TAC/TRECVID Streaming Multimedia KBP for AIDA 2018 Evaluation Plan V0.8

Last Updated: October 12, 2018 (V0.8)

Revision History

V0.1: March 30, 2018

- Initial release

V0.2: July 11, 2018

- Revised evaluation schedule
- Updated Ontology
- Update KE and KG representation
- Identified background KB (KBP 2009 Reference KB) and updated requirements for constrained and unconstrained conditions. Moved list of external resources to a separate file.
- Added more information about spans across different modalities and how they are represented: section 4
- Added clarification that audio data will not be part of month 9 evaluation: section 6.2.3
- Added example queries and Task 3 statement of information need
- Added example output format

V0.3: July 25, 2018

- Clarified that more than one HasName property is allowed for an entity/filler
- Added allowance for fringe entity, relation, and event types for each KE and each image or video shot.
- Clarified that "document" is the same as a "root", and "document element" is the same as a "child" in LDC's source corpus packages
- Added more detailed description of shot IDs and keyframes, and how keyframes
- Added description of the how executable SPARQL queries will "match" an entry point to a node in the system KB
- Clarified that audio is part of the pilot evaluation inasmuch as it is part of a video
- Allow Task 2 systems to process Task 1 output for all documents as a single batch (i.e., the "streaming" requirement is not enforced for the pilot evaluation)
- Updated example Task 2 query and response
- Clarified that participants should return responses for both the simplified xml queries and the executable SPARQL queries.

V0.4: August 1, 2018

- Removed PDF files from the evaluation
- Reduced size of evaluation corpus from 90,000 documents down to 10,000 documents
- Detailed impact of restrictions on distribution of Twitter data
- Clarified that cross-lingual and cross-modal coreference of entities and events are required for all tasks.
- Allowed multiple entry point entities per query. Gave more examples of each type of query.

V0.5: August 8, 2018

- Specified additional requirements (beyond those defined in AIF) for knowledge graphs that will be evaluated by applying executable SPARQL queries.

- Added assessment and evaluation protocol for TA1 and TA2

V0.6: August 17, 2018

- Described "what if" hypotheses for Task 1b (Section 7.2.2)
- Tentatively revoked requirement that endpoints of KE edges be represented as AIF clusters (Section 7.2.2)
- Added more detail about assessment and evaluation protocol (Section 7.4.3)
- Updated format of Task 3 statement of information need, in order to more closely parallel the format for a simplified graph query (Section 8.1)
- Provided more detail about reference hypotheses and metrics for Task 3

V0.7: August 29, 2018

- Confirmed that executable SPARQL queries will expect KE edges in AIF to have endpoints that are members of clusters, but that are not themselves clusters.
- Confirmed that up to 2 spans may be returned as a single justification for an edge.
- Added text mentions to the mentions that should be returned in response to class level queries. Clarified that class level queries will target only those entity/filler types that are in scope as entry point entities/fillers.
- Clarified that the AIF hypotheses file should contain all of the information needed to represent the hypothesis graph, including provenance.
- Updated Task 3 alignment procedure to allow multiple system generated hypotheses to match as single reference hypothesis.

V0.8: October 12, 2018

- Clarified that a video span is the entire shot identified by the keyframe, and not just the key frame (which uniquely identifies the shot) (Section 4)
- For the pilot evaluation, allow participants to return 0 for all four coordinates of the bounding box to indicate that they are not attempting to localize a justification beyond the level of the whole image or shot (Section 4)
- Noted that NIST generally follows LDC's determination of coreference for entities/fillers, relations, and events. (Section 7.1)
- For the pilot evaluation, clarified that if none of the descriptors in a query come from the document that's being processed in Task 1, then the Task 1 system would not return any answers for that query. (Section 7.2.1)
- Provided a pointer to the query and response dtds (Section 7.2.2)
- Noted the relationship between edge queries and full graph queries representing an entire hypothesis graph (Section 7.2.2)
- Added more constraints for the restricted AIF format required by executable SPARQL queries: `aida:CompoundJustification` must be used for edge justifications, and should not be used for justifications for entities, fillers, events, or relation KEs; each video justification must be represented by aida:KeyFrameVideoJustification, with explicit keyframe ID and bounding box; each image justification must include an explicit bounding box. (Section 7.2.2)
- Changed relation assessment to follow the same protocol as event assessment (Section 7.4.3)
- Require that all Task 3 AIF graphs conform to the constrained version of AIF required by executable SPARQL queries. Because an entity/filler, relation, or event is represented as a cluster in the constrained version of AIF, the hypothesis must include clusterMember statements. (Section 8.1)
- Added Section 9 (submission procedure)
- Updated schedule and made task submission deadlines approximate. (Section 12)

1 Introduction

In scenarios such as natural disasters or international conflicts, analysts and the public are often confronted with a variety of information coming through multiple media sources. There is a need for technologies to analyze and extract knowledge from multimedia to develop and maintain an understanding of events, situations, and trends around the world, in order to respond to the situations.

The goal of DARPA's Active Interpretation of Disparate Alternatives (AIDA) Program is to develop a multi-hypothesis semantic engine that generates explicit alternative interpretations of events, situations, and trends from a variety of unstructured sources, for use in noisy, conflicting, and potentially deceptive information environments. This engine must be capable of mapping knowledge elements (KE) automatically derived from multiple media sources into a common semantic representation, aggregating information derived from those sources, and generating and exploring multiple hypotheses about the events, situations, and trends of interest. This engine must establish confidence measures for the derived knowledge and hypotheses, based on the accuracy of the analysis and the coherence of the semantic representation of each hypothesis. In addition, the engine must be able to communicate with its user to reveal the generated hypotheses and to allow the user to alter the hypotheses or to suggest new ones.

This document describes the specifications of the evaluation conducted by NIST to assess the performance of systems that have been developed in support of AIDA program goals. The streaming multimedia KBP track will ask systems to extract knowledge elements from a stream of heterogeneous documents containing multimedia sources including text, speech, images, videos, and pdf files; aggregate the knowledge elements from multiple documents without access to the raw documents themselves (maintaining multiple interpretations and confidence values for KEs extracted or inferred from the documents); and develop semantically coherent hypotheses, each of which represents an interpretation of the document stream.

Participation in the NIST Streaming Multimedia KBP (SM-KBP) evaluation is required for all DARPA AIDA performers responsible for the relevant technology areas in AIDA. Task 1a and Task 2 evaluation is also open to all researchers who find the evaluation tasks of interest. There is no cost to participate. Participants are encouraged to attend a post-evaluation workshop to present and discuss their systems and results at their own expense. Information and updates about the tasks and evaluation will be posted to the NIST SM-KBP website.[1]

Novel characteristics of the open evaluation tasks (Task 1a and Task 2) include:

- Task 1: Multimodal multilingual extraction and linking of information within a document
- Task 1 and 2: Processing of streaming input
- Task 1 and 2: Confidence estimation and maintenance of multiple possible interpretations
- Task 2: Cross-document aggregation and linking of information without access to original documents (in order to protect intellectual property rights of various data providers, etc.)

Novel characteristics of the AIDA program-internal evaluation tasks (Task 1b and Task 3) include:

- Document-level extraction and linking conditioned on "feedback hypotheses" providing context.

[1] https://tac.nist.gov/2018/SM-KBP/

- Generation of semantically coherent hypotheses, each representing a different interpretation of the document stream.

The SM-KBP tasks will be run at TAC/TRECVID 2018 as ***pilot*** evaluations whose goals are to test evaluation protocols and metrics and to learn lessons that can inform how subsequent evaluations will be structured. The purpose of the pilot is to exercise the evaluation infrastructure, not to test systems' performance. As such, the pilot intends to be flexible and at the same time to follow the protocol of the official evaluation.

It is expected that the SM-KBP track will be run for 3 evaluation cycles after the initial pilot evaluation:

- Pilot Month-9 (M9) Evaluation
  - Evaluation window September-October 2018
  - Workshop at TAC/TRECVID in Gaithersburg, MD in November 2018
- Phase 1 Evaluation 1 (short cycle)
  - Evaluation window March-April 2019
  - Workshop TBD (possibly end of June 2019)
- Phase 2 Evaluation 2 (18-month cycle)
  - Evaluation window August-September 2020
  - Workshop at TAC/TRECVID in Gaithersburg, MD in November 2020
- Phase 3 Evaluation 3 (18-month cycle)
  - Evaluation window March-April 2022
  - Workshop TBD (possibly end of June 2022)

2 Evaluation Tasks

Evaluation is over a small set of topics for a single scenario. For the 2018 pilot, the scenario is the Russian/Ukrainian conflict (2014-2015). Early in the evaluation cycle, all task participants will receive an ontology of entities, events, event arguments, relations, and SEC (sentiment, emotion, and cognitive state), defining the KEs that are in scope for the evaluation tasks.

There are three main evaluation tasks:

- Task 1 (TA1): Extraction of KEs and KE mentions from a stream of multi-media documents, including linking of mentions of the same KE within each document to produce a document-level knowledge graph for each document. Extraction and linking will be conditioned on two kinds of contexts:
  a. generic background context
  b. generic background context plus a "what if" hypothesis

  For specific Task 1 requirements, see section 7.
- Task 2: Construction of a KB by aggregating and linking document-level knowledge graphs produced by one or more TA1. For specific Task 2 requirements, see Section 7.
- Task 3: Generation of hypotheses from one or more KBs produced by TA2. For specific Task 3 requirements, see Section 8.

AIDA performers are required to participate in the tasks as outlined by their Statement of Work. Open participants (non-AIDA performers) may participate in Task 1a and Task 2.

The three tasks are structured as a pipeline, such that Task 1 participants are given a stream of raw documents and output a stream of document-level knowledge graphs (KGs), Task 2 participants are given the stream of knowledge graphs from Task 1 participants and outputs a corpus-level knowledge graph, and Task 3 participants are given the output of Task 2 participants and outputs a set of knowledge graphs representing hypotheses. However, for the 2018 pilot, only Task 1a output (and not Task 1b output) will be part of the pipeline. Task 1b evaluates a media analysis system's ability to utilize knowledge in the common semantic representation and the generated hypotheses as alternate contexts for the media analysis by altering their models or prior probabilities to enhance accuracy and resolve ambiguities in line with expectations from the context.

For the 2018 pilot exercise, communication between the different components of the pipeline will be by writing system output to and reading from a text file, but in future it may be via an API.

For each task, systems in the task are given input at the beginning of their evaluation window and must submit their output by the end of the evaluation window. Evaluation queries for each task will be applied to the task output after the close of its evaluation window. Evaluation will be by queries that probe and sample system output, with pooling and assessment of responses to those queries. Evaluation queries will be generated from knowledge elements from a set of reference hypotheses and will probe system output for those KEs and related KEs.

TA1 will be evaluated on selected documents in the input document stream. For the pilot evaluation, TA2 and TA3 will be evaluated only at one timepoint (at the end of the data stream).

Table 1: AIDA18 Tasks

Task	**Input**	**Output**
Task 1a	Stream of multi-media documents	Knowledge graph for each document, consisting of KEs found in the document
Task 1b	Stream of multi-media documents; hypothesis as context	Knowledge graph for each document consisting of KEs found in the document, conditioned on context
Task 2	Stream of document-level knowledge graphs from one or more TA1	KB aggregating document-level knowledge graphs into a single corpus-level knowledge graph
Task 3	KBs from one or more TA2; statement of information need for a topic, including core KEs	Set of semantically coherent knowledge graphs (hypotheses) that contain the core KEs

3 Ontology

Participants will be given an ontology of entities, relations and relation arguments, events and event arguments, and SEC (sentiment, emotion, and cognitive state), defining the knowledge elements that are in scope for the evaluation tasks. The ontology will be rich enough to represent informational conflicts in the data stream and the different hypotheses arising from different interpretations of the document stream. Knowledge graphs and evaluation queries will be expressed in the language of the ontology. Unless noted otherwise, all evaluation queries may ask about any and all of the KEs that are defined in the ontology. For the 2018 pilot evaluation, the ontology will be an extension of the DEFT Rich ERE ontology that is partially described in the LDC's documentation for "seedling" annotation on the Russian-Ukrainian scenario topics (AIDA_Seedling_Ontology_Info_V6.xlsx).[2]

In addition to the entity/filler types, relation types, event types and event argument roles defined by LDC's seedling ontology, the 2018 pilot ontology will include the following entity/filler properties that will allow a name or other formulaic string to be associated with particular entities/fillers in a knowledge graph in a way that's accessible to TA1/TA2/TA3:

HasName
Subject: An entity or filler of type {PER,ORG,GPE,FAC,LOC,WEA,VEH,LAW}
Object: a string that must be a name. LDC defines what qualifies as a name. For the pilot evaluation, in addition to the name of a particular Weapon, Vehicle, or Law (e.g., "U.SS.Enterprise" designating a single vehicle), the name of the class or model of a {WEA, VEH, LAW} entity will also be allowed as a name for the entity itself – e.g., "BUK-21" and "Toyota Corolla" are allowable names for a Weapon and Vehicle, respectively; in future, these class names might be predefined as part of the ontology itself. Each entity is allowed to have multiple HasName properties, one for each distinct name that is observed.

NumericValue
Subject: A filler of type {VAL,AGE}
Object: a numeric value

TextValue
Subject: A filler of type {RES,URL,TME,MON}
Object: a string (e.g., a normalized date)

No evaluation queries will target the HasName property directly (i.e., no queries will ask for all names of an entity/filler). However, the name strings in this relation will be used to help match an evaluation query entity/filler with an entity/filler node in a system's knowledge graph.

In the pilot evaluation, temporal properties of relations will *not* be queried for, but the Phase 1 evaluation is expected to include queries about temporal properties of both events and relations.

[2] NIST encourages task participants to share mappings between the AIDA ontology and ontologies that have been used in past evaluations (e.g., DEFT Rich ERE, TAC KBP, TRECVID SIN concepts).

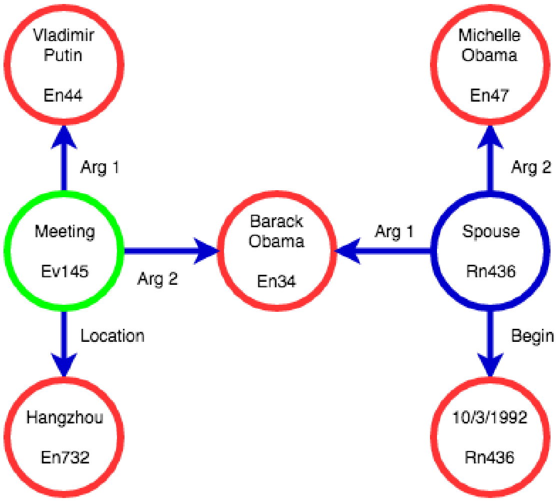
In addition to the KE types that are defined in the ontology, the AIDA program will define a set of **fringe types** that can be shared between TA1/TA2/TA3 but that will be ignored by the evaluation queries. For the pilot evaluation, in the absence of a program-wide set of fringe types, each pipeline of TA1/TA2/TA3 may use its own set of fringe types, which will be stored and passed around as privateData. In future evaluations, the fringe types may be incorporated into the public type assertions. When part of the public type assertions (in future evaluations), each fringe type should be a subtype of one of the Entity, Relation, or Event types in the ontology, and a fringe Relation or Event type must inherit all of the roles of its parent Relation or Event type.

4 Knowledge Elements (KE) and Knowledge Graphs

The AIDA common semantic representation of knowledge elements making up a knowledge graph is defined in the **AIDA Interchange Format (AIF)**. A **knowledge graph (KG**) represents all knowledge, whether it comes from the document stream or some shared background knowledge, or via insertion of knowledge by a human user. A **knowledge element (KE)** is a node or edge in a knowledge graph. Knowledge element types are defined in the ontology. A node in the knowledge graph represents an Entity/Filler, Event, or Relation, and an edge links an event or relation to one of its arguments.

A hypothesis is a connected knowledge graph that is N (or fewer) hops out from the facet KEs that are identified in the statement of information need for a topic. This knowledge graph will be a subset of the TA2-provided KB, though hypotheses may also contain additional KEs that are introduced via inference. Each hypothesis should have a confidence value. A set of hypotheses (each with its own confidence value) should be ranked. In addition, hypotheses must have a presentation format that allows a human user to view the hypotheses and alter the hypotheses or to suggest new ones.[3]

The formats for a TA1 knowledge graph, TA2 knowledge base, and TA3 hypothesis are defined by the AIDA Interchange Format (AIF), which is still undergoing some updates. However, AIF will represent the output of each TA1/TA2/TA3 in some form of a graph, with a node for each entity/filler, relation, and event.



[3] The presentation format will not be evaluated in the pilot evaluation.

Figure # 1: Entities/Fillers, Events, and Relations (TA1/TA2/TA3)

Each KE node or edge must contain one or more justifications. A **justification** for a KE is a set of spans that, taken together, show where the KE is asserted. A justification may come from the data stream, shared background knowledge, or human user. A justification for an entity/filler node, relation node, or event node is a single span and this will also be called a **mention** for that KE. A justification for an edge may contain multiple (up to 2) spans from different modalities as needed to establish the connection between the relation or event and its argument.

Entities/fillers that are in scope as query entry points must include ***all*** mentions in the document (for TA1) and the document stream (for TA2 and TA3) from text, image, and video. For the 2018 pilot, these entities are {Person, Organization, GeoPoliticalEntity, Facitliy, Location, Weapon, Vehicle}.

Given a document element ID, a **span** is defined to be one of the following (depending on the modality of the document element):

- Text: begin character offset and end character offset (defining a contiguous text span)
- Image: one bounding box
- Video: one bounding box in one key frame. The keyframe and bounding box will be used only to determine whether a query entry point mention matches a mention of an entity/filler in the KB. The video span is the entire shot identified by the keyframe; when assessing whether a video span provides justification for a KE, assessors will view and listen to the entire shot associated with the key frame.
- Speech: begin and end offsets, possibly in a speech segment ID *[Speech files are not included in the pilot evaluation]*
- PDF: page ID with one bounding box *[PDF files are not included in the pilot evaluation.]*

Therefore, a span is represented differently depending on the modality of the document element from which it came, as demonstrated in the following examples (these are not valid AIF spans, but are meant to illustrate the contents of the spans in AIF):

```
# Text span example:
<source> HC000T6GU </source> # document element ID
<start> 100 </start> # offset for start character, following the same convention as LTF files
<end> 104 </end> # offset for end character, following the same convention as LTF files
```

```
# Video span example:
<source> HC000021F </source> # document element ID
<keyframeid> HC000021F_101 </keyframeid>
<topleftx> 20 </topleftx> # x-coordinate of top left pixel of the bounding box
<toplefty> 20 </toplefty> # y-coordinate of top left pixel of the bounding box
<bottomrightx> 50 </bottomrightx> # x-coordinate of bottom right pixel of the bounding box
<bottomrighty> 50 </bottomrighty> # y-coordinate of bottom right pixel of the bounding box
```

```
# Image span example:
<source> IC000021F </source> # document element ID
<topleftx> 20 </topleftx> # x-coordinate of top left pixel of the bounding box
<toplefty> 20 </toplefty> # y-coordinate of top left pixel of the bounding box
```

<bottomrightx> 50 </bottomrightx> # x-coordinate of bottom right pixel of the bounding box
<bottomrighty> 50 </bottomrighty> # y-coordinate of bottom right pixel of the bounding box

For video and image spans:

- keyframeid for a video span is an extracted image (which we call a keyframe) and is distributed with the master shot boundary file. The master shot boundary file breaks up the video into multiple segments, called "shots", and the keyframe is an image within the shot that is representative (visually) of the shot. Currently there is a single keyframe per shot in the shot reference file. The keyframe ID is the same as the shot ID.
- bounding_box provides further localization of the entity (or other KE) that we are interested in inside a keyframe or image file. A bounding box is needed for further localization of the justification if, for example, the KE is a single person but the frame contains multiple people. A bounding box for an image (video keyframe or image document element) that includes the whole image is represented with <topleftx>0</topleftx>, <toplefty>0</toplefty>, <bottomrightx>image_width</bottomrightx>, <bottomrighty>image_height</bottomrighty>; for the pilot evaluation, participants will be allowed to return 0 for all four coordinates of the bounding box to indicate that they are not attempting to localize a justification beyond the level of the whole image or shot.

A shot contains a mention of a KE, such as Victor Yanukovych, if he is visually identifiable in any frame in the shot or if he is mentioned anywhere in the audio that's associated with that shot. If a video contains multiple shots that mention Victor Yanukovych (regardless of whether he is visible in the keyframe for that shot), then the system should return the keyframe for each of those shot IDs that contain a mention Victor Yanukovych (the keyframe identifies the entire shot, which is what will be assessed).

Audio is included in the pilot evaluation inasmuch as it is part of video. The video shot boundaries also segment the audio track of the video (though only imperfectly, since shot boundaries are based on visual cues rather than audio cues). Even though there will be no entry points targeting the audio part of a video, systems should still "search" the audio track for mentions of KEs. If a KE is mentioned in the audio part of a shot, systems should return the keyframe ID (representing the shot); at assessment time, LDC will watch and listen to the entire shot to determine if there is a mention of the KE either in the visual or audio track.

A justification for a KE must contain pointer(s) to text/image/video/speech/pdf span(s) (and possibly background KB) that, taken as a whole, are needed to justify the KE. The justification for a relation argument or event argument edge may include spans coming from multiple media. For example, the justification asserting that a BUK missile was the instrument of an attack on Flight MH17 might include a text span saying that a BUK launcher was in the vicinity of the crash, plus an image span showing a missile being fired at the time of the crash; LDC would assess whether the justification as a whole provided sufficient evidence for the KE asserting that the BUK missile was the cause of the crash.

A KE is allowed to contain **private data** that are accessible to TA1/TA2/TA3, but only if the private data does not contain document-level content features. Allowable private data include:

- fringe type(s) for the KE
- a vectorized representation of the KE, which cannot grow as the number of mentions/justifications for the KE increases, and from which a raw document (or significant portions thereof) cannot be recoverable.
- the number of documents that justify the KE

- time stamps of justification documents
- fringe type(s) for each image or shot, to describe features that are not represented explicitly in the seedling ontology. For example:
    - Physical.LocatedNear.Inside(Arg1_Type=Person.Soldier, Arg2_Type=Facility.Hospital)

The KE is not allowed to contain any strings from document text except for the strings in the HasName, NumericValue, and TextValue properties. In particular, KEs for filler types {SID, COM, SEN, CRM, BAL, TTL}, which do not allow HasName, NumericValue, and TextValue properties, will not be allowed to contain any strings from text; it is expected that participants will use fringe types to encode any additional information about these fillers.

5 Training Data

For the 2018 pilot evaluations, a set of 6 training topics will be provided by the LDC for training prior to the evaluation period.

1. Crash of Malaysian Air Flight MH17 (July 17, 2014)
2. Flight of Deposed Ukrainian President Viktor Yanukovych (February 2014)
3. Who Started the Shooting at Maidan? (February 2014)
4. Ukrainian War Ceasefire Violations in Battle of Debaltseve (January-February 2015)
5. Humanitarian Crisis in Eastern Ukraine (July-August 2014)
6. Donetsk and Luhansk Referendum, aka Donbas Status Referendum (May 2014)

A training corpus of 10,000 documents will be released by LDC and will include between 1200 and 1500 topic-relevant and/or scenario relevant documents; the remaining documents may or may not be relevant to the scenario.

Training and evaluation data that is specific to the SM-KBP pilot task will be listed on the SM-KBP data page (https://tac.nist.gov/2018/SM-KBP/data.html). Additional training/development data, as listed in the 2018 TAC Evaluation License Agreement, is also available by request from the LDC. To obtain any training/development/evaluation data that is distributed by the LDC, all evaluation participants must register for the TAC/TRECVID Streaming Multimedia KBP track, submit the Agreement Concerning Dissemination of TAC (or TRECVID) Results, and submit the 2018 TAC Evaluation License Agreement found on the NIST SM-KBP website.[4]

6 Evaluation Data

Data for 3 topics in the scenario will be provided as evaluation data at the start of the respective evaluation windows.

6.1 Data Format and Structure

Datasets will be released by the LDC.

[4] AIDA program performers do not need to submit the 2018 TAC Evaluation License Agreement if they are participating only in SM-KBP.

6.2 Task 1 System Input File Format

Approximately 10,000 evaluation documents will be released to Task 1 participants at the start of the Task 1a evaluation window. AIDA has several input source file formats. Details are in the documentation in the source corpus packages released by LDC.

6.2.1 Input multi-media source format

A multi-media document consists of one or more document elements, where each document element has a single modality (text, audio, video, or image). A document is a “root” in LDC’s source corpus package, and a document element is a “child”. LDC annotators see all child assets associated with a root UID at the same time, as they annotate the document. Similarly, Task 1 participants should process all document elements in the same document at the same time, and provide coreference/KE unification across the document elements in the document.

Each document (“root”) has a single primary source markup file (PSM.xml) that preserves the minimum set of tags needed to represent the structure of the relevant text as seen by the human web-page reader.

6.2.2 Input text source format

Each document (“root”) has a single text document element. The input text source data is provided in raw text format (.rsd) and a segmented/tokenized format (.LTF.xml). The LDC LTF common data format conforms to the LTF DTD referenced inside the test files. An example LTF file is given below.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE LCTL_TEXT SYSTEM "ltf.v1.5.dtd">
<LCTL_TEXT>
  <DOC id="NW_ARX_UZB_164780_20140900" tokenization="tokenization_parameters.v2.0" grammar="none"
raw_text_char_length="1781" raw_text_md5="1511bf44675b0256adc190a7b96e14bd">
    <TEXT>
      <SEG id="segment-0" start_char="0" end_char="31">
        <ORIGINAL_TEXT>Emlashni birinchi kim boshlagan?</ORIGINAL_TEXT>
        <TOKEN id="token-0-0" pos="word" morph="none" start_char="0" end_char="7">Emlashni</TOKEN>
        <TOKEN id="token-0-1" pos="word" morph="none" start_char="9" end_char="16">birinchi</TOKEN>
        <TOKEN id="token-0-2" pos="word" morph="none" start_char="18" end_char="20">kim</TOKEN>
        <TOKEN id="token-0-3" pos="word" morph="none" start_char="22" end_char="30">boshlagan</TOKEN>
        <TOKEN id="token-0-4" pos="punct" morph="none" start_char="31" end_char="31">?</TOKEN>
      </SEG>
      <SEG id="segment-1" start_char="33" end_char="61">
        <ORIGINAL_TEXT>Pereyti k: navigatsiya, poisk</ORIGINAL_TEXT>
        <TOKEN id="token-1-0" pos="word" morph="none" start_char="33" end_char="39">Pereyti</TOKEN>
        <TOKEN id="token-1-1" pos="word" morph="none" start_char="41" end_char="41">k</TOKEN>
        <TOKEN id="token-1-2" pos="punct" morph="none" start_char="42" end_char="42">:</TOKEN>
        <TOKEN id="token-1-3" pos="word" morph="none" start_char="44" end_char="54">navigatsiya</TOKEN>
        <TOKEN id="token-1-4" pos="punct" morph="none" start_char="55" end_char="55">,</TOKEN>
        <TOKEN id="token-1-5" pos="word" morph="none" start_char="57" end_char="61">poisk</TOKEN>
       </SEG>
       ...
    </TEXT>
  </DOC>
</LCTL_TEXT>
```

6.2.3 Input audio source format

LDC will provide audio files in mp3 format. LDC will provide automatic speech segmentation for audio files, which each speech segment defined by a start and end timestamp. The exact algorithm is TBD but will likely follow constraints on the minimum and maximum allowable length for a segment. **Please note that speech data will not be used for the pilot evaluation as it was not a target of collection for the seedling data.**

6.2.4 Input video source format

LDC will provide full video files in mp4 format, including the audio for the videos.

In addition, LDC will provide Task 1 participants with a master shot boundary file (.msb) that gives, for each video file:

1) a shot ID for each shot in the video (e.g, IC0019MUS_1, IC0019MUS_2, …., for the shots in video IC0019MUS)
2) the start and end times of each shot segment in the raw video
3) the start and end frames of each shot segment in the raw video

Each shot will be represented by one keyframe. An actual image file (.png) will be provided for each keyframe.[5] Because there is exactly one keyframe per shot, the keyframe ID will be the same as the shot ID. Generally, shot boundaries are detected such that the keyframe is supposed to be representative of the entire visual shot; however, because shot boundary detection is automatic, this may not always be the case. The shot boundaries also segment the audio track of the video (though only imperfectly, since shot boundaries are based on visual cues rather than audio cues).

Because the terms of usage of Twitter data do not allow NIST/LDC to distribute tweets and derived data (including keyframe files), no keyframe files will be provided for Twitter; however, the master shot boundary file will still include valid shot boundaries for video document elements from Twitter. A system returning a span from a Twitter video must provide a keyframe ID (shot ID) and a bounding box for a while frame from the video, represented with <topleft>0,0</topleft> and <bottomright>image_width,image_height</bottomright>. Note that even though NIST will not provide a keyframe file for the Twitter video, the image_width and image_height for the missing keyframe are defined in the video itself.

6.2.5 Input image source format

LDC will provide image files in .bmp, .svg, .gif, .jpg, .png format.

6.3 Twitter File List for Task 1

The terms of usage of the Twitter data require that only the URLs of the tweets can be redistributed, not the actual tweets. Tweets can be deleted at any given time. Participants are encouraged to harvest the tweets as soon as possible upon receipt of the evaluation data. As such, to distinguish between no output due to deleted tweets from no output due to a system's inability to produce the results, each team participating in Task 1 is required to submit a file list along with their system output to indicate the source data availability. For consistency, use the file list distributed with the source corpus (called '`twitter_info.tab`') and add a new field at the end of each line to indicate the file availability (`TRUE` or `FALSE`).

[5] A keyframe will be provided for **each** shot ID, not just selected shots or videos that have video entry points.

7 Task 1 and Task 2 Evaluation Specifications

7.1 Task Definition

TA1 will process one document at a time (single document processing) and produce a set of KEs for each input document from the document stream. This will be referred to as a document-level knowledge graph. A KE represents a single entity, a single event or a single relation. The KE maintains a cluster (or a node) of all mentions (from within the document for a TA1 system, and across the corpus for TA2 system) of the same entity, and a cluster (or node) of one or more mentions of each event or relation.

An entity cluster should group together entity mentions that are referring to the same real-world entity. The same is true with events and relation clusters, though the definition of equality (coreference) may be fuzzier than for entities.

- For entities/fillers and events, NIST follows LDC's determination of coreference as defined by what mentions and document-level entities/fillers or events they have linked to the same global (topic-level) entity/filler or event node_id in their mini KB (see data/*/*_mini-KB.tab in LDC2018E45); we assume a one-to-one mapping between LDC's node_id and the clusters that participants should be producing.
- For relations in the pilot evaluation, NIST will sometimes splitting LDC's relation node_id into multiple relation node_ids, one for each pair of arguments in the binary relation, because LDC sometimes conflates multiple binary relations into a single node_id. For example, in T101, LDC has a single node_id (E0094) to represent the "relation" "BUK (E0084) owned by ? (X)", where X can take on different values; in this case we create a different relation node_id for each value of X.

A document may contain multiple document elements in multiple modalities; therefore, cross-lingual and cross-modal entity and event coreference are required for all SM-KBP tasks.

Conceptually, TA1 must process each document in the order given in the document stream and must freeze all output for a document before starting to process the next document in the stream; however, because TA1 is stateless across documents (i.e., TA1 must process each document independently), in practice for the pilot evaluation, TA1 may choose to parallelize processing of documents for efficiency. NIST will evaluate output for only selected documents in the data stream, via pooling and assessment.

TA2 is given a stream of document-level knowledge graphs, containing output from each TA1 run that was submitted to Task 1, and incrementally constructs a KB from the KE stream. The document-level KB will indicate which TA1 run produced each KE, and TA2 may choose to use KEs from any single TA1 knowledge graph. (It is expected that participants will established TA1-TA2 pairings well in advance of the evaluation window in order to be able to communicate using mutually agreed upon private data.) ***However, TA2 will not have access to the raw document stream.*** In an operational setting, TA2 should be prepared to output the current state of its KB at any point in the stream (i.e., after receiving TA1 KEs from any document in the stream). However, for the pilot evaluation, TA2 may process the entire stream of document-level knowledge graphs as a batch and output a single KB at the end of the stream; only this final KB will be evaluated. In future, if TA2 needs to be evaluated at more than one timepoint, the evaluation timepoints will be designated in advance, and TA2 will process batches of document-level knowledge graphs delimited by those timepoints and output KB snapshots at those timepoints.

TA2 will submit the entire KB to NIST (to be used for subsequent analysis) at end of the Task 2 evaluation window. Additionally, NIST will send TA2 queries to TA2 teams, and each team must apply those queries to its own TA2 KB and return their responses to NIST.

7.2 Evaluation Queries

The goal of AIDA is to produce semantically coherent hypotheses graphs from a data stream. TA1 and TA2 will be evaluated on whether they provided KEs that are connected in such a way as to produce each semantically coherent hypothesis found in the data stream. SM-KBP will evaluate Tasks 1, 2, and 3 using structured queries that probe the output graph(s) for the task to search for a subgraph (or individual KE) that is salient to some reference hypothesis. A node KE represents an entity/filler, event, or relation, and an edge KE links an event or relation to one of its arguments. Systems must not only find the edges in the reference hypothesis, but also correctly coreference the endpoints (nodes) of those edges to produce the connected graph representing the reference hypothesis.

As in the TAC Cold Start track, participants should not construct knowledge graphs in response to queries; rather, they should first construct their knowledge graphs based on the input for the task, and then apply the queries to their knowledge graphs to get responses that will then be assess and evaluated.

For the pilot evaluation, NIST will give the evaluation queries to TA1/TA2/TA3 participants to apply to their output knowledge graphs. It will then be the responsibility of each participant to apply the queries to their individual (frozen) knowledge graphs and return the responses to NIST. In future evaluations, it is expected that participants will not receive any evaluation queries; rather, participants will submit their output knowledge graphs to NIST, and NIST will apply the evaluation queries to all the submissions.

For the pilot evaluation only, limited manual intervention is allowed to correct systematic bugs that are detected after producing system output or responses to queries. Participants are allowed to manually check their knowledge graph and responses to evaluation queries and correct bugs in their systems to avoid submitting knowledge graphs and responses that contain errors due to some systematic error. Such manual intervention will not be allowed in future evaluations, when participants will have to freeze their systems before processing input.

7.2.1 Query Entry Points

Each query starts with one or more entry points, where each **entry point** represents an entity or filler and includes one or more **descriptors** that identify the entry point entity/filler and allows it to be aligned with an entity/filler in a knowledge graph. An entry point descriptor may be a name string or a mention of the entity/filler in the data. The entry points ground the query to specific entities/fillers (e.g., MH17 Vehicle, Donetsk GPE) and thus constrain responses so that they involve only those entities/fillers.

Any entity/filler that allows a HasName relation (except for LAW) may be used as a query entry point entity; therefore, all mentions of {PER,ORG,GPE,FAC,LOC,WEA,VEH} must be extracted and coreferenced within a document (for TA1) and across the entire document collection (for TA2 and TA3). An entry point mention will be a single span and may come from text, image, or video. A video entry point mention will be a keyframe and bounding box and will therefore provide only a visual mention of the entity/filler. Audio mentions (from video or speech document elements) will not be used as entry points for the pilot evaluation; systems may extract entity/filler mentions from audio as needed to justify a KE but are not required to extract and provide coreference for all mentions from audio.

In order to mitigate errors due to an entry point descriptor not matching any mentions or name strings in the KB, NIST intends to issue each query multiple times, each with a single entry point descriptor per entry point entity/filler, in order to allow systems a greater chance at matching the query entry points against the KB nodes.

For both the simplified xml format and the executable SPARQL format, participants will be given a query that has a single entry point descriptor per entry point entity, and for each entry point descriptor they should find a single node in their KB that best matches that entry point descriptor. (In future evaluations, NIST may allow m>1 different nodes to be returned for an entry point descriptor; where each combination of matched nodes would generate a different response to the query.) Executable SPARQL queries will use exact match of keyframe ID in a video entry point descriptor against the keyframe ID for the video mentions of the entities/fillers in the system KB. Executable SPARQL queries will require exact string match for name string entry point descriptors, and IOU (intersection over union) of at least 0.8 for a match against an entry point descriptor that is a mention span.

For the 2018 pilot, it's reasonable for Task 1 participants to ignore descriptors if they don't come from the document that is being processed; therefore, if none of the descriptors in a query come from the document that's being processed (and none of the name string descriptors appears in the document), then the Task 1 system would not return any answers for that query. (In future – and perhaps only in the very distant future – one could imagine using a function to convert the query entry point descriptor into a vector representation that could be compared with the vector representation of any entity node in the KB, regardless of which document contains mentions of that entity.)

For the simplified xml queries, participants may optimize their matching function in whatever way they wish to determine the "best" node to return for each entry point entity (e.g., fuzzy matching of strings via edit distance, or fuzzy matching of spans using some criteria other than IOU >= 0.8). For a simplified xml query with a video entry point descriptor, participants may use whatever method they wish to "match" the keyframe for the entry point against an arbitrary frame that the system extracted from a shot from a video; for example, a participant's matching function might generate a vectorized representation of the entry point keyframe and compare it against the vectorized representations of the entities/fillers in the system KB.

Additionally, for the xml query format, participants will also get graph queries containing multiple descriptors per entry point entity, representing a compound description of the entry point entity. For these xml queries with a compound description of an entity, participants should find a single node in their KB that best matches the compound description as a whole; here, participants are allowed to choose how to optimize their matching function to determine the "best" node to return, given multiple descriptors – it could be the node that matches the greatest number of descriptors, or the matching function may also consider the confidence of linking each matched mention to that node. The node that is returned should "match" at least one descriptor for the entry point entity.

Only keyframes (which are provided with the master shot boundary file) will be used as video entry point descriptors. For the pilot evaluation, LDC will ensure that the entry point keyframe or image contains a clear visual representation of the query entity/filler and no clear visual representation of any other entity/filler of the same type – so it's clear which Person is the query entity, and the entire keyframe/image would be the "bounding box".

Because the terms of usage of Twitter data do not allow NIST/LDC to distribute tweets and derived data (including keyframe files), queries will not include any video entry points from Twitter.

Task 1 systems should extract all mentions (image ID, keyframe ID, or text span) for each distinct entity/filler that is a valid entry point. For the purposes of class-level queries (measuring entity type detection in video and image), query application will return all mentions of entities/relations/events that have been tagged with a particular type in the KB (e.g., all image IDs and keyframe IDs that serve as a mention for a Vehicle).

7.2.2 Query Types and Formats

The queries in general will test a system for its effectiveness in determining the presence of a knowledge element or knowledge graph in the document collection, where a document may contain multiple document elements, and each document element can be text, video, image, or audio. Broadly, queries may be one of three types:

1. **Class level queries:** The query will provide a type from the ontology, and the teams will be asked to return all mentions of the class *corresponding to the given type*. Mentions from text, image and video should be returned. For example, the query may ask for all mentions of an entity/filler type such as "Weapon", "Vehicle", or "Facility", or all mentions of an event type such as "Conflict.Attack", "Conflict.Demonstrate", "Movement.TransportPerson", "Contact.Meet". For the pilot evaluation, class level queries will ask for only the 7 entity/filler types that are valid for entry points (Person, Organization, Geopolitical Entity, Facility, Location, Weapon, Vehicle). [This corresponds to the TRECVID Semantic Indexing task.].

2. **Instance level queries (a.k.a. "zero-hop queries"):** The query will provide a mention of an entity or filler from the ontology, and the teams will be asked to return all mentions of **that particular entity/filler**. For e.g., the query may ask for all mentions of "Jack Bauer" referred to in document 32 at offset 100-110. [This corresponds to the TRECVID Instance Search Task.].

3. **Graph queries:** The query will be composed of a combination of ontology types and their instances and ask for a connected graph with at least one edge.

    a. Edge query: Asks for an edge of a particular type, grounded in one or more entry points. For example, "Conflict.Attack events where Conflict.Attack_Target is MH17" (one endpoint of the edge is grounded in an entry point), or "GeneralAffiliation.APORA relations where GeneralAffiliation.APORA_Affiliate is BUK 332 and GeneralAffiliation.APORA_Affiliation is Russia" (both endpoints of the edge are grounded in entry points).
    b. Generic graph query: Asks for a graph with nodes connected by specific edge labels, and one or more nodes grounded in an entry point(s). For example, "A graph containing a Conflict.Attack event that has Conflict.Attack_Target as MH17 and Conflict.Attack_Place as Donetsk; the graph must also contain a Movement.TransportArtifact event that has Movement.TransportArtifiact_Instrument as MH17 and Movement.TransportArtifict_Origin as Donetsk; the graph should include the

Movement.TransportArtifact_Destination (ungrounded node) if available." Generic graph queries can be used to probe a knowledge graph for an entire reference hypothesis, or some subgraph of the reference hypothesis.

Teams will be provided queries in two formats, which are intended to be semantically equivalent (except that the Simplified format will include additional queries containing a compound description of each entry point entity):

1. Simplified: Simplified query in an XML format that teams may apply to their KBs using any automatic technique that they choose. These simplified queries will be expressed in the domain ontology and are intended to be human-readable but will not be executable using standard tools.
2. Executable: Executable SPARQL query that teams should apply using a dockerized tool provided by NIST; subsequently, NIST would use the same tool to apply executable queries in a uniform way to all KBs from all teams.[6]

The dtds for class level, zero-hop, and graph queries and their responses are provided on the SM-KBP 2018 guidelines page (https://tac.nist.gov/2018/SM-KBP/guidelines.html).

Each graph query is based on an underlying hypothesis graph. The graph queries ask for either the full hypothesis graph or a single edge from the graph; the full graph queries have a query ID starting with "AIDA_GR_2018", while the single-edge graph queries have a query ID starting with "AIDA_EG_2018".

For the 2018 pilot evaluation, there is only one entry point entity per graph query, though each entry point entity may have more than one descriptor. Full graph query IDs have the following naming convention: AIDA_GR_2018_<graphID>_<entrypointID>_<descriptorID> . A descriptorID of 0 means that the query has a compound description containing multiple descriptors for the entity; queries with a non-zero value for descriptorID will have a single descriptor for the entry point entity.

Edge ("EG") queries enable NIST's query application tool to efficiently apply small single-edge SPARQL queries to a knowledge graph, with the aim of reconstructing most of the full graph using the results from SPARQL. NIST is providing SPARQL queries for only the smaller single-edge graph queries (and not the full graph queries).

Task 1a requires extraction of knowledge elements conditioned upon a shared background context (referred to as "*unconditioned query*"), while Task 1b requires extraction of knowledge elements conditioned upon a specific hypothesis ("what if" analysis) in addition to the shared background context (referred to as "*conditioned query*").

The "what if" hypothesis for Task 1b will be represented in AIF and will include type information for each node and role labels for each edge in the hypothesis graph. Entity/filler nodes will include their

[6] NIST is developing executable SPARQL queries that can be applied uniformly to all KBs (regardless of which team produced the KB) for the pilot evaluation, because NIST must apply evaluation queries to all teams' KBs in subsequent program evaluations, when it is expected that teams will **not** be applying queries themselves. Participants are encouraged to propose alternative matching functions to be adopted by NIST for the AIDA program as a whole in the future. If the AIDA program is unable to agree on a single representation and method of applying queries to KBs, an alternative might be for each team to provide NIST with a docker container to apply queries to their team-specific KB representation and output the responses in a NIST-defined format.

hasName, textValue, and numericalValue properties; these strings can come from any of the scenario language (Ukrainian, Russian, English). For the pilot evaluation, the "what if" hypotheses 1) will be the size of a topic-level reference hypothesis from Task 3 and 2) are intended to represent user-constructed hypotheses with no provenance for any of the nodes or edges. In future evaluations, the "what if" hypothesis 1) may be smaller than an entire topic-level hypothesis and 2) might be an automatically-generated hypothesis from Task 3 that is fed back to Task 1b, and thus may contain provenance from the data.

Executable SPARQL queries assume the following representations of TA1/TA2/TA3 knowledge graphs in AIF; teams applying executable SPARQL queries should ensure that their AIF knowledge graphs conform to these requirements, which adds constraints beyond what is generally applicable in AIF:

- An entity/filler, event, or relation KE node must be represented by an AIF cluster, even if there is only one member of the cluster. However, endpoints of KE edges (assertions having an event role or relation role as predicate) should be members of clusters rather than clusters themselves.
- A cluster is **not** hierarchical; rather, the node must be represented as a single-level cluster whose members are not clusters themselves.
- The executable SPARQL query will interpret the confidence of linking a span to a cluster, to be the product of the confidence of the span (inside justifiedBy) and the confidence of membership to the cluster.
- Each edge justification is allowed to have up to two spans and must be represented by `aida:CompoundJustification`, even if only one span is provided. If more than two spans are provided for a single CompoundJustification, then the SPARQL query will take two spans at random. `aida:CompoundJustification` must be used only for edge justifications, and not for justifications for entities, fillers, events, or relation KEs.
- Each video justification must be represented by aida:KeyFrameVideoJustification, with explicit keyframe ID and bounding box.
- Each image justification must include an explicit bounding box.

7.3 System Output Format

Systems will be required to submit the following to NIST:

1. The entire knowledge base (KB) in AIF format, and
2. A response file produced using one of the following approaches:
    a. Apply the executable SPARQL queries as-is to the system output using a docker provided by NIST. Note that this approach is not recommended for the very large TA2 KBs, as the SPARQL queries may not be optimized for speed.
    b. Modify the executable SPARQL queries before applying them to the system output using the team's own tools, and return responses in the simplified xml response format.
    c. Apply the simplified xml queries to the system output using the team's own tools, and return responses in the simplified xml response format.

TA1 and TA2 teams will be applying all queries using their own infrastructure for the pilot evaluation. Responses files submitted by the teams will be used for evaluation. NIST, at a later time, will separately apply SPARQL queries to the submitted KBs in order to verify that the file submitted in response to the executable query can be reproduced.

Please note the following constraints on a response to a graph query:

(1) Edge IDs defined in the query and those in the responses must match.
(2) A variable in the query (e.g., "?crash_event", "?target") should be bound to exactly the same team-internal node ID within each response. The same variable may be bound to different team-internal node IDs across responses, except that the variables for each entry point entity (e.g., "?target") must be bound to the same team-internal node ID across responses; i.e., the entrypoint variable is allowed to match only a single node in the system KB. In future evaluations, participants may be allowed to return up to m=3(?) different nodes for an entry point descriptor; where each combination of matched nodes would generate a different response to the query.
(3) Multiple responses should be returned even if only one of the variables is bound to a different team-internal node ID.
(4) A response may or may not contain all edges.
(5) The graph (nodes and edges) returned must be connected and include exactly one node for each variable representing an entry point entity/filler.
(6) For a given edge, bindings for both variables representing endpoints must be returned.
(7) The justification for an edge must contain three elements:
    a. subject_justification, containing exactly one span that is a mention of the subject (entity/filler, relation, or event)
    b. object_justification containing exactly one span that is a mention of the object (entity/filler, relation, or event)
    c. edge_justification containing up to two spans, connecting the subject to the object via the predicate
(8) For any single justification, the spans in that justification may come from different document elements, but all document elements must come from the same document.

7.4 Pooling, Assessment and Evaluation Metrics

The details of pooling, assessment and evaluation metrics will be made available in a future version of the evaluation plan.

7.4.1 Class-based queries

All mentions returned in response to a class-based query requesting a particular type (e.g., "Weapon") in a "core" subset of the evaluation documents, will be assessed by LDC for correctness. Evaluation will be F1 of Precision and Recall.

7.4.2 Zero-hop queries

All mentions returned in response to a zero-hop query requesting mentions of a particular entity/filler (e.g., "Vladimir Putin") in a "core" subset of the evaluation documents, will be assessed by LDC for correctness. Evaluation will be F1 of Precision and Recall.

7.4.3 Graph queries

Graphs returned in response to graph queries will be broken into assessment triples (subject justification, object justification, predicate justification) that will be assessed by LDC:

- Event argument edge: assessment triple will be (subject justification, object justification, edge justification), where subject is an event of a particular type t (e.g., Conflict.Attack) and object is an argument having a particular role r (e.g., Confict.Attack_Attacker)
- Binary relation: For a given relation node R of type r (e.g., Physical.Resident), having two edges, edge1=(Arg1, R) (e.g., Physical.Resident_Resident) and edge2=(Arg2, R) (e.g., Physical.Resident_Place), the assessment triple will be (Arg1 justification, Arg2 justification, edge1 justification + R justification + edge2 justification). Executable SPARQL queries will assume that the confidence of the edge justification for the binary relation is the product of the confidences of edge1 justification, R justification, and edge2 justification.

For event or relation argument edges, LDC will:

1) Assess predicate justification j for whether an entity/filler mention is an argument in a particular role r (e.g., Target) for some event or relation type t (e.g., Conflict.Attack); the subject justification (event or relation mention) is simply a handle for the event or relation, and LDC may decide to ignore the event or relation mention and look at only the predicate justification to ensure that it refers to some event or relation of type t.
    a. Does the predicate justification j provide sufficient evidence that some (unlocalized) entity/filler e mentioned in j has role r in an event or relation of type t? Note that e does not have any localization in j; it is sufficient for j to include some mention of an entity e having role r in an event or relation of type t.
    b. If yes, is the object justification linkable to e?
    c. If yes, then proceed to steps 2 and 3 for assigning global KB IDs to the event or relation, and the argument
2) For each assessment triple that has a correct predicate justification (1a is YES), AND for which 1b is YES, assign a global KB ID to the object justification (which has been coreferenced with the argument e).
3) For each assessment triple that has a correct predicate justification (1a is YES), AND for which 1b is YES, cluster the argument-role pairs into event or relation frames (and assign a global KB ID to each event or relation) Each event or relation frame represents a single corpus-level event or relation.

For a given predicate justification j in document d, the executable SPARQL query will select the subject (or object) mention in d that has highest confidence of being linked to the subject (or object) node. The highest confidence mention might be an image, or video shot, or text mention.

To assist in clustering, justifications for endpoints that are entities/fillers should be mentions for which LDC is able to determine the correct KB ID, such as a name (preferred) or nominal mention rather than a pronominal mention. However, because systems are not specifying whether a text mentions is a NAMed, NOMinal, or PROnominal mention, the executable SPARQL query may end up selecting a pronominal mention as the subject (or object) justification if such mentions are returned as mentions for the entity/filler node; thus, it is preferable that systems do not include pronominal mentions in the set of mentions for the entity/filler, or else include pronominal mentions with lower confidence than named or nominal mentions in the same document.

On the other hand, the endpoint justifications should also be linkable to the arguments shown in the predication justification; therefore, participants who choose to apply the simplified xml queries or a version of the SPARQL that have been customized for their KB representation, might also consider proximity to the predicate justification when deciding which mention to return for the endpoint justifications. (For the pilot evaluation, executable SPARQL queries will not consider proximity to the predicate justification to be a factor when deciding which mention to return as the subject justification or object justification.)

NIST will report several scores at three levels:

1) Relation/Event Argument Extraction: Are the predicate justifications correct? If a system returns multiple justifications for an edge, LDC will assess k=3(?) justifications having highest confidence edge justification per edge; only one of these predicate (edge) justifications needs to be correct in order for the predicate justification for the edge to be correct. Score is F1 of Precision and Recall of (correct) edges.
2) Edge Salience: Do the triples with correct predicate justifications represent relations between the entities and events of interest? An edge is salient if any of its justifications represents the relation between the entities and events of interest. Score is Recall of salient edges.
3) Graph Connectivity: Are the endpoints of salient edges correctly coreferenced?

8 Task 3 Evaluation Specifications

8.1 Task Definition

Input to TA3 is the full KB (produced by one TA2) plus a statement of information need that covers the facets of interest for a topic. A topic might be something like “the shooting down of MH17” or “the assassination of person X”. Given the TA2-produced KB and an “information need”, TA3 systems should return all hypotheses relevant to the information need, focusing on the facets requested in the statement of information need, but also returning relevant KE that are up to k hops from the entities that directly address the facets in the hypothesis.

TA3 output consists of a connected subgraph of the KB (with possible additional KE nodes derived from inference) with no KE-level disjunctions.[7] TA3 hypotheses must be in the format defined by AIF, where a hypothesis is represented by a aida:Hypothesis object with an aida:hypothesisContent property listing the node KEs (including clusterMember) and edge KEs that comprise the hypothesis graph.

[7] Per the AIDA program BAA, TA3 is also responsible for producing a human-readable form of their output hypotheses, and NIST may consider how to leverage that requirement to support the evaluation assessment needs for future evaluations.

Hypothesis confidence may be a function of the number of KEs that are “weak” (“weak” might be defined as hedged for reference hypotheses), and whether removing weak KEs results in a disconnected graph. The truth value of KEs and credibility of information sources are both out of scope for the purposes of determining hypothesis confidence.

Information needs will be conveyed via a structured representation that looks similar to a simplified graph query for TA2, but will have different semantics (i.e., given the information need, the output of the Task 3 system should be a set of hypotheses represented in AIF). For example, the information need for the topic “MH17 crash” would provide a graph depicting a Conflict.Attack event that has Conflict.Attack_Target=“MH17”, including some provenance to disambiguate names and ground the KE(s) to the source documents. (N.B. TA3 will not have access to the source documents, but will have access to document element IDs, text offsets, shot IDs, etc., in the provenance from TA2, and may use the provenance in the topic representation as entry points into the TA2 KB, if desired.) The KE for some of the arguments of Conflict.Attack would be designated as “facet” KEs for the hypotheses. A good hypothesis must contain the “facet” KEs, plus any additional information relevant to the facet KEs. The size of the hypothesis is constrained so that the distance between each entity/filler node and the facet entity/filler nodes can be at most k hops (likely k=1); an edge between an entity/filler node and a relation/event node counts as half a hop, so the hypothesis is allowed to have entities/fillers that are one hop away from one of the facet entities/fillers. The facets in the statement of information need will request information to answer questions such as “who, what, when, where, how?” regarding the topic.

The AIF hypothesis file should contain all of the information needed to represent the hypothesis graph. In particular, it should contain all KEs (entities/fillers, relations, events, relation arguments, and event arguments) that are used in the hypothesis, including provenance for each of these KEs. In order to align TA3 hypotheses with reference hypotheses, NIST requires that all AIF graphs conform to the constrained version of AIF required by executable SPARQL queries, defined at the end of Section 7.2.2. Because an entity/filler, relation, or event is represented as a cluster in the constrained version of AIF, the hypothesis must include clusterMember statements.

An example statement of information need is provided below for training topic T101:

```
<information_need id="AIDA_M09_TA3_T101">
 <frames>
  <frame id= "AIDA_M09_TA3_T101_F1">
   <edges>
       <edge id="AIDA_M09_TA3_T101_F1_1">
        <subject> ?crash_event </subject>
        <predicate> Conflict.Attack_Target </predicate>
        <object> ?crash_target</object>
       </edge>
       <edge id="AIDA_M09_TA3_T101_F1_2">
        <subject> ?crash_event </subject>
        <predicate> Conflict.Attack_Place </predicate>
        <object> ?crash_place </object>
       </edge>
       <edge id="AIDA_M09_TA3_T101_F1_3">
        <subject> ?crash_event </subject>
        <predicate> Conflict.Attack_Attacker </predicate>
        <object> ?crash_attacker </object>
       </edge>
       <edge id="AIDA_M09_TA3_T101_F1_4">
        <subject> ?crash_event </subject>
        <predicate> Conflict.Attack_Instrument </predicate>
        <object> ?crash_instrument </object>
       </edge>
   </edges>
  </frame>
  <frame id="AIDA_M09_TA3_T101_F2">
   <edges>
       <edge id="AIDA_M09_TA3_T101_F2_1">
        <subject> ?transport_event </subject>
        <predicate> Movement.TransportArtifact_Agent </predicate>
        <object> ?transport_agent </object>
       </edge>
       <edge id="AIDA_M09_TA3_T101_F2_2">
        <subject> ?transport_event </subject>
        <predicate> Movement.TransportArtifact_Artifact </predicate>
        <object> ?crash_target </object>
       </edge>
       <edge id="AIDA_M09_TA3_T101_F2_4">
        <subject> ?transport_event </subject>
        <predicate> Movement.TransportArtifact_Destination </predicate>
        <object> ?crash_place </object>
       </edge>
   </edges>
  </frame>
 </frames>
 <hops>   1  </hops>
 <entrypoints>
  <entrypoint>
   <node> ?crash_target </node>
   <enttype> Vehicle </enttype>
   <string_descriptor>
       <name_string> MH-17 </name_string>
   </string_descriptor>
   <image_descriptor>
       <doceid> IC0011TFO </doceid>
       <topleft> 20,20 </topleft>
       <bottomright> 50,50 </bottomright>
   </image_descriptor>
  </entrypoint>
  <entrypoint>
   <node> ?crash_place </node>
   <enttype> GeopoliticalEntity </enttype>
   <video_descriptor>
```

```
          <doceid> IC0019NAK </doceid>
          <keyframeid> IC0019NAK_101 </keyframeid>
          <topleft> 20,20 </topleft>
          <bottomright> 50,50 </bottomright>
    </video_descriptor>
   </entrypoint>
  </entrypoints>
</information_need>
```

Each frame will be evaluated with the disjunctive semantics (each subgraph extracted for a single frame represents a hypothesis). In the example, the first frame requests hypotheses in which MH17 crashed due to an attack, whereas the second frame is intended to represent hypotheses in which MH17 crashed due to pilot error. For each frame in the statement of information need, you should return as many different matching hypotheses as you can find.

The k-hop constraint is intended to provide the desired minimum distance on hypotheses that are returned. Systems can return bigger hypotheses, but NIST/LDC will assess justifications only up to k hops out from the facets (e.g., Conflict.Attack_Instrument) that are explicitly requested.

When aligning an entrypoint in the statement of information need against the nodes in the TA2 KB, if you will subsequently be applying executable SPARQL queries to extract KEs (from hypotheses) that will have their justifications assessed, then you should use the default matching function for executable SPARQL queries for TA1 and TA2; if you want to instead use the simplified XML queries or your own transformed version of the SPARQL queries, then you can do whatever fuzzy matching you want to align the entrypoint with the TA2 KB when generating hypotheses.

8.2 Performance measurement

Factors contributing to semantic coherence are based on Grice’s Maxims:

1) **the maxim of quantity**, where one tries to be as informative as one possibly can, and gives as much information as is needed, and no more,
2) **the maxim of quality**, where one tries to be truthful, and does not give information that is false or that is not supported by evidence, and
3) **the maxim of relation**, where one tries to be relevant, and says things that are pertinent to the discussion,
4) **The maxim of manner**, when one tries to be as clear, as brief, and as orderly as one can in what one says, and where one avoids obscurity and ambiguity.

Semantic coherence decreases if the hypothesis:

- Includes KEs that are not supported/grounded anywhere in the documents [maxim of quality]
- Includes KE’s that are irrelevant (i.e., the LDC assessor can’t figure out what role the KE has in the hypothesis; KE does not contribute to understanding the topic) [maxim of relation]
- Includes logically contradictory KEs [maxim of quality/manner]
- Is missing some KEs that are supported in the documents and that would make the hypothesis more informative. [maxim of quantity]

Due to resource limitations, the pilot evaluation will not address the maxim of relation (i.e., the evaluation will likely ignore KEs in the system-generated hypothesis that cannot be aligned with a KE in the reference hypothesis). However, NIST will explore the possibility of using LDC’s judgment of which KEs “contradict” or are “not relevant” to a hypothesis, to penalize submitted hypotheses that include these non-relevant KEs.

In order to measure adherence to the maxim of quality (i.e., whether the KE’s asserted in the hypothesis are justified/grounded in the documents), TA3 will apply the same graph queries as TA2 to each of their hypotheses graphs; this will result in edges and edge justifications that can then be assessed by LDC.

Because finding and aggregating correct support for KEs is the primary responsibility of Tasks 1 and 2 rather than Task 3, there will also be a diagnostic metric for Task 3 that does not require KEs to have correct justifications in the documents; instead, this diagnostic metric will align each submitted hypothesis with a reference hypothesis and report recall (and, to a lesser extent, precision) of KEs in the aligned pair of hypotheses.

The truth value of KEs and credibility of information sources are both out of scope for the purposes of evaluating semantic coherence. It is expected that confidence values would include information related to the uncertainty of the correctness of the extracted KE (or KEs); it is also expected that the confidence of a hypothesis will be lower if one or more KEs are only (always) hedged.

8.3 Evaluation Metrics

This section describes the scoring metrics that will be used to evaluate the list of hypotheses generated by TA3 participant systems.

Automated hypothesis evaluation:
The annotators will generate a list of reference hypotheses for each topic. In response to information need request, TA3 participants will return a similar list of hypotheses. Each hypothesis is a connected graph with no disjunctions (i.e. no multiple interpretations); a good hypothesis should contain as many of the facet KE's as possible from the statement of information need. The entry point of each such graph will be an argument for the main event or relation of the topic. For instance, in topic T101 of the seedling dataset, the facet KE's represent particular arguments in the Conflict.Attack event having MH17 as the target, or the Movement.TransportArtifact event having MH17 as the Artifact.

For the pilot evaluation, NIST will use the following procedure to generate a topic-level reference hypothesis out of two or more of LDC's query-level hypotheses (each from a different query):

1. Keep only KE's that have a global kbid. [This constraint is needed in order to make the hypothesis graph have a single connected component, since KE's with only document-level (but not global) KB IDs will result in disconnected subgraphs]
2. keep only non-negated edges (and their endpoints) that are marked as "fully relevant" in LDC's hypothesis annotations
3. keep only "facet" edges, relation/event nodes, and entity/filler nodes that are relevant to the facet(s) in the statement of information need; plus "place" and "time" arguments of events; plus edges and nodes that are one hop out from a facet entity/filler (where an edge between an entity/filler node and a relation/event node counts as a half hop)
4. MANUALLY delete KEs that produce self-contradiction (e.g., "Russia supplied rebels with BUK launcher" and "Buk launcher stolen from Ukrainian Military" cannot both be asserted in the same semantically coherent hypothesis graph); clean up other annotation errors as time permits

Because Step 4 for creating topic-level hypotheses requires applying additional manual effort to the types of hypothesis annotations that are being provided in the seedling annotations, it is expected that only a handful of these cleaned up topic-level reference hypotheses will be available for the pilot evaluation. If resources allow, NIST may generate smaller topic-level hypotheses that include only the facets (and not nodes that are one hop out from the facets), as these smaller graphs are more likely to be semantically coherent given the protocol for creating query-level hypotheses in the seedling annotation.

Following the T101 example, the scoring protocol will be as follows:

Order system generated hypotheses in descending order by hypothesis confidence.
For each system generated hypothesis, find the best matching reference hypothesis, such that would result in the highest overlap. Once the match is found, do not remove the matched hypothesis from the list of reference hypotheses; instead, repeat the matching step on the next system generated hypothesis, allowing a reference hypothesis to match more than one system generated hypothesis.

In the ideal case the matched system and reference hypotheses will be identical, however there are several possible reasons for differences that will be addressed in scoring:

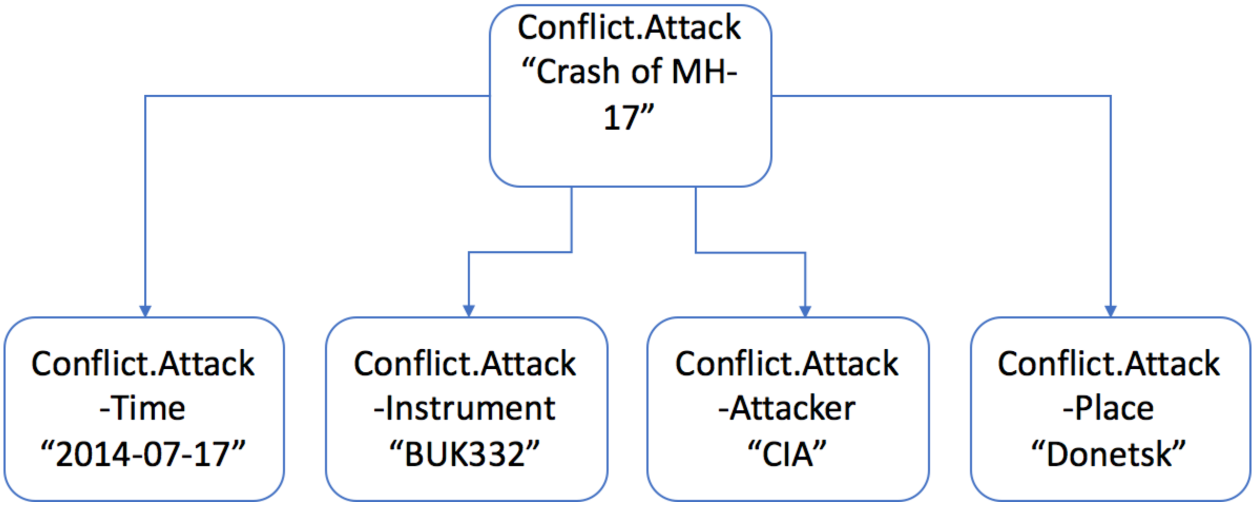
1) Miss - the TA3 graph is missing nodes and edges that are present in the reference

2) False Alarm - the TA3 graph has nodes and edges that are erroneous (aren't present in the reference)
3) The TA3 graph has nodes and edges that are salient and justified by input documents, but were missed by the annotators and therefore are absent in the reference hypothesis.

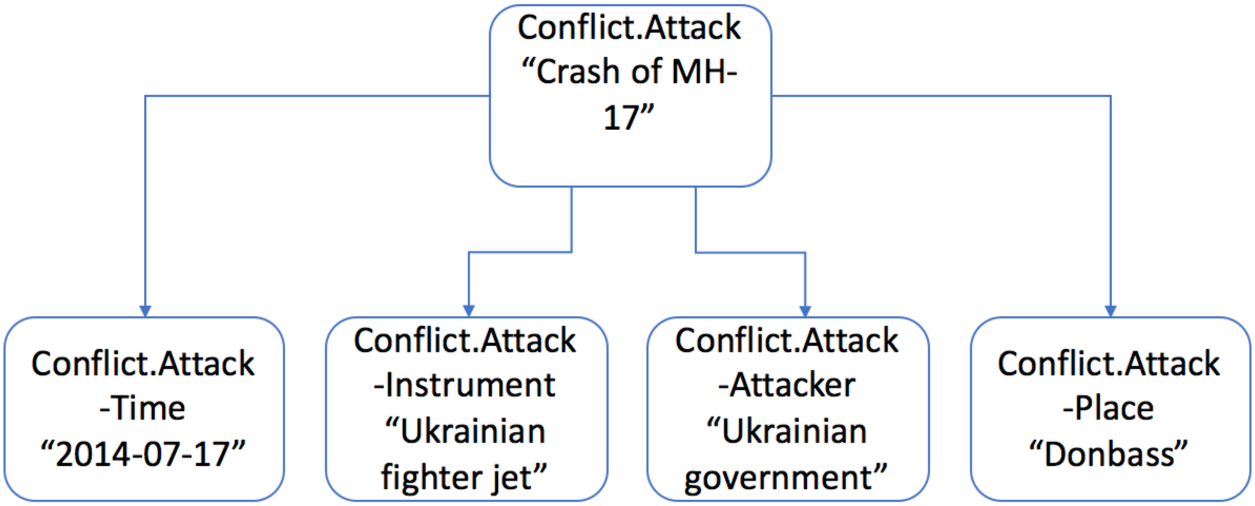
Overlap will be evaluated by reporting the miss, false alarm, true positive, true negative rate. If information that is missing in the reference hypothesis is discovered, it may be augmented and the submission will be rescored. However, due to limited resources for the pilot evaluation (including incomplete reference hypotheses), NIST will ignore KEs in the system-generated hypotheses that cannot be aligned with KEs in the reference hypothesis.

The overlap score will be measured with precision and recall of returned KE's for each hypothesis. The precision and recall will be weighted by the number of hops from the core KE. One of the weighting options is the inverse of distance squared from the core KE, i.e. KE's one hop away from the entry node will have a weight of 1, two hops away will have a weight of ½, three hops away the weight of ¼ and so on.

Example of reference subgraph is:


Conflict.Attack "Crash of MH-17"
Conflict.Attack -Time "2014-07-17"
Conflict.Attack -Instrument "BUK332"
Conflict.Attack -Attacker "CIA"
Conflict.Attack -Place "Donetsk"

And the best matching subgraph produced by the system is:



In this hypothesis, “Crashed in” KE is a false positive because the location is wrong, while “placed by” KE is a miss since it is missing.
A match of reference KE element to system KE element is not binary, but rather a range between 0 and 1 and is determined by the overlap of the list of entity mentions of the reference KE and the system KE. This overlap is determined by percent of reference mentions that the system was able to correctly extract.

In addition to the alignment metrics described above, the justifications for the KEs in the hypothesis will also be assessed for correctness, in order to evaluate the TA1-TA2-TA3 pipeline as a whole. For each reference hypothesis, LDC will assess justifications for the system generated hypotheses that match the reference hypothesis and that have highest confidence. The justifications that are assessed will be the ones that TA3 returns in response to graph queries that they have applied to each of their hypothesis graphs.

9 Submission procedure

Each team is allowed to submit up to 4 runs for each of Task 1a and Task 1b. For Task 2 (and similarly for Task 3), each team is allowed to submit up to three runs for each run that it uses as input from the previous stage of the pipeline.

In order to keep track of which modules were responsible for each of the 3 stages of the end-to-end pipeline, please name your runs using the following convention for run IDs:

Tasks 1a,b: <TA1-TeamID>_<TA1-RunID> (e.g., “BBN_1”, “BBN_2”, “BBN_3”, “BBN_4”)

Task 2: <TA1-TeamID>_<TA1-RunID>.<TA2-TeamID>_<TA2_RunID> (e.g., “BBN_1.Colorado_1”, “BBN_2.Colorado_1”, “BBN_3.Colorado_1”,...)

Task 3: <TA1-TeamID>_<TA1-RunID>.<TA2-TeamID>_<TA2-RunID>.<TA3-TeamID>_<TA3-RunID> (e.g., "BBN_1.Colorado_1.UTAustin_1")

If you're combining input from multiple runs in the previous stage of the pipeline, please separate those runs by "-". For example, a Task 2 run from Colorado that uses Task 1a runs from both JHU and Michigan might be named:
JHU_1-Michigan_1.Colorado_1

For a given run (e.g, the "BBN_1.Colorado_1" for Task 2), you will submit your system output (KBs) separately from your responses to the evaluation queries; however, both the KB(s) and the associated query responses should have the same run ID. Query responses for each run should be uploaded directly to NIST (https://tac.nist.gov/protected/2018/SubmissionForms/index.html).

AIDA performers must submit their KBs via S3 buckets that have been provided for AIDA, while other participants must upload their KBs directly to NIST (https://tac.nist.gov/protected/2018/SubmissionForms/index.html). AIDA performers who have already uploaded KBs to S3 should provide a RunID.mapping.txt file showing which KB directories/files correspond to which run ID.

For each Task 1a run:

- The KB submission should be a compressed tarball (.tgz or .zip) of a single directory named with the run ID, with one document-level KB file for each document in LDC2018E62

The responses to queries should be a compressed tarball (.tgz or .zip) of a single directory (named with the run ID), with 3 xml response files per document. Please name your response files <DocumentID>.{class_responses,zerohop_responses,graph_responses}.xml (e.g., "IC0015PZ4.class_responses.xml"). For class_responses and zerohop_responses, Task 1a participants may choose to return responses for only the "core" documents listed in LDC2018E62.coredocs.txt, available on the SM-KBP 2018 data page: https://tac.nist.gov/2018/SM-KBP/data.html

For each Task 1b run:

- The KB submission should be a compressed tarball (.tgz or .zip) of a single directory (named with the run ID) containing one subdirectory for each feedback hypothesis, named with the hypothesisID; each subdirectory contains one document-level KB file for each document in LDC2018E62
- The responses to queries should be a compressed tarball (.tgz or .zip) of a single directory (named with the run ID), containing one subdirectory for each feedback hypothesis, named with the hypothesisID; each subdirectory contains 3 xml response files per document. Please name your response files <DocumentID>.{class_responses,zerohop_responses,graph_responses}.xml (e.g., "IC0015PZ4.class_responses.xml"). For class_responses and zerohop_responses, Task 1b participants may choose to return responses for only the "core" documents listed in LDC2018E62.coredocs.txt, available on the SM-KBP 2018 data page: https://tac.nist.gov/2018/SM-KBP/data.html

For each Task 2 run:

- The KB submission should be a compressed tarball (.tgz or .zip) of a single KB file in AIF for all documents in LDC2018E62. The KB file should be named <runID>.ttl (e.g., "BBN_1.Colorado_1.ttl")

- The responses to queries should be a compressed tarball (.tgz or .zip) of a single directory (named with the run ID), with 3 xml response files. Please name your response files TA2.{class_responses,zerohop_responses,graph_responses}.xml (e.g., "TA2.class_responses.xml")

For each Task 3 run:

- The KB submission should be a compressed tarball (.tgz or .gz) of a single directory (named with the run ID), containing one subdirectory for each frame ID for each statement of information need; each of these subdirectories should be named with the frame ID and should contain one file per hypothesis satisfying the statement of information need, where the hypothesis fits into the frame given by the frame ID. Please name your file <hypothesisID>.ttl, where hypothesisID is the name of the aida:Hypothesis object in your AIF graph.
- The responses to queries should be a compressed tarball (.tgz or .gz) of a single directory (named with the run ID), containing one subdirectory for each frame ID for each statement of information need; each of these subdirectories should be named with the frame ID and should contain 3 xml response files per hypothesis. Please name your files <hypothesisID>.{class_responses,zerohop_responses,graph_responses}.xml, where hypothesisID is the name of the aida:Hypothesis object in your AIF graph.

10 Training Conditions

For each evaluation task, there are two training conditions, *constrained* and *unconstrained*, that differentiate the training/development material that are allowed.

- **Constrained** – The intent of the *constrained* training condition is to test multilingual multimodal systems using a fixed set of resources; no other materials (i.e., parallel text, speech corpora, etc.) are permitted. Teams wishing to use private (unshared) resources must do so under the unconstrained condition. The constrained training condition is optional for the SM-KBP 2018 pilot but in future evaluations will be **required for each task participated in**.
- **Unconstrained** – The intent of the *unconstrained* training condition is to measure performance gained when additional publicly available data are allowed (outside of what is described in the document "AIDA 2018 Resources for Constrained Training Condition"). Teams can mine for additional data but should not mine for post-scenario data that relates to the scenario. The unconstrained training condition is **optional but encouraged**.

The AIDA use case requires analysis of information about events, situations, and trends as they unfold. However, the evaluation will be based on a historical scenario (that has already happened), to make it possible to collect data for system training and testing. To mimic that the scenario has not happened yet, the **Time Machine Principle** requires that systems should not collect or use data about the scenario (beyond the training data specifically provided by the LDC for the scenario) as that would constitute "knowing the future". In a live situation, information about the scenario will develop over time, and systems will get to learn more about it. This is being simulated by the data stream that will be given to Task 1 participants during the Task 1 evaluation window.

For the pilot evaluation in the Russian/Ukrainian scenario, the time machine principle prohibits using any resources that are developed or derived from data from November 26, 2013 or later.

Mining for data from the internet is allowed as long as the team can categorize its data by date and can roll back to the time before the scenario[8]. As a general rule, under both the constrained and unconstrained conditions, participants are **not** allowed to use any resources which could be used to effectively replace the input corpus as a source of information about the scenario, its topics and its component KEs. For example, a rich Wikipedia article about the shooting down of Flight MH17 could provide much information about the KEs and informational conflict that is part of that topic (without the need to directly process multimedia and multi-lingual input data). Or, access to such a Wikipedia article could allow the participants to prime their processing of the input data with very rich information provided by such articles/resources.

However, many resources would not cause concerns of this type. There is a range of different types of resources available that provides various kinds of lexical or other information that may be permitted in part or in full (GeoNames, WordNet, and many others). For any resources that are not already publicly available via the internet, performers should share those with the program (so that other performers can leverage them as well). A list of resources allowed under the constrained training condition is listed in the document "AIDA 2018 Resources for Constrained Training Condition".

Teams that choose to submit system output for only the unconstrained condition will be allowed to ignore the time machine principle for the pilot evaluation, as it is only relevant for the constrained data condition. However, it is expected that some version of the time machine principle will be included in future evaluations.

11 Procedure and Requirements for Participation

11.1 Registration for the TAC/TRECVID Streaming Multimedia KBP Track

11.2 Submission Requirements

All teams (DARPA performers and open participants) are required to participate in the constrained training condition and are encouraged to participate in the unconstrained training condition. For cross-team comparison, NIST will use the best scoring submissions for each task, under each training condition.

The only time replacing an existing submission is allowed is when it is determined the submission has a bug, at which time, teams will need to contact NIST to enable resubmission. Submissions that do not pass validation will not count toward the submission limit.

At each submission, teams are recommended to provide a short description of their submissions when they upload their system output. At the conclusion of the evaluation, all teams are required to submit a more formal system description that covers their submissions for all tasks the team are participating in. Teams can download the template for the system description on the NIST AIDA website.

Please refer to the section on Submission Procedure for the requirements on how to package the system output for a given task into a submission file.

[8] If teams cannot roll back, they cannot use the data in the constrained training condition. Teams will be allowed to use it in the unconstrained condition if and only if they can demonstrate performance difference due to knowledge of the future.

11.3 Evaluation Rules Requirements

The evaluation is an open evaluation where the test data is sent to the participants who will process and submit the output to NIST. As such, the participants have agreed to process the data in accordance with the following rules:

- The participant agrees not to investigate the evaluation data. Both human/manual and automatic probing of the evaluation data is prohibited to ensure that all participating systems have the same amount of information on the evaluation data.
- The participant agrees to process at least the constrained training track for each of the selected tasks.
- The participant agrees to the rules governing the publication of the results.

11.4 Guidelines and Rules Governing Publication of Evaluation Results

This evaluation follows an open model to promote interchange with the outside world. At the conclusion of the evaluation cycle, NIST will create a report that documents the evaluation. The report will be posted on the NIST web space and will identify the participants and the scores from various metrics achieved for task.

The report that NIST creates should not be construed or represented as endorsements for any participant's system or commercial product, or as official findings on the part of NIST or the U.S. Government.

The rules governing the publication of the TAC/TRECVID evaluation results are similar to those used in other NIST evaluations.

- Participants are free to publish results for their own system, but participants must not publicly compare their results with other participants (ranking, score differences, etc.) without explicit written consent from the other participants.
- While participants may report their own results, participants may not make advertising claims about winning the evaluation or claim NIST endorsement of their system(s). Per U.S. Code of Federal Regulations (15 C.F.R. § 200.113): *NIST does not approve, recommend, or endorse any proprietary product or proprietary material. No reference shall be made to NIST, or to reports or results furnished by NIST in any advertising or sales promotion which would indicate or imply that NIST approves, recommends, or endorses any proprietary product or proprietary material, or which has as its purpose an intent to cause directly or indirectly the advertised product to be used or purchased because of NIST test reports or results.*
- All publications must contain the following NIST disclaimer:

  NIST serves to coordinate the evaluations in order to support research and to help advance the state- of-the-art. NIST evaluations are not viewed as a competition, and such results reported by NIST are not to be construed, or represented, as endorsements of any participant's system, or as official findings on the part of NIST or the U.S. Government.

12 Schedule (tentative)

Milestone	***Date***
Initial version of evaluation plan published	Mar 30, 2018
Registration deadline	July 15, 2018
Task 1a Evaluation (given data stream, some queries; output KGs)	September 10-16, 2018

Task 2 Evaluation (given KG stream, some queries; output KB)	September 17 - 30, 2018
Task 1a Apply all queries	September 17 – ~October 15, 2018
Task 1b Evaluation (given data stream, hypotheses, some queries; output KGs)	September 28 – ~October 4, 2018
Tasks 1b and 2 Apply all queries to frozen system output	October 5 - ~October 15, 2018
Task 3 Evaluation (given KB, information need; output list of hypotheses with confidences)	October 5 – ~October 15, 2018
Task 3 Apply all queries to frozen system output	~October 16-20, 2018
Partial preliminary scores released to individual teams	Mid November, 2018
Deadline for short system descriptions	October 15, 2018
Deadline for workshop presentation proposals	October 15, 2018
Notification of acceptance of presentation proposals	October 20, 2018
Deadline for system reports (workshop notebook version)	November 1, 2018
TAC (and TRECVID) 2018 Workshop at NIST	November 13-14, 2018
Deadline for system reports (final proceedings version)	February 15, 2019