AFRL-RI-RS-TR-2024-073

CMU CHRONOS-KAIROS FINAL SYSTEMS DESCRIPTION

CARNEGIE MELLON UNIVERSITY

JUNE 2024

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

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

NOTICE AND SIGNATURE PAGE

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

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

AFRL-RI-RS-TR-2024-073 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /
JOHN SPINA
Work Unit Manager

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

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

REPORT DOCUMENTATION PAGE

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
		START DATE	END DATE
JULY 2024	FINAL TECHNICAL REPORT	AUGUST 2019	FEBRUARY 2024

4. TITLE AND SUBTITLE

CMU CHRONOS-KAIROS FINAL SYSTEMS DESCRIPTION

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
FA8750-19-2-0200	N/A	62303E
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**
N/A	N/A	R2XC

6. AUTHOR(S)

Teruko Mitamura, David R. Mortensen, Alex Hauptmann, Yiming Yang, Graham Neubig, Anatole Gershman, Alan Black, Zhiqi Cheng, Susan Holm, Yukari Yamakawa

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh PA 15218

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/RIED 525 Brooks Road Rome NY 13441-4505	AFRL/RI	AFRL-RI-RS-TR-2024-073

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

The CMU CHRONOS system addresses both TA1 and TA2 components of DARPA's KAIROS program, including generating, generalizing, composing and specializing schemas from multimedia sources (text, audio, image, video), as well as representing and reasoning over the induced schemas in a knowledge base.

This report from the CMU CHRONOS team describes the CHRONOS systems pipeline. The pipeline includes schema visualization and curation tool called Eratosthenes; practices for schema library creation and curation, a system for extracting events, entities, and relations from text, video, and speech; a schema matching an instantiation system; and a prediction system.

15. SUBJECT TERMS

Knowledge-directed AI, Reasoning over schemas, understanding complex events, symbolic reasoning

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT	b. ABSTRACT	C. THIS PAGE		
U	U	U	SAR	42

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
JOHN SPINA	NA

Contents

List of Figures

List of Tables

1 SUMMARY

The CMU CHRONOS system addresses both TA1 and TA2 components of DARPA's KAIROS program, including generating, generalizing, composing and specializing schemas from multimedia sources (text, audio, image, video), as well as representing and reasoning over the induced schemas in a knowledge base.

This report from the CMU CHRONOS team describes the CHRONOS systems pipeline. The pipeline includes schema visualization and curation tool called Eratosthenes; practices for schema library creation and curation, a system for extracting events, entities, and relations from text, video, and speech; a schema matching an instantiation system; and a prediction system. More detailed descriptions of each module development are found in the quarterly reports from October 2019 through June 2023.

Since work on KAIROS Plus is still on-going, we will not describe it in this report. The KAIROS Plus has developed a large language model-based schema induction system that works in conjunction with human curators to rapidly produce large libraries of high-quality schemas across many domains. We plan to send a separate report at the end of December 2023.

2 INTRODUCTION

The CMU CHRONOS (Chronological and Hierarchical Reasoning Over Newly Observed Schemas) system consists of two pipelines with several components. In TA1, there is a extraction system for text, which can handle English, Russian and Spanish language texts. CHRONOS also includes a multimedia (including speech) extraction system. An important human-computer interaction component is the schema visualization and curation tool called Eratosthenes. This tool makes it easier to see complicated schema structures and curate them (when necessary) via human feedback. We also describe human schema curation methods and procedures.

On the TA2 front, we present a system that matches extracted entities, events, and relations against schemas from an existing library. Evaluation showing that the matching instantiation subsystem improves over past work is presented. The instantiated schemas that result from instantiation are then passed to a prediction subsystem, which infers missing events. This prediction subsystem is shown to perform better than the previous state of the art.

In the next section, we describe each module mentioned above in more detail. We then present evaluation and discussion of the subsystems.

3 METHODS, ASSUMPTIONS, AND PROCEDURES

3.1 Schema visualization and curation tool: Eratosthenes

Eratosthenes is the schema curation and visualization tool for the KAIROS-CHRONOS team at CMU. It is a web application with a graphical user interface for interacting with schemas defined in SDF. Its development spanned the length of the project, growing in tandem with the program to satisfy the needs of the whole team, especially our human curators.

3.1.1 Role

Schema Visualization Eratosthenes enables someone with minimal familiarity with SDF to view and interact with a schema. This makes the tool useful not only for curators and evaluators but also for anyone working with SDF since it is far easier to visually intuit the structure of a schema than to read text. Visualization works for Graphs G and TA2 output as well as for individual schemas and schema libraries.

The visualization in Eratosthenes is intended to be transparent in addition to intuitive. By this, we mean that the visualization should clearly correspond to the SDF and introduce minimal abstractions or layers of indirection. This is important on two levels. First, it ensures that anyone familiar with SDF will easily see why each visual element is the way it is. Second, for people who do not interact directly with SDF (e.g., curators), it is critical that Eratosthenes represents the semantics of SDF as best it can; "smoothing over" the SDF too much can make it seem like the schema is representing that the actual SDF is not.

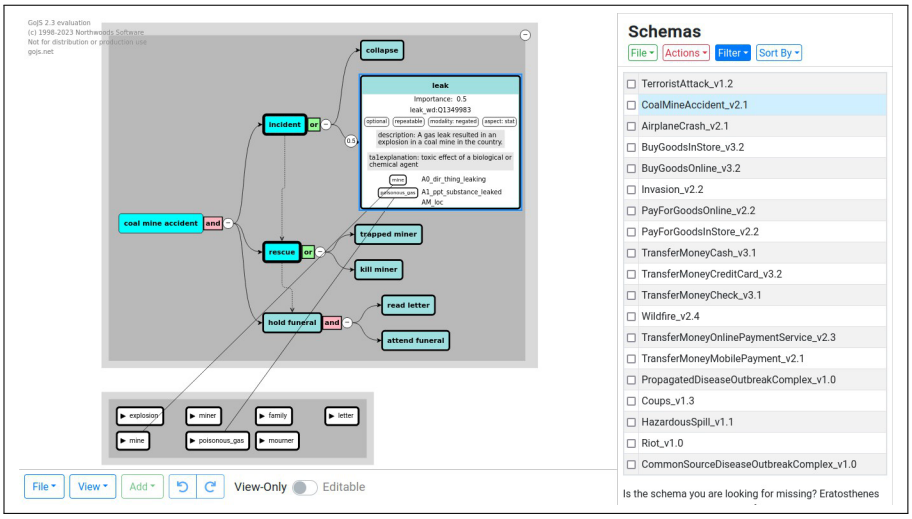

Figure 1: A screenshot of Eratosthenes' user interface.

User Interface The Eratosthenes interface (shown in Figure 1) comprises three main elements: the schema viewer, the schema tool bar, and the schema management menu. The schema viewer is the star of the show as it is responsible for actually visualizing the schema. The basic structure of the viewer is that there are two groups of nodes: events and entities. Events are represented as a left-to-right tree where the tree hierarchy represents parent-child relationships. Before-after relationships between children are represented with arrows and vertical ordering. Events can be can be expanded to show details like description, confidence, and the participants and their roles. Entities, in turn, are displayed below the events and have links displayed between them and the events they participate in (these are hidden until hovering over the relevant event or entity to reduce visual clutter). Subtrees can be collapsed in larger schemas in order to make them more manageable.

The schema toolbar provides the user with additional functionality related to the schema. For example, the user can toggle the schema from "View-Only" to "Editable" if he/she intends to edit the schema; this helps prevent accidentally editing/overwriting schemas as well as preventing multiple users from editing the same schema at once. Finally, the schema toolbar has a JSON viewer which displays the SDF for the current schema. This JSON viewer is hyperlinked

with the visualization itself such that you can click on elements in the graph and snap to the relevant JSON (and vice-versa).

Finally, the schema management menu allows the user to navigate to different schemas and interact with the collection of schemas as a whole (see "Schema Management" for more information).

Schema Editing Editing schemas in Eratosthenes follows the same graphical user interface paradigm. Events and entities can be added by right clicking the relevant areas of the visualization, selecting "Add Event" (or "Add Entity"), and entering information into a brief dialog (e.g., event name, type, description). Links between events and entities are added by clicking and dragging. Other properties of events and entities can be edited by right clicking on the object itself. Users are also provided with features like the ability to undo and redo edits. All edits made to the schema stay on the user's browser until the user clicks a save button at which point the schema is updated in the server's database.

Subschemas One way in which Eratosthenes significantly extends SDF for the sake of easier curation is through the use of "subschema references". Our curators found that certain subtrees of schemas were frequently used and reused by various schemas (e.g., paying for goods, medical response) and that having multiple copies of these subtrees in different schemas was unmanageable. Hence, we introduced the ability to reference one schema from another. For example a "paying for goods" (sub)schema might be referenced in two larger schemas, "eating at a restaurant" and "grocery shopping". This way, the "paying for goods" schema can be curated on its own without having to worry about keeping multiple copies in sync. Since SDF cannot represent this natively, Eratosthenes will insert the expanded subschemas into the final schema library when it is exported for delivery.

Schema Management Eratosthenes additionally serves as a repository for CMU's curated schema library as well as other frequently referenced schemas (e.g., induced schemas, other teams' libraries). Since Eratosthenes is a browser-based web application, it is easy to share schemas by simply sending the link to the desired page on Eratosthenes. Schemas can be deleted, duplicated, exported as JSON or ZIP files, and validated against the CACI validator. It is also possible to assign and filter by various "tags" (e.g., `finished_curating` or `induction_pipeline_output`) to schemas to more easily manage a large number of schemas in the system.

Although curated schemas are delivered as entire libraries, Eratosthenes maintains each individual schema in its own SDF document for visualization and editing. Once curation is finished, the library is packaged into a single SDF document and exported for delivery (since this output is SDF, it can still be visualized in Eratosthenes). This packaging step also allows for any final transformations to the Eratosthenes-internal SDF to be applied (e.g., references to subschemas being replaced with the explicit schema).

3.1.2 Technical

At its core, Eratosthenes is a single-page web application which connects to an API server providing application logic and a central database of schemas. A browser-based application provides numerous advantages such as no installation for users, centralized management of data, and a wide variety of functionality provided through JavaScript libraries.

Frontend The frontend of Eratosthenes is a React app written in TypeScript, using the GoJS library for the schema visualization itself[1]. Using TypeScript makes the application more robust by adding static type checking while remaining compatible with the vast selection of libraries written in JavaScript. The function of the client-side application is to request SDF files from the API server and display these to the user. Any edits made to the SDF will update a local copy of the document until a user "saves" it at which point the server-side copy is updated to match the client's edited version. Eratosthenes employs a simple locking system that transparently prevents multiple users from editing the same schema at the same time.

Backend The backend of Eratosthenes is a Python application which uses the Falcon web server framework served by Gunicorn and nginx, backed by a SQLite database. Generally speaking, the backend of Eratosthenes is intended to be lightweight, minimalist, and easy to understand. Most of the functionality of the application is focused in

[1]GoJS is a proprietary framework which is free to use only for "evaluation" (i.e., internally). We are currently in the process of replacing GoJS with a free and open source visualization framework that will not only make Eratosthenes free to distribute but also more performant and extensible.

the frontend to keep the application responsive and interactive, freeing up the backend to focus primarily on the management of data.

Python's popularity and flexibility make it a good choice for the lightweight and dynamic needs of the KAIROS program. Mypy is used to add static typing to Python to ease development and minimize the number of trivial bugs. Falcon, gunicorn, nginx, and SQLite all generally follow a minimalist design pattern prioritizing simple configuration and rapid development while still providing more than enough performance for the program's needs.

Development & Deployment Development of Eratosthenes is supported by the Nix package manager which creates reproducible builds across TypeScript, Python, shell scripts, and beyond. This is done while configuring the TypeScript code with npm and the Python code with Poetry, allowing developers unfamiliar with Nix to still easily work with the individual components. Nix works somewhat analogously to virtual environments in Python except across all packages (i.e., all the way down to the libc) meaning that it does rely at all on system packages. This makes it simple to use the exact same packages for development and production regardless of what Linux distribution is used.

Eratosthenes is deployed in production via Docker Compose, allowing for simple installation, migration, and updating. Docker Compose is used to orchestrate two separate Docker containers: one using nginx to serve the frontend JavaScript bundle and receive API requests and the other running the Python server for processing API requests and managing the database. Docker images are built directly using Nix, bypassing the needs for Dockerfiles. This results in a production environment that, package-wise, is identical to the development environment and only contains the packages that are needed (i.e., no need to work from a base Alpine or Ubuntu image).

3.2 Human Schema Curation Methods in CHRONOS

3.2.1 Schema Creation Procedure

Human curated schemas were initially developed by curators reading the articles provided by the LDC on specific examples of a topic, then the curators created a schema based on their human understanding of the more general complex event. Curators use our in-house schema editor and output viewer, Eratosthenes, which is loaded with the XPO Overlay of Wikidata Qnodes for events and their arguments (Spaulding et al., 2023). Qnodes are needed to represent both events and entities in Eratosthenes. Curators create an event, add a definition and an example sentence, set repeat, optional, and negation flags, link the arguments in the event to appropriate entities, arrange the events in a hierarchy with AND/OR/XOR gates, and add optional chronological links. The CMU system only uses chronological relations, so no other relations Qnodes are loaded into Eratosthenes. The curators find Qnodes for all pertinent events described in the articles. Curators may add events not specifically mentioned in the articles, if they align with the curator's understanding of the world. Curators are free to use whichever entity Qnodes they feel appropriate and are not restricted by Eratosthenes to only XPO Overlay entity Qnodes. When the scenario documents are available from MITRE, the curators supplement or reorganize their schemas based on input from the scenarios, if necessary. If the scenario document is available before or at the same time as the articles, the scenario is used as a basis to create a schema, with confirmation from the articles and the curator's own human understanding.

The curators are free to search online to assist in finding the appropriate XPO Overlay event Qnodes, when a good match is not initially found for a general language concept from articles, scenarios, or human understanding of the world. New candidate terms are then searched for in the XPO Overlay. Some internet resources that assist curators to find appropriate terms in the XPO Overlay are:

- Clicking on subclass links in a Wikidata entry to find ancestors of a particular Qnode.
- Writing a SPARQL query in the Wikidata Query Service to find all *subclass of* (P279) relations for a particular Qnode to find its children.[2]
- Searching online synonym resources to find other terms to represent a similar meaning.

Eratosthenes restricts human curators to use event Qnodes from the XPO Overlay. When curators feel an important event Qnode is missing from the Overlay even after searching for synonyms, candidate Qnodes are sent to the XPO Working Group to consider adding into the Overlay.

[2]SPARQL is the standard query language and protocol for Linked Open Data on the web or for RDF triplestores. Wikidata Query Service is the Wikimedia implementation of a SPARQL server and can be used to query Wikidata, for instance finding all Qnodes with a subclass of (P279) relation with another Qnode. See `https://query.wikidata.org/`

Since entity Qnodes are not restricted to the Overlay in Eratosthenes, the entity Qnodes from CMU's Schema Library were more closely examined to determine if they or any of their subclass ancestors were in the XPO Overlay. For entities not in the XPO Overlay and with no ancestors in the Overlay, the Qnode was changed to a better Qnode already in the XPO Overlay, if possible. If there was no appropriate entity in the Overlay, a request was sent to the XPO Working Group to add the entity Qnode to the Overlay.

3.2.2 Annotation

Annotation Process and In-house Annotated Dataset An in-house annotated dataset was created for graph *G* generation. The source documents included Quizlets released by LDC and publicly available documents which curators identified from Wikinews. Curators used the brat annotation tool (Stenetorp et al., 2012) with the XPO Overlay containing Wikidata Qnodes.

To annotate, curators:

- identify a relevant event in the document and tag it with the appropriate Qnode
- set the modality of event
- look for entities that fill the argument roles of the event
- tag participating entities
- link entities to events using argument roles
- add coreferent event and coreferent entity links
- add subevent (parent-child) links between events
- add temporal relation (after) links between events

The XPO Overlay helped curators choose Qnodes for events and entities, because it provided a map corresponding to Qnode concepts as well as argument roles for events. Regarding the Qnodes for entities, the curators referred to Wikidata to search for relevant Qnodes if none were found in the Overlay.

Challenges One of the challenges of in-house annotation was to determine the granularity of annotation. Because the source documents detailed particular scenarios (e.g., IED bombing at a particular time and place), annotations initially included too many details in which, for example, multiple agents reported the same events in different ways. This influenced the output quality at the early stages of graph *G* generation. Another challenge was to determine the significance/relevance of events. An article contains many events, but only the events that would be included in a schema should be tagged. Events in a schema are generic enough to be applicable to all instances of a scenario.

The original annotation in the in-house dataset was very dense, which resulted in noise for graph *G* generation. We revised the annotation dataset by (1) narrowing the annotation focus to the events and entities relevant for schemas and (2) removing non-significant events such as reporting events, to improve the graph *G* output. We also made sure that all the events had two or more arguments to increase the quality of output. These modifications helped graph *G* contain more salient and informative event steps.

One of the biggest challenges for annotation was changing from the LDC tagset to Wikidata Qnodes of the XPO Overlay. The change from using a relatively small tagset to the large and constantly changing Wikidata world was at first very daunting. The XPO Overlay Working Group helped greatly by curating a smaller set of events with argument roles.

3.2.3 Output Evaluation

TA2 Output Curators are able to view TA2 output in Eratosthenes. The events and their arguments in the matched schema are linked to the events and arguments in the graph *G*. Eratosthenes utilizes different colors and link types to ease user visualization. Curators evaluated the TA2 output using the following metrics:

- check that the appropriate schema was chosen from the schema library

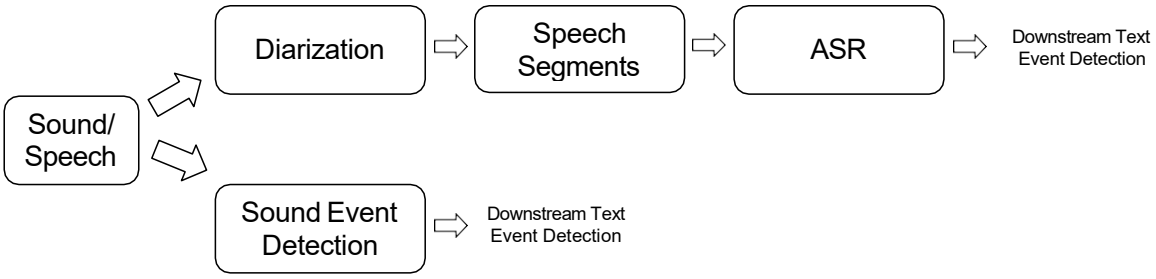

Figure 2: Overview of the Kairos Speech Processing System

- check that the matched schema event was actually the same kind of event as the GraphG the event
- count the total number of matched events and the number of correctly matched events
- review appropriateness of the predicted events given the hierarchy and AND/OR/XOR gates
- count the total number of predicted events
- count the number of predicted which included generate arguments

3.3 Extraction of Graphs G from Speech

3.3.1 System Overview

The Kairos Speech Processing System detects low-level audio events and transcribes speech in audio files using automatic speech recognition (ASR). As illustrated in Figure 2, the Kairos Speech System consists of two main processing pipelines: sound event detection and speech recognition. The input audio signal first goes through sound event detection to identify important auditory events like gunshots, explosions, screams, etc. This provides crucial clues about events that may not be visually discernible. In parallel, the input audio passes through a speaker diarization module to extract speech segments. These segments then go through automatic speech recognition (ASR) to transcribe the verbal speech into text. The outputs of both the sound event detection and ASR modules are then sent to the Kairos text and multimedia event detection subsystem for further analysis and integration with visual inputs.

3.3.2 Sound Event Detection

The Sound Event Detection (SED) Pipeline aimed to extract low-level audio event primitives, which were the basic units for schema generation and instantiation. For this purpose, we utilized sound event detection and localization models (Wang, 2018) trained on the AudioSet dataset (Gemmeke et al., 2017). This dataset contains a diverse collection of 527 audio event classes such as explosions, alarms, people gathering, fire, etc. Although the models were trained on AudioSet, they generalized well to in-domain data corresponding to KAIROS. Specifically, our audio event detection pipeline had two stages. First, we performed a coarse event search where 10s chunks of audio were analyzed to detect high-level variations in audio events by examining the top-5 model predictions. This allowed the identification of chunks containing events of interest, such as an “explosion” between 40-50s. The chosen coarse chunks were then sent for more refined processing and detection of specific audio events. In this stage, we took 1s chunks of audio and re-ran the pipeline for high-precision event labeling and localization. There were several challenges to audio event detection. The event ontology of AudioSet did not fully match the KAIROS ontology. We remapped the AudioSet ontology to the KAIROS classes and fine-tuned the models towards in-domain events.

3.3.3 Diarization and Automatic Speech Recognition

A typical KAIROS audio file could contain information as both audio events and human speech. To extract the information stored in the human speech we used an automatic speech recognition pipeline to convert the speech to text and passed it on to the text event detection and sequencing team. For any given audio file we followed a two-step process for extracting the transcriptions:

First, we ran out-of-the-box diarization tools which detected speech frames along with speaker change and separated out noise and other sound segments. The diarization toolkit was based on the Kaldi offline transcriber (Aluma¨e, 2018) which had been modified and improved using more training data (Le Franc et al., 2018). The speech segments were then filtered out using the output of the diarization tool and passed to the automatic speech recognition model. We had two different kinds of speech recognition models in order to handle robust recognition in English or different languages. The key differences in the two models were:

- An ASPiRE LF-MMI (Povey et al., 2016) model trained on 10,000+ hours of English data.
- End-to-End CTC model (Li et al., 2020) trained on 6000+ hours of speech data collected in various languages around the globe.

For the current scenario when we only had models in English, Spanish, and Russian, we used just the first model but we could quickly adapt to any target language using the second model.

In addition to the challenge of extending support in multiple languages, there were several other challenges to these ASR models, especially when used for downstream NLP tasks. ASR often missed named entities as having a large vocabulary was difficult. Code-mixing in bilingual communities also posed a problem due to the international scope of KAIROS. Punctuations and capitalizations were unreliable in ASR but could provide important cues for downstream tasks. We investigate solutions to alleviate these problems as follows.

Event Focused ASR When general ASR systems were adopted by organizations or used for downstream semantic tasks there was a need to recognize domain-specific events/entities. Existing models employing this idea relied on hybrid ASR systems where a language model reranker helped prioritize entities. However, they required cumbersome relation databases and did not use acoustic signals. We trained a CTC-based acoustic model that increased alignment likelihoods for target transcriptions while reducing likelihoods of incorrect alignments.

Visual Feature Incorporation We prepared a ”SLIDES” dataset to support open-vocabulary end-to-end speech recognition using visual context. The goal was to recognize new, unknown words provided visually in videos. The dataset contained 1783 lecture videos with aligned slides. Baseline models without context gave 22.92% word error rate (WER) while a simple model incorporating visual context improved performance by 3.46% absolute.

Russian Recognition We conducted a literature review to identify the OpenSTT dataset and ESPnet2 model as optimal choices for Russian recognition. We integrated the pre-trained model into the pipeline, following the existing code structure. Testing on 20 Russian videos related to COVID-19 and explosives demonstrated effectiveness. The updated pipeline is in the CMU-KAIROS GitHub repository. Evaluation on the OpenSTT validation set gave WER of 31.3% for phone calls, 19.4% for audiobooks, and 18.9% for YouTube videos.

3.4 Extraction of Graphs G from Text

Figure 3 provides a schematic overview of our multimedia information extraction system. In this section, we describe the modules that extract graphs from multilingual text documents (Figure 4).

3.4.1 Multilinguality

For Russian and Spanish language texts, we first translate them into English using state-of-the-art systems. We merge the translated texts with other English documents to constitute the input for our extraction pipeline. Additionally, we extract word-level alignments between original language texts and translated English text using the `awesome_align` toolkit (Dou and Neubig, 2021). We use these alignments to remap the source offset and length for extracted entity and event mentions. Our final graphs include the offset and length information from the original language text. We use Helsinki-NLP translation models from their huggingface repository. We use `opus-mt-es-en` and `opus-mt-ru-en` for Spanish and Russian language texts respectively.

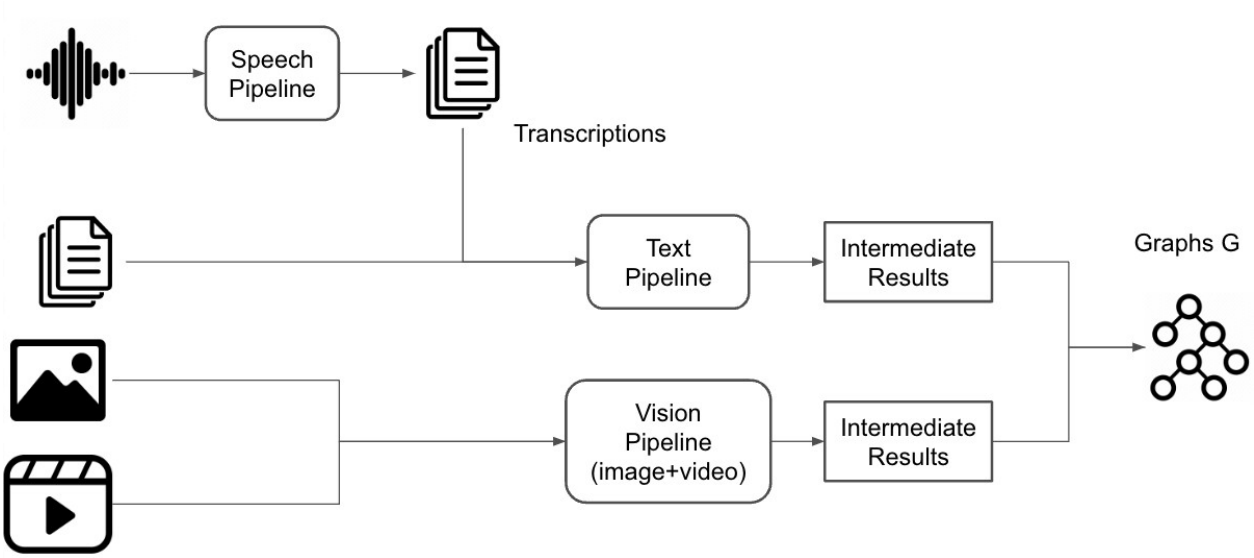

Figure 3: Graph *G* extraction overview

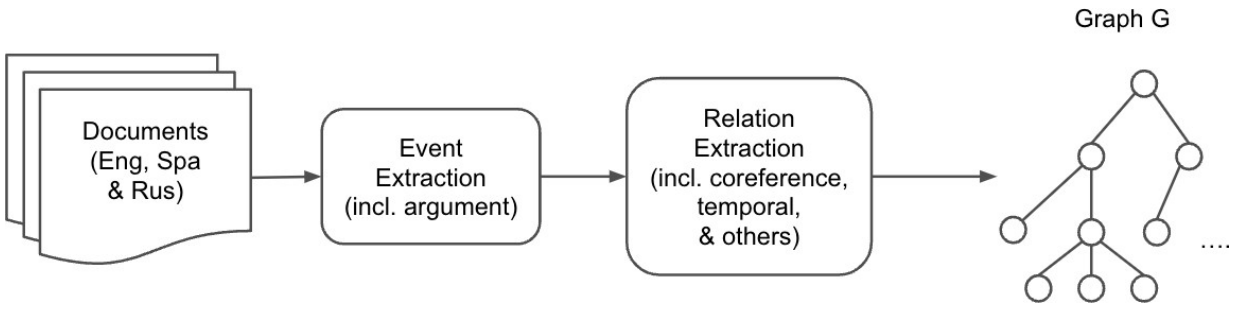

Figure 4: Extraction of Graphs G from text

3.4.2 Event Span Identification

Event step identification is the task of identifying the event span from input text. We developed two models for this task to better match with schemas while keeping its recall.

Base model One model, which we call the base model, is a fine-tuned pre-trained language model, RoBERTa-large (Liu et al., 2019), where the task is formulated as sequence tagging. This model is trained on our in-house annotated data.

Guided model The other model, which we refer to as the guided model, is developed to steer the event span extraction to focus on schema-related events. Inspired by Wang et al. (2022), we extract all event types mentioned in target schemas and convert each type into a query based on its description and frequent trigger. The guided pipeline consists of two stages. In the first stage, each query is paired with every input sentence through a discriminator model to predict if the query is a candidate event type mentioned in that sentence. In the second stage, each sentence is paired with every candidate query to perform sequence tagging for guided span extraction.

3.4.3 Event Qnode Linking

Event qnode linking is the task of linking qnodes to events. We compile the list of qnodes that are likely to appear in the KAIROS project through our annotation process. We fine-tune a pre-trained language model, RoBERTa-large, with a multi-class classification loss. The finetuning is conducted with our in-house annotated data.

3.4.4 Event Realis Detection

Event realis detection is the task of detecting a realis/modality of events. The model is a fine-tuned RoBERTa-large with a multi-class classification loss. We use our in-house annotated data to fine-tune the model.

3.4.5 Event Argument Extraction

Event argument extraction is the task of extracting event arguments, given events. We formulate the task as extractive question answering, following Du and Cardie (2020). More specifically, given a passage and a question, a model is to identify an argument span(s) from the passage. One of the advantages of the formulation is that it can utilize argument role names as additional information in a part of a question, to identify argument span(s). We first tried to use the publicly available data, e.g., ACE 2005 dataset (Walker et al., 2006), however, we found that their limited ontologies and the differences in target domains are not ideal for the KAIROS projects. Instead, we use our in-house annotated data and fine-tune RoBERTa-large with a sequence tagging loss.

3.4.6 Event Salience Identification

Event salience identification is the task of scoring each extracted event by its importance and relevance to the main content of the document, with the goal of further condensing and denising the extracted information. For each event, we extract its global linguistic features (e.g. frequency, position, etc.) and summarization features (whether or not it appears in the model-generated document summary). We train a salience model based on the extracted features to assign a salience score to each event. Thresholding on the salience score allows for flexible pruning on less important events.

3.4.7 Time Expression Linking

Time expression linking is the task of linking time expressions to events. Following the event argument extraction model, we formulate this task again as extractive question answering. For each event, the model is prompted with two types of inputs that search for its start time and end time from its passage. We fine-tune RoBERTa-base, with a sequence tagging loss, using TempEval3 (UzZaman et al., 2012) as the annotated data.

3.4.8 Time Expression Normalization

After time expression linking, we normalize each identified time expression into the "xsd:dateTime" or "xsd:duration" format. We use the combination of SUTime (Chang and Manning, 2012) and HeidelTime (Stro¨tgen and Gertz, 2016).

3.4.9 Entity typing

For entity typing, we use a sequence-to-sequence model trained on the ultra-fine entity typing dataset (UFET; Choi et al. 2018). Our model takes an entity mention as input (and its context) and employs constrained beam search to autoregressively generate multiple types. The raw sequence probabilities associated with the predicted types are then transformed into confidence scores using a novel calibration method. On the UFET dataset, our method outperforms the previous state-of-the-art in terms of F1 score and calibration error, while achieving an inference speedup of 50x. Our system also presents few-shot and zero-shot transfer abilities to specialized domains, illustrating its effectiveness in KAIROS scenarios.

3.4.10 Coreference

Within-document entity resolution For each input document, we extract entity mention spans and coreference clusters using the end-to-end coreference system from Lee et al. (2018). We use a SpanBERT-large model fine-tuned on OntoNotes 5.0 (Weischedel et al., 2013) from the allennlp-models library.[3]

Cross-document entity resolution We concatenate documents within each complex event to create a single long meta-document. The meta-document is segmented to fit into the model input length constraints. We use the same end-to-end coreference system from Lee et al. (2018) for our cross-document resolution. After identifying the clusters of corefering entity mentions, we first perform a match between the entity mention spans identified by the above end-to-end system and our CHRONOS entities. We retain clusters that include CHRONOS entities.

Event coreference resolution We use a single model to perform both within-document and cross-document event coreference resolution. For the extracted CHRONOS event mentions, we first compute mention embeddings using a fine-tuned sentence-transformer model (bert-base-uncased based).[4] We construct the sentence input by including special tokens to indicate event mention and argument token boundaries. We extract each mention embedding by mean pooling over the mention tokens. To extract coreference clusters, we perform agglomerative clustering with cosine similarity and single linkage. Our model is trained on the standard ECB+ dataset (Cybulska and Vossen, 2014) for within- and cross-document coreference resolution.

3.4.11 Event Temporal Ordering

Event Temporal Ordering is the task of identifying the chronological order between/among events. We formulate the task as extractive question answering, instead of common pairwise relation classification, to mitigate the label unbalance issue in annotated data. We fine-tune RoBERTa-large with a sequence tagging loss. The fine-tuning data is our in-house annotated data. After identifying the pairwise temporal relations, we employ ILP to enforce the consistency of relation predictions, e.g., removing cycles.

3.4.12 Graph G construction

To construct the final CHRONOS text graphs, we aggregate events from all the documents belonging to a complex event. We use the cross-document coreference clusters of events (or entities) to select the most common mention as the representative mention. Every mention of the event (or entity) in our final graph refers to this mention. We aggregate the participant information across all mentions of an event. We combine the document-level temporal relations using the coreference links, while discarding any cyclic relations.

3.4.13 Event Step Description Generation

Event Step Description is the natural sentence description that describes each event step in text graphs. In the final CHRONOS text graphs, we add a description to each event using a fine-tuned model with a semantic role labeling (SRL) dataset. In the inference time, we take an event and its arguments with their roles as input from each event step, and the model generates a natural sentence that contains the information given in its input. In the training phase, we use the CoNLL 2005 (Carreras and Ma`rquez, 2005) dataset, but in the reverse manner from the normal SRL task:

[3] https://github.com/allenai/allennlp-models
[4] https://github.com/UKPLab/sentence-transformers

we use the annotation, i.e., parsed sentences as input and their original sentences as output. The base model is a T5-base (Raffel et al., 2020).

3.5 Extraction of Graphs G from Multimedia

3.5.1 System Overview

The Multimedia Event Detection system is designed to detect events from raw multimedia data and fuse the results with text event detection. As illustrated in Figure 5, the system consists of the following key components:

1. **Filtering and Preprocessing**: This module performs initial processing of raw video, image, audio, and text data. Techniques including frame extraction, noise removal, resizing, and compression are applied to normalize the multimedia inputs. Media files are decoded into standardized formats based on their type for easier processing in later stages.

2. **Object Detection**: This module detects and localizes objects present in visual content using models such as YOLOv3, SSD, Faster R-CNN, and Mask R-CNN. A multi-scale, multi-modal framework is implemented to handle objects across a wide range of scales. Approaches including attention mechanisms and few-shot learning enable detecting niche object categories with limited training data.

3. **Face Detection**: This module focuses specifically on detecting human faces in visual content. The RetinaFace model with ResNet backbone is utilized to efficiently and accurately detect faces. An anchor-based scheme further improves localization performance.

4. **Scene Recognition**: This module categorizes the overall scene type, such as Gas station or Hospital, based on visual appearance. An initial Places365 model provides scene classification, while 3D CNNs add temporal context from videos to disambiguate confusing categories.

5. **Text Recognition**: This module detects and recognizes text characters in videos and images (such as subtitles, logos, signs, etc.). A two-stage pipeline using EAST for localization and CRNN for recognition is implemented. Specialized techniques handle challenging text such as curved or low-resolution.

6. **Chart Recognition**: This module parses charts and tables into structured data representations. The ChartReader framework detects chart elements using a transformer model and applies vision-language models to generate textual descriptions.

7. **Event Detection**: This module utilizes the self-labeled mapping rule between detected multimedia entities and Kairos ontology to identify events and their semantic parameters. Tokenization, embeddings, and transformer-based sequence labeling are used to detect events and their respective arguments.

8. **Coreference**: This module generates a unified multimedia graph G by resolving coreference across modalities for facial, entity, and event detection.

3.5.2 Preprocessing

We implemented a multi-stage preprocessing pipeline to convert the raw dataset into a cleaned format ready for downstream tasks like object detection and scene classification, similar to existing video preprocessing frameworks (Nguyen et al., 2017; Ngo et al., 2017; Cheng et al., 2016, 2017a,b). The pipeline executes the following steps. First, the dataset is unpacked into a *data* directory containing media subfolders for each type like GIF, PNG, JPG, SVG, and MP4. The *parent_children.tab* metadata file maps filenames to topic labels through a dictionary lookup. Different media types are converted to JPG images - GIF animations become JPG sequences, PNG and SVG are converted directly, valid JPGs are copied, and any corrupted files are skipped. The JPG images are then copied into subfolders named by topic, based on the filename-topic mapping. Finally, keyframes are extracted from the MP4 videos into a separate folder using keyframe extraction techniques (Ngo et al., 2017; Cheng et al., 2017a).

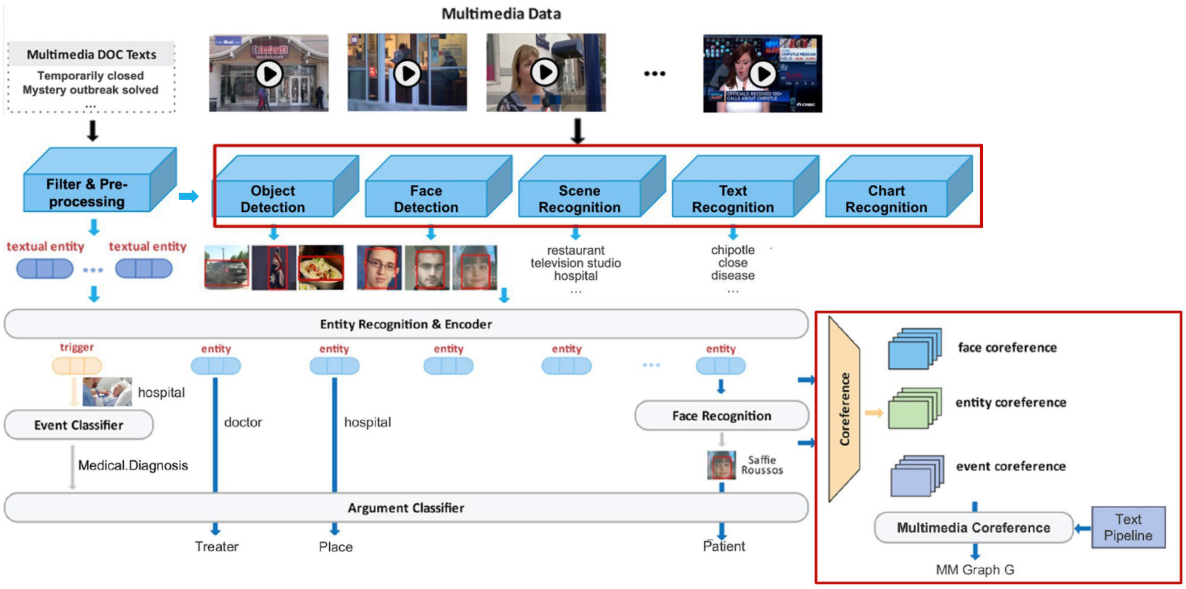

Figure 5: Multimedia Event Detection System Architecture

3.5.3 Object Detection

Our object detection module employs a multi-scale and multi-modal framework to robustly identify objects in complex multimedia data. For real-time detection of large, salient instances, we utilize single-stage models including YOLOv3 (Redmon and Farhadi, 2018) and SSD (Liu et al., 2016). These prioritize speed by directly predicting class labels and bounding boxes in a single inference pass. To complement this, two-stage models like Faster R-CNN (Ren et al., 2015) and Mask R-CNN (He et al., 2017) are incorporated to accurately localize smaller or heavily occluded objects through region proposal and deeper feature extraction.

To enhance localization for tiny objects, feature pyramid networks (Lin et al., 2017) are used to leverage both semantically strong high-level and spatially precise low-level features across the scale. For reasoning about object parts and attributes, sequential modular networks (Cheng et al., 2018) break down detection into specialized sub-tasks. Further alternate semantic attention refinement (Cheng et al., 2022b) is used to focus feature extraction on the most relevant regions. For niche categories with limited training data, we fine-tune base models on custom datasets collected in-house. Domain-specific augmentation and few-shot learning techniques (Hu et al., 2021) help efficiently adapt models to new classes. To precisely segment objects, mask predictors are added to extract per-pixel masks rather than coarse bounding boxes.

By combining one and two-stage models across multiple architectures, our framework achieves an optimal balance of speed, accuracy, and flexibility needed for real-world multimedia analysis. The system is designed to incorporate future state-of-the-art techniques as they emerge to continuously improve detection capabilities.

3.5.4 Face Detection

Face detection is implemented using the RetinaFace model (Deng et al., 2019). RetinaFace is a single-shot anchor-based face detector that adopts a ResNet-50 (He et al., 2016) or MobileNet (Howard et al., 2017) backbone for efficient feature extraction. The model aggregates multi-scale convolutional feature maps to detect faces across a spectrum of scales and leverages an anchor-based scheme (Bao et al., 2023) for classification and regression. Specifically, we initialize a *FaceDetector* class with the pretrained RetinaFace weights and specify the target GPU device ID. The FaceDetector takes an image directory as input and outputs face bounding boxes to a txt file for each image. Emptying the GPU cache after each inference avoids excessive memory accumulation. RetinaFace achieves state-of-the-art performance for face detection. By using the pretrained model, we avoid expensive training and obtain an efficient and accurate face detection module to facilitate subsequent pipelines.

3.5.5 Scene Recognition

Scene recognition is a key technique to provide contextual understanding of images and videos. We utilized ResNet-50 (He et al., 2016) model pretrained on Places365 dataset (Zhou et al., 2017a) to classify images into semantic scene types such as airport, bedroom, coast, etc. However, single image recognition lacks temporal context that can help disambiguate confusing categories. To address this, we incorporated temporal context from video inputs using 3D convolutional neural networks (Nguyen et al., 2017; Ngo et al., 2017). By modeling appearances and motions jointly across consecutive frames, this approach disambiguated confusing scene pairs like "highway" vs "city street" based on multi-attribute learning of scenes (Huang et al., 2018; Cheng et al., 2018). Furthermore, we investigated transformer-based architectures to model relationships between objects and global scene context (Cheng et al., 2022b). By combining local object interactions with the overall scene category, these networks helped resolve ambiguous scenes where multiple semantic interpretations were possible from a single image.

3.5.6 Text Recognition

Text detection and recognition in images and videos were critical for understanding visual content. We implemented a two-stage pipeline consisting of text localization followed by text recognition. For localization, we utilized an EAST model to detect text regions and regress bounding boxes around them (Zhou et al., 2017b). The detected regions were cropped and passed to a CRNN model for optical character recognition to transcribe the text (Shi et al., 2016). This framework enabled us to detect and recognize printed and handwritten text in natural images and videos.

Additionally, we further improved the accuracy of text detection and recognition in several aspects. For localization, incorporating text contours and spatial context beyond horizontal bounding boxes was used to improve detection of irregular text (Long et al., 2018). For recognition, attention mechanisms and transformer models were employed for curved and warped text (Raisi et al., 2021; Cheng et al., 2023). We pretrained on synthetically generated text to improve generalization to new fonts, styles, and languages (Gupta et al., 2016). Furthermore, we utilized visual context in a multimodal approach to help resolve ambiguities in low-resolution or occluded text (Cheng et al., 2023).

3.5.7 Chart Recognition

Charts conveyed important information but posed comprehension challenges due to diverse chart types and intricate components. We presented ChartReader, a unified framework for chart analysis (Cheng et al., 2023). It utilized a transformer to detect chart elements and a pretrained vision-language model for chart-to-table/text/QA. By automatically learning chart patterns from annotated data, ChartReader eliminated manual rule engineering and improved accuracy.

We applied ChartReader (Cheng et al., 2023) to analyze charts in FDA drug safety reports. The model was pretrained on synthetic charts and then fine-tuned on FEARS datasets, outperforming prior heuristic and OCR methods. A data variable replacement technique enhanced cross-task generalization. By extracting structured data from charts, ChartReader reduced manual review effort of safety reports. Our framework provided a crucial step towards universal chart understanding. Moreover, ChartReader integrated readily with leading LLMs like T5 (Raffel et al., 2020) and TaPas (Herzig et al., 2020), extending their capabilities for analyzing charts in documents.

3.5.8 Event Detection

We first explored mapping diverse real-world multimedia events to KAIROS ontology across various scenarios including IEDs, pandemics, and scenarios collected by our CMU scheme induction team. Through manual annotation of 141 videos considering visual and audio cues, followed by cross-validation, we verified that KAIROS ontology can generally describe multimedia data across these diverse scenarios. However, some limitations exist in covering obscure events, and distinguishing similar events remains challenging. Based on the initial ontology mapping exploration, we further defined extraction rules to extract structured event representations from multimedia data. We proposed 21 new event primitives and provided feedback to the Kairos ontology working group on issues encountered during annotation. Finally, 5 event primitives of our suggestions have been adopted in the latest KAIROS ontology.

We then utilized these defined extraction rules to detect events in multimedia data and output structured event representations. The rules help infer missing arguments and better match event primitives to improve detection accuracy. Specifically, we formulate this as the following sequence labeling task. (1) The input consists of previous multimedia entity detection results (i.e., objects, scenes, faces, text, tables). These are tokenized into subword units using Byte-Pair Encoding (BPE). BPE represents rare and unknown words as common subword sequences. We augment the token

embeddings with positional encodings to capture order information. Learned entity-type embeddings are also added to incorporate entity knowledge. (2) The token representation is passed through a transformer model like BERT(Devlin et al., 2018) to generate context-aware token encodings. The transformer captures long-range dependencies via multi-headed self-attention. We initialize from a pretrained checkpoint and fine-tune with our collected event extraction rules to adapt the model. (3) A classification layer on top of the transformer outputs predicts an event trigger label for each token. It computes a distribution over the trigger class vocabulary. During training, cross-entropy loss between predicted and true labels is minimized. At inference, the highest probability trigger is assigned to each token. (4) The identified trigger tokens are then combined with entity pointers in the text to extract structured event representations. Each event consists of the trigger, event type, and entity arguments with roles.

3.5.9 Coreference

The coreference module integrates information across text, images, and videos to construct coreference of detected multimedia events. The module performs two key functions. (1) *Facial Coreference*. The facial coreference function matches the faces of the same individual across different images and videos. It extracts convolutional facial features from detected faces and computes cosine similarity between these features. Highly similar facial features indicate the same person, enabling coreference linking of that individual's face across different images and videos. (2) *Entity and Event Coreference*. The entity and event coreference function establishes relationships between textual entity mentions, visual entity depictions, and referenced events. We first fine-tuned a pretrained CLIP model (Radford et al., 2021) on all the domain-specific data from Quizlet to adapt its multimodal representations. This adaptation allows our fine-tuned model to precisely connect semantic concepts between text and images from our dataset. It enables identifying when a textual mention and visual depiction refer to the same underlying entity or event.

3.6 Matching and Instantiation

Matching and instantiation is the process of matching a schema from the schema library (provided by the TA1 team) and its events with the events in a graph *G* (extracted from documents and provided by the extraction component). This first involves instantiating one of the schemas or subschemas in the schema library, or selecting which schema is the best match for the graph *G*. Once this is accomplished, the task consists of matching events in the selected schema with their appropriate matches among the graph *G* events extracted from documents. Entities that are arguments of schema events are also matched with graph *G* entities.

For example, a document about a wildfire in Hawaii might align with a Natural Disaster schema in the schema library. (I.e. That schema would be instantiated.) Following the instantiation, a "call emergency personnel" event in the schema might match with a graph *G* event describing a call to the Maui fire department. The "emergency personnel" participant of the former event might match with the "Maui fire department" participant of the latter.

Events in schema libraries and graphs *G* are organized in a highly structured format. In schemas, parent events are split into children events that designate them. For instance, the parent event of washing dishes could be realized by its children events of scrubbing the dishes and rinsing them. Events also have temporal information. Some events are necessarily preceded by other events (or necessarily succeeded by them). For instance, walking into a restaurant would precede ordering a meal. Events also have logical relations. When all children events are necessary to realize the parent event, they are connected by an AND-gate. When only one or more of the children events are needed, they are connected by an OR-gate. When exactly one event can be present (no more, no less), we use an XOR-gate.

3.6.1 Approaches

Our approach to instantiation evolved over time. In 2020-2021, our team relied on string-based similarity metrics to match events from schemas and graphs *G*. In 2021-2022, we made the pivot to using a sentence transformer to encode semantic information about both events and to calculate matching similarity that way.

Throughout 2021-2022 our system encoded summary information about each event, such as its name as description. We also provided context for each event from temporal relationships in the schema and graph *G* structure. If two events had similar temporal context (preceding and succeeding events), they could be more likely to match.

Unfortunately this approach had some major drawbacks. It did not consider non-temporal relations between events. It forced the events' tree structure into a temporal chain, losing information. And it resulted in long, noisy temporal contexts for each event. An illustration is give in Figure 6 below.

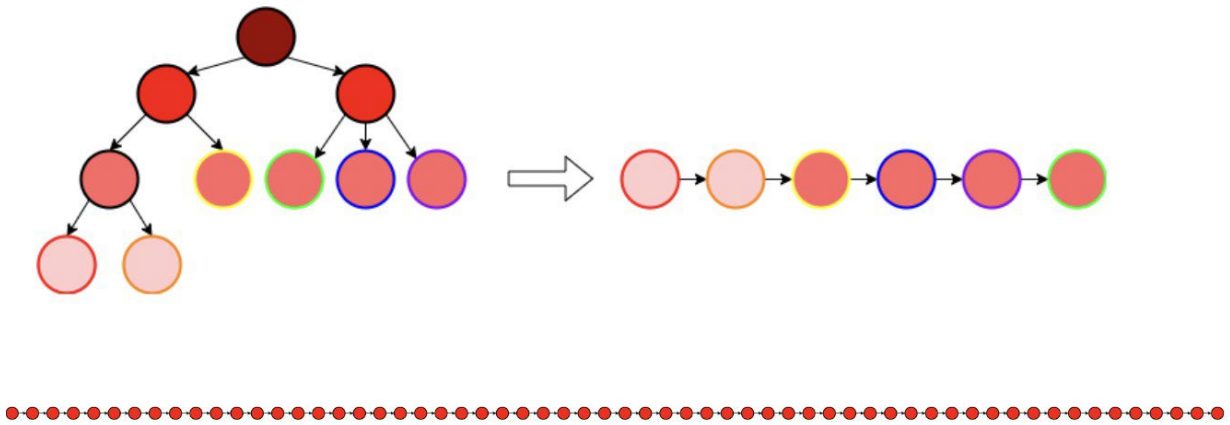

Figure 6: Linearization of a schema.

In 2022-2023 we modified our system to mitigate these shortcomings. Our new system considers multiple types of event relations, illustrated in Figure 7 below.

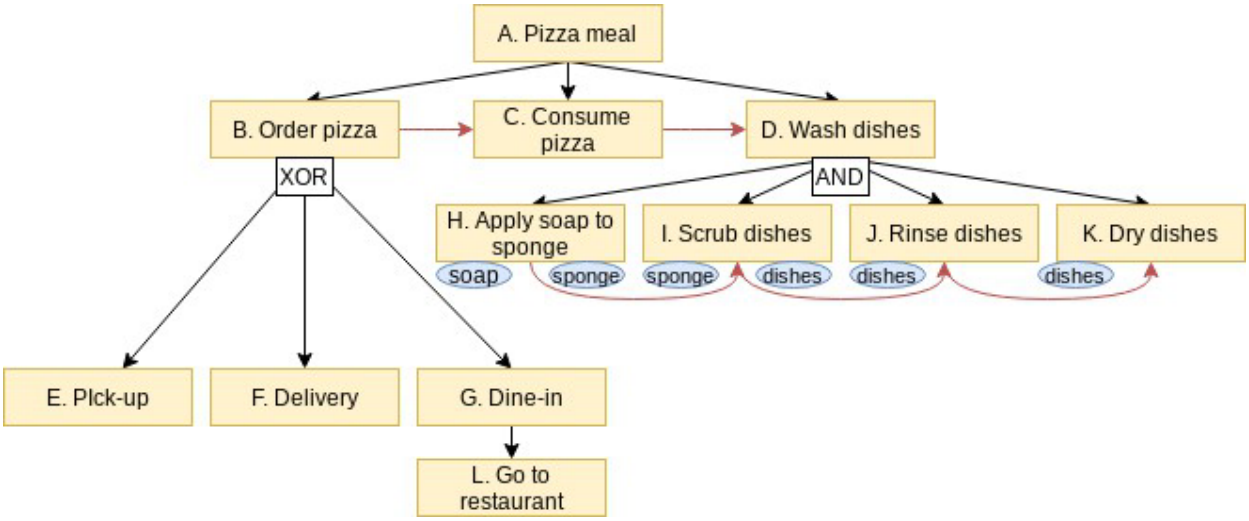

Figure 7: Current schema representation.

This example “Pizza meal” schema illustrates multiple vital event relationships that affect the probability of an event matching.

- Hierarchical relations
  - If the child “Rinse dishes” matches, “Wash dishes” is more likely to match
- Temporal relations
  - If temporal predecessors “Apply soap to sponge,” “Scrub dishes,” and “Rinse dishes” match, “Dry dishes” is more likely to match
- Event-to-participant relations
  - Matching “sponge” or events with “sponge” increases the likelihood that “Apply soap to sponge” matches

- Logical relations
  - If "Dry dishes" matches, then AND-sibling "Scrub dishes is more likely to match
  - Inversely, if "Delivery" matches, then XOR-sibling "Pick-up" is less likely to match
  - By the same reasoning, if "Delivery" matches, then XOR-niece "Go to restaurant" is less likely to match

In our new approach, we extract information from events regarding all of these relations and use them to calculate different similarity scores. These scores all factor into the final matching similarity score (as illustrated in Figure 8 below). Scores still depend on representations from a sentence transformer.

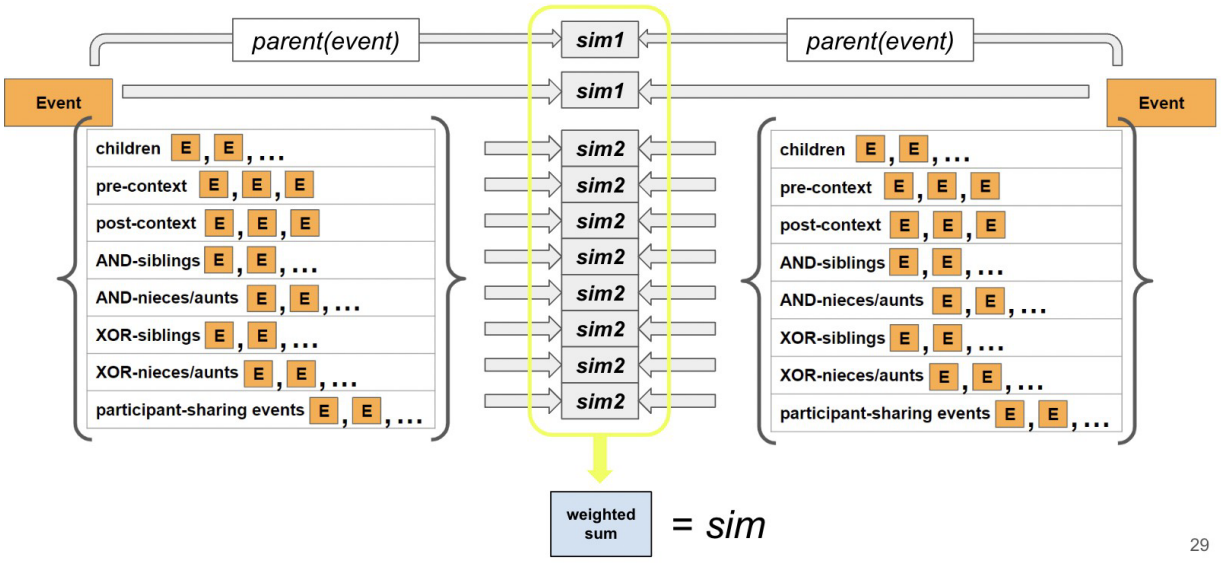

Figure 8: Calculating similarity scores

This new approach mitigated the disadvantages discussed earlier and resulted in surprisingly good matches by harnessing more information about each event. However, a major drawback to it is that of tuning. These similarity scores that combine to produce the final score do not have equal importance. We assign them each a weight coefficient accordingly. However, this means the number of possible weight coefficient combinations is very large and difficult to tune optimally. Many of our performance setbacks are due to difficulty finding this optimal coefficient configuration.

3.7 Event and Entity Prediction

3.7.1 Problem Definition

Predicting events over the schema graph is formulated as a node classification task. More specifically, given an instantiated event graph, the goal is to determine whether other unmatched schema events within this graph could potentially occur.

Formally, we consider an instantiated graph $G = (V, E)$, where V represents the set of event nodes and entity nodes, and E represents the set of links, including event-event temporal links, event-entity links, and entity-entity links. Some nodes of the instantiated graph correspond to real events or entities in the graph G, denoted as I, which is the set of matched schema events. Our task of event prediction involves classifying whether each node in the remaining schema event nodes, represented by the set $V \setminus I$, is a missing event given the instantiated graph (labeled as positive or negative).

3.7.2 Existing Methods

Three main methods exist for the event prediction task.

AddAll This simple heuristic method treats each event node in the set $V \setminus I$ as a predicted missing event in the instantiated graph G.

AddPath This is another heuristic used in our previous pipeline. It identifies all paths in the instantiated graph G, starting from the root node to leaf nodes, containing at least one matched event $u \in I$ and treats all nodes $u \in V \setminus I$ as missing events.

Schema-GCN This is a Graph Convolutional Network (GCN)-based method used by the RESIN team. This method iteratively propagates node information over linked structures during representation learning, followed by an MLP layer for predicting missing events.

3.7.3 Our Approach

We began our work by examining the limitations of existing methods. The AddAll method treats all unmatched event nodes in the instantiated event graph as missing events, which, in most cases, results in a significantly low precision score. The AddPath method, on the other hand, relies on event chains extracted from the graph. However, this approach proved quite challenging in our implementation when dealing with more complex and structured schema graphs. It fails to recognize the hierarchies inherent in the schema graph and ignores the logic gate in the graph. As for the Schema-GCN method, it does not adequately consider the logic gates defined in the graph.

Given these analyses, we develop a new approach that addresses these shortcomings. Unlike the AddAll and AddPath methods, our approach leverages a schema-guided prediction stage using a graph neural network designed for schema graphs (see Figure 9). This allows us to better utilize the structural information in the schema graph. We also introduce a constrained prediction stage that implements logic gates and hierarchies, ensuring our predictions follow the constraints defined in the schema graph. Lastly, our method includes an argument coreference stage, which uses coreference entity links and instantiated entities to generate reasonable arguments for predicted events.

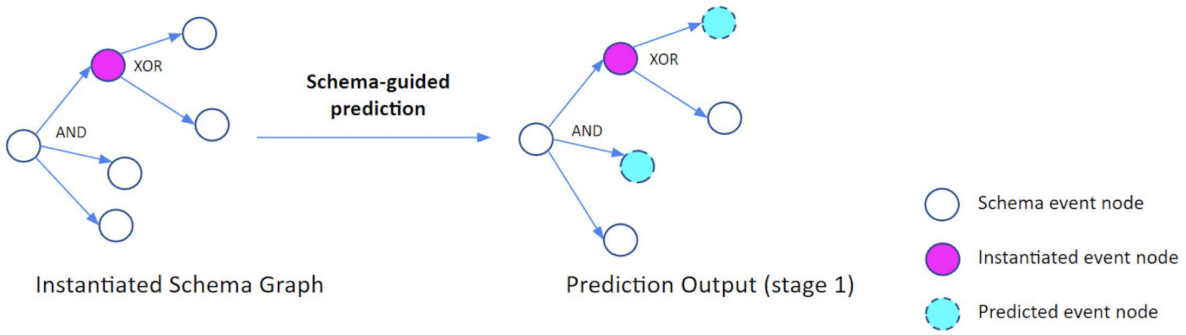

Figure 9: Schema-guided prediction

Stage 1: Schema-guided prediction

- In this stage, we utilize a trained graph neural network designed specifically for schema graph. The input is the instantiated schema graph.
- This process uses the graph representation and the node representation learned from the GCN to score and select from unmatched events in the instantiated event graphs, leading to the first-stage prediction output.

Stage 2: Constrained prediction For the second stage, we implement two constraints regarding logic gates and hierarchies, as shown in Figure 10.

- First, child-to-parent propagation dictates that if a node is predicted or matched, its parent node will also be predicted.

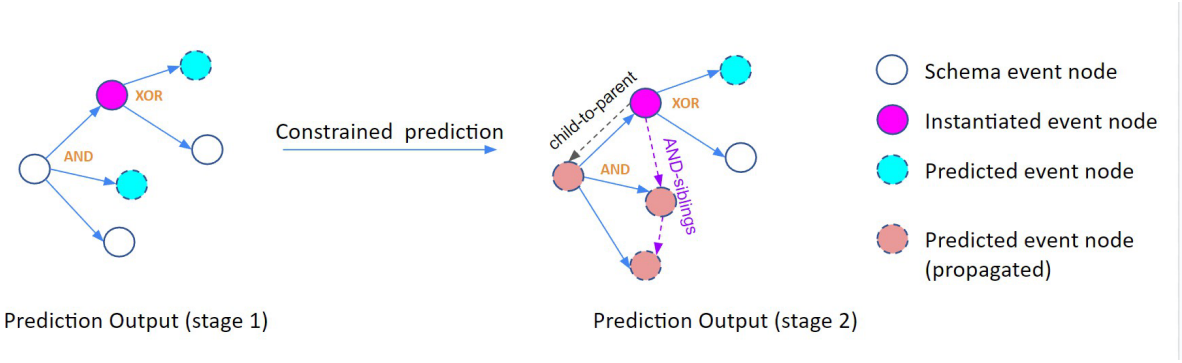

Figure 10: Constrained prediction

- Second, the AND-siblings propagation ensures that if a predicted node has AND siblings, all of its siblings will also be predicted.
- We continue to apply this constrained prediction approach iteratively until no further nodes can be predicted.

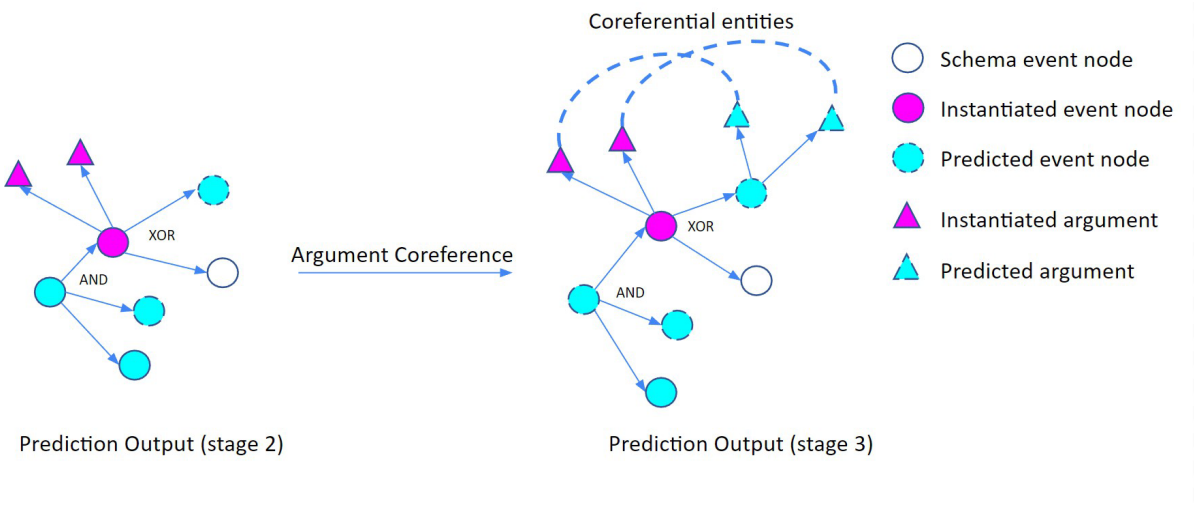

Figure 11: Argument coreference

State 3: Argument coreference

- In this phase (see Figure 11), we utilize the coreference entity links specified in the schema and the instantiated entities obtained from the instantiation outputs to generate predictions for the arguments associated with our predicted events. This constitutes our final prediction output.

4 RESULTS AND DISCUSSION

4.1 Human Curation

4.1.1 Results and Discussion

Human Knowledge Resources

- CMU's schema library went through many versions over the course of the project. The final schema library contains 12 schemas. They are:
  - airplane crash
  - buy goods in store
  - buy goods online
  - coal mine accident
  - common source disease outbreak complex
  - coup
  - hazardous spill
  - invasion
  - propagated disease outbreak complex
  - riot
  - terrorist attack
  - wildfire
- The total number of documents annotated through the course of the project is 588. Of these annotations, 288 use Qnodes and 300 use the LDC tagset.
- Human curators evaluated results from CMU TA2 output and from CMU system output using other team's Schema Libraries. The number of output examined by human curators for the various phases are:
  - Phase 1 - 1
  - Phase 2a - 42
  - Phase 2b dry run - 25
  - Phase 2b final - 40

4.2 Matching and Instantiation

We found that our matching and instantiation approach found suitable matches much of the time as part of our evaluation prior to the last PI meeting. Examples of schemas with suitable, passable, and failed matches are depicted in Figure 12 below.

As mentioned earlier, we ran into issues with weight coefficient tuning affecting matching and instantiation. These issues were pronounced in testing for the Phase2B evaluation. Some results from testing our own system are below. The number of matched events is adequate for both CMU and SBU TA1 schema libraries. The human evaluated matching accuracy is the proportion of matches deemed of high quality by human evaluators, in correctly instantiated schemas. These values are passably high for this measurement. But the proportion of correct instantiations is too low, resulting in entirely wrong schemas too often. (In the case where we used SBU TA1 schema libraries, this was in part due to the configuration of their library.) This trend can potentially be improved by tuning of the weight coefficients in the instantiation process.

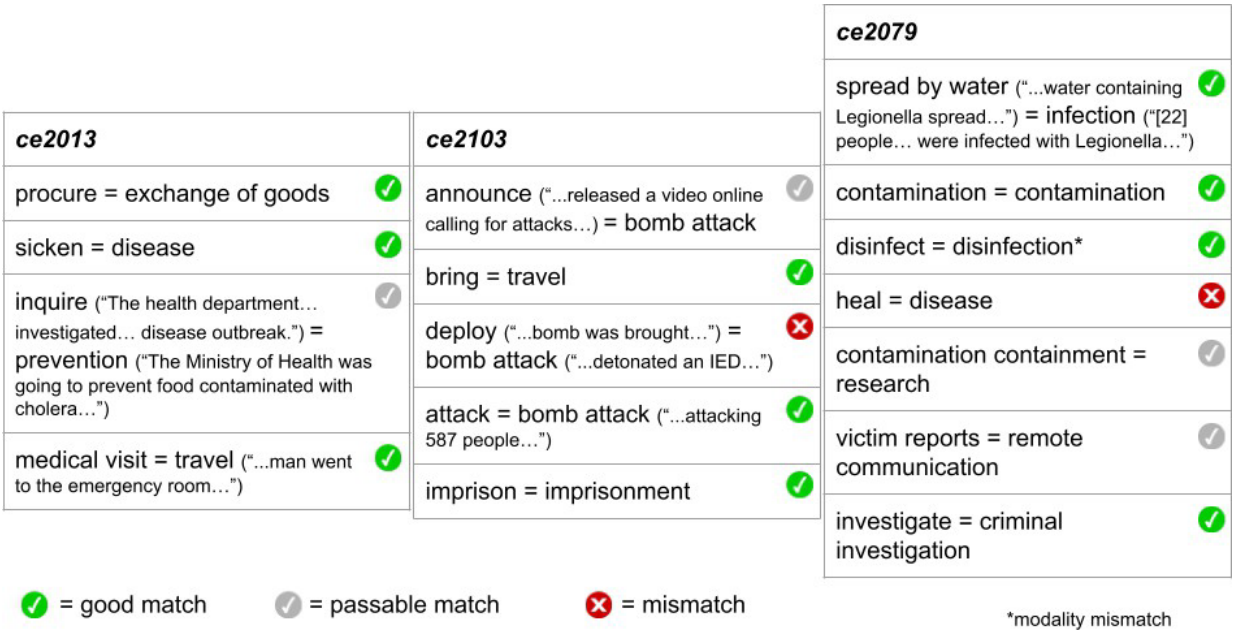

Figure 12: Good, possible, and bad matches.

Table 1: Evaluation results for matching and instantiation

Phase2B: CMU TA2, Task 2		
Number of graphs G evaluated		15
	CMU TA1	SBU TA1
Correct instantiations	7 / 15	5 / 15
Number of events matched	13.6	8.8
Human evaluated matching accuracy	0.68	0.63

4.2.1 Software
4.2.2

Our software for matching and instantiation is available through the provided docker system or via our private GitHub repository: `https://github.com/CHRONOS-KAIROS/matching_instantiation`

Our software pipeline is divided into a series of components, outlined below. Extraction of information from schemas and Graphs G Before matching and instantiation, we reorganize data in the schema and graph G JSON files. The purpose of this is to easily access information about each event's relational context (temporal context, hierarchical context, logical context, participant context, etc.) during the matching process.

Script for this: `extract_event_info.py`

Example commands:

```
python extract_event_info.py --in-file <json_filepath> --out-dir <output_directory> --graph --create-outlinks
python extract_event_info.py --in-file <json_filepath> --out-dir <output_directory> --create-outlinks
```

Args:

- –in-file (str): path to input JSON file. This will be for an unprocessed schema or a graph G.
- –out-dir (str): path to output directory (i.e. the directory that the program will write processed schemas or graphs G to).
- –graph (`set_True`): Use this flag if you are pre-processing a graph G. Omit it if you are pre-processing a

schema library. With the graph G option, a different main function will be called to account for structural differences between the file types.

- –relations (`set_True`): Use this flag if we are relying on temporal relations for temporal information, rather than outlinks. If this flag is set, relations are used and outlinks are ignored. Without this flag, outlinks are used and relations are ignored. We do not use this flag in our current implementation.
- –create-outlinks: Use this flag if, rather than simply relying on relations or outlinks, we want to use the relations in the unprocessed files and convert them to outlinks in the processed files. We added this flag because it became a requirement for us to use outlinks rather than relations in our TA2 outputs, for valid SDF. But the TA1 team and Team G were not required to use outlinks, so they stayed with temporal relations (which we had all been using previously). This flag is necessary in our current implementation.

Extracted information for each event is stored under privateData. As outlined earlier, we extract: event parent, event children, temporal predecessors (pre-context), temporal successors (post-context), AND-siblings, AND-nieces and aunts, XOR-siblings, XOR-nieces and aunts, and participant-sharing events.

Note: We remove privateData at the end of the instantiation process because it is not valid SDF anymore. We actually save instantiated outputs with privateData still in them, though, because we need it to produce the Human Readable Format (HRF). We sent both outputs with and without privateData to the prediction team so that they can produce final outputs both with privateData (for HRF) and without it (for SDF validation).

4.2.3 Matching and instantiation

Script for this: `matching_instantiation_phase2b.py`

Matching and Instantiation involves schema event grounding across a set of extracted elements from multiple documents. These documents are multimodal in nature and the matching and instantiation module involves finding the best match available between schema and graph G events, given its temporal context and various other attributes.

Event Matching We defined inter-event similarity as the weighted sum of multiple similarity measurements: text-based proximity of names and descriptions, proximity of children and parents, proximity of siblings and relatives under an AND gate, dissimilarity of siblings and relatives under an XOR gate, proximity of arguments, proximity of temporal predecessors and successors, and proximity of events sharing the same arguments. In this manner, our definition of event similarity incorporates both temporal, hierarchical, and argumental relations. There are hyperparameters in this matching as we need to weight different relations. This was done by manually analyzing output of various schema-graphs G combinations. Final score weighs direct event relations like event's name, description, etc., more than the other relations like parent event, AND sibling, etc.

Entity Matching We use transformer based models to generate a joint embedding for various entity attributes like name, description, role and use similarity between them to select the best entity match.

Top Schema Matching We select the top schema based on the number of matches and total number of events in that subschema.

4.2.4 End-to-end script

Main script: `run_ta2.sh`

Example command: `bash run_ta2.sh <schema input dir> <graph G input dir> <output dir> <log d`

This script runs:

```
extract_event_info.py
matching_instantiation_phase2b.py
postprocess.py
```

. . . for all graph G and schema library combinations, using the JSON files in the directories provided.

Arguments are positional.

- Output JSON files (instantiated schemas) are written to the `output_dir`
- Logs are written to the `log_dir`. These are helpful for troubleshooting.
- Input schema library and graph G JSON files (uninstantiated) are in the provided input directories.

4.2.5 Other helpful scripts

- `spot_validate.sh`: Used to call the CACI validator API to validate a single file (usually applied in a bash loop to do all files at once, though there is a better way to do this). Currently not working since the validator is having problems.
- `postprocess.py`: This script takes the outputs from `matching_instantiation_phase2b.py` (which have privateData) and removes privateData, writing outputs to a new directory. The outputs containing private-Data are saved in a directory with the suffix `_PD`.
- `Dockerfile`: used to compile the docker system
- `requirements.txt`: all required libraries for pip installation (if not using docker).

4.3 Event and Argument Prediction: Major Results

4.3.1 Inspection of Prediction of Event of Interests

We followed the evaluation procedure outlined by the NIST Phase 2b Dry Run Report, focusing on the predictions of events of interest. Specifically, for each graph G, NIST omits certain events of interest, and our system is tasked with predicting these events. We conducted a manual evaluation on three specific graphs (CE2013, CE2103, CE2079) and reported the recall scores. The recall score is calculated as the ratio of predicted events of interest to the total events of interest, with a higher score indicating better performance.

Table 2: Precision and recall of prediction systems on three complex event datasets

graph G	ce2013		ce2103		ce2079	
Schema Library	CMU	SBU	CMU	SBU	CMU	SBU
AddPath (our previous system)	2 / 3	2 / 3	1 / 2	0 / 2	1 / 3	1 / 3
Ours (our latest version)	**3 / 3**	**3 / 3**	1 / 2	**1 / 2**	**3 / 3**	**3 / 3**
Schema-GCN (RESIN)*	n / a				RESIN	ISI
					2 / 3	0 / 3

4.3.2 Inspection of Unexpected Prediction

We also conducted a manual inspection of unexpected prediction to examine if there are any predicted events violate the logic gate constraints or should not have been predicted based on the evidence in the context of graph G. This evaluation primarily aims to provide an understanding of the precision of the prediction. It is worth noting that defining precision in our task is challenging due to the lack of annotations. The score is calculated as the ratio of unexpected predicted events to total predicted events, with a lower score indicating better performance.

Table 3: Unexpected predictions. The full sheet is also available.

graph G	ce2013	ce2020	ce2024	ce2075	ce2094	
Ours	(our latest version)	0 / 27	0 / 26	0 / 24	0 / 25	0 / 27

4.3.3 Conclusion

We have presented an approach to event prediction over schema graphs that amalgamates schema-guided prediction, constrained prediction, and argument coreference. Our method has demonstrated promising results, outpacing existing techniques in recall scores while generating fewer unexpected predictions. This achievement underscores the prowess of our method in tackling intricate and structured schema graphs and in adeptly forecasting missing events. Our future endeavors will concentrate on enhancing the recall of our predictions and broadening the method's applicability to novel scenarios.

We posit that the representation of schemas as programs (Madaan et al., 2022b; Gao et al., 2023) holds significant potential. Viewing schema induction and prediction through the lens of program synthesis and code completion can be transformative. Recent studies have highlighted the promise of such methods in complex reasoning tasks (Zhao et al., 2023).

Finally, leveraging the latest techniques from feedback-driven structure rectification (Tandon et al., 2022; Madaan et al., 2022a) can be instrumental in refining auto-generated schemas and populating missing information.

5 CONCLUSIONS

This reports has presented a description of the CMU CHRONOS system, which includes a schema visualization and curation tool—called Eratosthenes; practices for schema library curation; subsystems for extracting events, entities, and relations from text and multimedia; a schema matching and instantiation subsystem; and an event/entity prediction subsystem.

Work on the CMU CHRONOS system was halted prematurely due to a program decision. This document presents its current state. However, funding for the KAIROS Plus project allows us to continue work on automatic schema induction, which is proceeding apace. Our KAIROS Plus model produces a large library of high-quality schemas from many domains using an interaction between LLM-driven automatic processes and human curation. A report presenting these results will be submitted later this year.

6 REFERENCES

Tanel Aluma¨e. 2018. kaldi-offline-transcriber. GitHub repository.

Siddhant Arora, Alissa Ostapenko, Vijay Viswanathan, Siddharth Dalmia, Florian Metze, Shinji Watanabe, and Alan Black. 2021. Rethinking end-to-end evaluation of decomposable tasks: A case study on spoken language understanding. *Interspeech*.

Xu Bao, Zhi-Qi Cheng, Jun-Yan He, Chenyang Li, Wangmeng Xiang, Jingdong Sun, Hanbing Liu, Wei Liu, Bin Luo, Yifeng Geng, et al. 2023. Keyposs: Plug-and-play facial landmark detection through gps-inspired true-range multilateration. *arXiv preprint arXiv:2305.16437*.

Xavier Carreras and Llu´ıs Ma`rquez. 2005. Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 152–164, Ann Arbor, Michigan. Association for Computational Linguistics.

Angel X. Chang and Christopher Manning. 2012. SUTime: A library for recognizing and normalizing time expressions. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3735–3740, Istanbul, Turkey. European Language Resources Association (ELRA).

Zhi-Qi Cheng, Qi Dai, Hong Li, Jingkuan Song, Xiao Wu, and Alexander G. Hauptmann. 2022a. Rethinking spatial invariance of convolutional networks for object counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19606–19616, Los Alamitos, CA, USA. IEEE Computer Society.

Zhi-Qi Cheng, Qi Dai, Siyao Li, Teruko Mitamura, and Alexander G. Hauptmann. 2022b. Gsrformer: Grounded situation recognition transformer with alternate semantic attention refinement. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, page 3272–3281, New York, NY, USA. Association for Computing Machinery.

Zhi-Qi Cheng, Qi Dai, Siyao Li, Jingdong Sun, Teruko Mitamura, and Alexander G Hauptmann. 2023. Chartreader: A unified framework for chart derendering and comprehension without heuristic rules. *arXiv preprint arXiv:2304.02173*.

Zhi-Qi Cheng, Yang Liu, Xiao Wu, and Xian-Sheng Hua. 2016. Video ecommerce: Towards online video advertising. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 1365–1374.

Zhi-Qi Cheng, Xiao Wu, Siyu Huang, Jun-Xiu Li, Alexander G Hauptmann, and Qiang Peng. 2018. Learning to transfer: Generalizable attribute learning with multitask neural model search. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 90–98.

Zhi-Qi Cheng, Xiao Wu, Yang Liu, and Xian-Sheng Hua. 2017a. Video ecommerce++: Toward large scale online video advertising. *IEEE transactions on multimedia*, 19(6):1170–1183.

Zhi-Qi Cheng, Xiao Wu, Yang Liu, and Xian-Sheng Hua. 2017b. Video2shop: Exact matching clothes in videos to online shopping images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4048–4056.

Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. 2018. Ultra-fine entity typing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 87–96, Melbourne, Australia. Association for Computational Linguistics.

Agata Cybulska and Piek Vossen. 2014. Using a sledgehammer to crack a nut? lexical diversity and event coreference resolution. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4545–4552, Reykjavik, Iceland. European Language Resources Association (ELRA).

Siddharth Dalmia, Brian Yan, Vikas Raunak, Florian Metze, and Shinji Watanabe. 2021. Searchable hidden intermediates for end-to-end models of decomposable sequence tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1882–1896, Online. Association for Computational Linguistics.

Jiankang Deng, Jia Guo, Yuxiang Zhou, Jinke Yu, Irene Kotsia, and Stefanos Zafeiriou. 2019. Retinaface: Single-stage dense face localisation in the wild. *arXiv preprint arXiv:1905.00641*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2021. GSum: A general framework for guided neural abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4830–4842, Online. Association for Computational Linguistics.

Zi-Yi Dou and Graham Neubig. 2021. Word alignment by fine-tuning embeddings on parallel corpora. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2112–2128, Online. Association for Computational Linguistics.

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683, Online. Association for Computational Linguistics.

Steven Y. Feng, Varun Gangal, Dongyeop Kang, Teruko Mitamura, and Eduard Hovy. 2020. GenAug: Data augmentation for finetuning text generators. In *Proceedings of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 29–42, Online. Association for Computational Linguistics.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021a. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online. Association for Computational Linguistics.

Steven Y. Feng, Jessica Huynh, Chaitanya Prasad Narisetty, Eduard Hovy, and Varun Gangal. 2021b. SAPPHIRE: Approaches for enhanced concept-to-text generation. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 212–225, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Jinlan Fu, Xuanjing Huang, and Pengfei Liu. 2021. SpanNER: Named entity re-/recognition as span prediction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7183–7195, Online. Association for Computational Linguistics.

Jinlan Fu, Pengfei Liu, and Graham Neubig. 2020. Interpretable multi-dataset evaluation for named entity recognition. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6058–6069, Online. Association for Computational Linguistics.

Varun Gangal, Steven Y. Feng, Malihe Alikhani, Teruko Mitamura, and Eduard Hovy. 2022. Nareor: The narrative reordering problem. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):10645–10653.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Pal: Program-aided language models. In *International Conference on Machine Learning*, pages 10764–10799. PMLR.

Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. 2017. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 776–780. IEEE.

Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. 2016. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2315–2324.

Kaiming He, Georgia Gkioxari, Piotr Dolla´r, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Jonathan Herzig, Paweł Krzysztof Nowak, Thomas Mu¨ller, Francesco Piccinno, and Julian Martin Eisenschlos. 2020. Tapas: Weakly supervised table parsing via pre-training. *arXiv preprint arXiv:2004.02349*.

Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Ting-Yao Hu, Zhi-Qi Cheng, and Alexander G Hauptmann. 2021. Subspace representation learning for few-shot image classification. *arXiv preprint arXiv:2105.00379*.

Ting-Yao Hu and Alexander G. Hauptmann. 2021a. Pose guided person image generation with hidden p-norm regression. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 2423–2427.

Ting-Yao Hu and Alexander G. Hauptmann. 2021b. Statistical distance metric learning for image set retrieval. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1765–1769.

Siyu Huang, Xi Li, Zhi-Qi Cheng, Zhongfei Zhang, and Alexander Hauptmann. 2018. Gnas: A greedy neural architecture search method for multi-attribute learning. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 2049–2057.

Adrien Le Franc, Eric Riebling, Julien Karadayi, Yun Wang, Camila Scaff, Florian Metze, and Alejandrina Cristia. 2018. The aclew divime: An easy-to-use diarization tool. In *Interspeech*, pages 1383–1387.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 687–692, New Orleans, Louisiana. Association for Computational Linguistics.

Xinjian Li, Siddharth Dalmia, Juncheng Li, Matthew Lee, Patrick Littell, Jiali Yao, Antonios Anastasopoulos, David R Mortensen, Graham Neubig, Alan W Black, et al. 2020. Universal phone recognition with a multilingual allophone system. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8249–8253. IEEE.

Tsung-Yi Lin, Piotr Dolla´r, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125.

Pengfei Liu, Jinlan Fu, Yang Xiao, Weizhe Yuan, Shuaichen Chang, Junqi Dai, Yixin Liu, Zihuiwen Ye, and Graham Neubig. 2021a. ExplainaBoard: An explainable leaderboard for NLP. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 280–289, Online. Association for Computational Linguistics.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. 2016. Ssd: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision*, pages 21–37. Springer.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Yixin Liu, Zi-Yi Dou, and Pengfei Liu. 2021b. RefSum: Refactoring neural summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1437–1448, Online. Association for Computational Linguistics.

Yixin Liu and Pengfei Liu. 2021. SimCLS: A simple framework for contrastive learning of abstractive summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 1065–1072, Online. Association for Computational Linguistics.

Shangbang Long, Xin He, and Zhongfei Mark Yaofu. 2018. Textsnake: A flexible representation for detecting text of arbitrary shapes. In *Proceedings of the European conference on computer vision (ECCV)*, pages 19–35.

Aman Madaan, Dheeraj Rajagopal, Niket Tandon, Yiming Yang, and Eduard Hovy. 2021a. Could you give me a hint ? generating inference graphs for defeasible reasoning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5138–5147, Online. Association for Computational Linguistics.

Aman Madaan, Shruti Rijhwani, Antonios Anastasopoulos, Yiming Yang, and Graham Neubig. 2020a. Practical comparable data collection for low-resource languages via images. In ICLR 2020 workshop for AfricaNLP.

Aman Madaan, Amrith Setlur, Tanmay Parekh, Barnabas Poczos, Graham Neubig, Yiming Yang, Ruslan Salakhutdinov, Alan W Black, and Shrimai Prabhumoye. 2020b. Politeness transfer: A tag and generate approach. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1869–1881, Online. Association for Computational Linguistics.

Aman Madaan, Niket Tandon, Peter Clark, and Yiming Yang. 2022a. Memory-assisted prompt editing to improve gpt-3 after deployment. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2833–2861.

Aman Madaan, Niket Tandon, Dheeraj Rajagopal, Peter Clark, Yiming Yang, and Eduard Hovy. 2021b. Think about it! improving defeasible reasoning by first modeling the question scenario. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6291–6310, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Aman Madaan and Yiming Yang. 2021. Neural language modeling for contextualized temporal graph generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 864–881, Online. Association for Computational Linguistics.

Aman Madaan, Shuyan Zhou, Uri Alon, Yiming Yang, and Graham Neubig. 2022b. Language models of code are few-shot commonsense learners. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1384–1403.

Chong-Wah Ngo, Zhi-Qi Cheng, and Xiao Wu. 2017. Minimizing risk in video hyperlinking. In *2017 TREC Video Retrieval Evaluation (TRECVID 2017)*.

Phuong Anh Nguyen, Qing Li, Zhi-Qi Cheng, Yi-Jie Lu, Hao Zhang, Xiao Wu, and Chong-Wah Ngo. 2017. Vireo@ trecvid 2017: Video-to-text, ad-hoc video search and video hyperlinking.

Jiefu Ou, Adithya Pratapa, Rishubh Gupta, and Teruko Mitamura. 2023. Hierarchical event grounding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(11):13437–13445.

Daniel Povey, Vijayaditya Peddinti, Daniel Galvez, Pegah Ghahremani, Vimal Manohar, Xingyu Na, Yiming Wang, and Sanjeev Khudanpur. 2016. Purely sequence-trained neural networks for asr based on lattice-free mmi. In *Interspeech*, pages 2751–2755.

Adithya Pratapa, Antonios Anastasopoulos, Shruti Rijhwani, Aditi Chaudhary, David R. Mortensen, Graham Neubig, and Yulia Tsvetkov. 2021a. Evaluating the morphosyntactic well-formedness of generated texts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7131–7150, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Adithya Pratapa, Rishubh Gupta, and Teruko Mitamura. 2022. Multilingual event linking to Wikidata. In *Proceedings of the Workshop on Multilingual Information Access (MIA)*, pages 37–58, Seattle, USA. Association for Computational Linguistics.

Adithya Pratapa, Zhengzhong Liu, Kimihiro Hasegawa, Linwei Li, Yukari Yamakawa, Shikun Zhang, and Teruko Mitamura. 2021b. Cross-document event identity via dense annotation. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 496–517, Online. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Zobeir Raisi, Mohamed A Naiel, Georges Younes, Steven Wardell, and John S Zelek. 2021. Transformer-based text detection in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3162–3171.

Abhilasha Ravichander, Siddharth Dalmia, Maria Ryskina, Florian Metze, Eduard Hovy, and Alan W Black. 2021. NoiseQA: Challenge set evaluation for user-centric question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2976–2992, Online. Association for Computational Linguistics.

Joseph Redmon and Ali Farhadi. 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Nathaniel Robinson, Cameron Hogan, Nancy Fulda, and David R. Mortensen. 2022a. Data-adaptive transfer learning for translation: A case study in Haitian and jamaican. In *Proceedings of the Fifth Workshop on Technologies for Machine Translation of Low-Resource Languages (LoResMT 2022)*, pages 35–42, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Nathaniel Romney Robinson, Perez Ogayo, Swetha R. Gangu, David R. Mortensen, and Shinji Watanabe. 2022b. When is TTS augmentation through a pivot language useful? In *Interspeech 2022*.

Baoguang Shi, Xiang Bai, and Cong Yao. 2016. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2297–2306.

Elizabeth Spaulding, Anatole Gershman, Rosario Uceda-Sosa, Susan Brown, James Pustejovsky, Peter Anick, and Martha. Palmer. 2023. The darpa wikidata overlay: Wikidata as an ontology for natural language processing. In preparation.

Pontus Stenetorp, Sampo Pyysalo, Goran Topic´, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. brat: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, Avignon, France. Association for Computational Linguistics.

Jannik Stro¨tgen and Michael Gertz. 2016. Domain-sensitive temporal tagging. In *Domain-Sensitive Temporal Tagging*, pages 47–83. Springer.

Niket Tandon, Aman Madaan, Peter Clark, and Yiming Yang. 2022. Learning to repair: Repairing model output errors after deployment using a dynamic memory of feedback. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 339–352.

Priyam Tejaswin, Dhruv Naik, and Pengfei Liu. 2021. How well do you know your summarization datasets? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3436–3449, Online. Association for Computational Linguistics.

Naushad UzZaman, Hector Llorens, James Allen, Leon Derczynski, Marc Verhagen, and James Pustejovsky. 2012. Tempeval-3: Evaluating events, time expressions, and temporal relations. *arXiv preprint arXiv:1206.5333*.

Vijay Viswanathan, Graham Neubig, and Pengfei Liu. 2021. CitationIE: Leveraging the citation graph for scientific information extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 719–731, Online. Association for Computational Linguistics.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. *Linguistic Data Consortium, Philadelphia*, 57:45.

Sijia Wang, Mo Yu, Shiyu Chang, Lichao Sun, and Lifu Huang. 2022. Query and extract: Refining event extraction as type-oriented binary decoding. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 169–182, Dublin, Ireland. Association for Computational Linguistics.

Yun Wang. 2018. Polyphonic sound event detection with weak labeling. *PhD thesis*.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. OntoNotes Release 5.0. *LDC2013T19, Philadelphia, Penn.: Linguistic Data Consortium*.

Brian Yan, Siddharth Dalmia, David R. Mortensen, Florian Metze, and Shinji Watanabe. 2021. Differentiable allophone graphs for language-universal speech recognition. In *22nd Annual Conference of the International Speech Communication Association, INTERSPEECH 2021*, Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, pages 356–360. International Speech Communication Association. Funding Information: We thank Xinjian Li and Awni Hannun for helpful discussions. This work was supported in part by grants from National Science Foundation for Bridges PSC (ACI-1548562, ACI-1445606) and DARPA KAIROS program from the Air Force Research Laboratory (FA8750-19-2-0200). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. Publisher Copyright: Copyright © 2021 ISCA.; 22nd Annual Conference of the International Speech Communication Association, INTERSPEECH 2021 ; Conference date: 30-08-2021 Through 03-09-2021.

Zihuiwen Ye, Pengfei Liu, Jinlan Fu, and Graham Neubig. 2021. Towards more fine-grained and reliable NLP performance prediction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3703–3714, Online. Association for Computational Linguistics.

Wenting Zhao, Mor Geva, Bill Yuchen Lin, Michihiro Yasunaga, Aman Madaan, and Tao Yu. 2023. Cutting-edge tutorial: Complex reasoning over natural language. In *The 61st Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, page 11.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2017a. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Shuyan Zhou, Pengcheng Yin, and Graham Neubig. 2022a. Hierarchical control of situated agents through natural language. In *Proceedings of the Workshop on Structured and Unstructured Knowledge Integration (SUKI)*, pages 67–84, Seattle, USA. Association for Computational Linguistics.

Shuyan Zhou, Li Zhang, Yue Yang, Qing Lyu, Pengcheng Yin, Chris Callison-Burch, and Graham Neubig. 2022b. Show me more details: Discovering hierarchies of procedures from semi-structured web data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2998–3012, Dublin, Ireland. Association for Computational Linguistics.

Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. 2017b. East: an efficient and accurate scene text detector. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5551–5560.

A APPENDIX A — PUBLICATIONS AND PRESENTATIONS

Siddhant Arora, Alissa Ostapenko, Vijay Viswanathan, Siddharth Dalmia, Florian Metze, Shinji Watanabe, and Alan Black. 2021. Rethinking end-to-end evaluation of decomposable tasks: A case study on spoken language understanding. *Interspeech*.

Zhi-Qi Cheng, Qi Dai, Hong Li, Jingkuan Song, Xiao Wu, and Alexander G. Hauptmann. 2022a. Rethinking spatial invariance of convolutional networks for object counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19606–19616, Los Alamitos, CA, USA. IEEE Computer Society.

Zhi-Qi Cheng, Qi Dai, Siyao Li, Teruko Mitamura, and Alexander G. Hauptmann. 2022b. Gsrformer: Grounded situation recognition transformer with alternate semantic attention refinement. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, page 3272–3281, New York, NY, USA. Association for Computing Machinery.

Siddharth Dalmia, Brian Yan, Vikas Raunak, Florian Metze, and Shinji Watanabe. 2021. Searchable hidden intermediates for end-to-end models of decomposable sequence tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1882–1896, Online. Association for Computational Linguistics.

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2021. GSum: A general framework for guided neural abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4830–4842, Online. Association for Computational Linguistics.

Zi-Yi Dou and Graham Neubig. 2021. Word alignment by fine-tuning embeddings on parallel corpora. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2112–2128, Online. Association for Computational Linguistics.

Steven Y. Feng, Varun Gangal, Dongyeop Kang, Teruko Mitamura, and Eduard Hovy. 2020. GenAug: Data augmentation for finetuning text generators. In *Proceedings of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 29–42, Online. Association for Computational Linguistics.

Steven Y. Feng, Jessica Huynh, Chaitanya Prasad Narisetty, Eduard Hovy, and Varun Gangal. 2021b. SAPPHIRE: Approaches for enhanced concept-to-text generation. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 212–225, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021a. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online. Association for Computational Linguistics.

Jinlan Fu, Pengfei Liu, and Graham Neubig. 2020. Interpretable multi-dataset evaluation for named entity recognition. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6058–6069, Online. Association for Computational Linguistics.

Jinlan Fu, Xuanjing Huang, and Pengfei Liu. 2021. SpanNER: Named entity re-/recognition as span prediction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7183–7195, Online. Association for Computational Linguistics.

Varun Gangal, Steven Y. Feng, Malihe Alikhani, Teruko Mitamura, and Eduard Hovy. 2022. Nareor: The narrative reordering problem. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):10645–10653.

Ting-Yao Hu and Alexander G. Hauptmann. 2021a. Pose guided person image generation with hidden p-norm regression. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 2423–2427.

Ting-Yao Hu and Alexander G. Hauptmann. 2021b. Statistical distance metric learning for image set retrieval. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1765–1769.

Pengfei Liu, Jinlan Fu, Yang Xiao, Weizhe Yuan, Shuaichen Chang, Junqi Dai, Yixin Liu, Zihuiwen Ye, and Graham Neubig. 2021a. ExplainaBoard: An explainable leaderboard for NLP. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 280–289, Online. Association for Computational Linguistics.

Yixin Liu, Zi-Yi Dou, and Pengfei Liu. 2021b. RefSum: Refactoring neural summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1437–1448, Online. Association for Computational Linguistics.

Yixin Liu and Pengfei Liu. 2021. SimCLS: A simple framework for contrastive learning of abstractive summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 1065–1072, Online. Association for Computational Linguistics.

Aman Madaan, Amrith Setlur, Tanmay Parekh, Barnabas Poczos, Graham Neubig, Yiming Yang, Ruslan Salakhutdinov, Alan W Black, and Shrimai Prabhumoye. 2020b. Politeness transfer: A tag and generate approach. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1869–1881, Online. Association for Computational Linguistics.

Aman Madaan, Shruti Rijhwani, Antonios Anastasopoulos, Yiming Yang, and Graham Neubig. 2020a. Practical comparable data collection for low-resource languages via images. In ICLR 2020 workshop for AfricaNLP.

Aman Madaan, Dheeraj Rajagopal, Niket Tandon, Yiming Yang, and Eduard Hovy. 2021a. Could you give me a hint ? generating inference graphs for defeasible reasoning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5138–5147, Online. Association for Computational Linguistics.

Aman Madaan and Yiming Yang. 2021. Neural language modeling for contextualized temporal graph generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 864–881, Online. Association for Computational Linguistics.

Aman Madaan, Niket Tandon, Dheeraj Rajagopal, Peter Clark, Yiming Yang, and Eduard Hovy. 2021b. Think about it! improving defeasible reasoning by first modeling the question scenario. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6291–6310, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Adithya Pratapa, Zhengzhong Liu, Kimihiro Hasegawa, Linwei Li, Yukari Yamakawa, Shikun Zhang, and Teruko Mitamura. 2021b. Cross-document event identity via dense annotation. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 496–517, Online. Association for Computational Linguistics.

Adithya Pratapa, Antonios Anastasopoulos, Shruti Rijhwani, Aditi Chaudhary, David R. Mortensen, Graham Neubig, and Yulia Tsvetkov. 2021a. Evaluating the morphosyntactic well-formedness of generated texts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7131–7150, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Adithya Pratapa, Rishubh Gupta, and Teruko Mitamura. 2022. Multilingual event linking to Wikidata. In *Proceedings of the Workshop on Multilingual Information Access (MIA)*, pages 37–58, Seattle, USA. Association for Computational Linguistics.

Jiefu Ou, Adithya Pratapa, Rishubh Gupta, and Teruko Mitamura. 2023. Hierarchical event grounding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(11):13437–13445.

Abhilasha Ravichander, Siddharth Dalmia, Maria Ryskina, Florian Metze, Eduard Hovy, and Alan W Black. 2021. NoiseQA: Challenge set evaluation for user-centric question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2976–2992, Online. Association for Computational Linguistics.

Nathaniel Robinson, Cameron Hogan, Nancy Fulda, and David R. Mortensen. 2022a. Data-adaptive transfer learning for translation: A case study in Haitian and jamaican. In *Proceedings of the Fifth Workshop on Technologies for Machine Translation of Low-Resource Languages (LoResMT 2022)*, pages 35–42, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Nathaniel Romney Robinson, Perez Ogayo, Swetha R. Gangu, David R. Mortensen, and Shinji Watanabe. 2022b. When is TTS augmentation through a pivot language useful? In *Interspeech 2022*.

Elizabeth Spaulding, Anatole Gershman, Rosario Uceda-Sosa, Susan Brown, James Pustejovsky, Peter Anick, and Martha. Palmer. 2023. The darpa wikidata overlay: Wikidata as an ontology for natural language processing. In preparation.

Priyam Tejaswin, Dhruv Naik, and Pengfei Liu. 2021. How well do you know your summarization datasets? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3436–3449, Online. Association for Computational Linguistics.

Vijay Viswanathan, Graham Neubig, and Pengfei Liu. 2021. CitationIE: Leveraging the citation graph for scientific information extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 719–731, Online. Association for Computational Linguistics.

Brian Yan, Siddharth Dalmia, David R. Mortensen, Florian Metze, and Shinji Watanabe. 2021. Differentiable allophone graphs for language-universal speech recognition. In *22nd Annual Conference of the International Speech Communication Association, INTERSPEECH 2021*, Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, pages 356–360. International Speech Communication Association. Funding Information: We thank Xinjian Li and Awni Hannun for helpful discussions. This work was supported in part by grants from National Science Foundation for Bridges PSC (ACI-1548562, ACI-1445606) and DARPA KAIROS program from the Air Force Research Laboratory (FA8750-19-2-0200). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. Publisher Copyright: Copyright © 2021 ISCA.; 22nd Annual Conference of the International Speech Communication Association, INTERSPEECH 2021 ; Conference date: 30-08-2021 Through 03-09-2021.

Zihuiwen Ye, Pengfei Liu, Jinlan Fu, and Graham Neubig. 2021. Towards more fine-grained and reliable NLP performance prediction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3703–3714, Online. Association for Computational Linguistics.

Shuyan Zhou, Pengcheng Yin, and Graham Neubig. 2022a. Hierarchical control of situated agents through natural language. In *Proceedings of the Workshop on Structured and Unstructured Knowledge Integration (SUKI)*, pages 67–84, Seattle, USA. Association for Computational Linguistics.

Shuyan Zhou, Li Zhang, Yue Yang, Qing Lyu, Pengcheng Yin, Chris Callison-Burch, and Graham Neubig. 2022b. Show me more details: Discovering hierarchies of procedures from semi-structured web data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2998–3012, Dublin, Ireland. Association for Computational Linguistics.

LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS

API Application Programming Interface

BERT Bidirectional Encoder Representations from Transformers

BPE Byte- Pair Encoding

CHRONOS Chronological and Hierarchical Reasoning Over Newly Observed Schemas

CLIP Contrastive Language-Image Pretraining

CMU Carnegie Mellon University

CRNN Convolutional Recurrent Neural Network

DARPA Defense Advances Research Planning Agency

EAST Easy, Attractive, Social, and Timely

GCN Graph Convolutional Networks

GPU Graphics processing unit

HRF Human Readable Format

IED Improvised Explosive Device

JSON JavaScript Object Notation

KAIROS Knowledge-directed Artificial Intelligence Reasoning Over Schemas

LLM Large Language Model

LDC Linguistic Data Consortium, University of Pennsylvania

NIST National Institute of Standards and Technology

NLP Natural Language Processing

R-CNN Region-Convolutional Neural Network

ROBERTa Robustly Optimized BERT Approach

SBU Stony Brook University

SDF Semantic Definition Format

SED Sound Event Detection

SQL Software Query Language

SRL semantic role labeling

SUTime Stanford Temporal Tagger

TA Technical Area

XOR Exclusively-OR

YOLO You Only Look Once