation is through the identification of *Y. pestis* culture from a patient sample. Confirmation of infection can be done by examining serum taken during the early and late stages of infection. To quickly screen for the *Y. pestis* antigen in patients, rapid dipstick tests have been developed for field use.[22]

Figure 7: Examples of Non-Named Concepts with Links in Wikipedia. From articles on Civil Unrest (upper) and Bubonic Plague (lower).

To improve the quality of qNode matches, we apply re-ranking models on top of KGTK's suggestions. Our re-ranking models are fine-tuned using found data that approximates the task of aligning a word in sentence (i.e., the event word in a sentence) in context with a word and its description. We explored using both PropBank example sentences and definitions and data harvesting (e.g., PropBank example sentences and their definitions, cross-links scraped from Wikipedia, for example the links as illustrated in Figure 7). The evaluation version of MASC incorporated suggestions based on a fine-tuned DeBERTa (He, 2020). In later work, we explored using more recent pretrained models as the base, specifically, Falcon-7b(Almazrouei, 2023) and Llama2-7b (Touvron, 2023). The more recent models achieved a higher classification accuracy on a small internally developed test-set of sentence-to-qNode decisions using less training data. The pane for reviewing/correcting automatic suggestions of qNodes for the event appears in Figure 8.



Event Name	Event Label	Importance
A bus driver picks up passengers	○ picking: process or act of choosing and collecting one entity among m... (Q105495461) ○ pickup truck: light-duty truck with an enclosed cab and an open cargo ar... (Q215601) ○ bus: large road vehicle for transporting people (Q5638) ○ transport: human-directed movement of things or people between locations (Q7590) ○ travel: movement of people between relatively distant geographical locati... (Q61509) ○ communication: act of conveying intended meanings from one entity or grou... (Q11024) ○ Other	High

Figure 8: A screenshot of 6 alternative Wikidata qNodes for the event "A bus driver picks up passengers."

3.4.3 Assign Arguments to Events

A schema contains not only events, but also arguments that play an important role in the description of the complex event (i.e., those arguments that help define the event itself, argument-entities that play a critical role across multiple events). Arguments are connected to events by argument roles. For instance, who/what is the agent in the event (the “Arg0”), who/what is the object acted on in the event (the “Arg1”), etc. In initial designs of the schema creation, the user had to drag a suggested argument to the proper role (ARG0, ARG1, etc.) in the interface. Later, ISI used an open-source semantic role labeler[7] (SRL) from the Allen Institute for Artificial Intelligence to extract potential arguments from the event description text, then we take advantage of the alignment between SRL arguments and the cross-program ontology to attach these arguments to roles in the event they were extracted from.

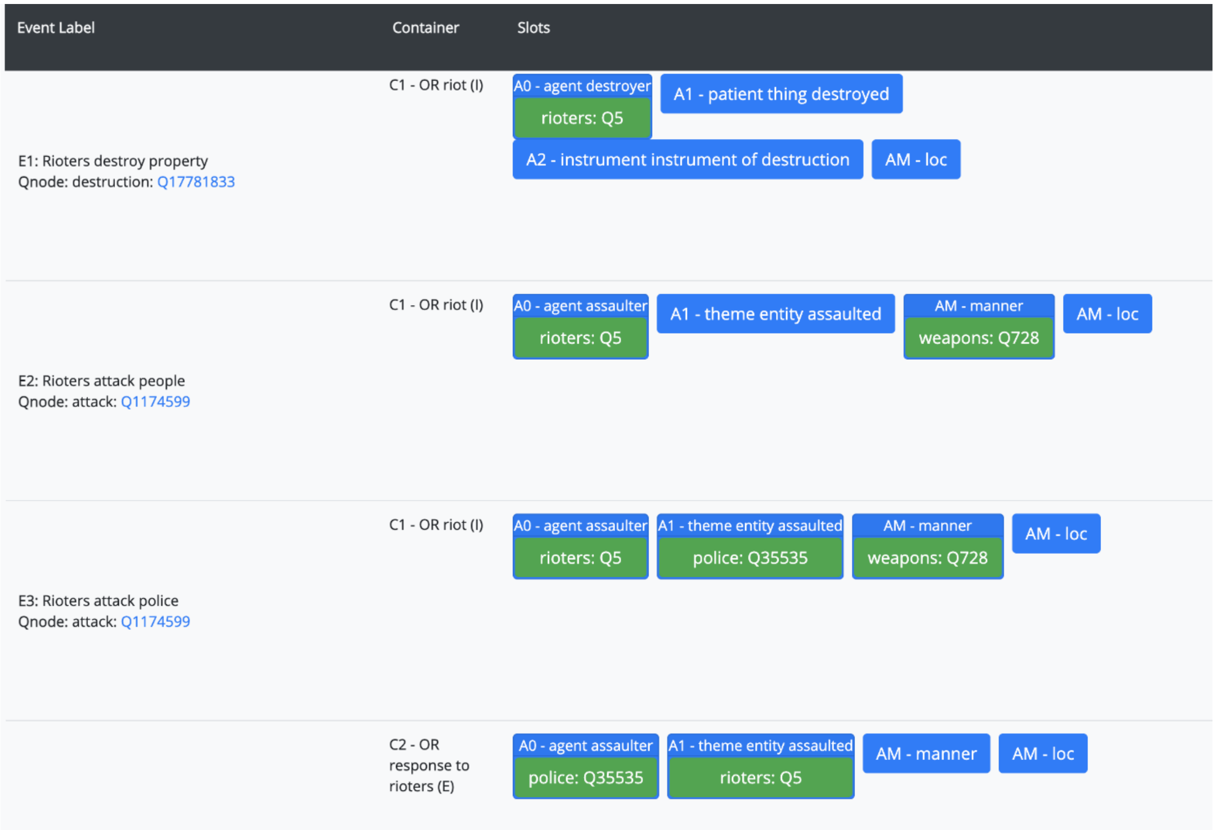
For example, if the user inputs “The rioters destroy public property” as the description of an event, the SRL model will extract [ARG0 - rioters] and [ARG1 - public property] as two arguments paired with semantic roles. Assuming the user chooses “destruction - Q17781833” as the qNode for the event, XPO defines this qNode as having the following argument roles:

- “A0 - agent, destroyer”
- “A1 - patient, thing destroyed”
- “A2 - instrument, instrument of destruction”
- “AM - location”

This new argument suggestion feature displays these suggestions in the relevant slots (A0 & A1, respectively) and allows the user to accept either, both, or none at all. The automatic assignment of arguments to events in a schema can be reviewed/corrected in MASC; see Figure 9.

---

[7] https://demo.allennlp.org/semantic-role-labeling/semantic-role-labeling

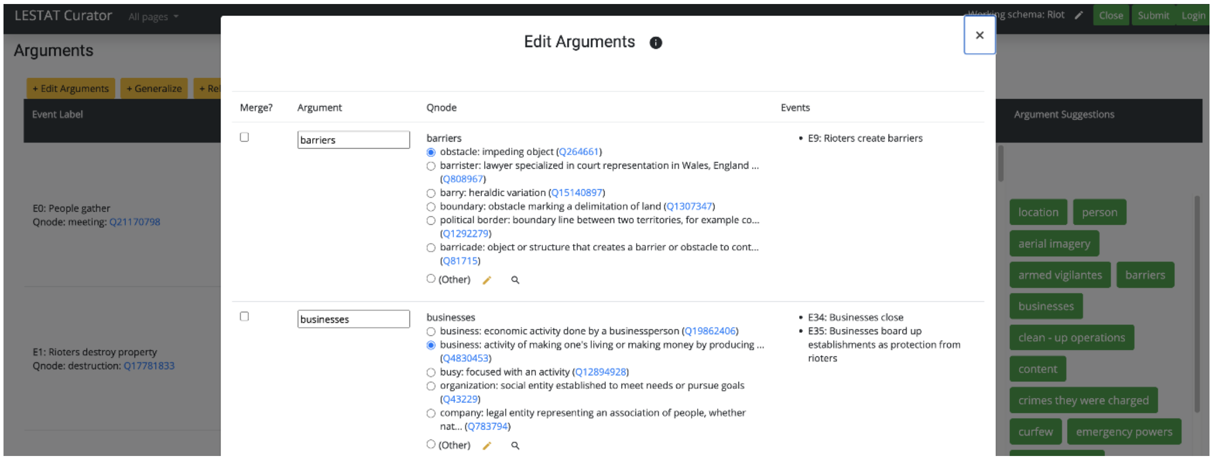


Event Label	Container	Slots
E1: Rioters destroy property Qnode: destruction: Q17781833	C1 - OR riot (I)	A0 - agent destroyer rioters: Q5 A1 - patient thing destroyed A2 - instrument instrument of destruction AM - loc
E2: Rioters attack people Qnode: attack: Q1174599	C1 - OR riot (I)	A0 - agent assaulter rioters: Q5 A1 - theme entity assaulted AM - manner weapons: Q728 AM - loc
E3: Rioters attack police Qnode: attack: Q1174599	C1 - OR riot (I)	A0 - agent assaulter rioters: Q5 A1 - theme entity assaulted police: Q35535 AM - manner weapons: Q728 AM - loc
	C2 - OR response to rioters (E)	A0 - agent assaulter police: Q35535 A1 - theme entity assaulted rioters: Q5 AM - manner AM - loc

Figure 9: Review pane in MASC to support review and correction of roles and their arguments for events in a schema

3.4.4 Suggest Wikidata qNode for Each Argument

Arguments, like events, must be assigned qNodes which complement the role assignment describe above. For arguments, qNode assignment serves to provide a context invariant type to the argument. As with other aspects of schema discover, MASC employs a suggestion-based approach to providing a curator access to potential argument qNodes. Suggestions use approaches similar to those described in Section 3.4.2 for event qNode assignment. Figure 10 illustrates the review/correction pane for argument qNodes.



E0: People gather
Qnode: meeting: Q21170798

E1: Rioters destroy property
Qnode: destruction: Q17781833

Merge?	Argument	Qnode	Events
☐	barriers	barriers ● obstacle: impeding object (Q264661) ○ barrister: lawyer specialized in court representation in Wales, England ... (Q808967) ○ barry: heraldic variation (Q15140897) ○ boundary: obstacle marking a delimitation of land (Q1307347) ○ political border: boundary line between two territories, for example co... (Q1292279) ○ barricade: object or structure that creates a barrier or obstacle to cont... (Q81715) ○ (Other)	- E9: Rioters create barriers
☐	businesses	businesses ○ business: economic activity done by a businessperson (Q19862406) ● business: activity of making one's living or making money by producing ... (Q4830453) ○ busy: focused with an activity (Q12894928) ○ organization: social entity established to meet needs or pursue goals (Q43229) ○ company: legal entity representing an association of people, whether nat... (Q783794) ○ (Other)	- E34: Businesses close - E35: Businesses board up establishments as protection from rioters

Figure 10: Pane for user to review automatically suggested Wikidata qNodes for arguments

3.4.5 Suggest Temporal Ordering of Events

When MASC suggests an event in either interactive mode or in batch mode, the preceding event before the suggested event is explicitly given. However, when a user suggests an event, temporal order may not be explicit. ISI explored alternatives for recommending temporal relations for events suggested by the user.

3.4.5.1 Pairwise Event Ordering

ISI investigated using GPT-3 in a zero-shot configuration for automatic temporal ordering in MASC. Using GPT-3 (with a hand-crafted prompt informed by literature on prompt engineering) predicted the order of randomly selected pairs of events within our schema library. Unfortunately, this approach proved very ineffective at predicting the correct order of events. ISI followed up with human annotation of temporal order between randomly selected pairs of events; the finding is that most event pairs were judged to be unordered.

ISI also investigated using an instance of BERT fine-tuned on the PeKO dataset[8] for temporal ordering. Since PeKO is based specifically on preconditions (i.e. events that are both temporally before and causally requisite for the following event), we believed this dataset would align well with the events in MASC scripts. However, the performance of this approach was not satisfactory and ultimately MASC does not incorporate it.

3.4.5.2 Event Order Annotation

In addition to ISI's several script curation efforts, we additionally organized for our engineers to review existing pairs of events found within our existing script library for temporal order. Annotators were asked to determine whether the pair of events definitionally occur in one particular order ("*Can A occur before B?*", "*Can B occur before A?*") and whether either event is a prerequisite for the other.

---

[8] https://stonybrooknlp.github.io/PeKO/

We noticed that the annotators marked a much larger share of event pairs as unordered than what would be indicated by the original relations found in the curated scripts. Meanwhile, agreement between annotators on pairs that most annotators thought were inherently ordered was relatively low. The results of this data collection process indicated to ISI that correct event ordering within scripts is at best very difficult to determine and potentially nebulous in definition.

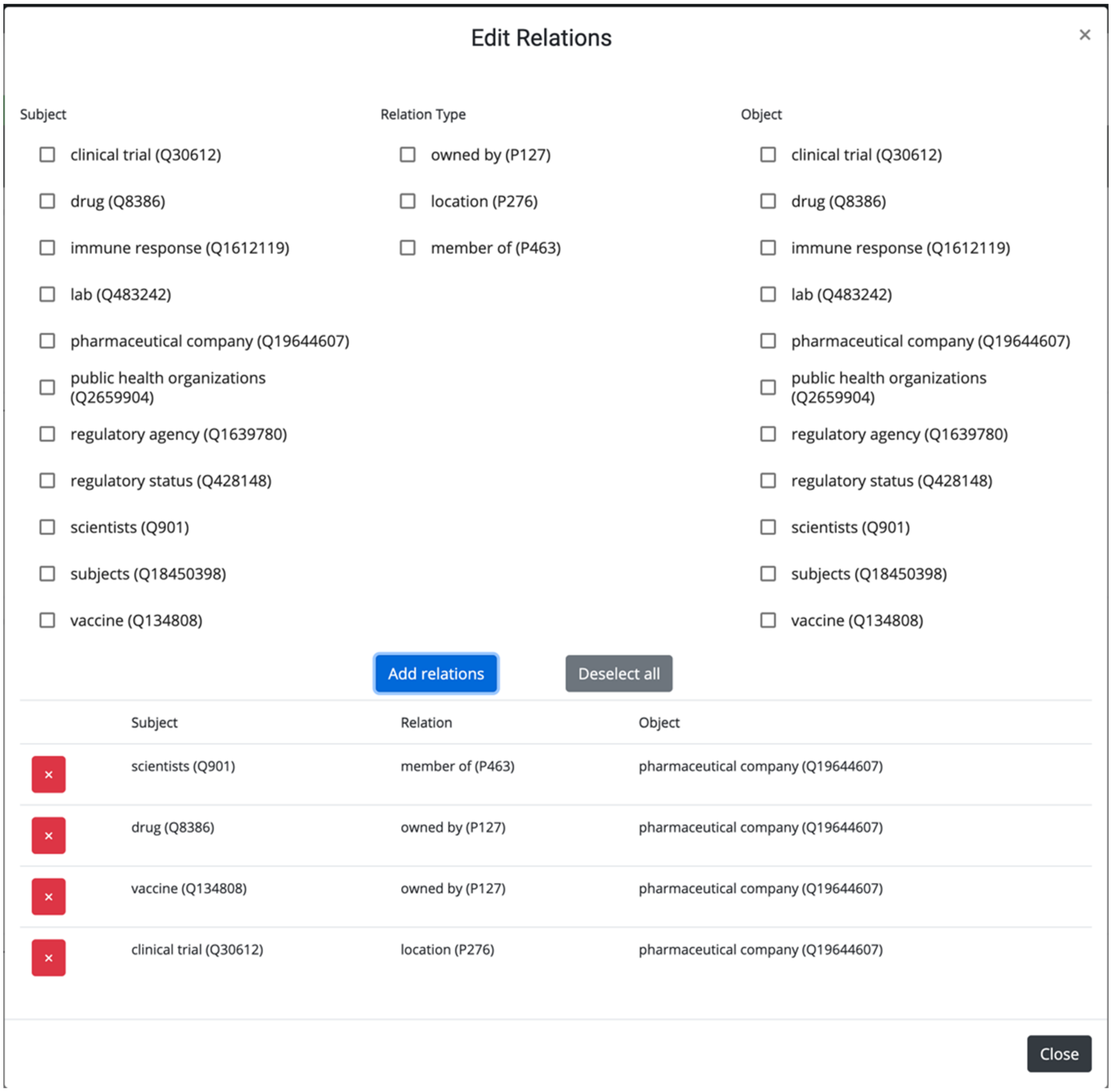
3.4.6 Argument Relations

With UIUC, ISI found that RESIN's matches of schemas and events to documents could be improved if ISI's schema library included argument relation information. Based on analysis, ISI focused on three primary relations: "owned by", "location", and "member of" for automatic prediction, since these were the most prevalent. We finetuned a DeBERTa MNLI model[9] on relations extracted from ACE 2005[10]. Our finetuning data was generated by input pairs of the reference span and a formatted string representing a natural language representation of the relation. For example, if the relation for the text "John works for the government" is ("John", "ORG-AFF:Employment", "the government"), the formatted text will be "'John' is employed by 'the government'". Given the span text and a formatted text representation of the relation, the model must predict ENTAILMENT if the relation is expressed in the SPAN, otherwise the MNLI model should predict NEUTRAL if it is not expressed. This model was tuned for precision so relations above a confidence threshold of 0.9 could be automatically added to a schema without explicit curator review. Since curators do still want to view these relations, and possibly add additional relations not captured by this automatic process, MASC's argument page was updated to include a new modal which both (a) displays previous or automatically added relations and (b) enables adding or removing relations from the schema. Figure 11 illustrate the result for the 'Clinical Trial' schema, relations may have been either manually added or exist from the automatic suggestion system.

---

[9] https://huggingface.co/microsoft/deberta-large-mnli

[10] https://paperswithcode.com/dataset/ace-2005


Edit Relations
×
Subject
☐ clinical trial (Q30612)
☐ drug (Q8386)
☐ immune response (Q1612119)
☐ lab (Q483242)
☐ pharmaceutical company (Q19644607)
☐ public health organizations (Q2659904)
☐ regulatory agency (Q1639780)
☐ regulatory status (Q428148)
☐ scientists (Q901)
☐ subjects (Q18450398)
☐ vaccine (Q134808)
Relation Type
☐ owned by (P127)
☐ location (P276)
☐ member of (P463)
Object
☐ clinical trial (Q30612)
☐ drug (Q8386)
☐ immune response (Q1612119)
☐ lab (Q483242)
☐ pharmaceutical company (Q19644607)
☐ public health organizations (Q2659904)
☐ regulatory agency (Q1639780)
☐ regulatory status (Q428148)
☐ scientists (Q901)
☐ subjects (Q18450398)
☐ vaccine (Q134808)
Add relations
Deselect all
Close

	Subject	Relation	Object
×	scientists (Q901)	member of (P463)	pharmaceutical company (Q19644607)
×	drug (Q8386)	owned by (P127)	pharmaceutical company (Q19644607)
×	vaccine (Q134808)	owned by (P127)	pharmaceutical company (Q19644607)
×	clinical trial (Q30612)	location (P276)	pharmaceutical company (Q19644607)

Figure 11: The Argument Relations modal with the schema's arguments in two columns and the relations available in the center. At the bottom of the modal, the existing relations in the schema can be found, along with a delete function.

3.4.7 Estimates of Importance of an Event

For the dry run prior to the final evaluation, LESTAT's schema library provided only limited, human estimates of importance/likelihood values to ISI's RESIN TA2 (prediction) system. The dry run showed how hard a task this is for human curation. The UIUC team reported that importance estimates were crucial for RESIN system performance both in automatically finding the best schema for a set of documents about some complex event and also for ranking the predictions of unseen events that might have occurred or might occur in the future.

Therefore, prior to the final evaluation, ISI began exploring automatic techniques for estimating importance. First, we developed a method of automatically estimating likelihood from corpus statistics based on the Wikidata qnodes in a schema, i.e., estimating p(event | schema). A very simple technique was implemented which first automatically mapped the label for a schema to a keyword(s), then automatically mapped an event qNode to keywords, and retrieved documents from a corpus that contain the schema keywords. For a schema S, let n = # events in schema S; e_i an event in S; D the set of documents involving schema S; C the corpus

- For each e_i, determine r_i = #(documents in D containing e_i) / #(documents in D)
- Let m be max r_i over all i in schema S
- To “normalize” across schemas and to smooth, uniformly shift the estimates between 0.95 (most likely) and 0.01 (not observed).

While intuitively an observation-based method seems appropriate, intuitions on the values computed varied greatly. It was not therefore integrated into the MASC curator.

Instead, ISI integrated an intrinsic measure of importance/likelihood. An individual was asked to manually “triage” schema events for 15 schemas into three categories: “highly likely”, ”highly unlikely”, and other. While this captured one person’s intuitions, two others annotated the same events; ISI explored a set of events for each of the 15 schemas where all three agreed that the event was highly likely and a set where at least two agreed that the event was highly likely.

An evaluation framework, Mean Average Precision, was employed. Specifically, the aforementioned set of events in which all three agreed and the set where at least two agreed were considered as alternative gold sets. Importance scores were generated using various methodologies like the aforementioned keyword-based approach, and compared against random baselines like sentence length, and certain BERT-based comparisons. The likelihood scores thus produced were compared against the gold sets and scored using Mean Average Precision on whether an event was highly likely or not.

Further, one person explored training a novel model based on quantum natural language processing to evaluate this task as a natural language inference task. Specifically, he manually annotated around 200 events into “highly likely” and “highly unlikely” classes. Further, each event in a given schema was cast as a premise of the natural language inference task, which was then compared against the rest of the events in the schema, sans this event. This was trained against the aforementioned manually annotated likelihood scores. The prediction by this trained model was also calculated against the gold sets using the evaluation framework mentioned above.

The best algorithm based on mean average precision in predicting highly likely events used SBERT to measure the similarity of the text of the event whose importance was to be estimated compared to the concatenation of the text descriptions of the other events in the schema.

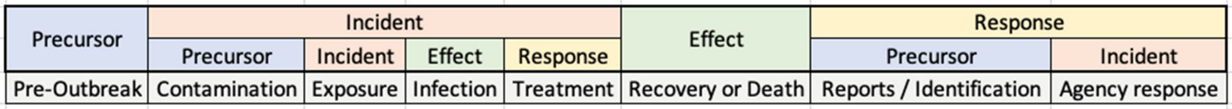
Since that measure had the best mean average precision, that was used as the estimate of importance in the final evaluation.

3.4.8 Structure and Hierarchy

The structure of schemas has evolved throughout curation activities within MASC. Earlier schemas tended to be more linear in their construction with branching limited to true simultaneous events (see section 3.4.5 for more details on this process). However, error analysis over various quizlets and dry runs with the UIUC TA2 system found that our schema temporal ordering often mismatched Graph G ordering. As a result of the limitations discovered with primarily linear schema curation, we revisited the approach for curation related to temporal structure. As a result of this review, we identified coupling between the temporal staging and logical structure of schemas. The two organization methods share a UI screen in MASC, encouraging a curator to actively consider both while organizing a schema. We explain temporal stages and named logical gates in more detail in the following sections then summarize how the two combine when supporting schema reuse across the larger library.

3.4.8.1 Temporally Anchored Stages

Our temporal stages follow a structure we call PIER. PIER breaks down temporal stages in a schema to distinct definitions with a shared grounding that each stage can be represented via a named logical gate (described below) of generally a logical *OR* combination[11]. Figure 12 illustrates the four named stages Precursor, Incident, Effect, and Response on an example 'Disease Outbreak' topic. We use this example to illustrate the definitions of PIER's various stages, the benefits of this less-linear structure, and the challenges herein. Additionally, while PIER presents the Effect stage occurring prior to the response stage, these two often overlap in practice.



Precursor	Incident				Effect	Response	
	Precursor	Incident	Effect	Response		Precursor	Incident
Pre-Outbreak	Contamination	Exposure	Infection	Treatment	Recovery or Death	Reports / Identification	Agency response

Figure 12: PIER example broken out by anchored stage on a 'Disease Outbreak' topic. The top-Incident stage contains an internal breakdown of PIER, demonstrating hierarchical reuse as a component of PIER layouts. The top-Response stage also demonstrates this hierarchical reuse utilizing only two of the PIER stages.

Precursor. The precursor temporal stage of a schema contains *events that must or may occur prior to the named topic.* As a result of our event prompting techniques, schemas in our final evaluation set contain minimal steps in this stage. Following the example in Figure 12, we note that precursor events aim to not be events needed to immediately set up the incident, but instead enable the conditions for such an incident to develop. For a disease outbreak an example precursor event sentence is "A refrigerator breaks down and goes unnoticed". This action creates an environment in which food can begin to spoil and become harmful.

Incident. The incident temporal stage of a schema contains *the immediate actions causing or resulting from the named topic.* The named topic usually come from the schema name, but a named gate can also be used. For the example in Figure 12, the top-level incident follows *outbreak* of a Disease Outbreak schema. This process is refined with a sub named topic of *infection* to enable specialization to various methods.

Effect. The effect temporal stage of a schema contains *the events after the named topic where existing agents continue to primarily serve as arguments.* The effect stage enables curators to separate actions following the incident, reducing the burden of more fine-grained temporal alignment. The events in this stage continue to relate to the *primary* agents in the Incident stage. In the Disease

---

[11] Top-level stage containers are rarely an *AND* operator due to schema's complex nature. Such logical *AND* stages often occurred in simpler schemas.

Outbreak example, the effects follow the furtherance of treatment of infected individuals to the resolution of either living or dying. The events in this stage tend towards the micro.

Response. The response temporal stage of a schema contains *the events after the named topic where external agents respond or react generally as temporally longer steps.* The response stage encompasses temporally longer events following the incident. Often this includes legal actions, after-action reviews, or recovery operations that span weeks-to-months. In practice this stage also introduces novel agents to the schema as new actors take on prominent roles. In the Disease Outbreak example such events may be "A government health agency recommends preventative measures", "A cure is researched and manufactured", or "Medical supplies are routed to hospitals to replenish consumed supplies".

3.4.8.2 Named Logical Gates

Within MASC, curators can visualize the stages via Named Logical Gates. These gates consist of (1) a human-interpretable string initialized at gate creation, (2) a Logical Operator from the set AND, OR[12], and XOR, and (3) a set of events or other named gates which the gate contains. The PIER stages discussed previously are semantically represented by a gate named for the stage. Figure 13 shows a simplified schema for the example PIER layout in Figure 12. Table 2 aligns the gate and event values to the example string. In schema gates C0, C1, C3, and C4 correspond to the four stages of PIER, with the temporal ordering shown as solid arrows between the corresponding gates.

C0 and C4 demonstrate an OR gate which enables the contained events to optionally occur. C1 and C3 demonstrate an XOR gate, indicating that only one of the available options may be instantiated. For C1, this limits an instantiated schema between a 'Disease Outbreak" subschema or a simplified 'Outbreak and Infection' default in C2. C2 demonstrates an AND gate, in this case all four events are required in an instantiated schema and must occur in a linear order. By utilizing these various logical gates and nesting them within each other (where C2 is contained by C1) curators can segment Schemas into increasingly granular segments for usability.

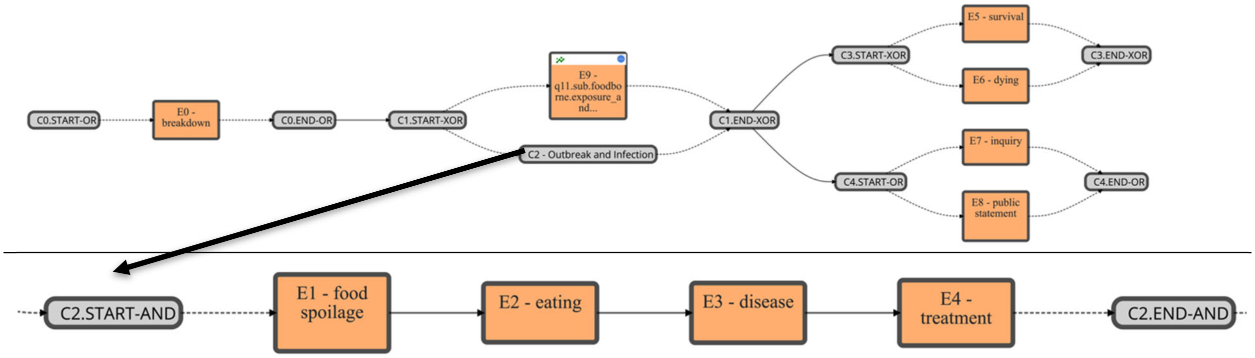


Figure 13: Above Figure 12 PIER's stage examples converted into a simplified schema. See Table 2 for the alignment of the event or gate ID to sentence examples. MASC visualizes the two separately for readability, we reproduce this layout here.

---

[12] OR in this case is defined as "0 or more" rather than the standard logical definition of "1 or more"

Table 2: Example schema alignment from IDs to qNode label and example sentence.

Event / Gate ID	qNode Label / Operator	Example Sentence
E0	Breakdown	A refrigerator breaks down and goes unnoticed
E1	Food spoilage	Spoiled food gets contaminated
E2	Eating	People eat the spoiled food
E3	Disease	People get sick with a foodborne illness
E4	Treatment	People receive treatment for the disease
E5	Survival	People recover from the illness
E6	Dying	People die from the illness
E7	Inquiry	A government agency investigates multiple outbreak reports
E8	Public Statement	A government agency announces a disease outbreak to the public
E9	Foodborne Subschema	Detailed foodborne infection subschema
C0	OR	Precursor
C1	XOR	Infection Options (Incident)
C2	AND	Outbreak and Infection
C3	XOR	Infected Persons Outcome (Effect)
C4	OR	Responses

3.4.8.3 Modular Reusable Sub-Schemas

Schemas intended for modular reuse need intentional framing to enable this capability. Generally, a schema on a new topic, even when PIER is implemented, is not immediately useful for reuse. The first problem for this reuse is the temporal scope of the entire schema. For example, consider a chain of natural disasters where an underwater earthquake causes a tsunami which leads into large coastal flooding. With complex individual schemas for each individual disaster (earthquake, tsunami, and flooding) one could imagine linking the three together internal to each other. However, format limitations imply that events (in the parent schema) following the subschema are after *all* event steps of the subschema. As a result, this can imply some strange temporal relations to internal effect and response stages. While this limitation can be worked around by breaking a schema up into modular schemas for each stage, this creates additional schemas in the library and loses the default connections and may result in curators being unaware they need to link multiple subschemas to achieve a valid story. This problem is best addressed through *Specialized Modular Sub-Schemas* where a single temporal stage from a schema is extracted and an XOR gate is used to link various specializations together. This approach was taken for the type of outbreak (airborne, animal borne, foodborne, etc) in Disease Outbreak.

The second problem we encountered is cross-topic reuse and contextual prioritization. From our final evaluation library, we included a *Hospitalization* schema which was created out of our Disease Outbreak curation but reused in most other topics. As a result, the hospitalization schema was wider rather than longer. However, despite the well-intentioned reuse. We noticed that linking contextual information from the parent schema (e.g. the mode of transportation to the hospital or the event causing the hospitalization), would be beneficial for better providing importance and likelihood values within the subschema as these values don't account for context outside of the local schema. A solution we may have considered for a phase focused on hierarchy would have been to create automatic variants of these reused subschemas linking in this context through automations.

In general, schema reuse requires intentionality that is hard to place into curating new schema topics for the first time. As a result, curating modular reusable subschemas becomes a review task within MASC for curators to identify stages of similar design and extract them into a reusable component. Both a subschema creation tool and a more complex schema merging tool exist within MASC to facilitate curators in creating these schemas.

3.4.8.4 Structured Schemas from GPT-3.5

After submitting our final evaluation schema library, we investigated how ChatGPT[13] inherently organizes schema-like content on similar topics. Two examples are shown below in Table 3. Intent was given in the prompt design to not ask for any specific structure of the output, past sentence-like events (reducing the total amount of content generated for each step). Surprisingly, these examples structure event steps similarly to the layout discussed above. ChatGPT organizes its results around various stages with additional detail within each phase. Such a discovery has prompted thoughts into how humans without our schema organization guidance and given a similar unstructured prompt may organize schema steps. Despite this well-organized output, schemas extracted from this method would likely still suffer from various problems of completeness (discussed in detail in Section 4.2.2).

Table 3: ChatGPT examples of self-organized schemas on Chemical Spills and Military Coups

Chemical Spill Schema[14]	**Military Coup Schema**[15]
Prompt: I'm building a schema of events related to a major chemical spill. Can you (1) suggest events, in the form of a sentence, which occur during this topic and (2) organize these events for me?	**Prompt:** I'm building a schema of events related to a Military Coup. Can you (1) suggest events, in the form of a sentence, which occur during this topic and (2) organize these events for me?
Response: Certainly, I can help you suggest events related to a major chemical spill and organize them into a logical sequence. Here are some events:	**Response:** Certainly, I can suggest events related to a military coup and organize them for you. A military coup typically involves

---

[13] ChatGPT-3.5 August 3rd , 2023 Version https://help.openai.com/en/articles/6825453-chatgpt-release-notes#h_c761ea2ff1

[14] https://chat.openai.com/share/ac4b46cd-ac7e-47e0-9bda-9137d7effa47

[15] https://chat.openai.com/share/36ce0463-4d52-4236-92ad-eeb28557b42b

1. **Chemical Storage**: Chemicals are stored improperly, leading to potential hazards.
2. **Spill Initiation**: The initial spill occurs due to a leak, rupture, or accident.
3. **Emergency Response**: First responders are alerted and dispatched to the spill site.
4. **Evacuation**: Nearby residents and workers are evacuated to ensure safety.
5. **Chemical Identification**: The spilled chemical is identified, including its toxicity and hazards.
6. **Containment Efforts**: Attempts are made to contain the spread of the chemical, such as using barriers or booms.
7. **Cleanup Operations**: Specialized teams work to clean up the spilled chemical.
8. **Health Assessments**: Affected individuals are assessed for injuries or exposure to harmful substances.
9. **Environmental Impact Assessment**: Experts evaluate the impact of the spill on the environment.
10. **Legal and Regulatory Actions**: Authorities investigate the cause and potential violations of regulations.
11. **Healthcare Responses**: Hospitals and medical facilities treat individuals affected by the spill.
12. **Long-term Monitoring**: Ongoing monitoring is established to assess the long-term effects of the spill.
13. **Public Notifications**: The public is informed about the situation and safety precautions.
14. **Community Support**: Community organizations and resources are mobilized to help affected people.

several key events leading up to and following the coup itself. Here's a list of events with brief descriptions, organized chronologically:

Events Leading to the Military Coup:

1. **Political Turmoil:** Widespread political unrest and discontent with the government's policies and leadership.
2. **Military Dissatisfaction:** Growing dissatisfaction within the military ranks due to alleged corruption or ineffectiveness of civilian leadership.
3. **Secret Planning:** Military officers and factions begin to secretly plan and discuss the possibility of a coup.
4. **Coup Leaders Emerge:** Key military leaders emerge as the driving force behind the coup, gaining support within the armed forces.

The Coup Takes Place:

5. **Seizure of Key Locations:** Coup plotters move to seize strategic locations like government buildings, media outlets, and communication centers.
6. **Announcement of Coup:** A military spokesperson declares the coup on national television, explaining their reasons for the takeover.
7. **Curfew and Martial Law:** A curfew is imposed, and martial law is declared, restricting civilian movement and freedoms.
8. **Arrest of Political Figures:** Key politicians and government officials are arrested or detained.
9. **Street Protests:** Protests both in favor and against the coup erupt on

Approved for Public Release; Distribution Unlimited.

15. **Recovery and Rehabilitation**: Efforts are made to restore the affected area and support affected communities.

Now, let's organize these events into a logical sequence:

Phase 1: Pre-Spill Preparation and Prevention

1. Chemical Storage
2. Regulatory Compliance

Phase 2: Spill Occurrence and Initial Response 3. Spill Initiation

4. Emergency Response
5. Evacuation

Phase 3: Spill Management and Mitigation 6. Chemical Identification

7. Containment Efforts
8. Cleanup Operations

Phase 4: Health and Environmental Assessment 9. Health Assessments

10. Environmental Impact Assessment

Phase 5: Post-Spill Actions and Recovery 11. Legal and Regulatory Actions

12. Healthcare Responses
13. Long-term Monitoring
14. Public Notifications
15. Community Support
16. Recovery and Rehabilitation

This sequence provides a structured overview of events related to a major chemical spill, from preparation to recovery and community support. Keep in mind that the specific details and order of events may vary depending on the circumstances and location of the spill.

the streets, potentially leading to violence.

Aftermath of the Coup:

10. **International Response:** The international community responds with condemnation, sanctions, or diplomatic efforts to restore civilian rule.
11. **Censorship:** Media censorship is enforced to control the flow of information and shape the narrative.
12. **Purges within the Military:** The coup leaders purge dissenting voices within the military to consolidate their power.
13. **Repression:** Crackdown on dissenting voices, activists, and opposition figures.
14. **Transition Government:** A transitional government is established, often with military leaders in key positions.
15. **Uncertainty and Instability:** The country faces a period of uncertainty and potential instability as the coup leaders try to establish control.

Please note that the specific events and their sequence can vary widely depending on the country and circumstances surrounding the military coup. This is a general outline to help you structure your schema

3.4.9 Generating SDF (Schema Data Format) JSON

We designed an internal YAML-based format to store our schemas, only using the program-wide SDF (Schema Data Format) JSON format to exchange schemas with other teams. We wrote a script to automatically convert a library of YAML files into an SDF file, which we only needed to run when delivering schemas. Although the improved readability of YAML helps humans directly read and edit schemas, the main advantage of having an internal format is that it insulated most of our work from the frequently changes and multiple redesigns of SDF over the program's length. Whenever there was an SDF change, we were usually able to accommodate by only changing the conversion script and not any of our other programs.

4.0 RESULTS AND DISCUSSION

NIST performed extrinsic evaluations of TA1 libraries as applied by TA2 algorithms. ISI's assigned TA2 partner was UIUC with its RESIN event prediction system. The sponsor also had the opportunity to conduct qualitative assessments of the schema library/repository and of the MASC tool (through the password-protected website). In this section, we first describe error analysis of a few examples which led to significant changes prior to the final evaluation and internal quantitative evaluations of specific AI components in MASC.

4.1 Error Analysis

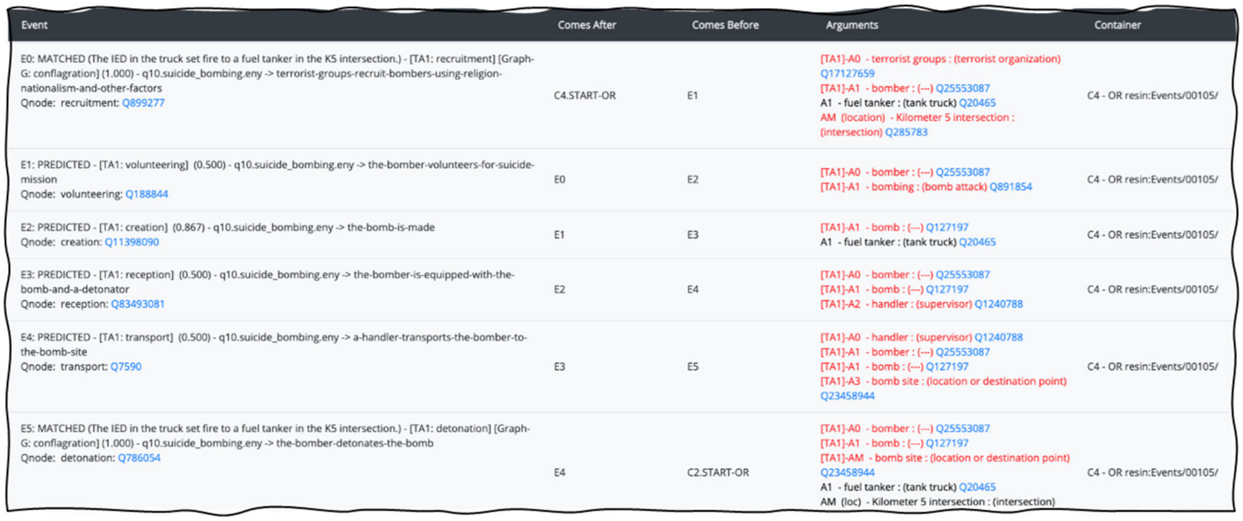
As part of preparation for the for the final evaluation, ISI assessed the effectiveness of the TA1 schema repository based on what UIUC's RESIN matched in the final dry run to determine:

- What errors occurred to discover ways that the library "confused" RESIN. In cases where RESIN's usage did not seem to make sense, those were reported to the RESIN team.
- Given the library, what would one expect a TA2 system to match? This could show gaps in the existing library independent of RESIN's specific instantiation approach in the dry run.

The analysis was based on four examples released by the Government near the beginning of 2023. Three were incidents of disease outbreak; one was an incident of terrorism. The release pertinent to TA1 approaches included a manually created input called Graph G for each incident. Each Graph G represented a set of events of the incident, e.g., the representation of event, entities participating in them, and relations between them. Each node of Graph G is a Wikidata qNode corresponding to a type of event or an entity. Relations (edges in the graph) correspond to the roles of arguments (e.g., ARG0 for the actor of an event and ARG1 for the object of an event) or to temporal relations (e.g., event E1 occurs before event E2). There were two forms of each Graph G: (1) the full graph as reported and (2) a partial graph where some reported events were explicitly omitted. The second is the input to a TA2 algorithm which uses a TA1 library of schemas. The events that are present in the second graph but not in the first represent events that the TA2 system should predict. Of course, a TA2 system may correctly predict events that were not reported; therefore human assessors must assess each event predicted by TA2 regarding its appropriateness in the context of the incident.

4.1.1 Errors Found

Note that this error analysis was based on UIUC running RESIN with the ISI schema library and with the Graph G input, which was manually produced by the Linguistic Data Consortium to factor out errors from automatic machine understanding of documents. Since RESIN's human readable format (HRF) at the time did not include predictions, RESIN output was loaded into the MASC curation tool to explore both the events that RESIN matched to an ISI schema and also the predictions made by RESIN. Below is a screenshot of a tabular view of the events RESIN matched to Graph G or predicted; any schema event not matched to Graph G was predicted by RESIN.



Event	Comes After	Comes Before	Arguments	Container
E0: MATCHED (The IED in the truck set fire to a fuel tanker in the K5 intersection.) - [TA1: recruitment] [Graph-G: conflagration] (1.000) - q10.suicide_bombing.eny -> terrorist-groups-recruit-bombers-using-religion-nationalism-and-other-factors Qnode: recruitment: Q899277	C4.START-OR	E1	[TA1]-A0 - terrorist groups : (terrorist organization) Q17127659 [TA1]-A1 - bomber : (---) Q25553087 A1 - fuel tanker : (tank truck) Q20465 AM (location) - Kilometer 5 intersection : (intersection) Q285783	C4 - OR resin:Events/00105/
E1: PREDICTED - [TA1: volunteering] (0.500) - q10.suicide_bombing.eny -> the-bomber-volunteers-for-suicide-mission Qnode: volunteering: Q188844	E0	E2	[TA1]-A0 - bomber : (---) Q25553087 [TA1]-A1 - bombing : (bomb attack) Q891854	C4 - OR resin:Events/00105/
E2: PREDICTED - [TA1: creation] (0.867) - q10.suicide_bombing.eny -> the-bomb-is-made Qnode: creation: Q11398090	E1	E3	[TA1]-A1 - bomb : (---) Q127197 A1 - fuel tanker : (tank truck) Q20465	C4 - OR resin:Events/00105/
E3: PREDICTED - [TA1: reception] (0.500) - q10.suicide_bombing.eny -> the-bomber-is-equipped-with-the-bomb-and-a-detonator Qnode: reception: Q83493081	E2	E4	[TA1]-A0 - bomber : (---) Q25553087 [TA1]-A1 - bomb : (---) Q127197 [TA1]-A2 - handler : (supervisor) Q1240788	C4 - OR resin:Events/00105/
E4: PREDICTED - [TA1: transport] (0.500) - q10.suicide_bombing.eny -> a-handler-transports-the-bomber-to-the-bomb-site Qnode: transport: Q7590	E3	E5	[TA1]-A0 - handler : (supervisor) Q1240788 [TA1]-A1 - bomber : (---) Q25553087 [TA1]-A1 - bomb : (---) Q127197 [TA1]-A3 - bomb site : (location or destination point) Q23458944	C4 - OR resin:Events/00105/
E5: MATCHED (The IED in the truck set fire to a fuel tanker in the K5 intersection.) - [TA1: detonation] [Graph-G: conflagration] (1.000) - q10.suicide_bombing.eny -> the-bomber-detonates-the-bomb Qnode: detonation: Q786054	E4	C2.START-OR	[TA1]-A0 - bomber : (---) Q25553087 [TA1]-A1 - bomb : (---) Q127197 [TA1]-AM - bomb site : (location or destination point) Q23458944 A1 - fuel tanker : (tank truck) Q20465 AM (loc) - Kilometer 5 intersection : (intersection)	C4 - OR resin:Events/00105/

Figure 14: RESIN's SDF Phase 2B Dryrun Output converted to display in MASC.

The leftmost column indicates whether the event was matched or predicted, the qNode of the event, and the qNode label; the two center columns indicate the preceding event and the following event; the next-to-rightmost column shows the arguments of the instantiated event; the rightmost column indicates a container that it appears in.

The items below show the two Graph G events correctly matched to one of our general disease outbreak schemas for the complex event corresponding to a cholera outbreak.

- prevention: Q1717246, *The Ministry of Health was going to prevent food contaminated with cholera from entering the Dominican Republic*,
- medical diagnosis: Q177719, *A Massachusetts woman and a 30-year-old man were diagnosed with cholera in Massachusetts and at Massachusetts General Hospital*

Any schema event not matched by Graph G was predicted by RESIN. Of those 8 predictions, ISI assessed that two were good. In the table below are three examples. The leftmost column gives the label and qNode for the event, plus the argument role (A1) with its qNode label and qNode; an assessment (*maybe*, *no*, or *yes*); and a comment which includes acknowledgement of how frequent this is in the corpus given the schema.

Table 4: Examples of assessments of three events predicted by RESIN

Event	Assessment	Comment
research: Q42240 A1 - disease: Q12136	Maybe	Does research: Q42240 include forensic analysis? Corpus shows it as rather frequent.
identification: Q3265221 A1 - cause of outbreak: Q2574811	Yes	Frequent in corpus.
development: Q7562091 A1 - treatment: Q179661	No	Treatment of cholera is known. No development. Not very frequent in corpus.

ISI also explored the Al-Shabaab bombing attack in Mogadishu. There were 25 events in Graph G, six of which RESIN matched to an ISI schema. Unfortunately, ISI judged that only one was correct; four of the six would have been appropriate predictions if RESIN had not incorrectly matched those events to Graph G. Of the 11 predictions made by RESIN, ISI assessed that seven were good predictions.

4.1.2 Desired Matches

Complementing our analysis of RESIN output, we manually matched Graph G to our schema repository for both the cholera outbreak and the El Shabaab bombing in Mogadishu. For the cholera outbreak, there was a schema that was a better match. That schema contains events for all the desired predictions for this complex event:

- Two are exact qNode matches
- The travel/waiting (as in serving food) match is less clear. Based on the sentence, it seems like a poor (human) qNode selection in Graph G.

The first two predictions came from the subschema `q8.sub.foodborne_illness_spread.jdlm` while the last came from the upper-level disease outbreak schema

In our analysis of the bombing incident in Mogadishu, the schema best matching graph G has three predictions which could match graph G. However, two of the three were not exact qNode matches, and thus might require human assessment rather than automatic scoring.

4.1.3 Conclusions from Error Analysis

Regarding ISI schemas, in the dry run repository, topically similar schemas varied greatly in size and specificity; therefore, the schemas seemed challenging for RESIN to select for instantiation. The repository provided only limited, curator-driven importance/likelihood values in the dry-run. The analysis showed that assessing importance/likelihood is a hard curation task! In fact, analysis of Graph G seemed to show that Graph G often included incidental facts of interest to newswire but not central to a schema for the complex event being reported.

The analysis led to the work on automatic estimates of importance/likelihood. At the time of the dry run, RESIN averaged their internal estimate of confidence and the repository's estimate of importance; based on the dry run, RESIN improved their prediction and ranking mechanism. UIUC updated RESIN's strategy; that improved schema matching.

Regarding Graph G, it was clear that qNode mismatches may hinder alignment. Incident-specific detail makes it difficult to understand where to draw the line for "how specific to be in a schema." For assessing predictions, the definition of a reasonable prediction remained elusive.

4.2 Schema Completeness and Machine Suggestion of Events Typical of a Complex Event

4.2.1 Machine Suggestion of Events while Creating an Initial Schema

Midway through KAIROS, ISI evaluated event suggestion using GPT-2 in two ways. *First*, we generated recommendations for five scripts created by human curators and ask the curators to accept relevant GPT-2 recommendations. We instructed them to accept recommendations even if the recommended events represent alternative paths (or are semantically redundant). With these instructions, the curators accepted 98% of GPT-2's recommendations. The high acceptance rate indicates that even with our simple setup for event recommendation using a language model, the system suggests domain-relevant events. For the *second* evaluation, we instructed the curators to accept only those GPT-2 recommendations that add to their existing script. In other words, they only accept events that add details to the scripts or supply some missing information. We instructed curators to reject recommendations for alternative script scenarios. With these instructions, curators accepted 23% of GPT-2's recommendations. This illustrated the feasibility of supplementing human knowledge with generations from language models.

Since MASC used GPT-2 after the human felt the script was complete, the machine identifies events previously overlooked by the human.

4.2.2 Assessing the Challenge of "Completeness" of a Schema

Our extensive experience in creating as many as 200 scripts over the last two years of the project convinced us that determining that a script is "complete" is quite challenging. Since we already had experience in using GPT-2 for suggesting events that a human curator might have overlooked, in August 2023 near the end of the project, ISI evaluated the effectiveness of the far richer, far larger OpenAI model GPT-4 (OpenAI, 2023), which unlike GPT-2, is software as a (paid) service.

Ciosici et al. (2021) and Zhang et al. (2023) examine their systems' performance on five case study scripts covering the following topics: evacuation (EVAC), ordering food at a restaurant (FOOD), searching for a job (JOB), receiving medical treatment (MED), and corporate mergers (MERGER). Both papers include curator (human) review of their scripts, thus the resulting output can be treated as fully correct in the eyes of a human reviewer. While neither paper asserted completeness of their scripts, one can compare these two sets of scripts to understand how much could be gained by adding information to an existing script. Table 5 shows the assessed overlap between the scripts that resulted from the two processes. In this analysis, one can treat as overlapping exact match in qNode selection for a pair of steps, (i.e., in Table 6, two steps with the qNode employing).

Table 5: Script comparison by library from Ciosici et al. (2021) and from Zhang et al. (2023).

Schema	Ciosici et al.	Zhang et al.	Overlap
Evacuation	11	11	0
Food	10	8	6
Job	13	12	3
Medical	5	12	2
Merger	11	10	1

Tellingly, the overlap is low: for three of the five scripts, at most 25% of the sub-events overlap in either direction. In no case is one script completely subsumed by the other. This hints that an initial round of script creation is often incomplete.

To further explore an example of potential incompleteness, Table 6 shows a selection of event steps from the JOB script. The outer columns list qNodes specific to each library while the center column lists qNode event steps present in both.

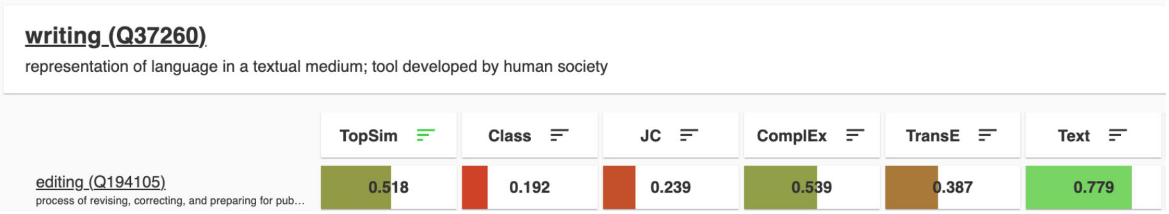
Table 6: Subset of event steps, shown as qNodes, from the JOB script from Ciosici et al. (2021) and from Zhang et al. (2023). Steps are shown without respect to temporal order.

Ciosici et al.	Overlap	Zhang et al.
Inquiry	Acceptance	Editing
Resignation	Employing	Search
Writing	Interview	Termination

Independently each script is good as an initial draft and both agree on core steps of accepting a job, beginning employment, and interviewing the candidate. However, the two diverge in the framing prior to the interview process and after a job has been accepted. While some of these differences can be accounted by event choice (e.g., editing a resume vs. writing a resume) with qNode-derived labels of "writing" and "editing", respectively. Relying on exact match of label provides a precise definition of overlap, but is a coarse measure since it can:

- Overcount overlap given the size and potential for synonymous terms to appear in Wikidata.
- Undercount overlap when the same event occurs in different contexts within a script.

Table 6 provides examples of non-overlapping labels, illustrating that semantic differences are common. One could imagine using similarity measures to try to refine this metric; however, identifying the threshold for such measures is challenging. Figure 15 shows a comparison of similarity metrics using the Knowledge Graph Tool Kit (KGTK)'s similarity interface (Ilievski et al., 2021) where only textual similarity has a high score.



writing (Q37260)

representation of language in a textual medium; tool developed by human society

	TopSim	Class	JC	ComplEx	TransE	Text
editing (Q194105) process of revising, correcting, and preparing for pub…	0.518	0.192	0.239	0.539	0.387	0.779

Figure 15: KGTK similarity metrics between the qNodes for writing and editing using various techniques.

We performed an additional analysis GPT-4's ability to complete an existing script. In this analysis, we formulated a prompt that incorporated the steps of an initial published script from either Ciosici et al. or Zhang et al. For example, the prompt for the earthquake script was as follows:

Prompt: *An earthquake (also known as a quake, tremor or temblor) is the shaking of the surface of the Earth resulting from a sudden release of energy in the Earth's lithosphere that creates seismic waves The step(s) so far (in no particular order) are as follows: 1. A volcano erupts setting up the potential of an earthquake 2. An earthquake occurs 3. People are injured by falling debris 4. Cleanup crews remove debris from roadways and other locations List additional steps in the same format and grammatical style:*

GPT's response was as follows:

Response (formatted):

1. Rescue teams search for survivors in the rubble

2. Medical teams provide first aid to the injured

3. Temporary shelters are set up for people who have lost their homes

4. Seismologists analyze the data to learn more about the earthquake

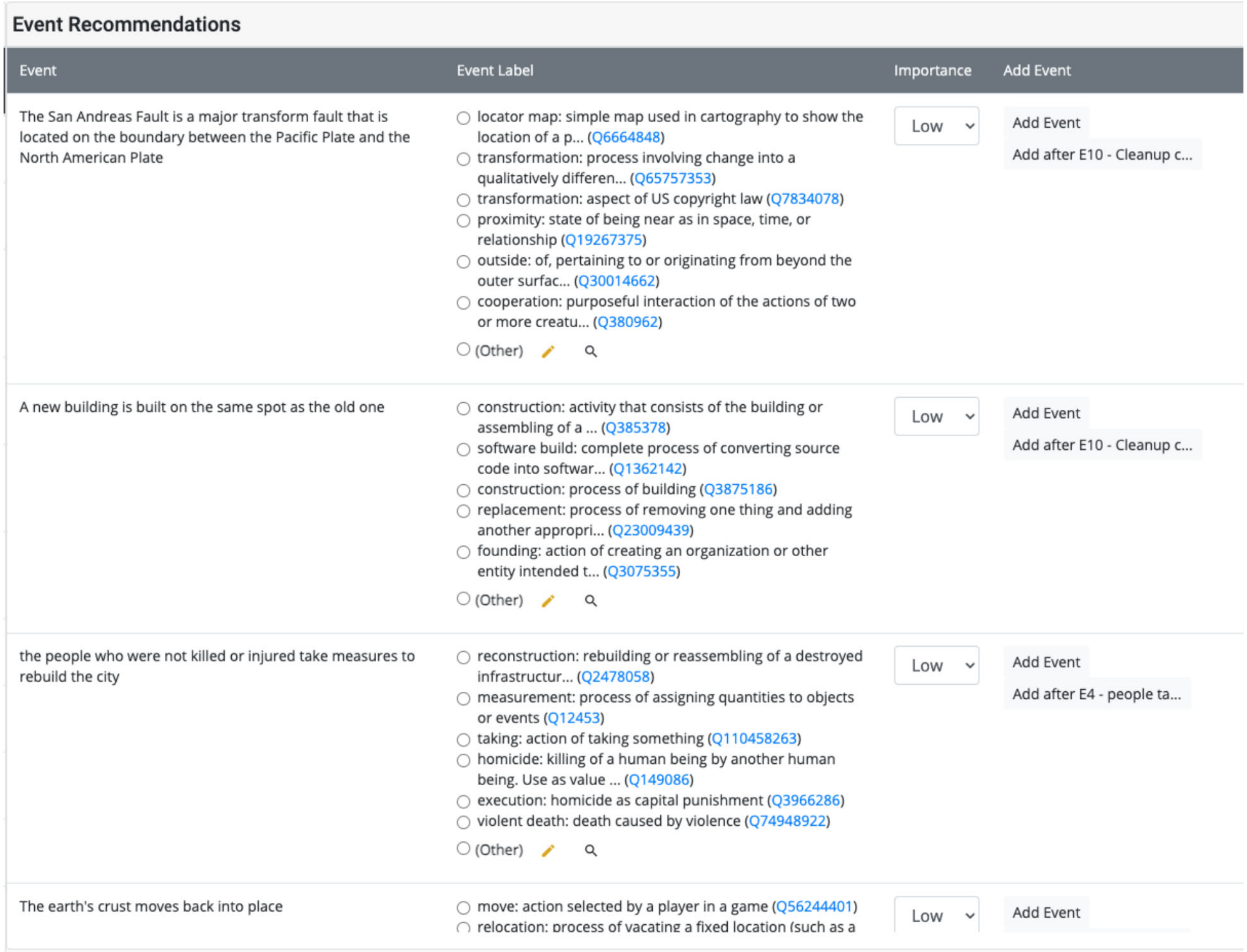
5. Government agencies assess the overall damage to infrastructure

6. Fundraising activities are organized to help with relief efforts

7. Reconstruction projects are initiated to rebuild damaged buildings and infrastructures

8. Preventive measures are taken to minimize the impact of future earthquakes

To evaluate the generated suggestions, we asked individuals with prior experience in script creation to review suggestions in the context of the pre-existing script (within the iMASC interface) and evaluate up to 20 suggestions per script for inclusion. Figure 16 illustrates the suggestion interface.



Event Recommendations

Event	Event Label	Importance	Add Event
The San Andreas Fault is a major transform fault that is located on the boundary between the Pacific Plate and the North American Plate	○ locator map: simple map used in cartography to show the location of a p... (Q6664848) ○ transformation: process involving change into a qualitatively differen... (Q65757353) ○ transformation: aspect of US copyright law (Q7834078) ○ proximity: state of being near as in space, time, or relationship (Q19267375) ○ outside: of, pertaining to or originating from beyond the outer surfac... (Q30014662) ○ cooperation: purposeful interaction of the actions of two or more creatu... (Q380962) ○ (Other)	Low	Add Event Add after E10 - Cleanup c...
A new building is built on the same spot as the old one	○ construction: activity that consists of the building or assembling of a ... (Q385378) ○ software build: complete process of converting source code into softwar... (Q1362142) ○ construction: process of building (Q3875186) ○ replacement: process of removing one thing and adding another appropri... (Q23009439) ○ founding: action of creating an organization or other entity intended t... (Q3075355) ○ (Other)	Low	Add Event Add after E10 - Cleanup c...
the people who were not killed or injured take measures to rebuild the city	○ reconstruction: rebuilding or reassembling of a destroyed infrastructur... (Q2478058) ○ measurement: process of assigning quantities to objects or events (Q12453) ○ taking: action of taking something (Q110458263) ○ homicide: killing of a human being by another human being. Use as value ... (Q149086) ○ execution: homicide as capital punishment (Q3966286) ○ violent death: death caused by violence (Q74948922) ○ (Other)	Low	Add Event Add after E4 - people ta...
The earth's crust moves back into place	○ move: action selected by a player in a game (Q56244401) ○ relocation: process of vacating a fixed location (such as a	Low	Add Event

Figure 16: A subset of batch event recommendations in MASC

Rather than relying on qNode overlap (as in Table 6), curators were instructed to make a more holistic judgment using both the natural language suggestions and the suggested qNodes and to accept steps that they considered novel and relevant with respect to the script. We asked the curators to consider novelty with respect to the Wikidata qNode label that anchors the event. We consider events that provide additional detail to be novel (e.g., "Person ingests spoiled food" provides additional detail in the context of an existing event "Person gets sick"). Events considered unusual but plausible are counted as relevant. The results of this process are shown in Table 7. Since the two human curators varied dramatically in which event suggestions should be accepted, an additional review of the accepted events were reviewed, focusing on the subset of steps that accepted by one curator and not by the other. Virtually all event suggestions were grammatical and relevant.

Table 7: Acceptance of event suggestions. Columns for Curators 1 and 2 are unique event suggestions added.

Schema	Library	Curator 1	Curator 2	Both
Evacuation	Ciosici et al.	0	11	3
	Zhang et al.	0	10	3
Food	Ciosici et al.	1	1	1
	Zhang et al.	2	3	1
Job	Ciosici et al.	1	5	2
	Zhang et al.	1	10	2
Medicine	Ciosici et al.	1	5	1
	Zhang et al.	0	10	2
Merger	Ciosici et al.	1	8	2
	Zhang et al.	1	5	3

The cases of disagreement further illustrate the challenge of completeness of a script and hint at the question of what should be included. There were multiple reasons why suggestion acceptance differed:

- **Redundancy**: An event suggestion is similar to an existing or suggested event step.
  Example: For (Zhang, FOOD), the initial script included "The person pays the bill". Curators differed as to whether they included the suggestion, "The person calculates the tip and includes it in the payment". The suggestion includes two events (calculation and payment). Payment is redundant, but one curator may have considered the suggestion of tip calculation sufficiently novel. This relates to granularity as described below.
- **Granularity**: An event suggestion describes a very specific sub-event of an existing or suggested event step.
  Example: For (Zhang, MED), the published script includes: the specialist prescribes a treatment plan and the patient begins the treatment plan. The curators disagreed about the addition of: "Possible side effects of the treatment are communicated to the patient". Perhaps this disagreement is about communication of side effects is subsumed in the prescription of a treatment plan.
- **Temporal scope**: Curators differed in their treatment of events that happened before or after the full pre-existing script.
  Example: In (Ciosici, FOOD) the curators differed in their acceptance of "Restaurant staff clear the table and prepare it for the next customer". In this case, the script description declared the script ends when the customer receives the food they ordered, so the suggestion was outside of the script's scope.

- **Reliance on not-yet included events**: In some cases, the suggestions could be relevant but were not well-connected to the rest of the script.
  Example: In (Ciosici, EVAC), curators differ in assessment of the suggestion of "Medical and auxiliary services are arranged at the shelters for the evacuees". However, no previous mention of shelters had appeared in the original version of the script. Perhaps the curator who rejected the suggestion believed that this was not a part of the script because it was too disconnected. Alternatively, the suggestion might inspire the curator to add other events about shelters so this suggestion fits better.

In summary, in comparing several scripts for the same complex events from two different publications, one can see that the overlap was surprisingly small. Generative AI (large language models) can be used to expand upon a pre-existing, curator-approved script. While no direct comparison between the GPT-4 suggestions used after the final evaluation and the GPT-2 suggestions used in the evaluation, it was qualitatively clear that GPT-4's suggestions are far better and (unlike GPT-2) rarely require filtering for relevance.

In this analysis of both pre-existing, independently created scripts and analysis of curator assessment about script additions, one can see that individuals differ in what they believe belongs in a script. As generative language models are incorporated into workflows that rely on the model's ability to produce general knowledge, the challenge of pre-defining completeness may mean that interactive workflows are required. While applications can often support this form of interaction, the lack of completion can present a challenge for evaluation. For the work presented here, a natural extension would be to explore multiple rounds of suggestions over increasingly detailed scripts to explore the progression of what a curator accepts. The related challenges of curator disagreement regarding granularity and redundancy hint at the need for script-like knowledge (and perhaps general-purpose "know how") to be represented in a way that a user can customize to the level of detail to meet their particular needs.

There is the opportunity of refinement for both suggestions that are prompted, e.g., designing prompts that specifically target not-yet included precursor events, and also how they are presented, e.g., incorporating techniques that identify similar yet non-identical steps and group them to assist the curators' decisions.

5.0 CONCLUSIONS

The LESTAT project resulted in MASC, a prototype that allowed for the human-machine collaboration in creating schemas that represent the structure of a complex event. Over the course of the program, we saw project-external improvements in the generative capabilities of large language models, such that the largest of the models provided well-structured, informative text that could serve as a description of a schema. However, to achieve the structured format of a schema, i.e., the labeled arguments, links to Wikidata in the form of qNodes, required additional special purpose processes.

Evaluating the quality, structure, and completeness of a schema remained a challenge throughout the project. Within our own error analysis, we found that for many complex events, the temporal order of most events is not prescribed. We also found that while there were a very small number strongly required events that were almost definitional (e.g., an explosion in a bombing complex event), that there were many events that fell into a grouping that felt reasonable to include in a schema but were not necessarily entailed by the presence of the complex event. It is possible that to sufficiently represent the variety within a complex event, a schema must represent many plausible, but uninstantiated sub-events when aligning with a particular instance of the complex event. As an interactive prototype, MASC provides a capability for exploring this representation challenge.

6.0 REFERENCES

Ebtesma Almazrouei, Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet et al. "The falcon series of open language models." arXiv preprint arXiv:2311.16867 (2023).

Manuel R. Ciosici, Joseph Cummings, Mitchell DeHaven, Alex Hedges, Yash Kankanampati, Dong-Ho Lee, Ralph Weischedel, Marjorie Freedman. Machine-Assisted Script Curation. Proceedings of NAACL-HLT 2021. 2021.

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.

Asaf Degani and Earl L. Wiener. 1993. Cockpit checklists: Concepts, design, and use. *Human Factors*, 35(2):345–359.

Atul Gawande. 2010. *The checklist manifesto: how to get things right*. Metropolitan Books, New York.

H Paul Grice. 1975. Logic and conversation, syntax and semantics. *Speech Acts*, 3:41–58.

Rujun Han, Qiang Ning, Nanyun Peng. Joint Event and Temporal Relation Extraction with Shared Representations and Structured Prediction. 2019. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*.

He, P., Liu, X., Gao, J., & Chen, W. (2020, October). DEBERTA: Decond-Enhanced BERT with Disentangled Attention. In *International Conference on Learning Representations*.

Filip Ilievski, Daniel Garijo, Hans Chalupsky, Naren Teja Divvala, Yixiang Yao, Craig Rogers, Ronpeng Li, Jun Liu, Amandeep Singh, Daniel Schwabe, and Pedro Szekely. 2020. KGTK: A toolkit for large knowledge graph manipulation and analysis. In *The Semantic Web – ISWC 2020*, pages 278–293, Cham. Springer International Publishing.

OpenAI. 2023. GPT-4 technical report. https://arxiv.org/abs/2303.08774

Roger C Schank and Robert P Abelson. 1977. *Scripts, plans, goals, and understanding: An inquiry into human knowledge structures*. Lawrence Erlbaum Associates.

Felix Stahlberg and Bill Byrne. 2019. On NMT search errors and model errors: Cat got your tongue? 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 3356– 3362, Hong Kong, China. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov et al. "Llama 2: Open foundation and fine-tuned chat models." arXiv preprint arXiv:2307.09288 (2023).

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.

Noah Weber, Leena Shekhar, Niranjan Balasubramanian, and Nathanael Chambers. 2018. Hierarchical quantized representations for script generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3783–3792, Brussels, Belgium. Association for Computational Linguistics.

Tianyi Zhang, Isaac Tham, Zhaoyi Hou, Jiaxuan Ren, Leon Zhou, Hainiu Xu, Li Zhang, Lara Martin, Rotem Dror, Sha Li, Heng Ji, Martha Palmer, Susan Windisch Brown, Reece Suchocki, and Chris Callison-Burch. 2023. Human-in-the-loop schema induction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics* (Volume 3: System Demonstrations), pages 1–10, Toronto, Canada. Association for Computational Linguistics.

Xiyang Zhang, Muhao Chen, and Jonathan May Salience-Aware Event Chain Modeling for Narrative Understanding. 2021*Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Approved for Public Release; Distribution Unlimited.

APPENDIX A – PUBLICATIONS

Manuel R. Ciosici, Joseph Cummings, Mitchell DeHaven, Alex Hedges, Yash Kankanampati, Dong-Ho Lee, Ralph Weischedel, Marjorie Freedman. Machine-Assisted Script Curation. Proceedings of NAACL-HLT 2021. 2021.

Rujun Han, I-Hung Hsu, Jiao Sun, Julia Baylon, Qiang Ning, Dan Roth, and Nanyun Peng. "ESTER: A machine reading comprehension dataset for reasoning about event semantic relations." In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7543-7559. 2021.

Manling Li, Qi Zeng, Ying Lin, Kyunghyun Cho, Heng Ji, Jonathan May, Nathanael Chambers, and Clare Voss. "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)*, pp. 684-695. 2020.

Mingyu Derek Ma, Jiao Sun, Mu Yang, Kung-Hsiang Huang, Nuan Wen, Shikhar Singh, Rujun Han, and Nanyun Peng. "EventPlus: A Temporal Event Understanding Pipeline." In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations,* pp. 56-65. 2021.

Alexander Spangher, Xiang Ren, Jonathan May, and Nanyun Peng. News Edits: A News Article Revision Dataset and a Novel Document-Level Reasoning Challenge. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2022.

Xiyang Zhang, Muhao Chen, and Jonathan May Salience-Aware Event Chain Modeling for Narrative Understanding. 2021*Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

APPENDIX B – SYSTEM DETAILS

In this appendix we describe the approach used to generate the schema library for the Spring 2022 evaluation. All schemas in the library were curated using a human-machine collaborative approach using ISI's MASC interface. By design, MASC suggests content which the curator can choose to accept, edit, or ignore. Curators included both the researchers working on MASC's development and 4 undergraduate students hired to assist with curation. The library incorporated schemas developed throughout the program, and thus the specific algorithms for suggestion varied. Curation was on-going throughout the effort and included both schemas relevant to the evaluation topics and schemas for events outside of the evaluation (e.g. for natural disasters).

For the Evaluation we tasked our curators with various improvements to our schemas. Given we knew the five topics for the Evaluation, we focused the bulk of our improvements and schema refactoring on schemas relevant to these topics. MASC's library browsing and search function (available as a schema index and shown in Figure 17) facilitates fast search and review of schemas based on their description and name.

Our refactoring consisted of the following steps:

(1) triage schemas by topic to identify a subset of our schema library relevant to the Evaluation,

(2) update schema event step arguments,

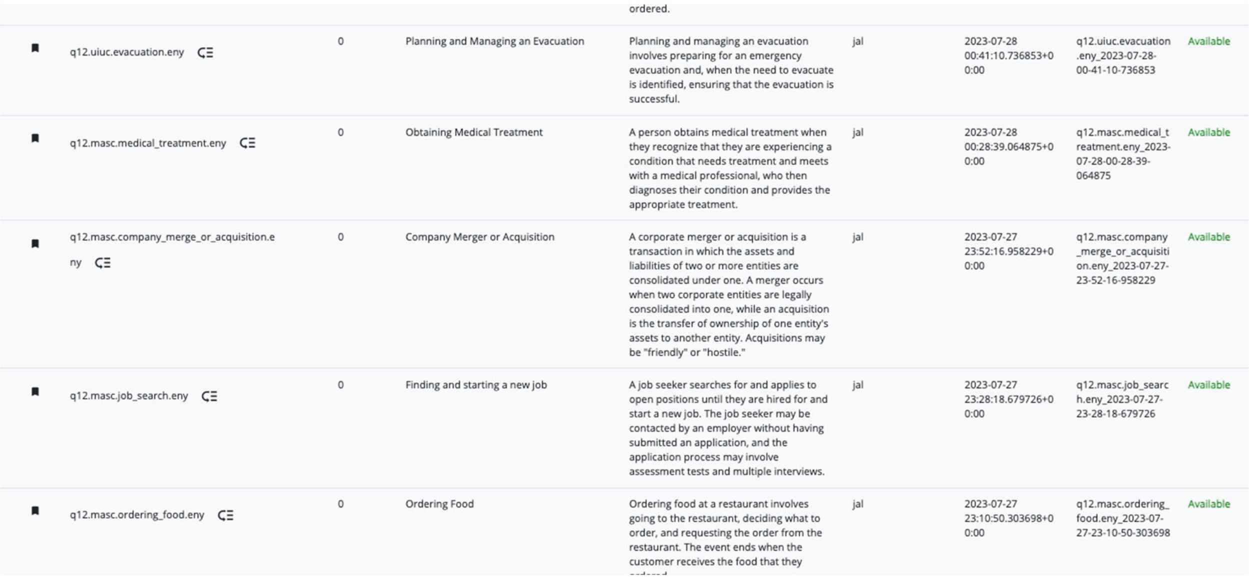
(3) revise the temporal and hierarchical structure to align with PIER (described in detail in Section 0),

(4) merge same-topic schemas together,

(5) perform a loose review of subschema usage across the library, and

(6) post-process the library to include automatic argument relations and SBert-based Likelihood scores.

The results of this process resulted in a final TA1 Schema Library consisting of 52 schemas for the five evaluation topics. This evaluation library is what we delivered to UIUC, our TA2 partner. An expanded library, including schemas which did not undergo this complete refactoring process consists of over 200 schemas and includes additional topics such as cyber-attacks, natural and human-made disasters, and criminal actions.



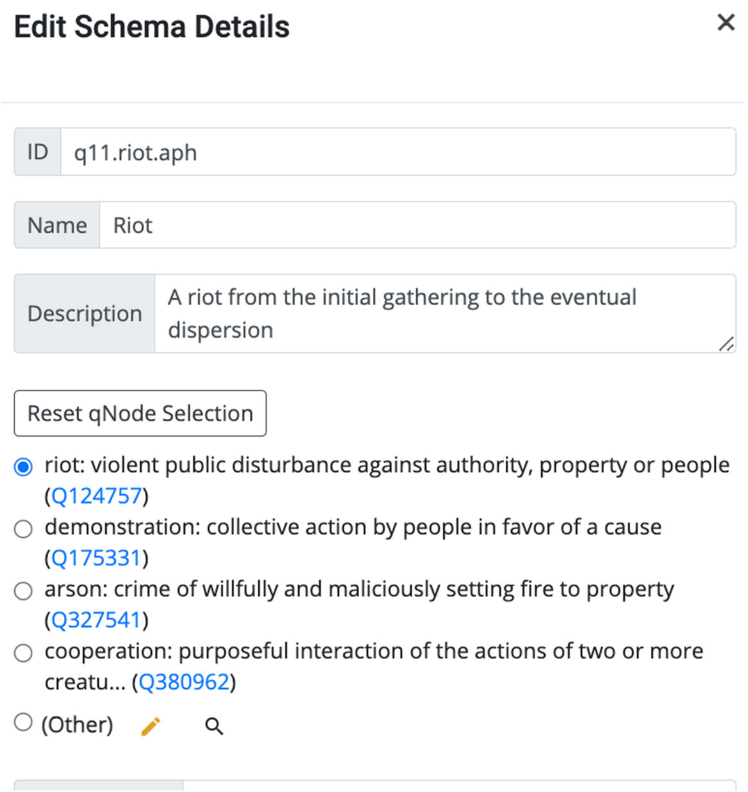
			ordered.				
q12.uiuc.evacuation.eny	0	Planning and Managing an Evacuation	Planning and managing an evacuation involves preparing for an emergency evacuation and, when the need to evacuate is identified, ensuring that the evacuation is successful.	jal	2023-07-28 00:41:10.736853+00:00	q12.uiuc.evacuation.eny_2023-07-28-00-41-10-736853	Available
q12.masc.medical_treatment.eny	0	Obtaining Medical Treatment	A person obtains medical treatment when they recognize that they are experiencing a condition that needs treatment and meets with a medical professional, who then diagnoses their condition and provides the appropriate treatment.	jal	2023-07-28 00:28:39.064875+00:00	q12.masc.medical_treatment.eny_2023-07-28-00-28-39-064875	Available
q12.masc.company_merge_or_acquisition.eny	0	Company Merger or Acquisition	A corporate merger or acquisition is a transaction in which the assets and liabilities of two or more entities are consolidated under one. A merger occurs when two corporate entities are legally consolidated into one, while an acquisition is the transfer of ownership of one entity's assets to another entity. Acquisitions may be "friendly" or "hostile."	jal	2023-07-27 23:52:16.958229+00:00	q12.masc.company_merge_or_acquisition.eny_2023-07-27-23-52-16-958229	Available
q12.masc.job_search.eny	0	Finding and starting a new job	A job seeker searches for and applies to open positions until they are hired for and start a new job. The job seeker may be contacted by an employer without having submitted an application, and the application process may involve assessment tests and multiple interviews.	jal	2023-07-27 23:28:18.679726+00:00	q12.masc.job_search.eny_2023-07-27-23-28-18-679726	Available
q12.masc.ordering_food.eny	0	Ordering Food	Ordering food at a restaurant involves going to the restaurant, deciding what to order, and requesting the order from the restaurant. The event ends when the customer receives the food that they	jal	2023-07-27 23:10:50.303698+00:00	q12.masc.ordering_food.eny_2023-07-27-23-10-50-303698	Available

Figure 17: Schema Index

In the following sub-sections, we describe the high-level interactions between MASC and the curator as well as the final models used to generate suggestions.

Schema Topic Description

Curators think of a topic for the schema and enter a name and description for that topic. As illustrated in Figure 18, the schema description has a curator defined ID, a human readable name, a short description, and a qNode. The qNode for the schema is suggested using the schema name field utilizing the event qNode recommendation for events (See section 3.4.2).


Edit Schema Details

ID: q11.riot.aph

Name: Riot

Description: A riot from the initial gathering to the eventual dispersion

● riot: violent public disturbance against authority, property or people (Q124757)

○ demonstration: collective action by people in favor of a cause (Q175331)

○ arson: crime of willfully and maliciously setting fire to property (Q327541)

○ cooperation: purposeful interaction of the actions of two or more creatu... (Q380962)

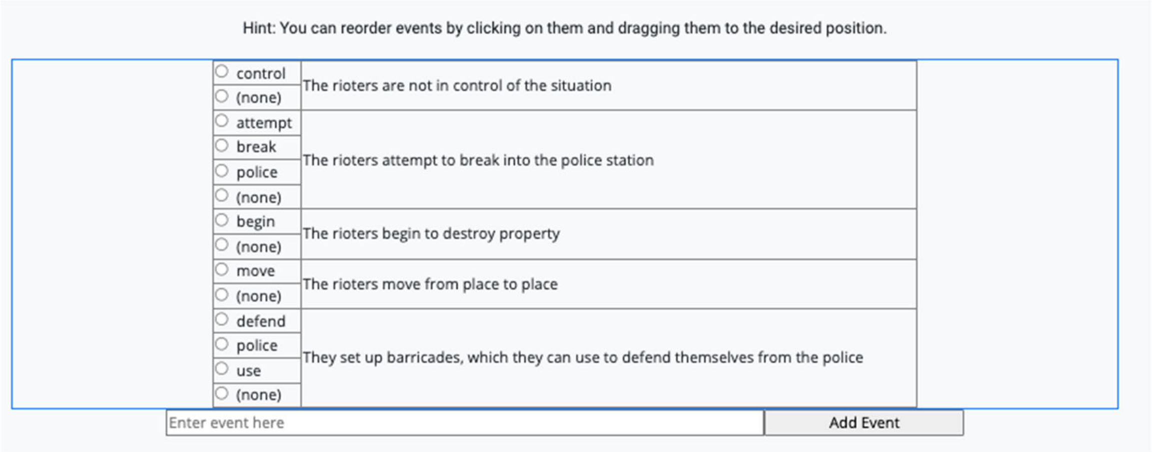
○ (Other)

Figure 18: Schema Description for Riot Schema

Step & Event Creation

The curation interface allows a curator to interact iteratively with suggestions for events to include in the schema. Suggestions are generated by prompting a generative model with current schema information (e.g. the schema name, the schema description and existing steps). Suggestions can be accepted, edited, or ignored. Curators build a set of relevant events using this interface. For the submitted schema library, we used a locally hosted instance of GPT-2 as the means of generation. Post-evaluation, MASC was extended to also enable direct calls to OpenAI's hosted instances of GPT-3.5 and 4.

Example suggestions appear in Figure 19. For some of the schemas in the library, as illustrated in Figure 19, MASC suggested a main event (e.g. *control* in the first row*)* in addition to the sentence-length description of the step (e.g. *The rioters are not in control of the situation)*. We introduced the event word suggestions in the final months before the evaluation and thus this feature influenced some, but not all schemas in the evaluation library. The intuition behind the feature was that the event selection could both improve QNode selection described below and assist the curators in focusing on introducing new, important events into a schema.

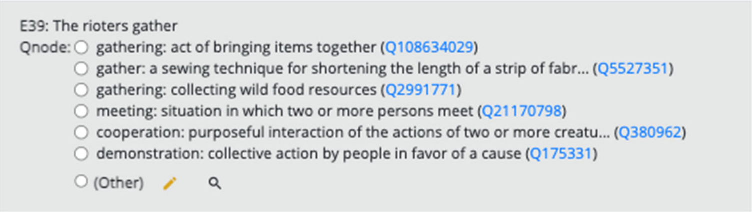


Hint: You can reorder events by clicking on them and dragging them to the desired position.

○ control ○ (none)	The rioters are not in control of the situation
○ attempt ○ break ○ police ○ (none)	The rioters attempt to break into the police station
○ begin ○ (none)	The rioters begin to destroy property
○ move ○ (none)	The rioters move from place to place
○ defend ○ police ○ use ○ (none)	They set up barricades, which they can use to defend themselves from the police

Figure 19: Suggestions of Steps for the Riot Complex Event

As events are added, the curator also selects a QNode for the event. QNode suggestions are derived via a combination of: (1) Identifying plausible events in the sentence using lexical processing (e.g. Part-of-Speech tagging, verb lists, nominalization lists); (2) Querying KGTK for plausible QNodes; (3) Reranking KGTK identified QNodes and common QNodes as idenfied by the core OWG subset using a fine-tuned version of DBERTa. As illustrated in Figure 20, the top-6 suggestions are presented to the curator as options. If the curator believes none of the suggestions are valid, they can search DWD directly using the magnifying glass icon.



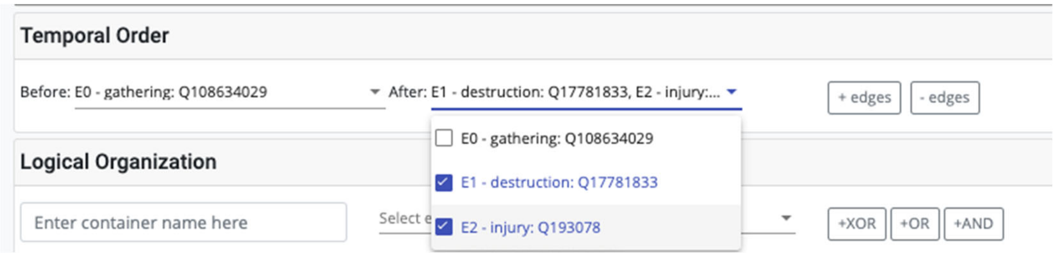
E39: The rioters gather

Qnode:
○ gathering: act of bringing items together (Q108634029)
○ gather: a sewing technique for shortening the length of a strip of fabr... (Q5527351)
○ gathering: collecting wild food resources (Q2991771)
○ meeting: situation in which two or more persons meet (Q21170798)
○ cooperation: purposeful interaction of the actions of two or more creatu... (Q380962)
○ demonstration: collective action by people in favor of a cause (Q175331)
○ (Other)

Figure 20: Suggested QNodes for the text "The Rioters Gather"

Temporal and Logical Organization

We explored several suggestion approaches to temporal and logical ordering throughout the program, but found that none were high enough precision to incorporate into the user interactions. Thus, temporal and logical ordering of events within a schema for evaluation processing is curator driven, using pull down menus with the curator (1) selecting groups of steps for which order was unknown and grouping them into named AND/OR/XOR gates (Figure 23) and (2) introducing temporal relationships between events/named logical gates (Figure 21, Figure 22). Figure 24 and Figure 25 illustrate how a curator sees the results of their decisions. The MASC interface requires a fully connected graph to save a schema, thus all nodes are connected through the combination of logical and temporal gates.


Temporal Order

Before: E0 - gathering: Q108634029

After: E1 - destruction: Q17781833, E2 - injury:...

☐ E0 - gathering: Q108634029

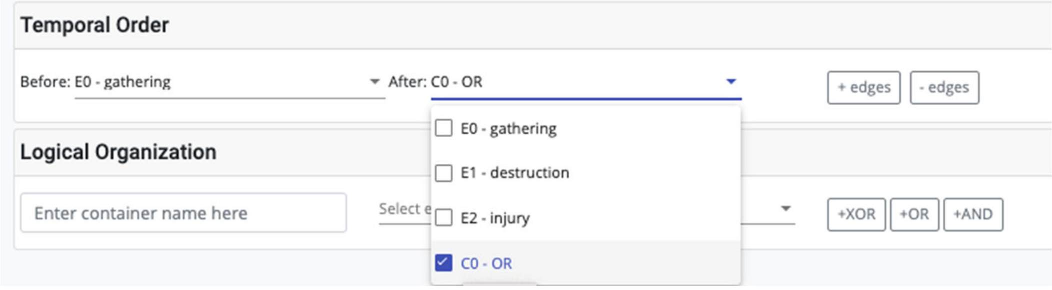
☑ E1 - destruction: Q17781833

☑ E2 - injury: Q193078

Logical Organization

Enter container name here

Figure 21: Adding Temporal Order without Logical Gates.


Temporal Order

Before: E0 - gathering

After: C0 - OR

☐ E0 - gathering

☐ E1 - destruction

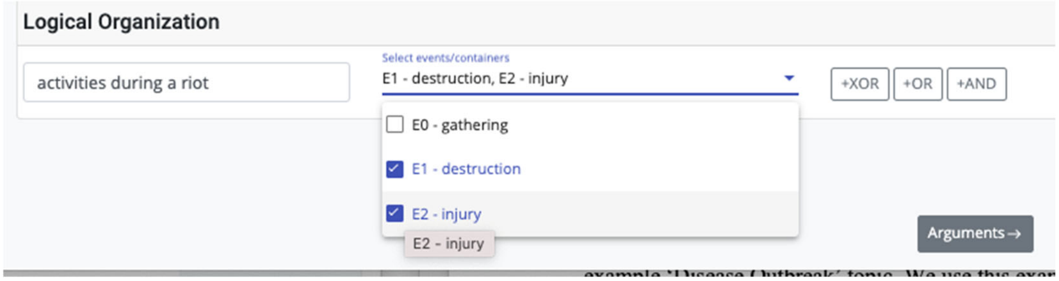
☐ E2 - injury

☑ C0 - OR

Logical Organization

Enter container name here

Figure 22: Adding Temporal Order Incorporating Logical Gates


Logical Organization

activities during a riot

Select events/containers: E1 - destruction, E2 - injury

☐ E0 - gathering

☑ E1 - destruction

☑ E2 - injury

Figure 23: Adding a Named Logical Gate

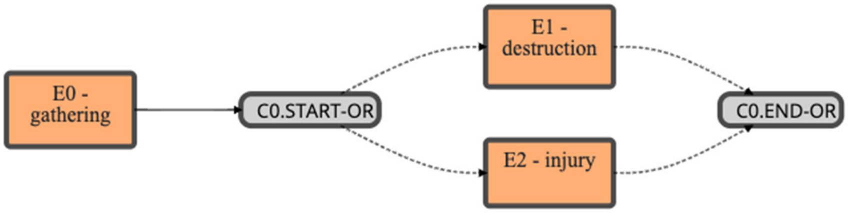
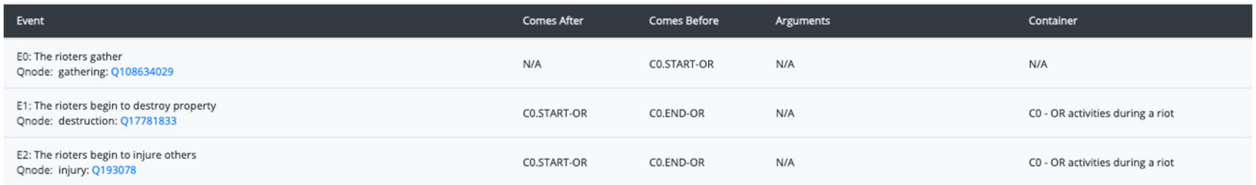


Figure 24: Graph Structure that Illustrates a Temporal Relation Between an OR Gate

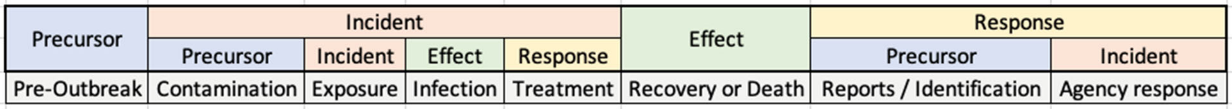


Event	Comes After	Comes Before	Arguments	Container
E0: The rioters gather Qnode: gathering: Q108634029	N/A	C0.START-OR	N/A	N/A
E1: The rioters begin to destroy property Qnode: destruction: Q17781833	C0.START-OR	C0.END-OR	N/A	C0 - OR activities during a riot
E2: The rioters begin to injure others Qnode: injury: Q193078	C0.START-OR	C0.END-OR	N/A	C0 - OR activities during a riot

Figure 25: Tabular View of Graph Structure with Columns for Temporal Information (Comes Before, Comes After) and the Named Logical Gates (Container)

To introduce an additional level of hierarchy, for the final evaluation we performed a suggestion driven approach to grouping steps within an event into Precursor, Incident, Effect, and Response (PIER). Each of these becomes a logical OR representing a stage within the larger complex events. For the submitted system, curators reviewed and corrected GPT-3's assignments of step to stage. The curators varied in whether the suggestion capability was preferred over simply performing the organization themselves.

Figure 12 illustrates the four named stages Precursor, Incident, Effect, and Response on an example 'Disease Outbreak' topic. We use this example to illustrate the definitions of PIER's various stages, the benefits of this less-linear structure, and the challenges herein. Additionally, while PIER presents the Effect stage occurring prior to the response stage, these two often overlap in practice.

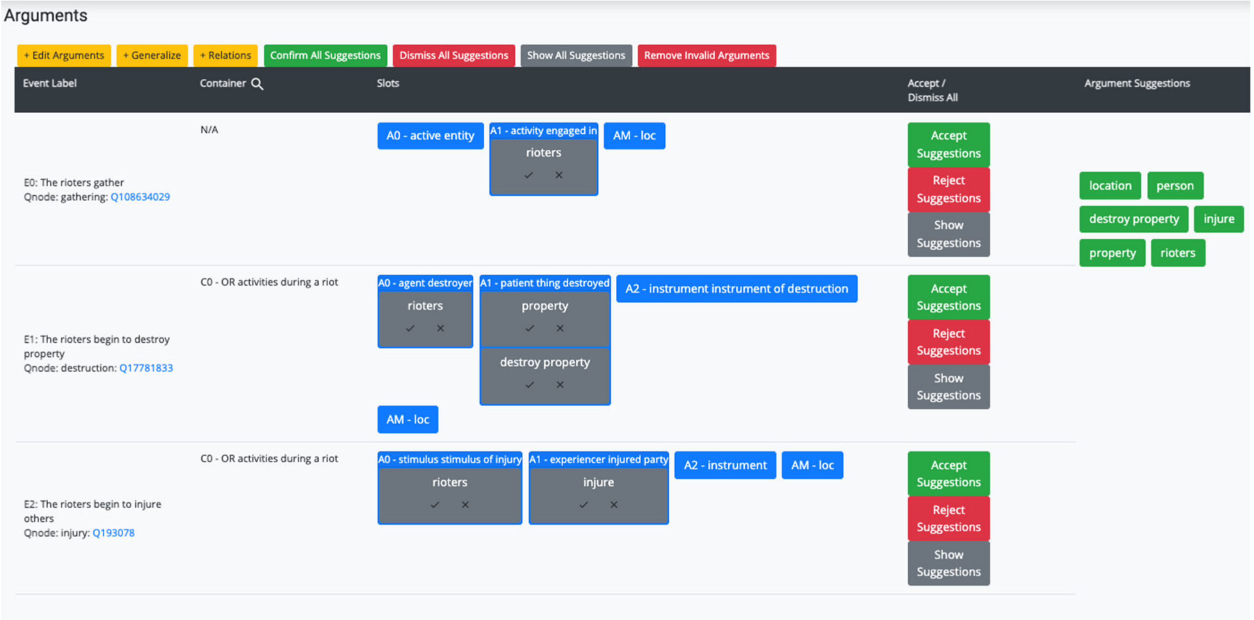


Precursor	Incident				Effect	Response	
	Precursor	Incident	Effect	Response		Precursor	Incident
Pre-Outbreak	Contamination	Exposure	Infection	Treatment	Recovery or Death	Reports / Identification	Agency response

Figure 26: PIER example broken out by anchored stage on a 'Disease Outbreak' topic. The top-Incident stage contains an internal breakdown of PIER, demonstrating hierarchical reuse as a component of PIER layouts. The top-Response stage also demonstrates this hierarchical reuse utilizing only two of the PIER stages.

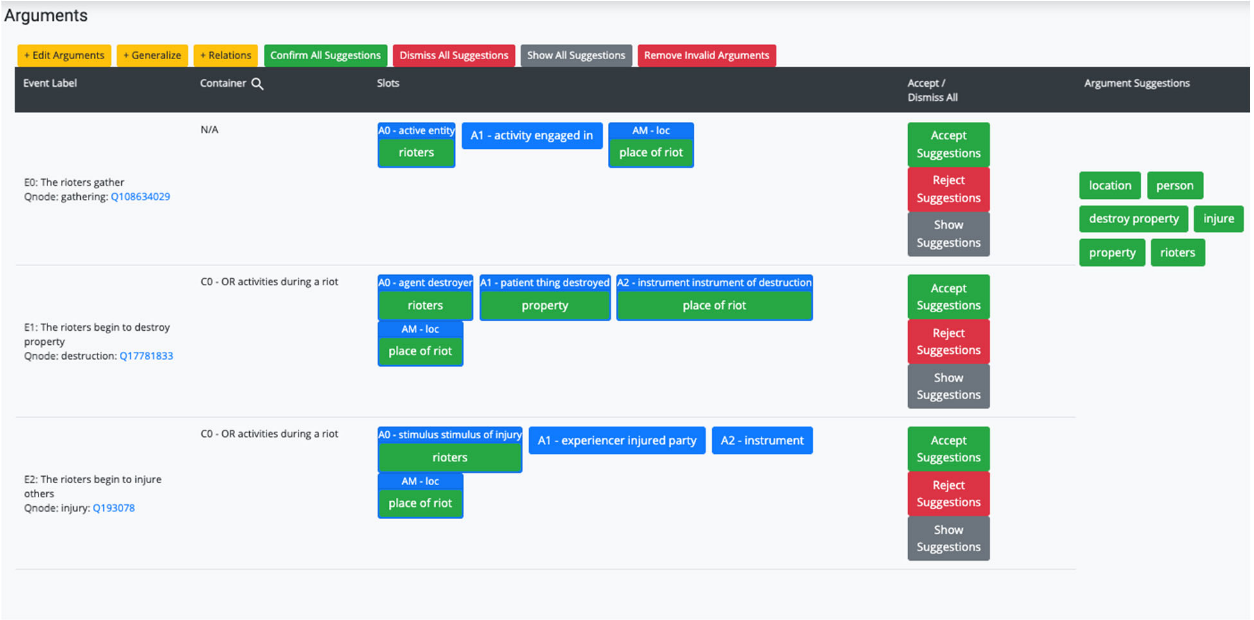
Arguments

The final required human-machine collaboration step in schema creation is the addition of arguments. Our approach to argument suggestion varied over the course of the program (and thus within the submitted schema library). The per-step argument slots (e.g. in the figures below *A0-active entity*) are determined by the Event QNode's alignment to established predicate roles using the OWG alignment. Initially, MASC suggested arguments for the schema as a whole and allowed the curator to drag these arguments to appropriate roles. This capability persists in the current version as the green boxes on the right of Figure 27 and Figure 28. In the final months before the evaluation, we extended the argument suggestion capability using semantic role labeling, and suggested arguments directly at the step level as in Figure 27. Curators have the option to select or reject argument suggestions. The new capability increases the number of arguments a curator is nudged to include, but for larger schemas introduces a greater need to incorporate coreference across steps. Figure 28 illustrates the result of curation to incorporate arguments. As illustrated here, the curator has the option of adding a new argument.



Event Label	Container	Slots	Accept / Dismiss All	Argument Suggestions
E0: The rioters gather Qnode: gathering: Q108634029	N/A	A0 - active entity A1 - activity engaged in: rioters ✓ ✕ AM - loc	Accept Suggestions Reject Suggestions Show Suggestions	location person destroy property injure property rioters
E1: The rioters begin to destroy property Qnode: destruction: Q17781833	C0 - OR activities during a riot	A0 - agent destroyer: rioters ✓ ✕ A1 - patient thing destroyed: property ✓ ✕; destroy property ✓ ✕ A2 - instrument instrument of destruction AM - loc	Accept Suggestions Reject Suggestions Show Suggestions	
E2: The rioters begin to injure others Qnode: injury: Q193078	C0 - OR activities during a riot	A0 - stimulus stimulus of injury: rioters ✓ ✕ A1 - experiencer injured party: injure ✓ ✕ A2 - instrument AM - loc	Accept Suggestions Reject Suggestions Show Suggestions	

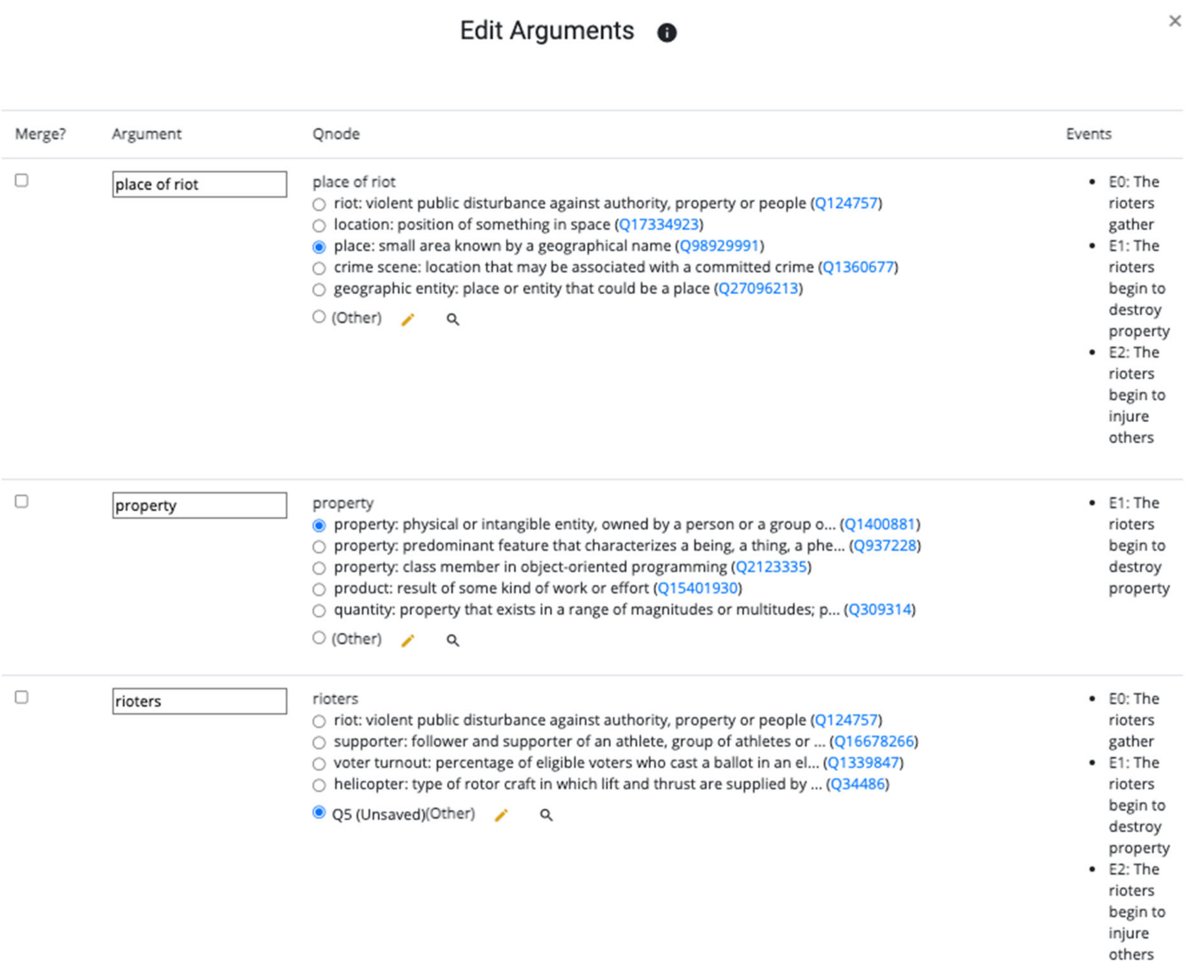
Figure 27: Arguments as Suggested



Event Label	Container	Slots	Accept / Dismiss All	Argument Suggestions
E0: The rioters gather Qnode: gathering: Q108634029	N/A	A0 - active entity: rioters A1 - activity engaged in AM - loc: place of riot	Accept Suggestions Reject Suggestions Show Suggestions	location person destroy property injure property rioters
E1: The rioters begin to destroy property Qnode: destruction: Q17781833	C0 - OR activities during a riot	A0 - agent destroyer: rioters A1 - patient thing destroyed: property A2 - instrument instrument of destruction: place of riot AM - loc: place of riot	Accept Suggestions Reject Suggestions Show Suggestions	
E2: The rioters begin to injure others Qnode: injury: Q193078	C0 - OR activities during a riot	A0 - stimulus stimulus of injury: rioters A1 - experiencer injured party A2 - instrument AM - loc: place of riot	Accept Suggestions Reject Suggestions Show Suggestions	

Figure 28: Selected Arguments

In addition to providing argument labels, each argument requires a QNode. QNode suggestion uses similar approaches to those described for Event QNode suggestion. Figure 29 illustrates Argument QNode suggestions.

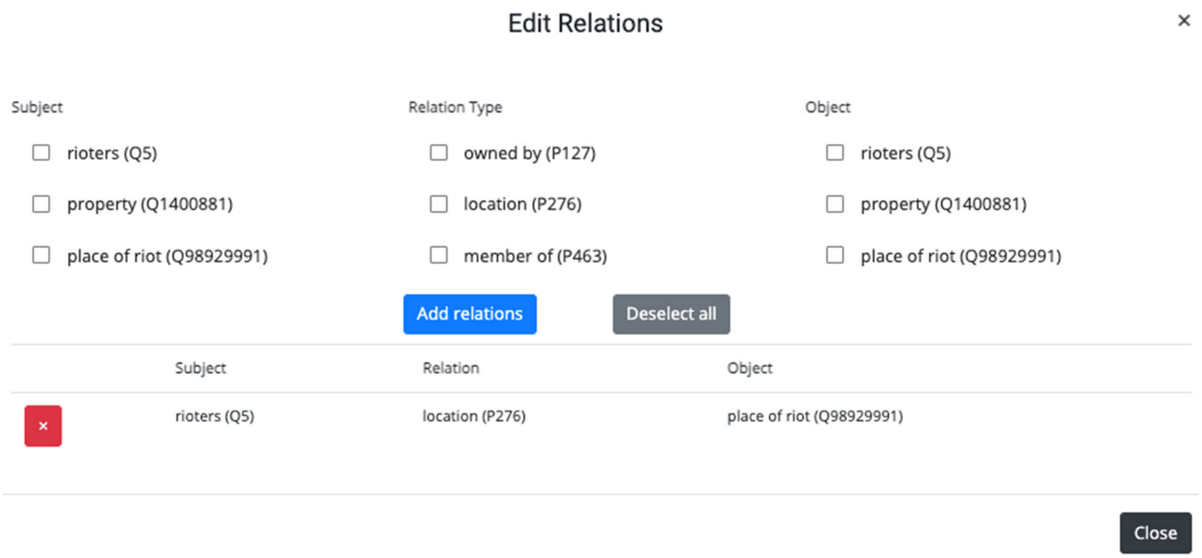


Merge?	Argument	Qnode	Events
☐	place of riot	place of riot ○ riot: violent public disturbance against authority, property or people (Q124757) ○ location: position of something in space (Q17334923) ● place: small area known by a geographical name (Q98929991) ○ crime scene: location that may be associated with a committed crime (Q1360677) ○ geographic entity: place or entity that could be a place (Q27096213) ○ (Other)	- E0: The rioters gather - E1: The rioters begin to destroy property - E2: The rioters begin to injure others
☐	property	property ● property: physical or intangible entity, owned by a person or a group o... (Q1400881) ○ property: predominant feature that characterizes a being, a thing, a phe... (Q937228) ○ property: class member in object-oriented programming (Q2123335) ○ product: result of some kind of work or effort (Q15401930) ○ quantity: property that exists in a range of magnitudes or multitudes; p... (Q309314) ○ (Other)	- E1: The rioters begin to destroy property
☐	rioters	rioters ○ riot: violent public disturbance against authority, property or people (Q124757) ○ supporter: follower and supporter of an athlete, group of athletes or ... (Q16678266) ○ voter turnout: percentage of eligible voters who cast a ballot in an el... (Q1339847) ○ helicopter: type of rotor craft in which lift and thrust are supplied by ... (Q34486) ● Q5 (Unsaved)(Other)	- E0: The rioters gather - E1: The rioters begin to destroy property - E2: The rioters begin to injure others

Figure 29: QNode Suggestions for the Selected Arguments

Argument Relations

As an additional feature of our schemas, we automatically incorporate automatically predicted relations between arguments and allow the curator to view and correct. We performed an analysis of potential arguments from the OWG overlay and identified ownership, location, and membership as being sufficiently frequent and characteristic of the schema in general (rather than an instance of the schema) to incorporate. We use a fine-tuned entailment base to automatically determine whether an argument pair is likely to hold some relation given the context of the schema. This approach leverages the simple inferences performed in a natural language entailment/inference case.



Subject

☐ rioters (Q5)
☐ property (Q1400881)
☐ place of riot (Q98929991)

Relation Type

☐ owned by (P127)
☐ location (P276)
☐ member of (P463)

Object

☐ rioters (Q5)
☐ property (Q1400881)
☐ place of riot (Q98929991)

	Subject	Relation	Object
×	rioters (Q5)	location (P276)	place of riot (Q98929991)

Figure 30: Argument Relation Suggestions

LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS

Definition / Term	Description
ASU	Arizona State University, a subcontractor on this effort.
BERT	Bidirectional Encoder Representations from Transformers
ChatGPT	Chat-based LLM hosted by OpenAI.
DWD	DARPA Wikidata
GPT-*	Specific generative LLMs hosted by OpenAI. A version number is specified when used rather than *. This report references versions 2, 3, 3.5, and 4.
Graph G	Graph G was created by the Linguistic Data Consortium to measure TA2 automatic capability to predict events not observed while factoring out errors of machine understanding of documents.
HRF	Human Readable Format, a format of schema presentation intended to be more interpretable by humans, rather than the format more appropriate for TA2 systems.
ISI	Information Sciences Institute
JSON	JavaScript Object Notation
KAIROS	Knowledge-directed Artificial Intelligence Reasoning Over Schemas, the DARPA program that included this effort.
KGTK	Knowledge Graph ToolKit
LESTAT	Learning Event Schema Temporally and Transmodally, the name of USC ISI's effort.
LDC	Linguistic Data Consortium at the University of Pennsylvania, which provided data for the KAIROS program.
LLM	Large Language Model
MASC	Machine-Aided Script Curator, the script-creation tool developed in this effort.
NIST	National Institute of Standards and Technology, the third party evaluator of KAIROS results.

NLP	Natural Language Processing
OWG	Ontology Working Group
PIER	Precursor. Incident. Effect. Response. An order of defined temporal stages for schema organization.
qNode	An identifier representing a concept in Wikidata.
RESIN	UIUC's engine that predicts likely events based on a schema repository and streaming input.
SBERT	Sentence-BERT, a modification to BERT to improve performance on semantic textual similarity tasks.
SDF	Schema Data Format, a JSON-based format for sharing schema libraries among performers.
SpaCy	An open-source NLP toolkit (https://spacy.io/)
SRL	Semantic role labeling, a technique that identifies roles that participants play in an event.
TA1	Technical Area 1, whose goal is to produce a human-curated library of schemas for complex events.
TA2	Technical Area 2, whose goals are to make predictions of events that are likely but have not been reported, given a stream of documents.
TEAR	Temporal Events and Relations, a system we developed for extracting events, their arguments, and temporal relations among them.
UCF	University of Central Florida, a subcontractor on this effort.
UCLA	University of California, Los Angeles, a subcontractor on this effort.
UIUC	University of Illinois at Urbana-Champaign
USC	University of Southern California
YAML	Yet Another Markup Language