CHRONOS: A Schema-Based Event Understanding and Prediction System

Maria Chang, Achille Fokoue, Rosario Uceda-Sosa, Parul Awasthy, Ken Barker, Sadhana Kumaravel, Oktie Hassanzadeh, Elton Soares, Tian Gao, Debarun Bhattacharjya, Radu Florian, Salim Roukos

IBM Research

{maria.chang, sadhana.kumaravel1, eltons}@ibm.com,

{achille, rosariou, awasthyp, kjbarker, hassanzadeh, tgao, debarunb, raduf, roukos}@us.ibm.com

Abstract

Chronological and Hierarchical Reasoning Over Naturally Occurring Schemas (CHRONOS) is a system that combines language model-based natural language processing with symbolic knowledge representations to analyze and make predictions about newsworthy events. CHRONOS consists of an event-centric information extraction pipeline and a complex event schema instantiation and prediction system. Resulting predictions are detailed with arguments, event types from Wikidata, schema-based justifications, and source document provenance. We evaluate our system by its ability to capture the structure of unseen events described in news articles and make plausible predictions as judged by human annotators.

Introduction

Analysis and prediction of newsworthy events is an important capability with potential uses in decision support, education, and intelligence gathering. Real-world application of this capability requires natural language understanding that can produce (or at least maintain) rich representations of interconnected events described across multiple documents. These event representations must take into account not only event types and event-argument structures but temporal and causal dynamics between events as well. In order to earn the trust of human stakeholders, predictions must be reliable and explainable. This use case and its challenges are the focus of the DARPA Knowledge-directed Artificial Intelligence Reasoning Over Schemas (KAIROS) program.

Modern generative large language models (LLMs) are capable of producing predictions, reasoning traces and explanations that are *believable* but not necessarily reliable or authentic. Symbolic reasoning and faithfulness are two well known limitations to LLMs [Valmeekam et al. 2022; Turpin et al. 2023], which can pose unacceptable risks in certain application settings. Prompting techniques can elicit explanations that can dramatically improve performance on downstream tasks, but there is evidence that LLM-generated explanations do not reflect the real factors influencing output [Turpin et al. 2023]. Furthermore, we must consider the possibility that the data being analyzed by the system is confidential, classified, and/or scarce. This suggests that rapid, training-free domain adaptation is a requirement as well.

We present CHRONOS, a schema-based event understanding and prediction system developed to address the goals of the DARPA KAIROS program. CHRONOS analyzes news articles with respect to *complex event schemas*, to classify narratives across multiple documents and to make detailed predictions about unobserved events. CHRONOS uses deep learning models for a variety of NLP tasks (machine translation, information extraction, semantic parsing, text classification) and symbolic knowledge structures to represent the compositional and hierarchical structure of complex events. We believe that the use of event schemas, the multi-document setting, and the justification/provenance requirements make this application unique. As an emerging application, we show that our system is able to correctly classify events across 5 distinct domains: terrorist attacks, disease outbreaks, violent coups d'état, riots, and hazardous spills. We also show that our system makes plausible predictions evaluated by human assessment.

Related Work

Multi-document summarization is related to the narrative extraction step of the KAIROS use case. Well known datasets in this area are: DUC 2004[1], TAC 2011[2], Multi-news [Fabbri et al. 2019], and Wikipedia Current Events Portal (WCEP) [Ghalandari et al. 2020]. Although many of these datasets have reference summaries that come from only one document [Wolhandler et al. 2022] and they do not have any structured event requirements.

Event argument extraction is a notoriously difficult problem and has received attention in various settings, most commonly within sentence event argument extraction [Walker et al. 2006], but also document-level [Li, Ji, and Han 2021; Ebner et al. 2020], and more recently open-vocabulary argument role prediction [Jiao et al. 2022].

Our work is most closely related to the RESIN project [Du et al. 2022; Wang et al. 2022], which is a schema-guided event prediction system for the same type of newsworthy event analysis use case. Our information extraction pipeline differs in its use of zero-shot text classification and AMR

Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

[1] https://duc.nist.gov

[2] https://tac.nist.gov

parses to extract event-argument structures. We extend this area of work by including additional complex event datasets and human judgements of prediction plausibility.

System Overview

The objective of the CHRONOS system is to understand complex events described across multiple documents and to provide meaningful predictions about those events. It consists of three main components: (1) a complex event schema library, (2) an event-centric information extraction pipeline, and (3) a schema instantiation and prediction system.

Complex Event Schemas

A *complex event schema* is an abstraction that describes the structure of a complex event type. Complex events involve multiple participants/entities and they persist over an interval of time such that their subevents are noteworthy. Complex event schemas are hierarchical and compositional – schemas can have specializations (e.g. a foodborne disease outbreak is a more specific version of a disease outbreak) and they can be composed of other schemas (e.g. the Olympic Games consist of many individual competitions, each having their own complex structure). This composition results in a tree-like structure, where the children are the subevents. A *schema library* is a set of complex event schemas.

A schema is *instantiated* when it is used to describe a particular instance of that schema, e.g. the United States Presidential Election of 2020 is an instantiation of an election schema. Schemas can facilitate inference and prediction because they capture the archetypical structure of observed events. That is, the presence of an event, entity, or relation in a schema indicates its presence in events of the same type, even when not directly observed.

Figure 1 shows a partial visualization for our event schema for hazardous spills. Each node represents a (sub)-event. The root event represents not just the spill itself, but all related, typical events, such as cleanup, containment, restoration, and so on. Dotted directed edges indicate subevent relationships (e.g. Prevention & Preparedness is a subschema of Hazardous Spill). Solid directed edges represent temporal relations (e.g. Spill Detection happens before Authorities Notified). Leaf nodes represent *primitive events*. Primitive events have roles specified with role naming conventions of Propbank [Palmer, Gildea, and Kingsbury 2005] and DARPA Wikidata (DWD) [Spaulding et al. 2023]. Importantly, entities that fill these roles are reused in other parts of the schema to facilitate co-references in the story. For example, the agent roles are shared for all sub-events of Prevention & Preparedness, meaning that the organization that evaluates the spill response is typically also in charge of developing prevention strategies. Note that type constraints and granularity are important here, as this role-sharing typically makes sense when agents are described at a higher level granularity, e.g. referring to the environmental agency evaluating the spill response rather than an individual person. Another aspect of the schema is that logical constraints are defined to facilitate prediction. Each parent event has either an AND, OR, or XOR logical gate. Logical gates indicate that the instantiation of an event entails instantiation of all of its sub-events (AND), at least one of its sub-events (OR), or exactly one (XOR) sub-event.

We have experimented with several approaches for creating complex event schemas that use a combination of automatic extraction (e.g. from Wikidata and Wikipedia), generation (via LLM prompting), and human curation. We leave the technical presentation of those methods for another paper. Here, we focus on the application that instantiates schemas that are supplied as part of a fixed, pre-populated schema library.

Information Extraction Pipeline

The inputs to our information extraction (IE) pipeline are (1) a possibly noisy multilingual batch of documents that describe a complex event, and (2) a schema library with schemas for newsworthy events across different domains (Figure 2). Our information extraction pipeline analyzes the batch of documents to produce an event graph that summarizes its main events, entities, and relations. Each event, entity, and relation is associated with a concept (either Q or P node) in Wikidata [Vrandečić and Krötzsch 2014]. Events and their arguments are also mapped to Propbank frames [Palmer, Gildea, and Kingsbury 2005]. These links enable integration with open knowledge resources, such as the recently developed DARPA Wikidata (DWD) [Spaulding et al. 2023]. Each document batch may by noisy in the sense that there may be irrelevant/distractor documents.

Document pre-processing. We pre-process all documents using in-house tools prior to English Information Extraction. The first step is language detection to determine the appropriate sentence segmentation and translation models. We do sentence segmentation in the source language, followed by sentence-level neural machine translation to English. This allows us to maintain sentence alignment to report sentence-level, source-language provenance in our output, as required by the program. The final pre-processing step is English tokenization so that all subsequent Information Extraction steps requiring tokenization are working with a shared representation.

Zero-Shot Text Classifier using Natural Language Inference (ZSTC). We use a zero-shot text classifier throughout the system to inform semantic interpretations. A zero-shot classifier allows us to specify target classes dynamically without needing to train a model on data specific to the task. Our classifier is a standard Natural Language Inference (NLI) model: a pre-trained language model (RoBERTa-Large [Liu et al. 2019] in our case) fine-tuned on the multi-genre textual entailment corpus (MultiNLI or MNLI) for the NLI task from the RepEval workshop [Williams, Nangia, and Bowman 2018]. Instances in the MNLI corpus are pairs of sentences with a label indicating whether the first sentence (the premise) *entails*, is *contradicted* by, or is *neutral* with respect to the second sentence (the hypothesis). NLI models can be used for zero-shot text classification by supplying the text to be classified as the premise, and a textual representation of a target class (i.e. class label or description, queryable from Wikidata) as the hypothesis. The target class whose textual representation has the highest En-

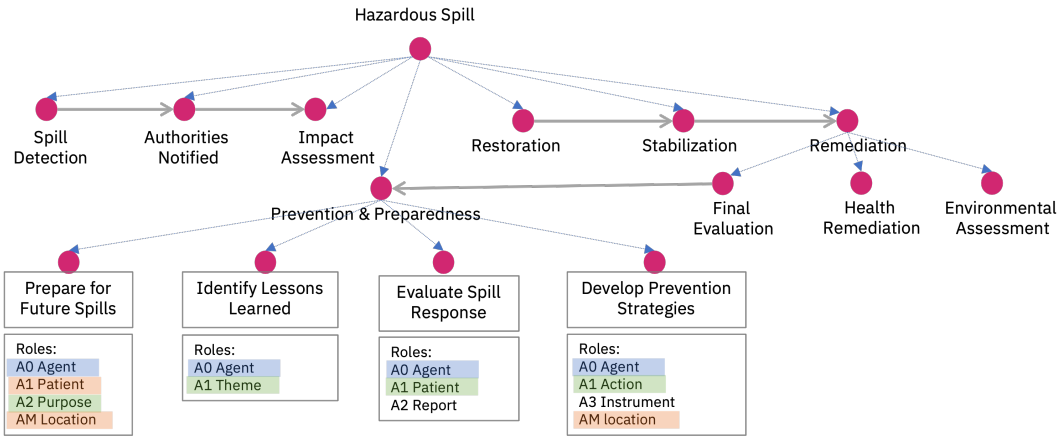

Figure 1: A partial illustration of the Hazardous Spill schema. Nodes are (sub-)events. Dotted edges are sub-event relations and solid edges are temporal relations. Roles highlighted in the same color are co-referent/shared across events.

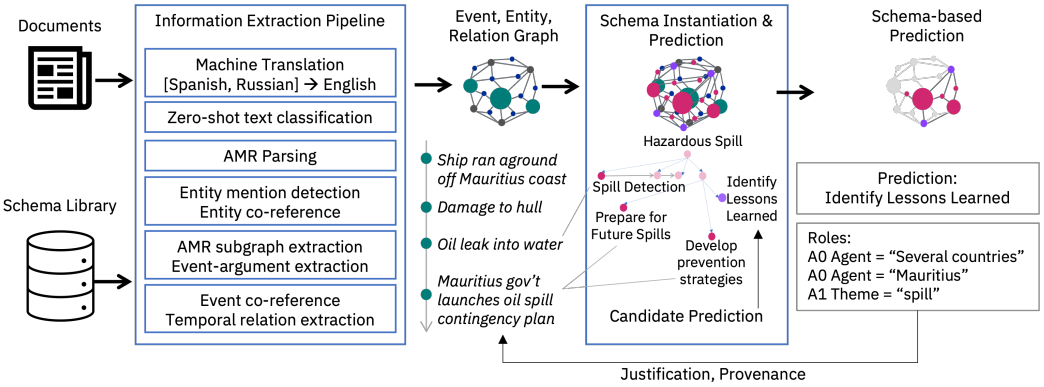

Figure 2: Overview of CHRONOS system.

tailment score is the predicted class for the text. We use this ZSTC model to assign Wikidata event classes to sentences. One advantage of this approach is that using an entirely different schema library (and hence a different event vocabulary) requires no re-training.

Structured Event Extraction. We first extract entity representations using entity mention detection [Moon et al. 2019], relation extraction [Ni et al. 2020], and coreference models trained on KLUE [Florian et al. 2010] and ACE05 [Walker et al. 2006] datasets. In parallel, we generate abstract meaning representations (AMR) [Banarescu et al. 2013] for sentences using a publicly available parser [Zhou et al. 2021]. At this stage in the pipeline we have event type labels at different levels of granularity. At the sentence level, we have Wikidata event classes from our ZSTC service. At the sub-sentence level, the AMR graph has verb frame nodes that can be interpreted as events, but there are typically many more verb frame nodes in a single sentence than there are salient events. To align these two representations we extract event phrases corresponding to AMR subgraphs where the root is a predicate, and find the most similar Wikidata qnode to it using SBERT embeddings of the qnode definition and examples [Reimers and Gurevych 2019]. Using the ZSTC results as a form of supervision, we only include the subset of such AMR subgraph-to-qnode pairs if the qnode also appears in the classes selected by ZSTC. We also use SBERT embeddings to map the entities in the AMR subgraphs to entity mentions extracted from text (with entity coreference resolved). Once we have these event representations, we resolve event coreference using a model based on [Yu, Yin, and Roth 2022]. We detect event modality using a model trained on the ACE05 dataset [Walker et al. 2006], and extract temporal relations between events using a model trained on the MATRES dataset [Ning, Wu, and Roth 2018]. At the end of the pipeline, we have a structured representation of the multi-document narrative that consists of (1) event-argument structures, where each event and argument has a type label from Wikidata, (2) entity-entity relations, where each relation is linked to a property (P node) in Wikidata, and (3) before/after relations between some events. Note that while linking to Wikidata is useful for downstream analysis and filtering, it is not required by the IE pipeline.

Schema Matching & Prediction

The schema matching and prediction module takes the event graph produced by the IE pipeline and the schema library as inputs and produces an *instantiated schema*. As mentioned above, the schema library is created through a separate process whose details will be presented in another paper. The instantiated schema includes three types of events: (1) *matched events*, which are events that were observed in the documents and matched to an event in the complex event schema, (2) *unmatched events*, which are events observed in text for which the system finds no corresponding schema event, but are still considered important to the overall story, and (3) *candidate predictions*, which we define as all schema elements that were not matched to an event observed in text. In this sense, the schema enables the system to "fill in the gaps" of the narrative and reduce the size of the overall event graph. This procedure occurs in three stages: (1) narrative classification, (2) fine-grained matching, (3) instantiation and prediction.

Narrative Classification. In the first stage, the goal is to find the schema that best matches the overall narrative of the event graph. We treat this as a multi-class classification problem, and reuse our ZSTC model to determine which root schemas are most likely entailed by the events extracted from source documents. Specifically, we obtain classification scores between each sentence in the event graph (i.e. from a source document) and each complex event schema (i.e. each class label of each root in the schema library). The class label with the greatest average score across all sentences is chosen as the best matching schema. This simple approach works surprisingly well and avoided errors that occurred in prior versions that took a bottom-up matching approach.

Fine-grained Matching. In the second stage, we align the events in the event graph to those in the schema tree. We again use the ZSTC model to compute classification scores between observed event descriptions and schema event descriptions. We also compute classification scores between observed event argument descriptions and candidate schema argument matches. We use a gap-thresholding technique on event classification scores (i.e. only keeping classifications above the largest score gap). It is possible for an observed event to match multiple schema events. These individual matches then propagate up the schema hierarchy, instantiating higher level schemas to build multiple graph-to-tree candidate matches. For example, a document batch describing a hazardous spill may have multiple instantiations of the hazardous spill schema (Figure 1) that differ in which entities were included as arguments.

Prediction. In the final stage, the system selects a subset of unmatched schema elements (those whose parent is instantiated and has an AND logical gate) as candidate predictions. At this stage we have multiple graph-to-tree matches. We retain the best scoring matches by using two score augmentation techniques. First, we penalize matches that violate temporal orderings found in the observed event graph. Second, we apply *importance weighting* to event match scores, such that matches between events that are important to the narrative or schema are given a score boost. The importance weights are determined by the ZSTC model, which helps us estimate which events most strongly entail a given root schema or observed narrative. We found that this approach greatly reduces the possibility that frequent but less salient events drive the entire schema match. For example, in a terrorist bombing schema, the presence of a detonation event is much more indicative of a bombing than a travel event or even an injury event, despite all three being present in the schema. The final predictions consist of all the candidate predictions from the highest scoring match. In the example in Figure 2, the resulting prediction is that lessons learned will be identified, its justification is the set of matches to its schema siblings, and its provenance is the source document and sentence "*Several countries*, including France and Japan, are also assisting *Mauritius*, which has activated its national contingency plan for oil *spills*." This justification and provenance trail is required by the program and, unlike an explanation elicited from a generative language model, its faithfulness is guaranteed.

Experiments

Complex Events Dataset

We evaluate our system on a dataset that consists of several smaller datasets created by the Linguistic Data Consortium (LDC) for the DARPA KAIROS program. The dataset is comprised of 34 unique complex events that span several different domains. Each complex event is accompanied by a small batch of relevant documents describing the event, such as news articles. Each document may be in English, Spanish, or Russian. We present statistics of the dataset in Table 1. Each story is a unique instance within the domain, e.g. the 2010 Deepwater Horizon Oil Spill could be a single story (with multiple documents) in the Hazardous Spill domain. These data will be made available through the LDC catalog under the research project name KAIROS [3]. Specifically, we use data corresponding to the Phase 1 Evaluation Data, Phase 2A Quizlet Data, Phase 2A Evaluation Data, and Phase 2B Evaluation Data packages of KAIROS.

Although there are no official training splits to these data, we froze development before gaining access to the 15 stories from the most recent LDC data package (Phase 2B Evaluation Data). All the stories in the Hazardous Spill, Riot, and Coup d'état domains (as well as some from all other domains) fall into this post-development category.

Task and Metrics

Our system analyzes each batch of documents and matches it to a schema event type. We asses the system's classification capabilities with schema instantiation accuracy (Acc), event match recall (MR-Events), and argument match recall (MR-Args). Event match recall is the proportion of extracted events that were matched to an event in the complex event schema. Argument match recall is the proportion of

[3] https://catalog.ldc.upenn.edu/search

Domain	# Stories	# Docs (per story)
Coup d'état	3	38 (12.67)
Disease outbreak (food)	4	47 (11.75)
Disease outbreak (general)	6	77 (12.83)
Hazardous Spill	3	40 (13.33)
Riot	3	44 (14.67)
Terrorist Attack (general)	5	64 (12.80)
Terrorist Attack (bombing)	10	120 (12.00)

Table 1: Descriptive statistics on Complex Events Dataset

Domain	Events	Entities	Relations
Coup d'état	227	431	148
Disease outbreak (food)	298	502	95
Disease outbreak (general)	318	517	120
Hazardous Spill	180	259	39
Riot	197	292	64
Terrorist Attack (general)	463	722	206
Terrorist Attack (bombing)	874	1380	346

Table 2: Total number of events, entities, and relations extracted by our IE pipeline. Across all domains, the system detected 389 unique event types, 50 unique entity types, and 8 unique relation types.

extracted arguments that were matched to roles in the complex event schema. These metrics give us a sense of coverage, but it is important to note that we do not aim for 100% match recall in this use case because source documents are much more detailed than the schema, so events that are relevant to a particular story but not to a generalized schema should not be matched. In other words, 100% recall would indicate that our schema is too detailed and therefore likely to produce implausible event predictions.

We assess the quality of our systems' predictions via human assessment. Human annotators judge the plausibility of each predicted event type (PP-Events) and each predicted argument (PP-Args). Plausibility is defined as the proportion of predictions that were rated plausible by one or both annotators. These judgements can be subjective, so we instructed annotators to answer the question: does the predicted event (or event argument) naturally follow from the general narrative of the story? For example, the prediction shown in Figure 2 was generated by our system after analyzing a batch of documents describing an oil spill in Mauritius. Annotators marked this prediction, that there would be some identification of lessons learned from the oil spill, as plausible. Annotators also marked each of the arguments shown as plausible: that Mauritius as well as other countries could identify lessons learned and that the lessons would be about the spill. Each complex event was annotated by two different annotators. Overall, annotators were in agreement for 88% of items (Cohen's kappa = 0.77).

Results

The CHRONOS system extracted a broad vocabulary of events for each of the 34 stories. Each event, entity, and relation is linked to a Wikidata label. Across all domains, the system detected 389 unique event types, 50 unique entity types, and 8 unique relation types. Totals for each domain are shown in Table 2.

The system instantiated the correct schema for most (32 out of 34) stories (Table 3). Both errors were in the general terrorist attack domain. In two out of four of those stories, the system incorrectly instantiated the terrorist attack bombing schema, when the stories were actually about mass shootings. In both cases, the mass shooting schema was the second best matching schema.

The match recall for events and arguments is moderate, with the system matching between 35-57% of extracted events and 35-54% of arguments to elements in the selected schema (Table 3). This is not very surprising considering that the event graphs for each story typically include hundreds of events, entities, and relations, some of which may be from distractor documents. Importantly, even if the event graph were at the perfect level of granularity, we would not aim for 100% recall because that would diminish our schema's ability to make predictions.

Table 4 shows metrics for predictions. Event prediction plausibility is nearly at ceiling, with the exception of the terrorist attack (general) domain, where the system instantiated the incorrect schema, leading to predictions having to do with explosives rather than shootings. However, predicting plausible arguments proved to be much more challenging. As noted above, unmatched schema elements are the source of all predictions, so plausibility of arguments depends on the appropriateness of the schema roles, the accuracy of the event-argument extraction from documents, and the schema instantiation system's ability to correctly match arguments. We experimented with different ways of filtering predictions arguments: (1) applying a simple confidence threshold, (2) applying a Wikidata superclass filter, and (3) a permissive filter that is the disjunction of 1 and 2.

A simple confidence threshold was determined by performing a confidence sensitivity analysis. We observed a substantial increase in plausibility when using a threshold of 0.01 and an ideal threshold around 0.1 (results were similar for thresholds between 0.05 and 0.15).

In many of the implausible predicted arguments, we found that the argument matching algorithm was likely too liberal, as we were seeing argument alignments that could clearly be ruled out with an analysis of the argument type labels. For example, in one of the Coup d'état stories, one of the predictions was about planning the coup, and one of the roles was *A0_ppt_thing_being_planned*, which had a schema constraint of *coup* (Q45382). However, the predicted argument was 'country' with *geographic entity* (Q27096213) as its type label. The Wikidata superclass filter queries the publicly available Wikidata query service to determine if the the type label for the extracted argument is a superclass of the type constraint for the schema element. This is similar in spirit to the text classification that occurs within the schema matching module via the ZSTC service, but instead of providing a score, allows us to explicitly rule out certain argument alignments. This filter greatly improves the plausibility rate for Hazardous Spill and Riot stories.

Lastly, we applied a permissive filter that only kept predictions that were above the confidence threshold or passed

Domain	Acc	MR-Events	MR-Args
Coup	100%	0.35	0.35
DOFood	100%	0.46	0.45
DOGen	100%	0.54	0.54
HazSpill	100%	0.57	0.52
Riot	100%	0.52	0.49
TerrAtk	50%	0.45	0.41
TerrIED	100%	0.45	0.42

Table 3: Match metrics across domains: root classification accuracy (Acc), average match recall for events (MR-Events), average match recall for event arguments (MR-Args).

Domain	Pr-Events	Pr-Args	PP-Events	PP-Args
Coup	6.33	4.67	100%	25%
DOFood	9.50	4.50	100%	46%
DOGen	8.00	6.50	100%	68%
HazSpill	6.33	8.00	100%	88%
Riot	6.00	4.00	100%	67%
TerrAtk	13.00	11.20	78%	46%
TerrIED	10.80	10.00	100%	59%

Table 4: Prediction metrics across domains: average number of predicted events per story (Pr-Events), average number of predicted arguments per story (PR-Args), prediction plausibility for events (PP-Events), and prediction plausibility for arguments (PP-Args). PP-Arg percentages are for predicted arguments that pass the permissive filter.

the Wikidata super class filter. The PP-Args numbers shown in Table 4 use this permissive filter. The impact of each of those filters is shown in Figure 3.

Challenges to Deployment

Our system underwent multiple evaluations under the DARPA KAIROS program where we received feedback, especially on the conciseness and clarity of system outputs. As mentioned above, the event graphs generated by our system often contain hundreds of entities, events, and relations. This presented a challenge on how to summarize the multi-document story, and how to present it in a human readable form. In addition to sheer size, graphical visualizations needed to accommodate multiple edges and distinguishing between conceptual relations between entities, compositional relations, and temporal relations between events, which became very challenging. As of this writing we employ a combination of textual (html outline) and graphical visualizations. This will likely evolve as the quality of predictions improve.

Related to visualization and usability, we are planning to integrate our pipeline with our schema curation tools. Currently these two operate independently, but combining them would enable a user to interactively analyze document batches and rapidly assess how changes to the schema library impact predictions.

In transitioning Information Extraction models into this

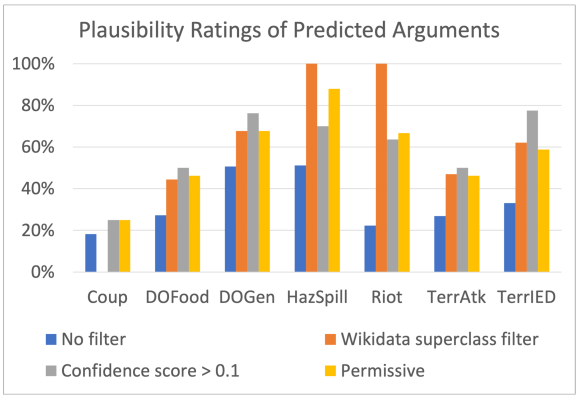

Figure 3: Percentage of predicted arguments that were rated plausible by at least one annotator.

application, we learned that IE vocabularies such as ACE/ERE/KLUE were insufficient for this use case. We gained expressivity by using larger vocabularies like Wikidata and Propbank for events. Utlimately, we moved to an almost entirely open vocabulary, where pre-trained models could be used to compute similarity and assign vocabulary labels as needed.

Moving forward, our biggest priorities are improving the quality of predictions, while maintaining faithful justifications and provenance links. We currently use plausibility as our prediction quality metric, but we would like to understand what other dimensions, such as how actionable a prediction is, might be useful. Currently our predictions can fall into the category of commonsense inferences. However, in a setting where predictions might inform some kind of intervention, then more sophisticated metrics might be required. In future implementations, LLMs may play a more direct role in generating predictions. If so, we will also explore if and how we can make guarantees related to LLM explanation faithfulness. We believe the field is moving in this direction and we hope that the KAIROS use case will further illustrate the importance of reliable, justifiable, and provenance linked predictions.

Acknowledgments

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA) KAIROS program. The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. Distribution Statement A (Approved for Public Release, Distribution Unlimited).

References

Banarescu, L.; Bonial, C.; Cai, S.; Georgescu, M.; Griffitt, K.; Hermjakob, U.; Knight, K.; Koehn, P.; Palmer, M.; and Schneider, N. 2013. Abstract Meaning Representation for Sembanking. In *LAW@ACL*.

Du, X.; Zhang, Z.; Li, S.; Yu, P.; Wang, H.; Lai, T.; Lin, X.; Wang, Z.; Liu, I.; Zhou, B.; Wen, H.; Li, M.; Hannan, D.;

Lei, J.; Kim, H.; Dror, R.; Wang, H.; Regan, M.; Zeng, Q.; Lyu, Q.; Yu, C.; Edwards, C.; Jin, X.; Jiao, Y.; Kazeminejad, G.; Wang, Z.; Callison-Burch, C.; Bansal, M.; Vondrick, C.; Han, J.; Roth, D.; Chang, S.-F.; Palmer, M.; and Ji, H. 2022. RESIN-11: Schema-guided Event Prediction for 11 Newsworthy Scenarios. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: System Demonstrations*, 54–63. Hybrid: Seattle, Washington + Online: Association for Computational Linguistics.

Ebner, S.; Xia, P.; Culkin, R.; Rawlins, K.; and Van Durme, B. 2020. Multi-Sentence Argument Linking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 8057–8077. Online: Association for Computational Linguistics.

Fabbri, A.; Li, I.; She, T.; Li, S.; and Radev, D. 2019. Multi-News: A Large-Scale Multi-Document Summarization Dataset and Abstractive Hierarchical Model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 1074–1084. Florence, Italy: Association for Computational Linguistics.

Florian, R.; Pitrelli, J.; Roukos, S.; and Zitouni, I. 2010. Improving Mention Detection Robustness to Noisy Input. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, 335–345. Cambridge, MA: Association for Computational Linguistics.

Ghalandari, D. G.; Hokamp, C.; Pham, N. T.; Glover, J.; and Ifrim, G. 2020. A Large-Scale Multi-Document Summarization Dataset from the Wikipedia Current Events Portal. *CoRR*, abs/2005.10070.

Jiao, Y.; Li, S.; Xie, Y.; Zhong, M.; Ji, H.; and Han, J. 2022. Open-Vocabulary Argument Role Prediction For Event Extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, 5404–5418.

Li, S.; Ji, H.; and Han, J. 2021. Document-Level Event Argument Extraction by Conditional Generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 894–908. Online: Association for Computational Linguistics.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692.

Moon, T.; Awasthy, P.; Ni, J.; and Florian, R. 2019. Towards Lingua Franca Named Entity Recognition with BERT. arXiv:1912.01389.

Ni, J.; Moon, T.; Awasthy, P.; and Florian, R. 2020. Cross-Lingual Relation Extraction with Transformers. arXiv:2010.08652.

Ning, Q.; Wu, H.; and Roth, D. 2018. A Multi-Axis Annotation Scheme for Event Temporal Relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1318–1328.

Palmer, M.; Gildea, D.; and Kingsbury, P. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Comput. Linguist.*, 31(1): 71–106.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Spaulding, E.; Conger, K.; Gershman, A.; Uceda-Sosa, R.; Brown, S. W.; Pustejovsky, J.; Anick, P.; and Palmer, M. 2023. The DARPA Wikidata Overlay: Wikidata as an ontology for natural language processing. In *Workshop on Interoperable Semantic Annotation (ISA-19)*, 1.

Turpin, M.; Michael, J.; Perez, E.; and Bowman, S. R. 2023. Language Models Don't Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting. *arXiv preprint arXiv:2305.04388*.

Valmeekam, K.; Olmo, A.; Sreedharan, S.; and Kambhampati, S. 2022. Large Language Models Still Can't Plan (A Benchmark for LLMs on Planning and Reasoning about Change). In *NeurIPS 2022 Foundation Models for Decision Making Workshop*.

Vrandečić, D.; and Krötzsch, M. 2014. Wikidata: A Free Collaborative Knowledgebase. *Commun. ACM*, 57(10): 78–85.

Walker, C.; Strassel, S.; Medero, J.; and Maeda, K. 2006. ACE 2005 multilingual training corpus. *Linguistic Data Consortium, Philadelphia*, 57.

Wang, H.; Zhang, Z.; Li, S.; Han, J.; Sun, Y.; Tong, H.; Olive, J.; and Ji, H. 2022. Schema-Guided Event Graph Completion. In *4th Conference on Automated Knowledge Base Construction*.

Williams, A.; Nangia, N.; and Bowman, S. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 1112–1122. New Orleans, Louisiana: Association for Computational Linguistics.

Wolhandler, R.; Cattan, A.; Ernst, O.; and Dagan, I. 2022. How "Multi" is Multi-Document Summarization? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 5761–5769. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Yu, X.; Yin, W.; and Roth, D. 2022. Pairwise Representation Learning for Event Coreference. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, 69–78. Seattle, Washington: Association for Computational Linguistics.

Zhou, J.; Naseem, T.; Fernandez Astudillo, R.; Lee, Y.-S.; Florian, R.; and Roukos, S. 2021. Structure-aware Fine-tuning of Sequence-to-sequence Transformers for Transition-based AMR Parsing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 6279–6290. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.