AFRL-RI-RS-TR-2021-205

DECOMPOSITIONAL SCHEMA INDUCTION AND PREDICTION

JOHNS HOPKINS UNIVERSITY

DECEMBER 2021

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

AFRL-RI-RS-TR-2021-205 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /
ROSANNE L. PELLI
Work Unit Manager

/ S /
SCOTT D. PATRICK
Deputy Chief
Intelligence Systems Division
Information Directorate

REPORT DOCUMENTATION PAGE

1. REPORT DATE	Report Type	3. DATES COVERED START DATE	END DATE
DECEMBER 2021	FINAL TECHNICAL REPORT	AUGUST 2019	JULY 2021

4. TITLE AND SUBTITLE
DECOMPOSITIONAL SCHEMA INDUCTION AND PREDICTION

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
FA8750-19-2-0034	N/A	62303E
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**
KIRS	00	03

6. AUTHOR(S)
Benjamin Van Durme and Aaron Steven White

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Johns Hopkins University
3400 N Charles Street
Baltimore MD 21218-2608

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)	11. SPONSOR/MONITOR'S REPORT NUMBER(S)
Air Force Research Laboratory/RIEA 525 Brooks Road Rome NY 13441-4505	RI	AFRL-RI-RS-TR-2021-205

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 was developed under DARPA KAIROS. The report summarizes work on the induction and use of schemas.

15. SUBJECT TERMS
Schemas, scripts, causality, association rule mining

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	C. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
U	U	U	SAR	49

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
ROSANNE L. PELLI	**N/A**

 PREVIOUS EDITION IS OBSOLETE. **STANDARD FORM 298 (REV. 5/2020)**
Prescribed by ANSI Std. Z39.18

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1.0 SUMMARY

The Johns Hopkins University was a TA1+TA2 performer team under the Defense Advanced Research Projects (DARPA) Knowledge-directed Artificial Intelligence Reasoning Over Schemas (KAIROS) Program. Senior members included Benjamin Van Durme (PI), Kyle Rawlins (coPI), Craig Harman, Mahsa Yarmohammadi, Joao Sedoc, Aaron White (coPI, University of Rochester), and Rachel Rudinger (University of Maryland).

Under TA1 (Schema Induction) our primary contributions included: a model for inducing schema information from corpora rooted in causal statistics; an approach to incrementally building up schemas through recognizing schema induction as an instance of Association Rule Mining; a novel user interface for the rapid curation of schemas with a human in the loop; and a built out workflow that combined these contributions into a practical framework for building schema libraries. This workflow resulted in extremely high quality schemas that were: (a) built out efficiently by hand, in the case of transforming descriptions from the Linguistic Data Consortium (LDC) into fully structured schema representations; (b) induced as partial structures that were then manually further instantiated. These hundreds of schemas are to our knowledge the highest quality schema library that emerged from KAIROS Phase 1, and were constructed entirely following the constraints provided by the program.

Under TA2 (Schema Inference) our primary contributions included: a decompositional analysis of temporal relations in language, with an associated corpus representing the largest amount of such annotations in the world; and a scalable framework for schema inference based on a combination of efficient indexing via Lucene and the reasoning capabilities of Probabilistic Soft Logic. To our knowledge, our inference framework in Phase 1 KAIROS best captured the constraints made explicit in the program, recognizing schemas as discrete objects with no underlying semantics beyond what was strictly defined in a shared program ontology, and making no assumptions on a shared content understanding toolchain used in the creation of the schemas and later use in inference.

In addition to our primary contributions that were guided by program evaluation requirements, we contributed to the scientific literature with a number of ideas salient to understanding and inferring events described in documents. These included: a series of advances in controllable text generation, with an experimental focus on the generation of causal statements; the development of a task we called "statutory (legal) reasoning"; successfully determining that large pre-trained language models contain a significant amount of human background knowledge; a method for rapidly augmenting annotated information extraction datasets; and finally, in collaboration with the Johns Hopkins University Bloomberg School of Public Health, we developed and launched online a software agent that was able to answer frequently asked questions about COVID during the onset of the global pandemic.

2.0 INTRODUCTION

The Johns Hopkins University was a TA1+TA2 performer team under the DARPA KAIROS Program. Senior members included Benjamin Van Durme (PI), Kyle Rawlins (coPI), Craig Harman, Mahsa Yarmohammadi, Joao Sedoc, Aaron White (coPI, University of Rochester), and Rachel Rudinger (University of Maryland).

As summarized in Sec. 3, we made contributions motivated by the shared program evaluations, in schema induction and curation, and schema inference. Other salient research was performed in the broader space of how to understand and reason with language content at the document level. Our contributions also included the following, which we cover in further detail in subsequent sections of this report.

1. **Causal Schema Induction –** a model for schema induction that induces *causal* connections between events, rather than the broader set of merely *correlated* ones. We illustrated that the mathematics of causal model induction better captures human plausible schemas than prior work based on Pointwise Mutual Information or Language Modeling.
2. **Schema Induction as Association Rule Mining –** Association Rule Mining was a popular topic in the 1990s in industry and academia, being the core pursuit of the emergent science of data mining. We drew connections from that line of algorithms to the strategies employed in contemporary work in schema induction.
3. **SchemaBlocks:** a visual programming interface for schema curation, allowing for a Human-in-the-loop workflow for a non-technical human, such as a crowdsource worker, to graphically refine automatically induced schemas.
4. **Schema Library Evaluation:** We proposed evaluation methods for evaluating the schema resources, and evaluated our schemas against three datasets, including against Chinese and Russian resources.
5. **Temporal Relation Extraction:** We developed a new model and dataset for extracting document-level complex event structure, including temporal relations and partial event coreference.
6. **Covid-19 Infobot:** In response to Covid-19 we directed effort to employ technologies from this and previous DARPA programs to deploy a live dialogue agent online that answered frequently asked questions about the virus, based on a collaboration with researchers in the Bloomberg School of Public Health.
7. **Paraphrastically Augmented FrameNet Parsing:** We demonstrated a process for iteratively paraphrasing with a human in the loop that led to rapid bootstrapping of event semantic resources.
8. **Tacit Assumptions in Language Models:** We demonstrated that out of the box contemporary contextual encoders contain *stereotypic tacit assumptions*, a.k.a. common sense.
9. **Diverse Causal Generation:** We constructed a resource of 300million English sentences that match basic causal patterns, e.g., “X because of Y”, and used this to develop multiple models that can take an input sentence and produce a series of sentences that represent possible causes and effects.
10. **Reasoning with Rules:** We developed a corpus that pairs brief descriptions of legal cases with a manually curated and simplified collection of tax laws, then manually encoded these in Prolog. Our experiments illustrate the difficulty in performing complex inference (legalistic/procedural reasoning) purely atop contemporary contextual encoders.

3.0 METHODS, ASSUMPTIONS, AND PROCEDURES

3.1 Causal Schema Induction

When does a sequence of events define an everyday scenario and how can this knowledge be induced from text? Prior works in inducing such *scripts* (schemas) have relied on, in one form or another, measures of correlation between instances of events in a corpus. We argued from both a conceptual and practical sense that a purely correlation-based approach is insufficient, and instead proposed an approach to script induction based on the causal effect between events, formally defined via interventions. Through both human and automatic evaluations, we showed that the output of our method based on causal effects better matches the intuition of what a script represents.

In work prior to KAIROS, we illustrated that Pointwise Mutual Information (PMI) led to results people found qualitatively interesting, but were poor models for performing fill-in-the-blank style "narrative cloze" inference. PMI leads to finding correlations that "makes sense" to people looking at them, but as an approach it focuses on correlations that may be very rare in practice: picking up on strong correlations between events, but being less concerned about the likelihood a given event may actually happen. PMI is not likely a good approach for a KAIROS-themed effort. We also previously illustrated that standard language modeling could be applied as a schema induction technique and that such models did quite well in fill-in-the-blank style tasks. This was to be expected, given this is the objective language models are trained for, but this observation had not been previously made. Unfortunately, language models are so focused on what is likely, that they may not pick up on what is most interesting: when asking a language model what is most likely to happen after someone is arrested it will predict that the person arrested is likely to say something. While true, this is not interesting, and misses the forest for the trees: when one is inducing schemas from document collections, we are not actually trying to memorize the most likely events to occur in a document, we are trying to ascertain underlying event structures in the world that help explain common event-event co-occurrences (as compared to common verb-verb co-occurrences in a document).

We therefore developed a model that posits potential latent events in a document that may help best describe what is explicitly reported. This causal model has the ability to pick up strong event-event correlations that are explicitly attested to, while also benefitting from base frequency in a corpus, as does a traditional language model. The following figure illustrates our notion of the "universe" giving rise to events over time, which may or may not then be mentioned in a document.

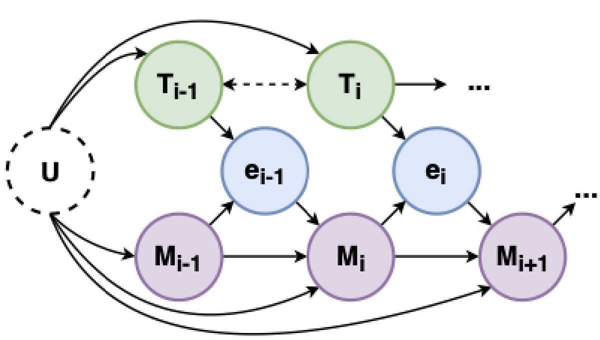

Figure 1: A universe unrolling sequentially over time, with events that sometimes are mentioned in text.

Through human evaluations we found that our proposed schemas were more plausible than Language Models, while also being more predictive than PMI. Some examples from the model are below, where we start with a target event (right hand side) and then ask each model which event most likely precedes it. These results indicate this approach as superior to prior approaches for inducing initial schema suggestions, which will then under KAIROS be modified by a human in the loop under TA1.

Causal	LM	PMI	Target
X tripped	*X came*	*X featured*	*X fell*
X lit	*X sat*	*X laboured*	*X inhaled*
X aimed	*X came*	*X alarmed*	*X fired*
X poured	*X nodded*	*X credited*	*X refilled*
X radioed	*X made*	*X fostered*	*X ordered*

Figure 2: Predicting what likely led to the target event, via three approaches.

3.2 Schema Induction as Association Rule Mining

In this project, we reduced the Schema Induction problem to Association Rule Mining (ARM), a time-tested count-based technique for finding frequently occurring groups of events. We show that the traditional PMI-based Script Induction used in initial schema induction approaches is a special case of ARM over rules of size 2. We generalized our findings to non-binary rules, showing a simple linear time inference algorithm. Finally, we conducted extrinsic and intrinsic evaluation of schemas induced by our approach and compare them to PMI-induced schemas.

This effort closed a hole in the literature: methods for schema induction via PMI can be viewed as ARM applied to database entries extracted from documents by parsing and coreference analytics. The approach was later combined with the causal model of 5.1, along with the SchemaBlocks workflow of 5.3 to induce large schema libraries, described next. Further details of this study can be found in Belyy and Van Durme (2020).

3.3 SchemaBlocks

We developed an interface to allow experts to easily construct schemas, which we call SchemaBlocks. SchemaBlocks is a Web-based tool that provides a way to display and modify the contents of a schema by representing its units – events and arguments, entity relations and types – as blocks, that can be stacked and nested. In addition to capturing schema events, participants, and their relations, the interface also enables the representation of entity coreference, event ordering, and mutually exclusive events. The following is an example schema as defined by SchemaBlocks.

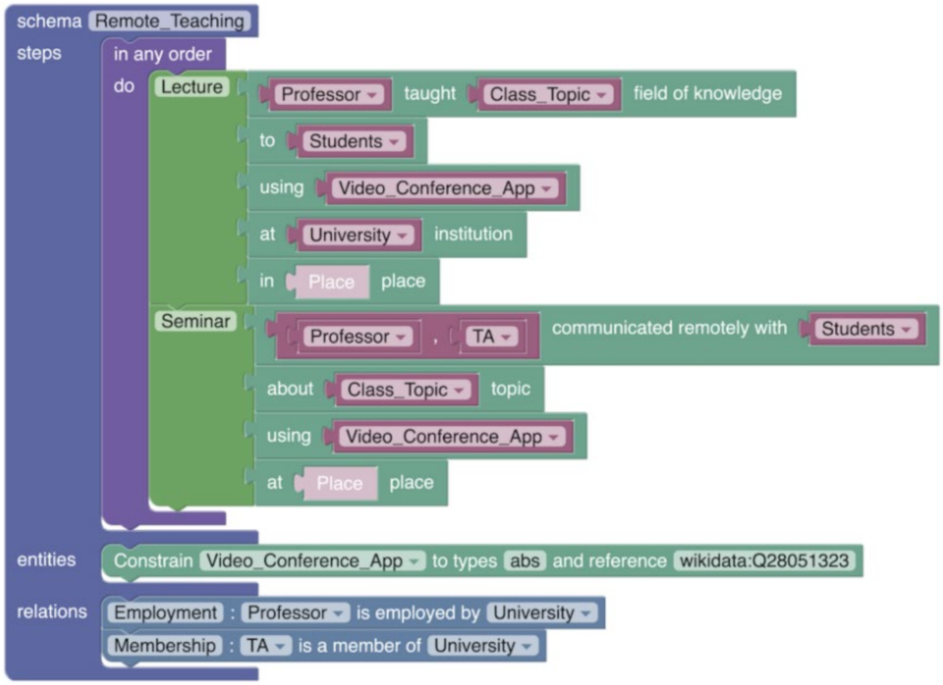

Figure 3: The SchemaBlocks interface.

Before annotating with SchemaBlocks, an annotator needs to familiarize themselves with the schema ontology we have defined, which establishes the vocabulary of blocks they can use to build schemas. In the interface, this is displayed as a dashboard, organized hierarchically for convenience. The following image shows as an example of all levels of the ontology hierarchy for the "Medical" event category of the KAIROS ontology adapted into the SchemaBlocks interface.

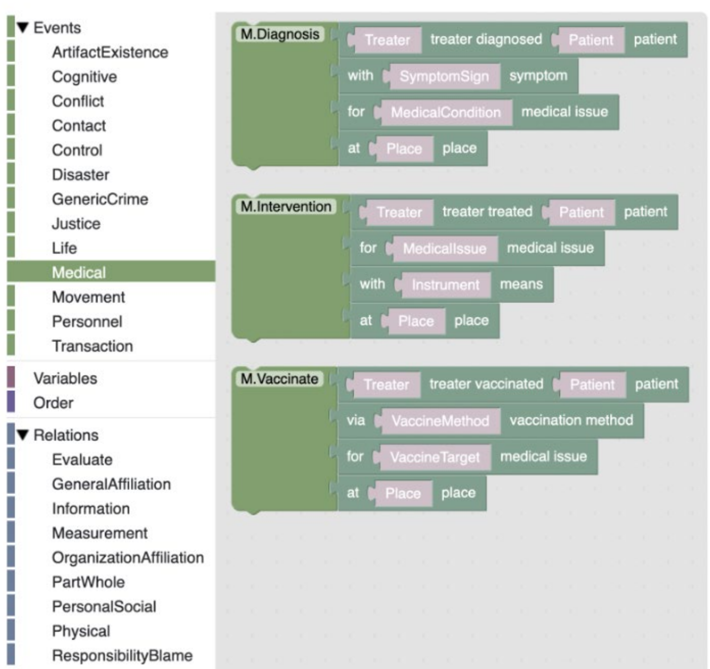

Figure 4: A schema from KAIROS.

The block interface here is flexible and could be adapted to a similar event ontology, such as FrameNet (Baker et al., 1998), ACE (Doddington et al., 2004) and ERE (Song et al., 2015). The repository with the SchemaBlocks user interface can be accessed at this link: https://github.com/AVBelyy/SchemaBlocks

SchemaBlocks is primarily based on the Google Blockly library. On top of the User Interface (UI) primitives provided by Blockly, we implement ontology-to-blocks and blocks-to-JSON converters to make user-created schemas usable in downstream tasks. We also ensure the schema entities have correct types by implementing continuous type checking and type inference in the UI. If a user breaks entity type constraints specified by the ontology, they are notified and the relevant entity blocks are highlighted until the error is fixed. Our choice of block-based representation is inspired by Scratch (Resnick et al., 2009), a prominent tool that engages children to learn the basics of programming. By allowing annotators to program schemas using ontology-specific blocks, as opposed to general-purpose text formats such as JSON or XML, we are also able to engage more experts with non-programming backgrounds and annotate schemas at a faster rate. The annotators in our study (undergraduate students with non-CS majors) found the interface easy-to-use and left overall positive feedback. To familiarize annotators with the interface, we provided them with a guide prior to running the annotation.

We used this interface as part of curating a library of schemas. We built a collection of schemas by enlisting several undergraduate annotators who we train to construct schemas using the SchemaBlocks interface. Though in principle an annotator may use the SchemaBlocks interface to build schemas entirely from scratch, in practice we provide for each schema some form of seed information for the annotator to build the schema off of. In particular, we have annotators use the SchemaBlocks interface to "flesh out" schemas from the following two resources:

- Linguistic Data Consortium (LDC) provided textual descriptions of 83 schemas
- Automatically induced “skeleton schemas”

LDC Complex Event Schemas -- In the first annotation round, annotators were provided with 82 textual descriptions of schemas from an LDC developed resource. The resource contains textual definitions for 82 schemas, or complex events. Each schema in the resource is given a title, a 2-3 sentence long description, specifications of the scope of the complex event (i.e. when and where the complex event should be considered initiated or finished), and the series of steps that defines the complex event/schema. Each step is defined with a title that specifies the event type of the step (in natural text, no event ontology is used), a short one-sentence description, and expected high level event types that may happen as subevents. The annotators are then tasked with “translating” these textual descriptions of schemas into a machine readable form via our SchemaBlocks interface. Relations and entity types are not specified in the textual descriptions, so annotators are instructed to annotate for relations that must be true throughout all steps of the schemas, as well as provide specific entity types and links to Wikidata. Annotators reported an average time of 30 minutes per schema to fully annotate, with 82 schemas being the product of this annotation task. The number of events in each of 82 schemas ranges from 2 to 10, with 6 being the median number of events.

Automatic Induction of Skelton Schemas -- In the second annotation round, annotators were tasked with fleshing out structures we call “skeleton schemas” which have been automatically mined from text corpora. (More specifically, skeleton schemas are argumentless event sequences which form an outline of a potential event schema.) Because the skeleton schemas are automatically induced from selected text corpora, we expect the resulting schemas to be relevant to the domain of the text from which they are mined; in this way, we allow the data to “speak for itself” with regards to what kinds of topics and scenarios we are targeting in our annotations.

The automatic system for skeleton schema induction combines two recent advances in schema induction: (1) an Association Rule Mining based algorithm presented in Belyy and Van Durme (2020) and described in Sec. 3.5.2, which efficiently finds all event subsequences which have enough support in the data, and (2) a script compatibility scoring model presented in Weber et al. (2020) which finds high quality subsequences output by the Association Rule Mining method and combines them together to form full skeleton schemas.

Given a skeleton schema, we import it into SchemaBlocks as a partially filled out schema of which only its events have been specified. We then present these partially filled out schemas to annotators and task them with the following:

- Determining what scenario the partially filled out schema is describing. This includes determining a name for the schema, as well as a brief textual description on what it is about.
- Determining what entities fill the slots of the given events in the schema, what types (coarse and fine-grained) they take on, and which slots are filled with co-referring entities.
- Determining what relations hold between the above defined entities. The criteria for annotating relations here is the same as before.

Given this annotation is designed to be similar to the LDC-based annotations, annotators who participated in the first annotation effort (all of them) required little extra training to complete this annotation, only a single one-hour training session. Again, annotators reported around a 30 minute average to annotate a schema. The end result of this fleshing out process is an additional 150 schemas. The number of events in this additional set of schemas ranges from 3 to 6, with 4 being the median number of events.

3.4 Schema Library Evaluation

To evaluate our schema library, we devised a measure that evaluates the match between a set of schemas and a corpus. We proposed a **dataset coverage** measure to evaluate such a match, given no additional labels. When the documents are provided along with schemas that apply to them, we additionally compute the **ranking** measures, which treat the matching between a document and a schema as a retrieval task where both can participate either as a query, or as an indexed document. To quantify what constitutes a match between documents and schemas, we define a similarity function $\mathrm{sim}(d, s) = |d \cap s| / |d|$ which counts how many events in a document d are matched by a schema s. For the purposes of this metric, we treat documents as multisets and schemas as sets of events. Thus, if a document d = {LIFE.INFECT : 2, MEDICAL.VACCINATE : 1} is matched with a schema s = {LIFE.INFECT, LIFE.DIE}, the similarity will be sim(d, s) = 2/3. We bucket the results by the number of identified events N_{events} in each document. We compute the final metrics using bootstrap over 100 samples and report the mean performance. To extract events from text documents, we use the FrameNet parser from the LOME IE system (Xia et al., 2021), which identifies FrameNet events and their arguments. We map the events to KAIROS ontology using a rule-based mapping.

We use the following three datasets in our evaluation:

- **LDC corpus** Under the DARPA KAIROS program, LDC has annotated 924 multilingual multimedia documents (covering images, audio, video, and text in English and Spanish) with KAIROS event types and a complex event (CE) label. The CE label indicates the complex event (from LDC2020E25) that best applies to a document. Each CE label is covered by 11 documents on average, each document having one CE label. Out of 924 documents, 921 have partial event annotations (event type, link to a complex event step, and a provenance link to a span/offset in a document) and 36 have complete annotations (with identified and provenance linked entities and relations). Given the sparsity of the latter, we opted to only use event type annotations in order to compute ranking-based metrics.
- **Gigaword** We pick (uniformly at random) a subset of 100K documents from the NY-Times portion of the Fifth Edition of the English Gigaword (Graff et al., 2003) corpus, spanning the New York Times news articles from years 1994–2010. This corpus is typically employed in the script induction literature. We use it to compute dataset coverage of our schemas over a large newswire corpus.
- **CC-News** Additionally, we employ the CCNews corpus (Nagel, 2016) which provides a wide array of news articles over multiple languages. We pick a random subset of 100K English-language articles from years 2016–2017. To evaluate crosslingual abilities of schemas, we also pick 100K news documents in Chinese and Russian from the same corpus, covering years 2016–2019. We use the cld3 library along with the "meta lang" field from the news source for language ID. The collection is de-duplicated in the following

way: first we run the clustering over all documents in a particular language. If an article is in a cluster with more than 20 documents we remove all documents in that cluster, otherwise we select one representative. Both Gigaword and CC-News are tokenized using spaCy v2.3.5 (Honnibal et al., 2020).

We devised three evaluation metrics: a **schema ranking**, **document ranking**, and **dataset coverage**.

Schema Ranking

Here we asked: how well can we predict the true Complex Event (CE) using the match between schema and document events as a ranking function? To answer this, we used documents from the LDC corpus, where each document d has precisely one CE label in that dataset. For each document d, we rank schemas according to sim(d, s) and report the average rank (lower score is better), Mean Reciprocal Rank (MRR, higher score is better), and Recall@10 (R@10, higher score is better) of the gold CE label. The following table shows the results of this evaluation against the 82 schemas on the LDC corpus using gold events:

Table 1: Ability to predict Complex Events, when we have gold event annotations.

	Schema ranking			Document ranking		Dataset coverage		
N_{events}	Avg Rank	MRR	R@10	R@30	nDCG	Cov@0.5	Cov@0.7	Cov@0.9
$[1; 5)$	26.4	0.112	0.244	0.387	0.246	0.960	0.852	0.797
$[5; 10)$	23.8	0.147	0.340	0.472	0.276	0.937	0.785	0.614
$[10; \infty)$	20.8	0.194	0.410	0.545	0.269	0.925	0.759	0.533
$[1; \infty)$	21.1	0.191	0.404	0.442	0.272	0.925	0.761	0.542

The performance tends to improve with longer documents, as the documents' descriptions (in terms of identified events) become richer and match a particular schema more precisely. We note, however, that perfect performance is not expected here: there are many contributing factors to a correct prediction of a schema other than matching the set of events. If this task was solved perfectly, this would render complex schema-based inference systems unnecessary, which we are not trying to show. Instead, we argue for event match to be a useful first step to narrow down a set of candidates. We also compare gold event annotations with IE extracted events:

Table 2: Ability to predict Complex Events, when we have automatic event annotations.

	Schema ranking			Document ranking		Dataset coverage		
N_{events}	Avg Rank	MRR	R@10	R@30	nDCG	Cov@0.5	Cov@0.7	Cov@0.9
$[1; 5)$	35.4	0.072	0.199	0.293	0.162	0.895	0.576	0.491
$[5; 10)$.	32.1	0.088	0.193	0.347	0.170	0.833	0.502	0.334
$[10; \infty)$	30.6	0.105	0.229	0.411	0.247	0.759	0.417	0.242
$[1; \infty)$	30.2	0.109	0.239	0.351	0.240	0.745	0.400	0.223

Document Ranking

In a similar vein, we asked: how well do event types allow us to rank documents given a schema as a query? We report Recall@30 and normalized discounted cumulative gain (nDCG) of the gold annotated documents. Similar to schema ranking, the ranking of documents improves with longer documents as they become more descriptive of a true complex event. Document ranking is reported in the two tables above.

Dataset Coverage

Finally, we investigated the question of how many documents are explained, or "covered" by at least one schema. We say that a document d is "t-covered" by a schema s if sim(d, s) ≥ t and define "coverage at t", or Cov@t, as the ratio of documents in the dataset t-covered by at least one schema. This measure does not require any CE label annotations, so we compute it for the LDC corpus (the two tables above), and **Gigaword** and **English CC-News**, in the two following tables, respectively:

Table 3: How many documents in the LDC corpus are covered by our schemas.

	82 schemas			82 + 150 schemas		
N_{events}	Cov@0.5	Cov@0.7	Cov@0.9	Cov@0.5	Cov@0.7	Cov@0.9
$[1; 5)$	0.887	0.531	0.425	0.975 (+10%)	0.637 (+20%)	0.509 (+20%)
$[5; 10)$	0.791	0.391	0.233	0.892 (+13%)	0.496 (+27%)	0.278 (+19%)
$[10; \infty)$	0.695	0.313	0.164	0.807 (+16%)	0.379 (+21%)	0.195 (+19%)
$[1; \infty)$	0.684	0.303	0.154	0.798 (+17%)	0.367 (+21%)	0.183 (+19%)

Table 4: How many documents in the Gigaword + English CC-News corpus are covered by our schemas.

	82 schemas			82 + 150 schemas		
N_{events}	Cov@0.5	Cov@0.7	Cov@0.9	Cov@0.5	Cov@0.7	Cov@0.9
$[1; 5)$	0.874	0.588	0.529	0.980 (+12%)	0.719 (+22%)	0.643 (+22%)
$[5; 10)$	0.784	0.450	0.303	0.915 (+17%)	0.558 (+24%)	0.368 (+21%)
$[10; \infty)$	0.708	0.376	0.224	0.850 (+20%)	0.472 (+26%)	0.272 (+21%)
$[1; \infty)$	0.720	0.392	0.246	0.860 (+19%)	0.490 (+25%)	0.299 (+22%)

One particular advantage of the KAIROS schema representation is that it is language-agnostic: given a schema automatically induced, say, over English documents, we can apply it to match with and rank documents in other languages. Thus, in addition to English, we also evaluate coverage for **Russian** and **Chinese** newswire documents, in the following two tables, respectively:

Table 5: Coverage on Russian newswire documents.

	82 schemas			82 + 150 schemas		
N_{events}	Cov@0.5	Cov@0.7	Cov@0.9	Cov@0.5	Cov@0.7	Cov@0.9
$[1;5)$	0.886	0.612	0.561	0.983 (+11%)	0.734 (+20%)	0.670 (+19%)
$[5;10)$	0.778	0.465	0.335	0.921 (+18%)	0.586 (+26%)	0.408 (+22%)
$[10;\infty)$	0.688	0.387	0.250	0.839 (+22%)	0.492 (+27%)	0.306 (+22%)
$[1;\infty)$	0.713	0.414	0.287	0.858 (+20%)	0.523 (+26%)	0.349 (+21%)

Table 6: Coverage on Chinese newswire documents.

	82 schemas			82 + 150 schemas		
N_{events}	Cov@0.5	Cov@0.7	Cov@0.9	Cov@0.5	Cov@0.7	Cov@0.9
$[1;5)$	0.875	0.589	0.528	0.981 (+12%)	0.718 (+22%)	0.639 (+21%)
$[5;10)$	0.776	0.460	0.314	0.924 (+19%)	0.582 (+27%)	0.387 (+23%)
$[10;\infty)$	0.699	0.408	0.251	0.877 (+25%)	0.531 (+30%)	0.314 (+25%)
$[1;\infty)$	0.713	0.422	0.271	0.885 (+24%)	0.545 (+29%)	0.337 (+24%)

We observe that the initial set of 82 schemas covers a substantial portion of the newswire corpora. Even in the most extreme case of long documents (Nevents ≥ 10) and high required coverage (Cov@0.9, meaning that 90% of documents' events need to match the events of at least one schema), around 20-25% of each document set is covered by our 82 schemas. Extending the schema library and adding additional 150 schemas boosts this result to 27-31%. The results are similar across all languages considered. This suggests feasibility of expanding schema-labeled corpora (such as the LDC corpus) to a much larger newswire corpora and demonstrates cross-lingual potential of schemas.

3.5 Temporal Relation Extraction

In this project, we approached extraction of temporal relations between events using discriminative models, and we investigated the possibility of inducing an optimal set of temporal/subevent relations from annotated data using generative models.

3.5.1 Continuous Temporal Relations

In this part of the project, we developed a novel framework for temporal relation representation that puts event duration front and center. Like standard approaches using the TimeML standard, we drew inspiration from Allen's (1983) seminal work on interval representations of time. But instead of annotating text for categorical temporal relations, we mapped events to their likely durations and event pairs directly to real-valued relative timelines. This change allowed us to better reason about the temporal structure of complex events as described by entire documents by giving a central role to event duration.

We collected a dataset—the Universal Decompositional Semantics Time (UDS-T) dataset, which is the largest publicly available temporal relations dataset to date (https://decomp.io/projects/time/)—using this approach on top of the English Web Treebank (Bies et al., 2012) portion of Universal Dependencies (UD-EWT; Silveira et al., 2014; De Marneffe et al., 2014; Nivre et al., 2015). The main advantages of UD-EWT over other similar corpora are: (i) it covers text from a variety of genres; (ii) it contains gold standard Universal Dependency parses; and (iii) it is compatible with various other semantic annotations which use the same predicate extraction standard (White et al., 2016; Zhang et al., 2017; Rudinger et al., 2018; Govindarajan et al., 2019). Table 7 the size of UDS-T against other temporal relations datasets.

Table 7: number of total events and event-event relations captured in various corpora

Dataset	#Events	#Event-Event Relations
TimeBank	7,935	3,481
TempEval 2010	5,688	3,308
TempEval 2013	11,145	5,272
TimeBank-Dense	1,729	8,130
Hong et al. (2016)	863	25,610
UDS-T	**32,302**	**70,368**

Annotators are given two contiguous sentences from a document with two highlighted event-referring expressions (predicates). They are then asked (i) to provide relative timelines on a bounded scale for the pair of events referred to by the highlighted predicates; and (ii) to give the likely duration of the event referred to by the predicate from the following list: *instantaneous, seconds, minutes, hours, days, weeks, months, years, decades, centuries, forever*. In addition, annotators were asked to give a confidence ratings for their relation annotation and each of their two duration annotation on the same five-point scale: *not at all confident* (0), *not very confident* (1), *somewhat confident* (2), *very confident* (3), *totally confident* (4). An example of the annotation instrument is shown in Figure 5.

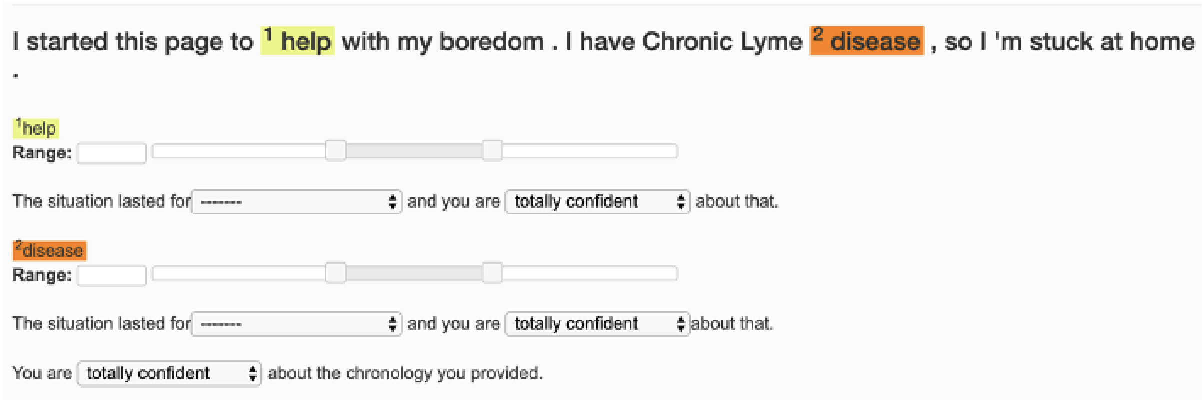

Figure 5: An annotated example using our protocol

We used this dataset to train models for jointly predicting fine-grained temporal relations and event durations. We report strong results on our data and show the efficacy of a transfer-learning approach for predicting categorical relations. The core model architecture is visualized in Figure 6.

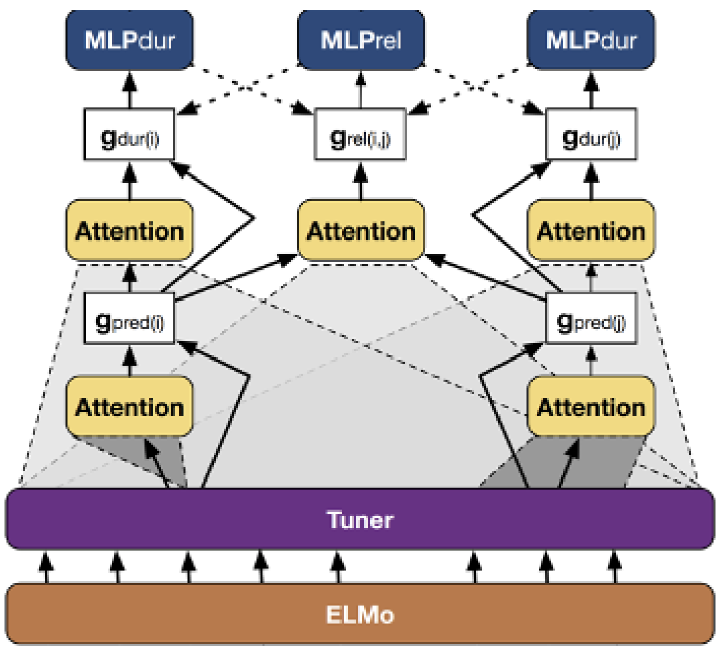

Figure 6: The architecture of our temporal relation extraction system

Within this core architecture, we experimented with different ways of modeling duration–enforcing ordinal constraints (binomial) or not (softmax)–and tying relation and duration prediction together–in the Dur→Rel architectures, we modified the relation representation in two ways: (i) adding the predicate's duration probabilities from the binomial model into the inputs for relation prediction, and (ii) not using the relation representation model at all; and in the Dur←Rel architectures, we use two modifications: (i) adding relation information as additional input for duration prediction, and (ii) not using the duration representation model The results are given in Table 8.

Table 8: Results on test data with different architectural assumptions

Model			Duration			Relation		
Duration	Relation	Connection	ρ	rank diff.	R1	Absolute ρ	Relative ρ	R1
softmax	✓	-	32.63	1.86	8.59	77.91	68.00	2.82
binomial	✓	-	37.75	**1.75**	13.73	77.87	67.68	2.35
-	✓	Dur ← Rel	22.65	3.08	-51.68	71.65	66.59	-6.09
binomial	-	Dur → Rel	36.52	1.76	13.17	77.58	66.36	0.85
binomial	✓	Dur → Rel	**38.38**	**1.75**	**13.85**	77.82	67.73	2.58
binomial	✓	Dur ← Rel	38.12	1.75	13.68	**78.12**	**68.22**	**2.96**

We found that most of our models are able to predict the relative position of the beginning and ending of events very well (high relation correlation) and the relative duration of events somewhat well (relatively low duration correlation), but they have a lot more trouble predicting relation exactly and relatively less trouble predicting duration exactly.

Further, the model that enforces ordinality constraints (binomial model) outperforms the model without such constraints (softmax model) for duration prediction by a large margin, though it has basically no effect on the accuracy of the relation model, with the binomial and softmax models performing comparably. This suggests that enforcing concavity in duration rank on the duration probabilities helps the model better predict durations.

Relatedly, connecting the duration and relation model does not improve performance in general. In fact, when the durations are directly predicted from the temporal relation model–i.e. without using the duration representation mode–the model's performance drops by a large margin, with the correlation down by roughly 15 percentage points. This indicates that constraining the relations model to predict the durations is not enough and that the duration representation is needed to predict durations well. On the other hand, predicting temporal relations directly from the duration probability distribution–i.e. without using the relation representation model---results in a similar score as that of the top-performing model. This indicates that the duration representation is able to capture most of the relation characteristics of the sentence. Using both duration representation and relation representation separately (model highlighted in blue) results in the best performance overall on the UDS-T development set.

In addition to our new dataset, we experimented with how well our model transfers to existing categorical datasets. We found that our system outperformed the F1-micro scores of all other systems on TimeBank-Dense. The results can be found in Table 9.

Table 9: F1 micro scores for event-event relations in TimeBank-Dense in our transfer learning experiments

System	**F1**
CAEVO	0.494
CATENA	0.519
Cheng & Miyao 2017	0.529
Ours	**0.566**

3.5.2 Temporal relation extraction via natural language inference

To further improve transfer across different temporal relation annotation formats, we investigated recasting a variety of temporal relation datasets into a natural language inference format. We created five new Natural Language Inference (NLI) datasets recasted from four existing temporal reasoning datasets: (i) TempEval3 (TE3; UzZaman et al., 2013); (ii) TimeBank-Dense (TB-D; Chambers et al., 2014); (iii) Richer Event Description (RED; O'Gorman et al., 2016); and described above (iv) UDS-Time (UDST; Vashishtha et al., 2019). These NLI datasets focus on two key aspects of temporal reasoning: (a) temporal ordering and (b) event duration. Across these datasets, we have more than a million NLI examples and we retain the training, development, and test splits from the original (for datasets in which such splits exist). Table 10 reports the total number of NLI pairs in each of our recast datasets.

Table 10: Recasted dataset statistics

Phenomenon	Dataset	# NLI Pairs
duration	UDS-Time	504,136
order	UDS-Time	562,944
order	TempEval3	27,240
order	TimeBank-Dense	11,910
order	RED	5,578

To generate hypotheses for our temporal ordering datasets, we created 8 templates which refer to the start-points and end-points of events in a pair of two events. The templates are shown in Table 11.

Table 11: NLI hypothesis templates

	Hypothesis Template	Entailing Relations TE3	TB-D	RED
1	X started before Y started	B,I,EB,IB,D	Bt,I	Br,C, EO?
2	X started before Y ended	B,I,II,S,IB,BB,BE,EB,D,E	Bt,I,II, At?	Br, C, EO, BO, S
3	X ended before Y started	B	?Bt?	Br?
4	X ended before Y ended	B,II,BE,IB,D	Bt,II	Br,BO?
5	Y started before X started	A,II,IA,E	At,II	EO?
6	Y started before X ended	A,I,II,S,IA,BB,BE,EB,D,E	At,I,II, Bt?	C, EO, BO, S, Br?
7	Y ended before X started	A	At?	-
8	Y ended before X ended	A,I,IA,BB	At,I	C, BO?

UDS-T directly annotates for the relation between start and end points of events in an event pair, making hypothesis generation with our templates straight-forward. In contrast, TE3, TB-D, and RED annotate event pairs for categorical temporal relations based on those proposed by Allen (1983). Using each category's definition, we mapped that category to a template predicate—a function from hypothesis templates to {entailed, not-entailed}—summarized in Table 11.

TE3, which is comprised of the TimeBank (Puste-jovsky et al., 2003) and AQUAINT (Graff) corpora, contains 13 temporal links: before(B), ibefore(IB), after(A), iafter(IA), isincluded(II), includes(I), begins(BE), begun-by(BB), ends(E), ended-by(EB), during(D), simultaneous(S), and identity. Each of these relations unambiguously maps to a template predicate.

TB-D uses a reduced set of relations: before(Bt), after(At), is included(II), includes (I), simultaneous(S), and vague (the last of which we ignore); as does RED: before(Br), begins-on(BO), ends-on(EO), contains(C), and simultaneous(S). This reduction results in the categories being ambiguous with respect to certain hypothesis templates. For instance, for Template 3 (X ended before Y started) knowing that X is before(Bt,Br) Y in the TB-D and RED sets does not give enough information about the ending point for X because these relations are not defined to have a strict ending boundary—in contrast to before(B) in TE3. We thus excluded hypothesis templates for ambiguous TB-D or RED relations. For RED, we collapsed relations with the same prefix into a single relation, e.g. before/causes, before/precondition is collapsed into Br.

To generate hypotheses for our temporal duration dataset, we created 18 hypothesis templates that refer to a range of likely durations for an event, based on two meta templates: (i) X did last or will last longer than LOWER-BOUND and (ii) X did last or will last shorter than UPPER-BOUND, where LOWER-BOUND and UPPER-BOUND range over a second, a minute, an hour, a day, a week, a month, a year, a decade, and a century. We recasted a single dataset—UDS-T—which (as described above) contains annotations for the duration of an event drawn from the following 11 labels.

For each event, we create two or four NLI pairs (depending upon the true label) to capture the duration information. The entailed hypothesis of the NLI pair takes a range of duration values derived from the gold duration label for the given event. The lower limit of the range is one rank less than the gold label—e.g. for minutes, the LOWER-BOUND is a second—and the upper limit is one rank greater than the gold label—e.g. for minutes, the UPPER-BOUND is an hour. Two entailed hypotheses were then generated from these two limits, one corresponding to the lower limit—longer than a second, and the other corresponding to the upper limit—shorter than an hour. The corresponding not-entailed hypotheses were then generated by inverting the entailed hypothesis—e.g. for minutes: shorter than a second and longer than an hour. In cases, where the gold duration label is instantaneous or forever, only one entailed and one not-entailed pair was created. We validated this dataset by sampling NLI pairs and asking participants to rate them, finding that in general the recasted items were valid and grammatical.

We use our recast datasets to explore how well different common classes of NLI models capture temporal reasoning. Specifically, we use three types of models: (i) neural bag of words (NBOW; Iyyer et al., 2015) (ii) InferSent (Conneau et al., 2017), and (iii) RoBERTa (Liu et al., 2019).[4] Our NBOW model represents contexts and hypotheses as an average of GloVe embeddings (Pennington et al., 2014). The concatenation of these representations is fed to a MLP with one hidden layer. The InferSent model encodes contexts and hypotheses independently with a BiLSTM and sentence representations are extracted using max-pooling. The concatenation of these sentences, their difference, and their element-wise product (Mou et al., 2016) are then fed to a MLP. For Roberta, we use a classification head on top of the pooled output of roberta-large to predict the labels.

In our experiments, we train and test these models on each recast temporal dataset. For each model, we include a hypothesis-only baseline to evaluate how much the datasets test NLI as opposed to just the likely duration and order of events in general. Additionally, we train each model on Multi-genre NLI (MNLI, Williams et al., 2018) and test the model on our datasets to see if the model learns temporal reasoning from a generic NLI dataset that does not necessarily focus on temporal reasoning.

Table 12 shows the accuracy of different models on our recast temporal datasets. We report the majority baseline (MAJ) of always predicting the label that appeared the most in training. We see that the models trained on MNLI perform poorly on our recast datasets, even worse than MAJ baseline in many cases. This indicates that the models trained on MNLI do not learn representations well enough to infer temporal reasoning in our datasets.

Table 12: NLI experiment results

Model	UDS-duration	UDS-order	TempEval3	TimeBank-Dense	RED
Majority	50.00	54.52	54.57	50.54	52.51
MNLI Baseline					
NBOW	43.37	33.83	34.13	35.36	34.02
InferSent	49.04	49.58	47.57	47.84	47.64
RoBERTa	50.28	55.01	54.87	51.01	52.97
Hypothesis-Only					
NBOW	84.96	54.52	54.57	50.54	52.51
InferSent	91.18	71.96	63.22	68.29	63.47
RoBERTa	91.52	71.22	**63.25**	68.83	52.51
Context and Hypothesis					
NBOW	82.45	54.52	54.57	50.54	52.51
InferSent	92.65	73.22	62.2	68.29	63.47
RoBERTa	**94.51**	**80.17**	54.57	**94.60**	**80.59**

The hypothesis-only models demonstrate an interesting limitation of NBOW and InferSent. Both NBOW and InferSent hypothesis-only models are as good as, or even better, than the normal models across all datasets. RoBERTa, however, improves when given the context, across all datasets, with TempEval3 as the exception. This suggests that RoBERTa embeddings are better able to capture the semantics of the context than NBOW and InferSent. In fact, NBOW and InferSent may just predict the label based on information about lexical entities in the hypothesis.

Context in duration for all three hypothesis-only models, achieve high accuracy on the NLI dataset based on UDS-Duration. Even RoBERTa seems to fail to capture anything extra from the context. To analyze this anomaly, we created a hypothesis- template based majority baseline inferred from the UDS-Duration train data and find that it achieves an 80.2% accuracy on the test set. This indicates that the data is skewed for each template, which might be caused by the skewed *minutes* duration label in UDS-T (roughly 28% of the UDS-T train set contains *minutes* as the true duration label, as seen in Figure 7).

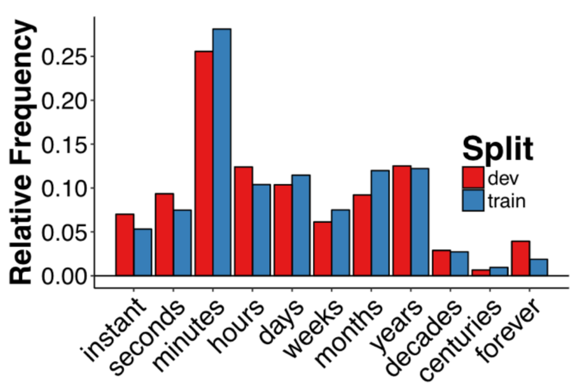

Figure 7: Distribution of duration labels

This template based majority prediction is noteworthy as the models pretrained on MNLI fail to infer the correct labels even when the labels are skewed per template. The neural models see a 10% gain in accuracy over the template-sensitive majority, indicating that the models are learning the range of durations for different entities. Another possible reason that the context does not help much for duration is that events often have a modal distribution for a duration label, similar to the explanation for the recast NER data in Poliak et al. (2018).

3.5.3 Beyond Temporal Relations

In this component of the project, we experimented with using latent variable models to understand document-level event structure. One part of this involved inducing document-level event timelines. For instance, the target for the document below would look like the timeline in Figure 8.

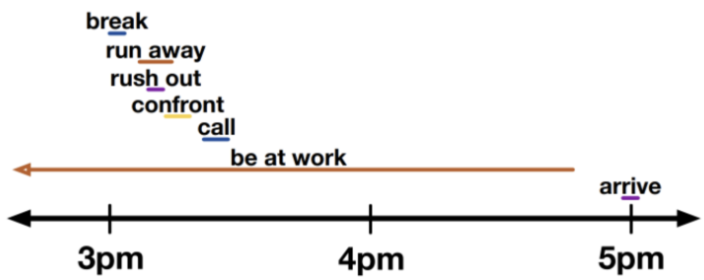

Figure 8: An example timeline

*At 3pm, a boy **broke** his neighbor's window. He was **running away**, when the neighbor **rushed** out to **confront** him. His parents were **called** but couldn't **arrive** for two hours because they **were still at work**.*

We found that the results of this model were underwhelming. We hypothesized that such a model would need additional information about the structure of complex events and might need to compress the continuous relations into optimal categorical relations. With this aim in mind, we investigated deriving document-level event structure (including temporal relations) from the joint distribution of properties relevant of event, entity, and semantic role types, in addition to temporal relation and duration information among other event-event relations.

This work specifically aimed to develop an *empirically derived* event structure classification. Where prior work takes a top-down approach—hand-engineering an event classification before deploying it for annotation—we took a bottom-up approach—*decomposing* event structure into a wide variety of theoretically informed, cross-cutting semantic properties, annotating for those properties, then *recomposing* an event classification from them by induction. To support this induction, we augmented existing annotations found in the Universal Decompositional Semantics (UDS)1.0 dataset, which covers the entirety of the English Web Treebank, with an array of inferential properties capturing fine-grained aspects of the temporal and aspectual structure of events.

One important extension this work made to UDS was the addition of document-level edges. A portion of a document-level UDS graph can be found in Figure 9.

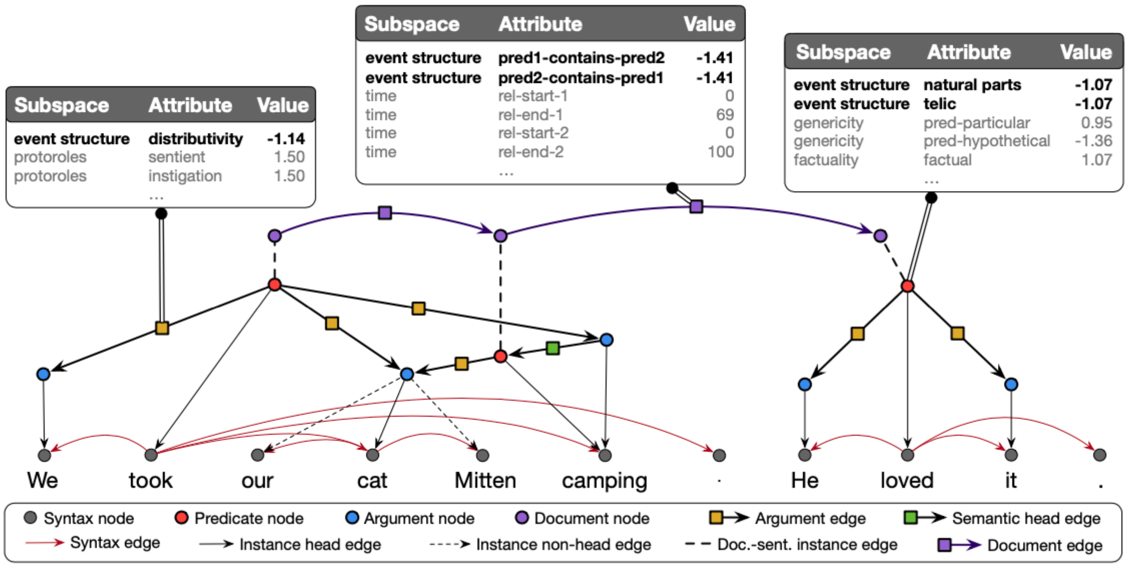

Figure 9: An example UDS2.0 syntactic and semantic graph with attributes

We annotated for the core event structural distinction not currently covered by UDS, breaking our annotation into three subprotocols. For all questions, annotators report confidence in their response to each question on a scale from 1 (*not at all confident*) to 5 (*totally confident*).

In the event-subevent protocol, annotators are presented with a sentence containing a single highlighted predicate followed by four questions about the internal structure of the event it describes. Q1 asks whether the event described by the highlighted predicate has natural subparts. Q2 asks whether the event has a natural endpoint.

The final questions depend on the response to Q1. If an annotator responds that the highlighted predicate refers to an event that *has* natural parts, they are asked (i) whether the parts are similar to one another and (ii) how long each part lasts on average. If an annotator instead responds that the event referred to does *not* have natural parts, they are asked (i) whether the event is dynamic, and (ii) how long the event lasts.

All questions are binary except those concerning duration, for which answers are supplied as one of twelve ordinal values: *effectively no time at all, fractions of a second, seconds, minutes, hours, days, weeks, months, years, decades, centuries* or *effectively forever*. Together, these questions target the three Vendler-inspired features (DYN, DUR, TEL), plus a fourth dimension for subtypes of dynamic predicates. In the context of UDS, these properties form a predicate node subspace, alongside FACTUALITY, GENERICITY, and TIME.

In the event-event protocol, annotators are presented with either a single sentence or a pair of adjacent sentences, with the two predicates of interest highlighted in distinct colors. For a predicate pair (p_1, p_2) describing an event pair (e_1, e_2), annotators are asked whether e_1 is a mereological part of e_2, and vice versa. Both questions are binary: a positive response to both indicates that e_1 and e_2

are the same event; and a positive response to exactly one of the questions indicates proper parthood. Prior versions of UDS do not contain any predicate-predicate edge subspaces, so we add document-level graphs to UDS to capture the relation between adjacently described events.

This subprotocol targets generalized event coreference, identifying *constituency* in addition to strict identity. It also augments the information collected in the event-subevent protocol: insofar as a proper subevent relation holds between e_1 and e_2, we obtain additional fine-grained information about the subevents of the containing event—e.g. an explicit description of at least one subevent.

In the event-entity protocol, we focused on the relation between the event described by a predicate and its plural or conjoined arguments, asking whether the predicate is distributive or collective with respect to that argument. This property accordingly forms a predicate-argument subspace in UDS, similar to PROTOROLES.

Table 13 shows the number of annotations per question and example sentences for each response. To our knowledge, our partial event coreference annotation is the largest to date.

Table 13: Example annotations from UDS-Event Structure dataset

	Annotation	Count (%)	Example
Event-subevent	Has natural parts	6,903 (23%)	The eighteen steps of the dance are done rhythmically
	Parts similar	4,498 (15%)	Israel resumed its policy of targeting militant leaders
	Parts dissimilar	2,158 (7%)	Fish are probably the easiest to take care of
	(Part duration)	(–)	(ordinal; not shown)
	No natural parts	23,069 (77%)	It had better nutritional value
	Dynamic	13,903 (48%)	I would like to informally get together with you
	Not dynamic	8,839 (29%)	I assume this is 12:30 Central Time?
	(Full duration)	(–)	(ordinal; not shown)
	Natural endpoint	6,031 (20%)	I will deliver it to you
	No natural endpoint	23,941 (80%)	If you know or work there could you enlighten me?
	total	29,984	(all event descriptions)
Event-event	P1, P2 identical	2,435 (6%)	All horses [. . .] are happy$_1$ & healthy$_2$ when they arrive
	P1, P2 disjoint	30,247 (80%)	I am often stopped$_1$ on the street and asked, 'Who does your hair . . . I LOVE$_2$ it'
	$P1 \subset P2$	1,832 (5%)	The office is shared with a foot doctor and it's very sterile$_1$ and medical feeling$_2$, which I liked
	$P2 \subset P1$	3,029 (8%)	It is a very cruel death$_1$ with bodies dismembered$_2$
	total	37,719	(pairs of event descriptions w/ temporal overlap)
Event-entity	Distributive	4,812 (50%)	the **pics** turned out ok
	Collective	4,876 (50%)	**we** draw on our many faith traditions to arrive at a common conviction
	total	9,710	(event descriptions with plural arguments)

Our goal in inducing event structural categories is to learn representations of event structure categories on the basis of annotated UDS graphs, augmented with the new UDS-E annotations. We aim to learn four sets of interdependent classifications grounded in UDS properties: event types, entity types, semantic role types, and event-event relation types. These classifications are interdependent in that we assume a generative model that incorporates both sentence- and document-level structure.

Semantics edges in UDS1.0 represent only sentence-internal semantic relations. This constraint implies that annotations for cross-sentential semantic relations—a significant subset of our event-event annotations— cannot be represented in the graph structure. To remedy this, we extended UDS1.0 by adding document edges that connect semantics nodes either within a sentence or in two distinct sentences, and we associated our event-event annotations with their corresponding document edge. Because UDS1.0 does not have a notion of document edge, it does not contain Vashishtha et al.'s (2019) fine-grained temporal relation annotations, which are highly relevant to event-event relations. We additionally add those attributes to their corresponding document edges. The algorithm in Equation 1 gives the generative story for our event structure induction model.

Equation 1: The generative story for our event structure model

$$
\begin{array}{l}
\text{Initialize queue } I\text{ ;}\\
\textbf{for } \textit{sentence } s \in \mathcal{S} \textbf{ do}\\
\quad \text{Initialize queue } J\text{ ;}\\
\quad \text{Enqueue } J \rightarrow I\text{ ;}\\
\quad \textbf{if } \mathrm{length}(I) > W \textbf{ then}\\
\quad\quad \text{Dequeue } I\\
\quad \textbf{for } \textit{predicate node } v \in \mathrm{predicates}(s) \textbf{ do}\\
\quad\quad \text{Sample event type } t_{sv} \sim \mathrm{Cat}\left(\boldsymbol{\theta}^{(\mathrm{event})}\right)\text{ ;}\\
\quad\quad \textbf{for } \textit{property } p \in \mathcal{P}_{\mathrm{event}} \textbf{ do}\\
\quad\quad\quad \textbf{for } \textit{annotator } i \in \mathcal{A}_{svp}^{(\mathrm{event})} \textbf{ do}\\
\quad\quad\quad\quad \text{Sample } x_{svpi}^{(\mathrm{event})} \sim f_p^i\left(\boldsymbol{\mu}_{t_{sv}}^{(\mathrm{event})}\right)\\
\quad\quad \text{Enqueue } \langle s, v\rangle \rightarrow J;\\
\quad\quad \textbf{for } \textit{argument node } v' \in \mathrm{arguments}(s, v) \textbf{ do}\\
\quad\quad\quad \text{Sample ent. type } t_{sv'} \sim \mathrm{Cat}\left(\boldsymbol{\theta}^{(\mathrm{entity})}\right)\text{ ;}\\
\quad\quad\quad \textbf{for } \textit{property } p \in \mathcal{P}_{\mathrm{ent}} \textbf{ do}\\
\quad\quad\quad\quad \textbf{for } \textit{annotator } i \in \mathcal{A}_{sv'p}^{(\mathrm{ent})} \textbf{ do}\\
\quad\quad\quad\quad\quad \text{Sample } x_{sv'pi}^{(\mathrm{part})} \sim f_p^i\left(\boldsymbol{\mu}_{t_{sv'}}^{(\mathrm{part})}\right)\\
\quad\quad\quad \textbf{if } v' \textit{ is eventive } \textbf{then}\\
\quad\quad\quad\quad \text{Enqueue } \langle s, v'\rangle \rightarrow J;\\
\quad\quad\quad \text{Sample role type } r_{svv'} \sim \mathrm{Cat}\left(\boldsymbol{\theta}_{t_{sv}t_{sv'}}^{(\mathrm{role})}\right)\text{ ;}\\
\quad\quad\quad \textbf{for } \textit{property } p \in \mathcal{P}_{\mathrm{role}} \textbf{ do}\\
\quad\quad\quad\quad \textbf{for } \textit{annotator } i \in \mathcal{A}_{svv'p}^{(\mathrm{role})} \textbf{ do}\\
\quad\quad\quad\quad\quad \text{Sample } x_{svv'pi}^{(\mathrm{role})} \sim f_p^i\left(\boldsymbol{\mu}_{r_{svv'}}^{(\mathrm{role})}\right)\\
\quad\quad \textbf{for } \textit{index pair } \langle s', v'\rangle \in \mathrm{flatten}(I) \textbf{ do}\\
\quad\quad\quad \text{Sample rel. type } q \sim \mathrm{Cat}\left(\boldsymbol{\theta}_{t_{sv}t_{s'v'}}^{(\mathrm{rel})}\right)\text{ ;}\\
\quad\quad\quad \textbf{for } \textit{property } p \in \mathcal{P}_{\mathrm{rel}} \textbf{ do}\\
\quad\quad\quad\quad \textbf{for } \textit{annotator } i \in \mathcal{A}_{svs'v'p}^{(\mathrm{rel})} \textbf{ do}\\
\quad\quad\quad\quad\quad \text{Sample } x_{svs'v'pi}^{(\mathrm{rel})} \sim f_p^i\left(\boldsymbol{\mu}_q^{(\mathrm{rel})}\right)
\end{array}
\tag{1}
$$

Figure 10 shows the resulting factor graph for the semantic graph shown above.

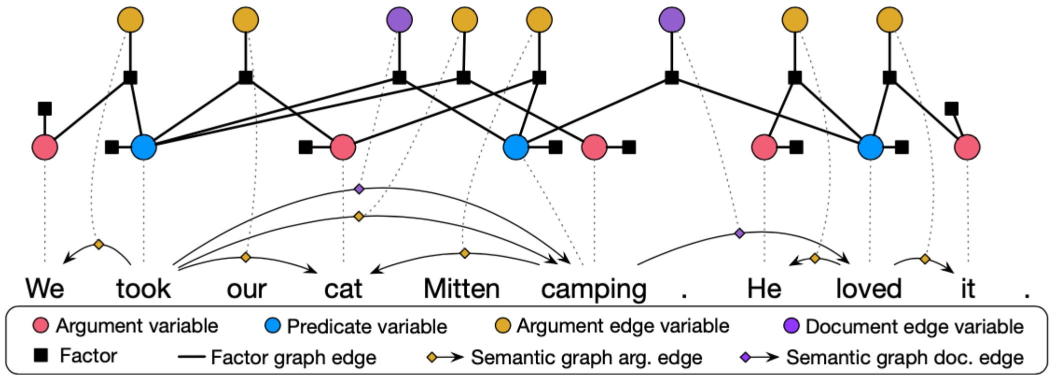

Figure 10: Factor graph for our event structure model

We fit our model to the training data using expectation-maximization. We used loopy belief propagation to obtain the posteriors over event, entity, role, and relation types in the expectation step and the Adam optimizer to estimate the parameters of the distributions associated with each type in the maximization step. As a stopping criterion, we compute the evidence that the model assigns to the development data, stopping when this quantity begins to decrease.

To select |Tevent|, |Tent|, |Rrole|, and |Rrel| for the algorithm above, we fit separate mixture models for each classification—i.e. removing all factor nodes—using the same likelihood functions as in the generative story. We then computed the evidence that the simplified model assigns to the development data given some number of types, choosing the smallest number such that there is no reliable increase in the evidence for any larger number. To determine reliability, we compute 95% confidence intervals using nonparametric bootstraps.

The event-event relation types we obtain track closely with approaches that use sets of underspecified temporal relations (Cassidy et al., 2014; O'Gorman et al., 2016; Zhou et al., 2019, 2020; Wang et al., 2020).

e1 starts before e2: *[. . .]the Spanish, Thai and other contingents are already* ***committed*** *to* ***leaving*** *[. . .]*
e2 starts before e1: *And I have to* ***wonder****: Did he* ***forget*** *that he already has a memoir[...]*
e2 ends after e1: *no, i am not* ***kidding*** *and no i don't want it b/c of the taco bell dog. i want it b/c it is really* ***small*** *and cute.*
e1 contains e2: *they* ***offer*** *cheap air tickets to their country [...] you may get excellent discount airfare, which may even* ***surprise*** *you.*
e1 = e2: *the food is good, however the tables are so* ***close together*** *that it feels very* ***cramped****.*

We believe these induced relation alongside the event structure classes and thematic role classes will be useful for capture document-level event structure, including temporal structure moving forward.

3.6 Covid-19 Infobot

The outbreak of the COVID-19 pandemic has caused tremendous amounts of suffering and deaths around the world. As the world sees more infected cases every day, the need and demand for **reliable and up-to-date** information on COVID-19 likewise increase every day. However, building automated systems to retrieve this information can be challenging, especially when there is a lack of annotated data. We designed an ordinal annotation interface that enabled public health experts to efficiently curate a high quality COVID-19 FAQ dataset. Our infobot and dataset is publicly available at https://covid-19-infobot.org/chat.

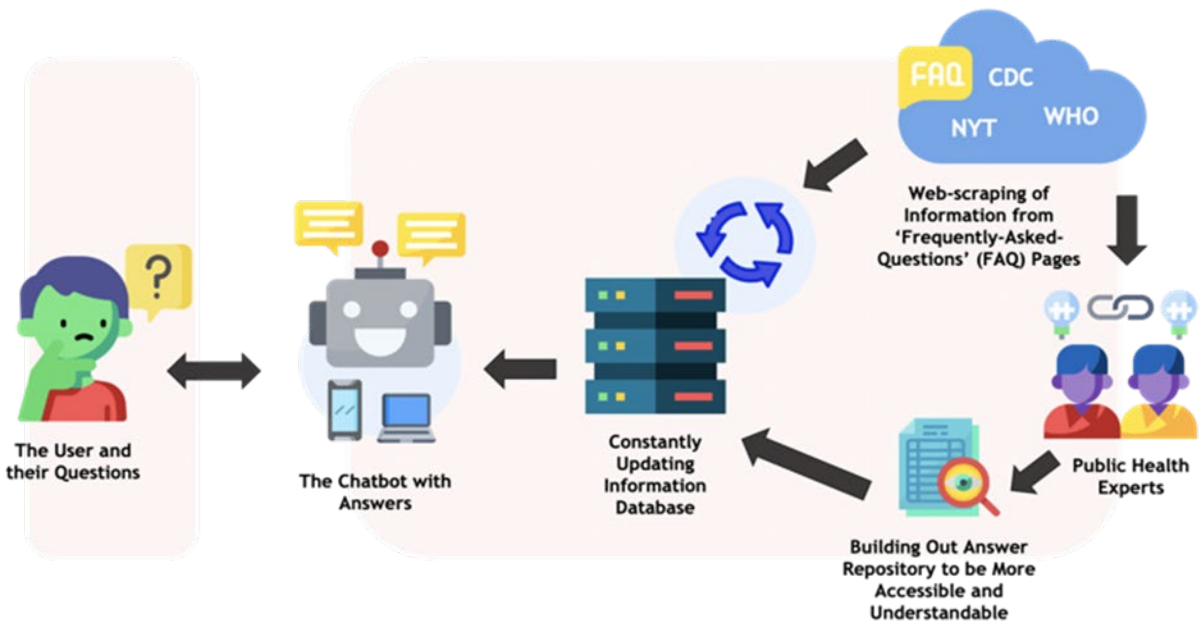

Figure 11: Diagram of our full process for collecting information and serving it to those concerned about COVID-19.

The COVID-19 pandemic deeply affected the lives of everyone throughout the world - death rates were rising, many schools were closed, and unemployment increased. More than with any other global health crisis, the public was constantly searching for latest developments of the pandemic to keep up with the newest information in order to protect themselves and their families. A well-informed public was deemed critical to managing the spread of COVID-19 and its impact on society. Unfortunately, misinformation was rampant in online media and has dire consequences. For example, incorrect information on the use of certain drugs for the prevention of COVID-19 has had fatal outcomes, and stigmatization caused by misinformation about certain communities as vectors of the virus undermines the long-term welfare of our society.

Therefore, providing accurate and timely information was imperative. To aid in this goal, we developed a natural language processing-backed informational infobot that provided an accessible Question and Answering system using trusted sources. When building our COVID-19 infobot, we overcome the following two challenges:

- Creating an accurate and up-to-date repository of answers as well as supervised data for machine learning models by public health officials.
- Building a domain-specific QA system for COVID-19 which returns accurate answers to users.

We addressed the first challenge by aggregating up-to-date factual information in the form of verified questions and answers about the pandemic. We employed three main aggregation efforts in tandem: 1) generating high quality and accurate information from domain experts, i.e. public health experts 2) continuously scraping frequently asked questions and answers from online trusted sources, e.g. newspapers and government agencies; 3) automatically ranking and manually aligning additional questions from social media with the scraped questions and answers in our dataset. For the second challenge, we examined the feasibility and effectiveness of using state-of-the-art machine learning methods to automatically rank and retrieve relevant answers to user queries.

Our effort has resulted in a publicly available dataset that currently contains over 2,200 Questions and Answers from more than 40 webpages, and a fully functional COVID-19 infobot powered by machine learning models trained on verified data sources. Users can interact with our infobot on different platforms such as Slack, Whatsapp, Facebook Messenger and website to access information about COVID-19, available care, and other topics of interest. To encourage future research in this area, we fully open-sourced our codebase and dataset. The dataset and the infobot are available at https://covid-19-infobot.org/.

We created our publicly available dataset of over 2,200 question-answer pairs by aggregating FAQs from trusted news sources. We chose websites to scrape based on three broad criteria: 1) the informativeness and trustworthiness of the website; 2) the ease of scraping frequently asked question-answer pair from the website; and 3) the number of questions and answers on the website. The first criterion was assessed by trained experts at the School of Public Health at John's Hopkins University School of Medicine.

By using a straightforward scraping process, we enable undergraduate students to contribute to our efforts. We developed a python library for students to easily add scrapers to our project. Our library requires each question-answer (and metadata) to be stored as a simple dictionary. The library then abstracts away adding this information to our set of question-answer pairs. Additionally, the library accordingly handles updating answers to questions in our dataset if a previously scraped website updates its information. Further documentation is available at https://github.com/JHU-COVID-QA/scraping-qas and we encourage others to join our efforts.

For each scraped question-answer pair, we extracted relevant metadata for our infobot and other NLP analytics. The metadata includes information about the source of each question-answer pair (we include both the source name and the link) and the date when the question-answer was last scraped from or updated on the website. Additionally, if the information on the website is targeted for a specific geographic area, we included that in our metadata as well.

As our understanding of COVID-19 rapidly evolved, trustworthy sources regularly updated the information they released. Therefore, each day, we automatically re-ran the web scrapers to find new information. This enabled us to add new question-answers or update answers to existing questions in our dataset.

If a previously scraped question-answer is removed from a website, we removed that example from our dataset. Question and answers that we removed from our dataset are still available in our history since we archive each day's dataset on our webpage. In turn, the quality of our dataset constantly evolved and improved.

The described effort resulted in a dataset that evolved daily. The June 2nd 2020 version contained over 2,200 questions and answers scraped from 40 websites. Our dataset contains some examples in different languages besides English, owing to deep-set scraping websites in multiple languages. Roughly 65% of our examples are in English. Websites might update or change how they store information. This is why the June 2nd version of our dataset contains just 1 example from the Delaware State Government webpage, while the May 20th version of our dataset contains 22 examples from this website.

Since the Internet contains many more questions that are not answered, we additionally collected questions from Twitter, Qorona, and CovidFaq and manually aligned them with the question-answer pairs in our dataset. We leveraged information retrieval techniques to match these unanswered questions with questions in our dataset and then rely on domain experts to verify each aligned question-question-answer pair. The following subsections provide details for each of these steps.

We downloaded 28 million tweets from the COVID-19 Twitter Dataset, extracted the questions from the tweets, sorted them by frequency, and discarded the questions that occurred less than four times. Then, we grouped semantically similar questions into 9,200 clusters. Next, we extracted the centers of the clusters and, using a state-of-the-art sentence re-writer, we generated three high quality paraphrases of each question. This resulted in a collection of over 27,000 unanswered questions about COVID-19 from Twitter. We worked with public health experts from Johns Hopkins Bloomberg School of Public Health to align these unanswered questions with our verified question-answer pairs. For each of these 27,000 questions, we used an extended model from our lab to determine the most similar answered questions in our dataset. For each unanswered Twitter question, we presented public health experts with the five most similar question answering pairs from our dataset. Based on a formal protocol developed by a senior Public Health researcher on our team (Figure 4), we asked the experts to determine, on a scale from 1 to 100, how relevant or similar the question-answer pair from our dataset is to the unanswered question.

For this annotation effort, we leveraged Turkle, open-sourced, locally hosted clone of Amazon Mechanical Turk developed by the JHU Human Language Technology Center of Excellence. As part of this protocol, expert annotators could indicate whether a question was not relevant to COVID-19 or whether an existing answer was no longer correct. We removed such labeled examples from our set. This effort results in 24,240 annotated QQAs, with over 18,000 examples judged to be less than 1% relevant, indicating that the majority of the questions extracted from Twitter are irrelevant to the answered questions in our dataset. These additional examples can be used to further train a chatbot to answer questions about COVID-19. More details on the data collection can be found in Poliak et al. (2020). Subsequent efforts explored the use of these materials for a hosted infobot.

3.7 Paraphrastically Augmented FrameNet Parsing

This work was aimed at data creators such as at the LDC, where an ontology is created and representative data must be built up quickly. In that setting we might write out a lexical unit and search for, or author directly an example sentence that the specialist annotates for the target semantics. At that point the data creator might find new examples that have the same lexical unit, or search for examples using new lexical units. In our conception, the data curator asks an underlying system for a new version of the same sentence just selected, but with a paraphrase of the original span that was annotated as evoking the given semantic unit. Based on this information the system should then automatically align the spans of text from the originating sentence to the new example, and then in further such paraphrastic examples the system should ensure that is does not simply re-use prior lexical units in its new variants. Incrementally a user may continue to paraphrase, vetting the new examples as for whether they are appropriate, then continuing, thereby enumerating a much wider set of words or phrases that evoke the target concept than they'd be able to do themselves by hand, efficiently.

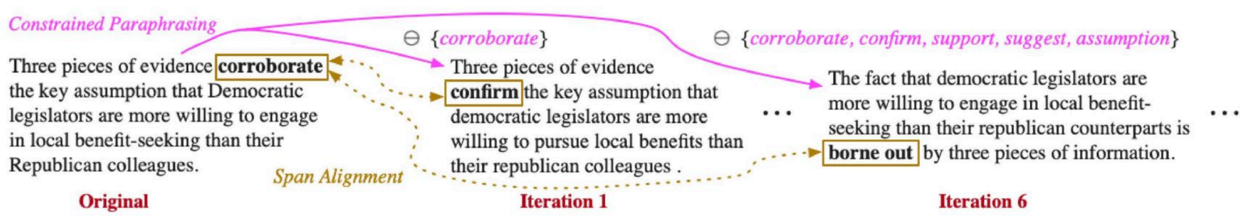

Figure 12: Framework for iterative paraphrastic augmentation illustrated on an actual system output. The original, manually annotated sentence contains a tag over the word ''corroborate'', which is then iteratively paraphrased.

We performed experiments with the FrameNet ontology to demonstrate downstream impact using this approach. FrameNet was well aligned to the event ontology designed for KAIROS Phase 1 and thus we were optimistic about this line of work in further improving our KAIROS extraction system.

In the following diagram we illustrate that our paraphrastic augmentation strategy is very successful in informing our extraction system when the amount of annotated data was especially sparse (useful for real world scenarios when the customer is defining their own ontology on the fly), but even at 100% employment of the FrameNet full-text annotations, our paraphrastic augmentation led to a many point gain in performance. Further details can be found in Culkin et al. (2021).

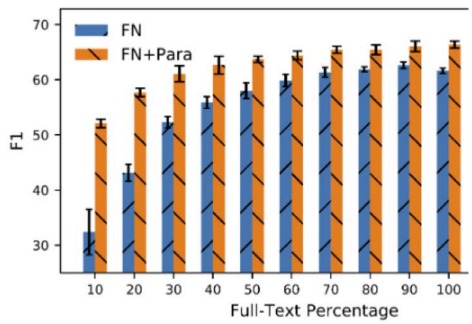

Figure 13: Improvements to FrameNet parsing based on augmentation.

3.8 Tacit Assumptions in Language Models

KAIROS is concerned with modeling expectations on events given the occurrence of other events, as captured in some schematic form. Treatment of these schemas relate significantly to discussions on general common sense knowledge. We performed a study on how much basic common sense knowledge was captured in the parameters of large pretrained language models.

Humans carry stereotypic tacit assumptions (STAs) (Prince, 1978), or propositional beliefs about generic concepts. Such associations are crucial for understanding natural language. We construct a diagnostic set of word prediction prompts to evaluate whether recent neural contextualized language models trained on large text corpora capture STAs. Our prompts are based on human responses in a psychological study of conceptual associations. We find models to be profoundly effective at retrieving concepts given associated properties. Our results demonstrate empirical evidence that stereotypic conceptual representations are captured in neural models derived from semi-supervised linguistic exposure.

Ellen Prince in 1978 laid out the idea of "stereotypical tacit assumptions" as essentially the common sense we bring to a conversation that helps language work: all the things we don't have to state explicitly because we know the listener shares a basic understanding of the world. This idea was very similar to the Minsky notion of a frame, which Schank then sub-divided into scripts and plans.

The following is borrowed from Prince, where she lays out examples of assumptions people can make of what other people already know:

```
(20)a. People have: parents      siblings   a spouse
                    relatives    a home     a job
                    a pet        a car      a hometown
                    a television a clock    neighbors
                                   :
                                   :
     b. Countries have: a leader  a president a queen
                        a duke    citizens    land
                        borders   a language  a history
                                   :
                                   :
```

Figure 14: Examples from Prince on Stereotypical Tacit Assumptions.

In psychology, McRae et al (2005) brought human subjects into a lab and asked them for the most common properties they could provide that described basic concepts like bears, or tables. While under a different terminology, these *norms* are a form of the STAs that Prince had in mind.

Prince claimed one needed STAs to understand language. Recently, many have claimed to have solved, or nearly solved, aspects of language based on contextual encoders. It is then reasonable to ask: do these encoders, e.g., BERT, possess STAs? While interesting in itself as a matter of science, it is KAIROS relevant because if basic properties are contained in contemporary encoders, then perhaps richer script-like information is also already captured in those same models?

Prompt	Model Predictions
A ___ has fur.	dog, cat, fox, ...
A ___ has fur, is big, and has claws.	cat, **bear**, lion, ...
A ___ has fur, is big, has claws, has teeth, is an animal, eats, is brown, and lives in woods.	**bear**, wolf, cat, ...

Figure 15: The concept 'bear' as a target emerging as the highest ranked predictions of a neural language model.

To find STAs, we took off-the-shelf neural masked language models (BERT and RoBERTa) and fashioned “probes”: natural language sentences with a missing word, where those sentences can be provided to a masked language model and it will provide back a score on all possible lexical items as to how likely the model believes a given word could fill the slot. A novel twist in our work is the construction of “concatenative” probes, where we had a series of properties about, e.g., a bear (derived from McRae et al), and we fashioned a longer and longer statement employing those properties (norms). We can then measure as a function of the number of properties, how likely the model finds the target concept (e.g., bear). We found that as the probe gets longer, the model becomes more confident in the target concept, and that strikingly one model in particular (RoBERTa-large) does quite well in absolute terms.

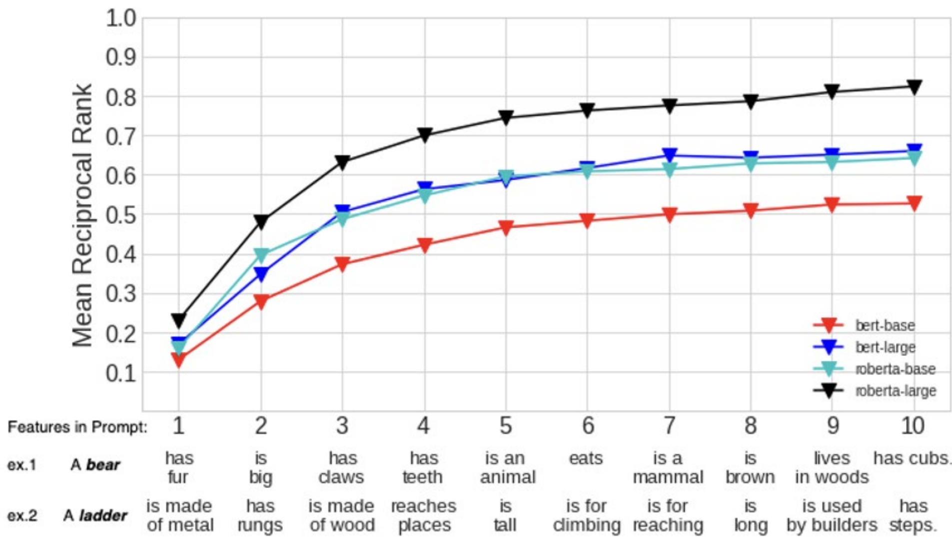

Figure 16: As more features of a class are made available the rank of answer improves.

We conclude from this study that basic conceptual knowledge is in fact contained in the largest of the contemporary encoders. More details appear in Weir et al (2020a).

From this finding we began an effort that coupled our Association Rule Mining (ARM) work described earlier, with probes over masked language models. We were interested in determining whether the schemas we are interested in KAIROS are already present in these masked language models, and "merely" need new efficient algorithms to extract this information out symbolically, as according to the requirements of the TA1/TA2 exchange. This effort then grew into our schema curation effort that coupled our causal language model with ARM, as reported earlier.

3.9 Diverse Causal Generation

We developed a conditional text generation framework that posits sentential expressions of possible causes and effects. This framework depends on two novel resources we develop in the course of this work: a very large-scale collection of English sentences expressing causal patterns (CausalBank); and a refinement over previous work on constructing large lexical causal knowledge graphs (Cause Effect Graph). Further, we extend prior work in lexically-constrained decoding to support *disjunctive* positive constraints. Human assessment confirms that our approach gives high-quality and diverse outputs. Finally, we use CausalBank to perform continued training of an encoder supporting a recent state-of-the-art model for causal reasoning, leading to a 3-point improvement on the Choice Of Plausible Alternative challenge set, with no change in model architecture.

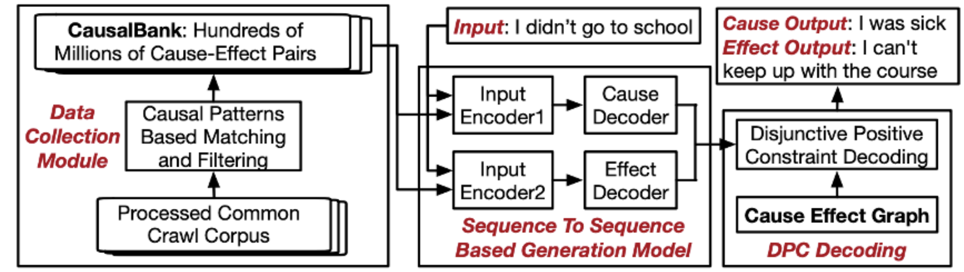

Figure 17: Conditional text generation framework.

There have been many prior works over the decades to scrape corpora for causal connections between events, through regular expressions matching on, e.g., "because of", "and this caused", "which led to", and so on. We employed a similar strategy, but against a very large collection of webtext, amassed as part of Philipp Koehn's long-running efforts at building massive bitexts and language models for many of the world's languages based on industrial grade web-scraping research. After pruning and filtering, we resulted in roughly 300million sentences that expressed a causal connection between two propositions. This text was then naively separated into two text passages: that coming before, and after, the matched causal connective. E.g.,

"[Ben bought ball bearings] because [he wanted to fix his bike]".

Causal Pattern
as, as a consequence/result of, as long as, because, because of, caused by, due/owing to, in response to, on account of, result from
accordingly, consequently, bring on/about, give rise to, induce, in order to, lead to, result in, prevent/stop...from, and for this reason, cause, for the purpose of, if...then, ,_so, so that, thereby, therefore, thus, hence

These text pairs were fed into a large scale neural MT (NMT) model, which are meant for mapping input sentences to output sentences. In contrast to NMT, in our case we want an input sentence to lead to a diverse range of outputs. There may be many reasons to, e.g., "buy ball bearings", and we want to output a variety of such reasons, not just the one that the model found most likely (the default behavior of an NMT system). In our prior work in paraphrasing via NMT, we know that such models lack diversity at "the top of the beam": even if you search over the k-best most likely outputs of such a system, the alternatives tend to be just small edits of the most likely output. We have prior work in various ways getting diverse outputs, including through randomly sampling outputs, as well as **lexically constrained decoding**. In lexically constrained decoding, one enforces hard constraints on words that either must, or must not, appear in the output, referred to as positive and negative constraints, respectively.

We adopted this previous framework here, contributing a novel extension of lexically constrained decoding with positive disjunctive constraints: rather than specifying one or more phrases that must appear, you can now give a set of items such that just one of those items need appear to satisfy the constraint. This is helpful when you are concerned with providing a constraint on an output wrt some lemma, e.g., "eat", but are not concerned which morphological inflection appears, e.g., "eating", "ate", "eats", etc.

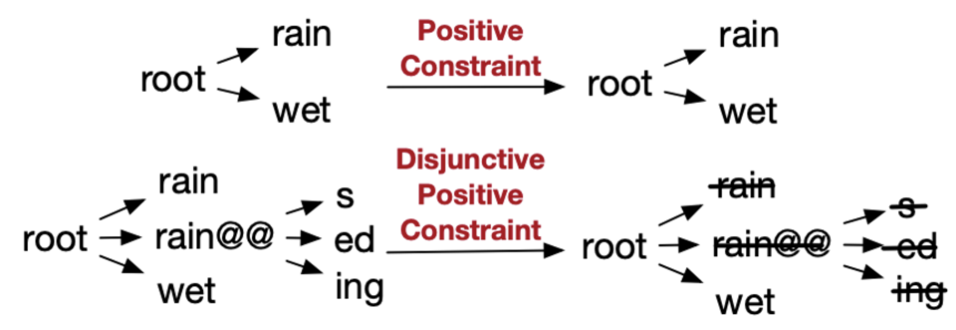

Figure 18: We extend lexically constrained decoding to allow for, e.g., morphologically inflected alternatives.

We paired this new capability with a large-scale data mining effort to build a causal lexical knowledge base: words that tend to co-occur in causal constructions.

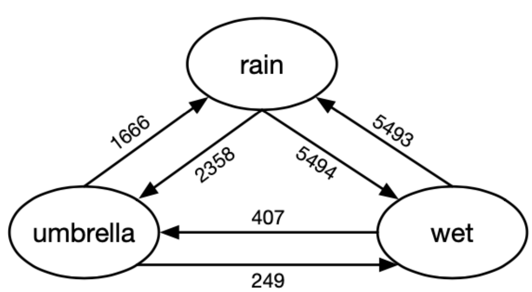

Figure 19: An example portion of our lexical causal correlation graph.

Combined, this allows us to take an input sentence, lookup the words in that sentence in our causal knowledge graph to get candidate causal terms. Those terms can serve as positive disjunctive constraints on our seq2seq rewriter, and we can thereby generate a series of candidate cause or effect sentences. More details can be found in Li et al. (2020).

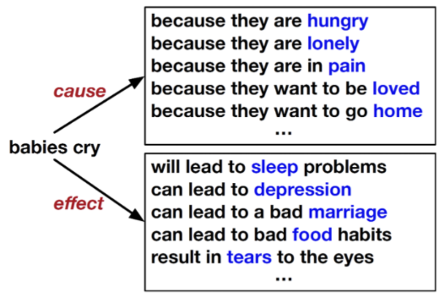

Figure 20: Example output from our model.

This work was followed by a project we called COD3S: COnstrained Decoding with Discrete Semantic Signatures, a novel method for generating semantically diverse sentences using neural sequence-to-sequence (seq2seq) models. Conditioned on an input, seq2seq models typically produce semantically and syntactically homogeneous sets of sentences and thus perform poorly on one-to-many sequence generation tasks. Our two-stage approach improves output diversity by conditioning generation on locality-sensitive hash (LSH)-based semantic sentence codes whose Hamming distances highly correlate with human judgments of semantic textual similarity. Though it is generally applicable, we apply COD3S to causal generation, the task of predicting a proposition's plausible causes or effects. We demonstrate through automatic and human evaluation that responses produced using our method exhibit improved diversity without degrading task performance.

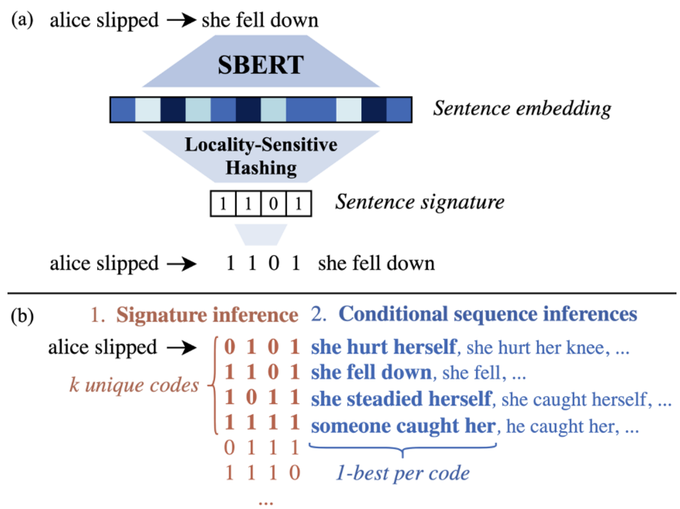

Figure 21: Example of semantic bit signatures as guidance on text generation.

The approach is to train a sequence to sequence model with the addition of a prefix on the target side that represents the semantic encoding of the part of "meaning space" we wish the decoder to focus on in its generation. At training time the model sees the signature, and the target sentence. At test time the model first decodes a set of signatures, with a constraint on a certain minimum hamming distance between signatures, and then we can take the one-best decode that continues from each signature.

We discretize sentences in the training via Locality-sensitive hashing (LSH; Indyk and Motwani, 1998) which maps high dimensional vectors into low-dimensional sketches for quick and accurate similarity comparison under measures such as cosine or Euclidean distance. We use the popular variant by Charikar (2002), which computes a discrete b-bit signature:

$$\cos(\vec{u},\vec{v}) = \frac{\vec{u}\cdot\vec{v}}{|\vec{u}||\vec{v}|} \approx \cos\left(\frac{\pi}{b}\sum_{i=1}^{b} \mathbb{1}\{\mathtt{LSH}_i(\vec{u}) \neq \mathtt{LSH}_i(\vec{v})\}\right)$$

Where Hamming distance between two LSH signatures approximates the cosine distance of the underlying vectors. We use the pretrained S-BERT encoder after internal experiments that verified cosine on its representations correlated strongly with human measures of similarity we were concerned with.

Our experiments focused again on CausalBank, where the resultant system allowed us to type in an event and then ask for likely causes, or effects, to that event as predicted by the model. Examples are provided below.

Cause Input: my favorite song came on the radio		
Bin Medoid	*I will try this version for sure*	*I was quite excited to finally experience it*
Ranked Predictions	**I decided to listen to it** I decided to hear it I figured I'd try it	**I was excited to hear it again** I was pleasantly surprised to hear it I'm glad to see it here
Effect Input: the executive decided not to hire the applicant		
Bin Medoid	*I knew that they expected it*	*they are what earn you cash*
Ranked Predictions	**they knew she was not qualified** they knew it would be a mistake she knew she had to	**they could not afford the payments** it would cost them money she was paid

Figure 22: Example outputs of COD3S model.

More details appear in Weir et al. (2020b)

We then extended these ideas to the document level, pairing with ideas in constrained decoding and explicit symbolic constraints derived from FrameNet.

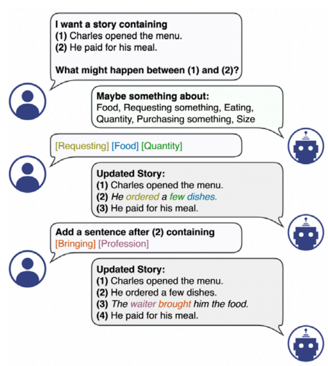

Figure 23: The proposed generation model, applied to the interactive story generation task.

A. Iterative Refinement

User: (I) Alice went to the grocery store. (II) [Commerce_buy]
System: (IIA) She bought all the ingredients for a cake. (IIB) She bought a new pair of shoes. (IIC) She bought a lot of fruits and veggies.
U: Choose (IIA) as (II) and infer content after (II)
S: [Food], [Cooking], [Ingredients], [Desirability], [Time_Collocation]
U: Choose [Desirability] [Cooking]
S: (I) Alice went to the grocery store. (II) She bought all the ingredients for a cake. *(III) She made the best cake she ever had.*
U: Infer content to replace (I)
S: [Food], [Deciding], [Social_Event], [Building], [Quantity]
U: choose [Social_Event]
S: *(I) Mary wanted to make a cake for her birthday.* (II) She bought all the ingredients for a cake. (III) She made the best cake she ever had.
U: Insert sentence about [Motion] at (II) and sentence about [Temporal_Collocation] at (IV)
S: (I) Mary wanted to make a cake for her birthday. *(II) She went to the store.* (III) She bought all the ingredients for a cake. *(IV) That afternoon, she baked the cake in the oven.* (V) She made the best cake she ever had.

B. Generation from Skeleton

User: (I) Bob went to Paris. (II) [Buildings] (III) [Experience] (IV) [Motion] 5 [Calendric_Unit]
System: (I) Bob went to Paris. *(II) He went to see the Louvre museum. (III) He saw lots of paintings and sculptures. (IV) He went to see the Eiffel Tower. (V) It was the best day of Bob's life.*

U: (I) Emma loved writing and wanted to be a writer. (II) [Deciding] (III) [Practice] (IV) [Publishing][Text] (V) [Fame]
S: (I) Emma loved writing and wanted to be a writer. *(II) She decided to enter a writing contest. (III) Emma's practice paid off and she won first prize. (IV) Emma was so proud of herself that she published a book. (V) Now Emma is a famous author.*

C. Diverse Candidate Generation

User: Infer candidates after (I) Charles slipped on a banana peel.
System: (IIA [Body_Parts]) He skinned his knee badly. (IIB [Motion_Directional]) He fell down on the ground. (IIC [Spatial_Contact]) He landed on his arm and broke it. (IID [Kinship]) His mother brought him to the emergency room. (IIE [Causation]) It caused him to fall down the stairs.

U: Infer candidates before (II) She hired a lawyer.
S: (IA [Submitting_Documents]) She filed a civil lawsuit. (IB [Trial]) She sued for breach of contract. (IC [Personal_Relationship]) She filed for divorce. (ID [Awareness]) She didn't know how to defend herself. (IE [Desiring]) she did not want to go to jail.

D. Counterfactual Story Rewriting

User: (I) Alec's daughter wanted more blocks to play with. (II) Alec figured that blocks would develop her scientific mind. (III) Alec bought blocks with letters on them. (IV) Alec's daughter made words with them rather than structures. (V) Alec was happy to see her developing her verbal ability.
Replace (II) with "Alec could not afford to buy new blocks for his daughter" and rewrite the last three sentences.
Parser: (III) [Containers] (IV) [Text_Creation] (V) [Emotion_directed]
System: (I) Alec's daughter wanted more blocks to play with. (II) Alec could not afford to buy new blocks for his daughter. *(III) Alec's daughter begged him to buy her blocks. (IV) Alec wrote a letter to Santa Claus himself. (V) She was very happy when he wrote back.*

Figure 24: A. Iterative story refinement, B. Surface realization from a frame semantic skeleton, C. Diverse candidate generation using model frame, D. Counterfactual story revision.

This effort allowed for taking examples of sequences of event types (schemas) and interactively expanding these into stories, where we can start with an arbitrary amount of the story pre-populated as text and then ask "what if" style questions that allows the model to hypothesize possible sentences that would complete the schema, based on the schema's top level constraints. The framework extended our prior work in lexically constrained decoding, allowing for specifying ordered sets of symbols that needed to appear in the output. For example, "Generate text such that one of the words *(bought, purchase, sold)* appears, followed by text that includes one of *(built, assembled, made)*". This was paired with a modification of a text infilling strategy, where a large corpus was preprocessed with a state of the art FrameNet semantic parser, and we learned to contextually substitute FrameNet frame symbols with the text that led to their prediction. When applying this in a text generation scenario, this allows us to generate arbitrary text, conditioned on text before and after the context we wish to instantiate, influenced by the kinds of FrameNet events we wish the text to evoke. We envisioned the combination of these two methods as a "post-TA2" capability to be added to our intended Schema Curation Workbench in later phases of the program, for a human in the loop mechanism for predicting events not explicitly mentioned in context. More details on this effort appear in Ou et al. (2021).

3.10 Reasoning with Rules

In early KAIROS Quizlets, we observed documents that were procedural. For example, instructions on how to modify a drone to carry a package. Such content represents a distinct style of language conveyance of schematic knowledge: rather than inducing schemas via abstracting from multiple examples of a complex scenario, this style of language requires direct interpretation, from a single document directly to a schema.

This motivated us in part to develop a corpus of legal documents: case descriptions and their results, coupled to legal statutes that govern those cases. We manually encoded a small set of cases into symbolic form, and encoded the statutes into prolog. By design, we can perfectly resolve our legal cases based on the prolog encoding, and then contrasted this against contemporary approaches that encode all salient text into dense representations and attempt to perform non-symbolic inference. We found in experiments that such non-symbolic approaches, even when coupled to state of the art contextual encoders, are not capable of well-addressing these legal reasoning tasks. This serves as a non-military, non-KAIROS domain that further motivates the consideration of explicitly symbolic representations as part of reasoning over content.

Research in this direction continued to pursue better versions of reasoning on legal statutory language, details of which appear in Holzenberger et al. (2020) and Holzenberger and Van Durme (2021).

4.0 RESULTS AND DISCUSSION

Regarding the performer evaluation in Phase 1 of the program, our team participated in the TA1 and TA2 components. For TA1, centered on the creation of a schema library, we approached the task with the understanding that: (a) there was allowance for a human in the loop, with no time constraints; and (b) knowledge of domain was provided ahead of time. The goal for TA1 was maximally enabling useful TA2 inferences. We therefore stated in program discussions with DARPA and other performers that the logical solution was to build a good user interface for creating schemas, and to provide the LDC definitions of the in-domain assumed schemas to humans for ensuring a maximally correct library. We did this, using tools described in earlier in the report, leading to a comprehensive library that well-captured the intended Phase 1 domain. To our knowledge, our schema library was optimal under the guidance provided for the evaluation. We believe that a human in the loop is a critical component of any future use of schemas if the intent is to mirror practical situations an analyst would employ KAIROS themed technologies. With this said, future evaluations might wish to quantify the amount of allowable human in the loop.

For TA2, we built a framework employing Lucene and Probabilistic Soft Logic. Given an arbitrary TA1 library, we would index the schemas as if they were individual documents, based on their specified event primitives. Given a knowledge graph on which to compute schema inference, we could treat that graph as a bag-of-events query, and retrieve those schemas from the library with the best number of matching event types. We then, for each such schema, employed a reasoning process through Probabilistic Soft Logic that matched events from the knowledge graph to the schema, as well as arguments. Rules were authored specific to the mechanics of the evaluation and agreed upon structures of the Phase 1 schemas, but not specific to the in-domain schema types. Related details can be found in Weber et al. (2021).

5.0 CONCLUSIONS

The JHU team under KAIROS pursued novel research in areas such as temporal understanding of language, causality, knowledge contained within large language models, methods for schema induction, and protocols for rapid human curation. Throughout our participation in the program we advocated for clear specifications of the goals of schemas and their use in inference, and in the practical power of clean user interfaces and the use of a human in the loop. The current state of the art in multi-modal conversion of unstructured to structured representations is still not sufficient for fully automatic reasoning, and even if it were, humans are not yet prepared to trust the result. We therefore pursued our work in the KAIROS program with the belief that these technologies would be part of an interactive analyst workflow, leading to a series of efforts along this theme.

6.0 REFERENCES

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1, pages 86–90.

Anton Belyy and Benjamin Van Durme. 2020. Script induction as association rule mining. In Proceedings of the First Joint Workshop on Narrative Understanding, Storylines, and Events, pages 55–62, Online. Association for Computational Linguistics.

Ryan Culkin, J. Edward Hu, Elias Stengel-Eskin, Guanghui Qin, and Benjamin Van Durme. 2021. Iterative Paraphrastic Augmentation with Discriminative Span Alignment. Transactions of the Association of Computational Linguistics.

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

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

Nils Holzenberger, Andrew Blair-Stanek and Benjamin Van Durme. 2020. A Dataset for Statutory Reasoning in Tax Law Entailment and Question. In Proceedings of the 2020 Natural Legal Language Processing (NLLP) Workshop.

Nils Holzenberger and Benjamin Van Durme. 2021. Factoring Statutory Reasoning as Language Understanding Challenges. In Proceedings of the Association of Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Li, Zhongyang and Ding, Xiao and Liu, Ting and Hu, J. Edward and Van Durme, Benjamin. 2020. Guided Generation of Cause and Effect. Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, (IJCAI-2020).

McRae, K., Cree, G. S., Seidenberg, M. S., & McNorgan, C. 2005. Semantic feature production norms for a large set of living and nonliving things. Behavior Research Methods, 37(4), 547–559.

Sebastian Nagel. 2016. Cc-news.

Jiefu Ou, Nathaniel Weir, Anton Belyy, Felix Yu, and Benjamin Van Durme. 2021. InFillmore: Frame-Guided Language Generation with Bidirectional Context. Proceedings of StarSem.

Adam Poliak, Max Fleming, Cash Costello, Kenton Murray, Mahsa Yarmohammadi, Shivani Pandya, Darius Irani, Milind Agarwal, Udit Sharma, Shuo Sun, Nicola Ivanov, Lingxi Shang, Kaushik Srinivasan, Seolhwa Lee, Xu Han, Smisha Agarwal, and João Sedoc. 2020. Collecting Verified COVID-19 Question Answer Pairs. Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020

Prince, E. F. 1978. On the function of existential presupposition in discourse. In Papers from the regional meeting of the Chicago Linguistics Society. chicago, ill (Vol. 14, pp. 362–376).

Mitchel Resnick, John Maloney, Andres Monroy-Hernandez, Natalie Rusk, Evelyn Eastmond, Karen ´ Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, et al. 2009. Scratch: programming for all. Communications of the ACM, 52(11):60–67.

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

Noah Weber, Rachel Rudinger, and Benjamin Van Durme. 2020. Causal inference of script knowledge. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).

Noah Weber, Anton Belyy, Nils Holzenberger, Rachel Rudinger and Benjamin Van Durme. 2021. Schema Curation via Causal Association Rule Mining. arXiv.

Nathaniel Weir, Adam Poliak, Benjamin Van Durme. 2020a. Probing Neural Language Models for Human Tacit Assumptions. Proceedings of CogSci. https://arxiv.org/abs/2004.04877

Weir, Nathaniel and Sedoc, Joao and Van Durme, Benjamin. 2020b. COD3S: Diverse Generation with Discrete Semantic Signatures. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP).

Patrick Xia, Guanghui Qin, Siddharth Vashishtha, Yunmo Chen, Tongfei Chen, Chandler May, Craig Harman, Kyle Rawlins, Aaron Steven White, and Benjamin Van Durme. 2021. LOME: large ontology multilingual extraction. CoRR, abs/2101.12175.

APPENDIX A – PUBLICATIONS AND PRESENTATIONS

Anton Belyy and Benjamin Van Durme. 2020. Script induction as association rule mining. In Proceedings of the First Joint Workshop on Narrative Understanding, Storylines, and Events, pages 55–62, Online. Association for Computational Linguistics.

Ryan Culkin, J. Edward Hu, Elias Stengel-Eskin, Guanghui Qin, and Benjamin Van Durme. 2021. Iterative Paraphrastic Augmentation with Discriminative Span Alignment. Transactions of the Association of Computational Linguistics.

Gantt, William, Lelia Glass, and Aaron Steven White. in press. Decomposing and Recomposing Event Structure. *Transactions of the Association for Computational Linguistics*.

Nils Holzenberger, Andrew Blair-Stanek and Benjamin Van Durme. 2020. A Dataset for Statutory Reasoning in Tax Law Entailment and Question. In Proceedings of the 2020 Natural Legal Language Processing (NLLP) Workshop.

Nils Holzenberger and Benjamin Van Durme. 2021. Factoring Statutory Reasoning as Language Understanding Challenges. In Proceedings of the Association of Computational Linguistics.

Li, Zhongyang and Ding, Xiao and Liu, Ting and Hu, J. Edward and Van Durme, Benjamin. 2020. Guided Generation of Cause and Effect. Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, (IJCAI-2020).

Jiefu Ou, Nathaniel Weir, Anton Belyy, Felix Yu, and Benjamin Van Durme. 2021. InFillmore: Frame-Guided Language Generation with Bidirectional Context. Proceedings of StarSem.

Adam Poliak, Max Fleming, Cash Costello, Kenton Murray, Mahsa Yarmohammadi, Shivani Pandya, Darius Irani, Milind Agarwal, Udit Sharma, Shuo Sun, Nicola Ivanov, Lingxi Shang, Kaushik Srinivasan, Seolhwa Lee, Xu Han, Smisha Agarwal, and João Sedoc. 2020. Collecting Verified COVID-19 Question Answer Pairs. Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020

Vashishtha, Siddharth, Adam Poliak, Yash Kumar Lal, Benjamin Van Durme, and Aaron Steven White. 2020. Temporal Reasoning in Natural Language Inference. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 4070–4078. Online: Association for Computational Linguistics.

Vashishtha, Siddharth, Benjamin Van Durme, and Aaron Steven White. 2019. Fine-Grained Temporal Relation Extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2906–2919. Florence, Italy: Association for Computational Linguistics.

Noah Weber, Rachel Rudinger, and Benjamin Van Durme. 2020. Causal inference of script knowledge. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).

Noah Weber, Anton Belyy, Nils Holzenberger, Rachel Rudinger and Benjamin Van Durme. 2021. Schema Curation via Causal Association Rule Mining. arXiv.

Nathaniel Weir, Adam Poliak, Benjamin Van Durme. 2020a. Probing Neural Language Models for Human Tacit Assumptions. Proceedings of CogSci. https://arxiv.org/abs/2004.04877

Weir, Nathaniel and Sedoc, Joao and Van Durme, Benjamin. 2020b. COD3S: Diverse Generation with Discrete Semantic Signatures. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP).

White, Aaron Steven, Elias Stengel-Eskin, Siddharth Vashishtha, Venkata Subrahmanyan Govindarajan, Dee Ann Reisinger, Tim Vieira, Keisuke Sakaguchi, Sheng Zhang, Francis Ferraro, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. 2020. The Universal Decompositional Semantics Dataset and Decomp Toolkit. In *Proceedings of The 12th Language Resources and Evaluation Conference*, 5698–5707. Marseille, France: European Language Resources Association.

Patrick Xia, Guanghui Qin, Siddharth Vashishtha, Yunmo Chen, Tongfei Chen, Chandler May, Craig Harman, Kyle Rawlins, Aaron Steven White, and Benjamin Van Durme. 2021. LOME: large ontology multilingual extraction. CoRR, abs/2101.12175.

LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS

ARM	Association Rule Mining
CE	Complex Event
COD3S	COnstrained Decoding with Discrete Semantic Signatures
DARPA	Defense Advanced Research Projects Agency
DOD	Department of Defense
LDC	Linguistic Data Consortium
LM	Language Model
LSH	Locality Sensitive Hashing
MNLI	Multi-genre Natural Language Inference
MRR	Mean Reciprocal Rank
NLI	Natural Language Inference
NMT	Neural Machine Translation
PMI	Pointwise Mutual Information
STA	Stereotypic Tacit Assumption
UDS	Universal Decompositional Semantics
UI	User Interface