TAC/TRECVID Streaming Multimedia KBP for AIDA 2019 Evaluation Plan V1.5

Last Updated: June 28, 2019 (V1.5)

Revision History

V1: March 8, 2019

- Initial release

V1.1: April 25, 2019

- Sections 3.1, 4.3: Each HasName, NumericValue, and TextValue property string is limited to 256 UTF-8 characters. HasName is allowed for {PER, ORG, GPE, FAC, LOC, WEA, VEH, LAW}. TextValue is allowed for {RES, MON, VAL}. NumericValue is not applicable in the 2019 annotation ontology.
- Sections 4, 9: For each task, participants must submit their KGs in NIST-restricted AIF (needed by NIST for evaluation); participants in Task 1a and Task 2 may submit an additional version of their KGs, in standard AIF (to communicate with other TAs within the pipeline).
- Sections 4.1, 4.3: The term "canonical" mention has been replaced with "informative" mention and its representation in AIF has changed.
- Sections 4.1, 4.3: Participants are not required to return all mentions of each entity. Instead TA1 and TA2 must return at most one informative mention per entity KE (cluster) per document. Furthermore, TA1, TA2, and TA3 must have exactly one informative mention for each aida:Entity, aida:Event, and aida:Relation that is an object of an AIF argument assertion, for each document that provides a justification for the argument assertion. For a given justification for an AIF argument assertion in a particular document, assessors will look at both the justification and the informative mention of the object in that document, to determine whether the AIF argument assertion is correct. If no informative mention is provided for the object for the document then, for the purposes of evaluation, it will be as if the justification for the argument assertion from that document did not exist in the knowledge graph.
- Sections 4.1, 4.3, 6.1.1: The `aida:sourceDocument` for a span must contain the document ID, and `aida:source` must contain the document element ID.
- Section 4.3: TA2 must link entities to the evaluation reference KB; `aida:link` with confidence must be used to assert that an `aida:Entity` and an `aida:SameAsCluster` can be coreferenced with an entity in the evaluation reference KB.
    a. `aida:link` must have one or more `aida:linkAssertion`; each `aida:linkAssertion` must have exactly one `aida:linkTarget` and exactly one `aida:confidence`
    b. TA2 zero-hop queries will look at the `aida:link` on the `aida:SameAsCluster` to determine which clusters (entity KEs) are asserted to be the same as the query entry point.
    c. TA2 graph queries will look at the `aida:link` on the `aida:Entity` to determine which AIF argument assertions have the query entry point as an argument.

- Section 4.3: Updated description of dockers to rank the lines of a SPARQL output file.
- Section 6.2: The evaluation reference KB will be released as LDC2019E43 (AIDA Phase 1 Evaluation Reference Knowledge Base) at the beginning of the Task 1a evaluation window, and Task 1 participants may also access and link to the evaluation reference KB (but are not required to).
- Section 7.3.1: Only named entities in the evaluation reference KB may be used as a query entry point entity for Task 2.
- Section 7.3.2: Updated description of zero-hop queries, pooling, and scoring
- Section 7.3.3: Updated description of Task 2 graph queries, pooling, and scoring
- Section 7.4: Updated description of Task 2 class and graph queries, pooling, and scoring.
- Section 8: Added two "oracle" evaluation conditions for TA3, resulting in 3 tasks for TA3.
- Section 8.1.1: Added description of statement of information need (SIN)
- Section 8.2: Corrected description of semantic coherence to remove reference to pairwise comparison of KEs; LDC will not provide any pairwise assessment of semantic coherence, but will remove a minimal set of edges and event/relation KEs to produce a graph that's semantically coherent.
- Section 8.2: Clarified that an event/relation cluster that has multiple types associated with it (as shown in the edge labels coming out of the cluster's members), will be split into multiple different events/relations – one for each type.
- Section 8.2.2: An event/relation KE that has no arguments with a correct justification will be counted as "Not Relevant" for the purposes of scoring relevance. If LDC assesses an event/relation KE to be "Fully relevant" based on arguments that have correct justification, but the KE includes an argument that has no correct justifications, NIST will consider the KE to be only "Partially relevant" for the purposes of scoring relevance.
- Section 8.2.3: An argument (edge) that has no correct justification will be considered "Incoherent". An event/relation KE that has no arguments with a correct justification will be considered "Incoherent".
- Section 8.2.4: After the official evaluation, LDC will perform an analysis on the additional hypotheses submitted by TA3 that weren't in any of the prevailing theories to see if the set of prevailing theories should be augmented in any way.
- Section 11: Updated schedule

V1.2: May 29, 2019

- Sections 4.3 and 8.2:
    - Section 4.3: If an argument assertion in a TA1 or TA2 AIF graph does not have any justification spans, it will be ignored by NIST for the purposes of the TA1 and TA2 evaluation, since NIST and LDC cannot evaluate the correctness of the argument assertion if it doesn't have any justification.
    - Section 8.2: A TA3 hypothesis graph is allowed to contain **some** argument assertions that don't have any justifications, if justification for the assertion would require inference and cannot be represented by pointing to any spans in the source documents. If an edge does not contain any justification spans, then it will be ignored for the purposes of evaluating correctness but will be included in the hypothesis that LDC assesses for relevance, semantic coherence, and coverage.
- Section 7.3.3: For some event/relation types (such as Government.Agreements), the role labels differ depending on the granularity of the type; because the 2019 annotation

ontology does not define the mapping of role labels between different granularities, the 2019 evaluation will not include any queries at query level 2 where the role for the coarse-grained event/relation type.subtype isn't explicitly defined for all of its finer-grained type.subtype.subtype.
- Section 7.4.2: Added description of pooling for TA1 graph queries.
- Section 9.1: For AIDA performers, each TA1 team must send output to at least two TA2 teams, and each TA2 team must accept input from two TA1 teams and send output to two TA3 teams. Each TA3 team must accept input from at least two TA2 teams.

V1.3: June 5, 2019
- Section 4: Added summary of the relationship between KEs and their representation in NIST-restricted AIF.
- Section 4.3: Updated description of LDC coreference to refer to the *kb_linking.tab table in LDC's 2019 annotations, which replaces the mini-KBs in LDC's 2018 seedling annotations.
- Section 4.3: Corrected numbering in numbered list (which was corrupted in V1.2) to match numbering in V1.1.
- Sections 7.3.2, 7.3.3, 7.4.1, 7.4.2: Updated description of input/output formats for confidence aggregation dockers to match requirements in NIST's default confidence aggregation tools.
- Section 8.1: Emphasized that the TA3 hypothesis should include relevant KE nodes and KE edges that are not explicitly included in the SIN.
- Section 4.3, 8.1: Clarified how NIST will handle cases in which the hypothesis graph submitted to NIST contains more than two justifications per edge or different values of importance for a single edge.
- Section 8.1: Relaxed requirements for what must be included in the `aida:Hypothesis` in 2019. To handle the case of a possible mismatch between the contents of the `aida:Hypothesis` and the RDF statements in the rest of the file, NIST will assume that the set of all RDF statements in the submitted .ttl file, outside of the `aida:Hypothesis`, defines the contents of the hypothesis graph; NIST will use the `aida:Hypothesis` only to define the importance of the hypothesis.
- Section 8.1.1: Cleaned up language describing semantics of temporal information.

V1.4: June 26, 2019
- Section 7.1: For the Task 1b hypothesis, each entity cluster may include an `aida:handle` and `aida:link`; additionally, the `aida:Entity` members of the entity cluster may include `aida:link` properties (but not `aida:hasName, aida:textValue, aida:numericalValue` in 2019).
- Section 7.1: For the 2019 evaluation, three hypotheses will be given as input for Task 1b.
- Section 7.3.3: For some fine-grained TA1 and TA2 graph queries, NIST will also generate a "back-off" query to evaluate the coarse-grained type.
- Section 4.3 and 8.1: For a TA3 hypothesis, all edges coming out of [members of] any event or relation cluster must be labeled with exactly the same event or relation type, subtype, and sub-subtype, and these must be the same type, subtype, and subsubtype for [all members of] the cluster itself.

- Section 8.2: For the 2019 Task 3 evaluation, H=14 hypotheses per SIN, N=25 events or relations per hypothesis, and E=7 edges per event.
- Section 8.2: Deleted last paragraph.

V1.5: June 28, 2019

- Section 4.1: Clarified that a justification for an edge may contain multiple spans from the same modality.

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 multilingual 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 automatically extracting knowledge elements (KE) from multiple languages and media sources to produce a knowledge graph (KG), 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 semantic coherence 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 multilingual multimedia sources including text, image, and video 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: 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.)

[1] https://tac.nist.gov/2019/SM-KBP/

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

- Task 1b: Document-level extraction and linking conditioned on "feedback hypotheses" providing context.
- Task 3: Generation of semantically coherent hypotheses, each representing a different interpretation of the document stream.

The SM-KBP tasks were run at TAC/TRECVID 2018 as ***pilot*** evaluations whose goals were to test evaluation protocols and metrics and to learn lessons that could inform how subsequent evaluations would be structured. It is expected that the SM-KBP track will be run for three more evaluation cycles:

- Phase 1 Evaluation 1
  - Evaluation windows May 2019
  - Workshop at TAC/TRECVID in Gaithersburg, MD in November 2019
- Phase 2 Evaluation 2 (18-month cycle)
  - Evaluation windows August-September 2020
  - Workshop at TAC/TRECVID in Gaithersburg, MD in November 2020
- Phase 3 Evaluation 3 (18-month cycle)
  - Evaluation windows April 2022
  - Workshop TBD (possibly end of June 2022)

2 Evaluation Tasks

Evaluation is over a small set of topics for a single scenario. The scenario for the SM-KBP 2019 tasks is the same as for the 2018 pilot: the Russian/Ukrainian conflict (2014-2015). Early in the evaluation cycle, all task participants will receive the **LDC annotation ontology** of entities, relations, and events; the annotation ontology defines the types of knowledge elements that LDC will annotate and that will be evaluated in the evaluation tasks.

There are three main evaluation tasks:

- Task 1 (TA1): Extraction of KEs and KE mentions from a stream of multilingual multimedia 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
- Task 2 (TA2): Construction of a knowledge graph by aggregating and linking document-level knowledge graphs produced by TA1; TA2 is given a reference knowledge base (KB) of entities at the beginning of the evaluation window and must also link an entity KE to the reference KB if such an entity already exists in the reference KB.
- Task 3 (TA3): Generation of hypotheses from a KB produced by TA2.

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 a reference KB of entities and the stream of knowledge graphs from Task 1 participants and output a corpus-level knowledge graph, and Task 3 participants are given some Statements of Information Need (SIN) and the output of Task 2 participants and output a set of knowledge graphs representing hypotheses. For SM-KBP 2019, Task 1b output is not part of the pipeline. Task 1b evaluates a media analysis system's ability to utilize knowledge in 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 SM-KBP 2019, as 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 whose format is defined by the AIDA Interchange Format (AIF), 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. NIST will apply evaluation queries to Task 1 and Task 2 KGs to probe and sample system output, with pooling and assessment of responses to those queries; these evaluation queries will be generated from KEs associated with a set of manually produced prevailing theories for the topics in the scenario and will probe system output for those KEs and types of KEs. The knowledge graphs (hypotheses) submitted for Task 3 will be evaluated in their entirety (subject to size constraints) and will be assessed for correctness of KE justifications, relevance of event and relation KEs, semantic coherence of KEs, and how well the set of hypotheses returned covers the prevailing theories.

TA1 will be evaluated on selected documents in the input document stream. For the 2019 evaluation, TA2 and TA3 will be evaluated only at one timepoint (at the end of the data stream).

Table 1: SM-KBP Tasks

Task	Input	Output
Task 1a	Stream of multilingual multimedia documents	Knowledge graph for each document, consisting of KEs found in the document
Task 1b	Stream of multilingual multimedia 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 TA1; evaluation reference KB of entities	Knowledge graph aggregating document-level knowledge graphs into a single corpus-level knowledge graph, with links to entities in the reference KB when applicable

Task 3	KGs from TA2; statement of information need (SIN) for each evaluation topic	Set of semantically coherent knowledge graphs (hypotheses) in response to the SIN

3 Ontology

All task participants will receive the **LDC annotation ontology**, which defines the entities, relations, and events that LDC will annotate and that will be evaluated. Additionally, participants will receive the **AIDA program ontology** defining additional types that can be shared as private data in a knowledge graph but that will not be directly evaluated. Together the program ontology and annotation ontology are expected to provide coverage of all the semantic categories needed for sharing information between TA1, TA2, and TA3.[2]

The annotation 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 for the topics in the scenario. KEs in the submitted knowledge graphs will be limited to the types specified in the annotation ontology. Unless noted otherwise, all evaluation queries may ask about any and all of the KEs that are defined in the annotation ontology.

The annotation ontology has three levels (type, subtype, subsubtype); annotators and participants should select the finest-grained level they can confidently label, backing off to a higher level if necessary. To specify a type at a specific level of granularity, the type label should be appended to the type label of its more coarse-grained parent, all the way up to the root:

1. top level type for the most coarse-grained level (e.g., PER)
2. type.subtype for the next level (e.g., PER.Politician)
3. type.subtype.subsubtype for the finest-grained level (e.g., PER.Politician.Governor)

The LDC annotation ontology and AIDA program ontology for SM-KBP 2019 are released as an Excel sheet (LDC_AIDAAnnotationOntologyWithMapping_V8.xlsx) that also contains a mapping between the two ontologies.

3.1 Ontological Properties

In addition to the type labels defined by the annotation ontology and AIDA program ontology, the following properties allow a name or other formulaic string to be associated with a particular entity in a knowledge graph in a way that's accessible to TA1/TA2/TA3. Each string is limited to 256 UTF-8 characters:

- **HasName**
  - Subject: An `aida:Entity` of type {PER, ORG, GPE, FAC, LOC, WEA, VEH, LAW}

[2] Therefore, team-specific "fringe" types in the 2018 pilot have been eliminated in 2019.

    - Object: a string that must be a name for an entity. Each entity is allowed to have multiple HasName properties, one for each distinct name that is observed.

- **NumericValue**
    - Subject: (none) [3]
    - Object: a numeric value

- **TextValue**
    - Subject: An `aida:Entity` of type {RES, MON, VAL}
    - Object: a string

No evaluation queries will target the HasName, NumericValue, or TextValue properties directly (e.g., no queries will ask for all names of an entity). Instead, these properties are intended to provide a way for TA1 to communicate limited string-valued properties to TA2 and TA3 to assist with coreference, etc.

Temporal properties of events and relations may also be shared between TA1/TA2/TA3, and dates will be normalized as YYYY-MM-DD according to LDC's annotation guidelines. In the SM-KBP 2019 evaluation, evaluation queries will ***not*** refer to temporal properties of events or relations; however, Task 3 Statements of Information Need may restrict events and relations to particular dates or date ranges.[4]

4 Knowledge Elements (KE) and Knowledge Graphs

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. A node in the knowledge graph represents an entity, relation, or event and will sometimes be referred to as an entity KE, relation KE, or event KE. An edge in the knowledge graph connects an event KE or relation KE to the KE representing one of the arguments of the event or relation (the argument may be an entity, relation, or event); the edge is labeled by the role that the argument has in the event or relation.

Each entity/relation/event KE in the knowledge graph has attributes associated with it, including the entity/relation/event type (e.g., "PER.Protestor", "Physical.LocatedNear.Surround", "Contact.Discussion") and the mentions (a.k.a. justifications) of the entity, relation, or event in the source corpus. Similarly, each edge in the knowledge graph has attributes associated with it, including the edge label (e.g., "Conflict.Yield.Retreat_Place"), and justifications for the edge in the source corpus. KE node types (e.g. "Contact.Discussion", "Physical.LocatedNear.Surround", "PER.Protestor") and edge labels (e.g., "Conflict.Yield.Retreat_Place") are defined in the annotation ontology.

[3] No entity types in the 2019 annotation ontology have a purely numeric value
[4] Future evaluations may include queries that directly evaluate temporal properties of events and relations.

A hypothesis is a connected knowledge graph that does not contain alternate interpretations. A hypothesis will be a subset of a TA2 knowledge graph (possibly with additional KEs that TA3 has added through inference), but TA3 will add an importance value for each hypothesis, each event or relation node in the hypothesis, and each edge in the hypothesis. The importance value encompasses the confidence in the individual KEs, the centrality of a KE to its hypothesis, and the degree to which the KE or hypothesis responds to the statement of information need. The most important hypotheses should also be a diverse set of hypotheses (rather than all being only slightly different from one another). Importance values will be used to limit the hypotheses and KEs that are assessed in the evaluation and may also help a user prioritize their attention when viewing hypotheses.

Knowledge graphs in SM-KBP must be expressed as reified RDF triples, as defined by the **AIDA Interchange Format (AIF).** Standard AIF defines the format that should be use by participants to communicate between Task 1, Task 2, and Task 3 systems. Additionally, NIST defines further restrictions on AIF, called **NIST restricted AIF**, which participants should follow when producing KGs that will be submitted to NIST for evaluation. For each task, participants must submit their KGs in NIST-restricted AIF (which NIST will query and evaluate); participants in Task 1a and Task 2 may submit an additional version of their KGs in standard AIF (to communicate with other TAs within the pipeline). It is the responsibility of participants within the same pipeline to communicate and agree among themselves about the semantics of the KG in standard AIF.

N.B.: This document uses the term "node" to refer to an entity KE, relation KE, or event KE in a knowledge graph (as described above in this section) or to a node in LDC's linking kb (i.e., an entity, relation, or event that has a kb_id entry in the *kb_linking.tab table of LDC's 2019 annotations). The term "node" in this document is ***never*** used to refer to an `aida:Entity`, `aida:Relation` or `aida:Event` in AIF or NIST-restricted AIF, nor is it used to refer to an IRI, a literal, or a blank node in a general RDF graph; this document talks about nodes (i.e., entity KEs, relations KEs, and event KEs) at a higher level of abstraction, where a node for a single entity KE (or relation KE or event KE) in a knowledge graph is representing by *multiple* RDF statements that together define the KE and its attributes, including type and justifications.

An entity KE, relation KE or event KE in a knowledge graph is represented in NIST-restricted AIF by an `aida:SameAsCluster` that can contain multiple members, where each member of the `aida:SameAsCluster` is an `aida:Entity`, `aida:Relation` or `aida:Event`, respectively.

- The set of type(s) for the entity KE (or relation KE or event KE) is the union of the set of types asserted for each `aida:Entity` (or `aida:Relation` or `aida:Event`) that is a member of the `aida:SameAsCluster` representing the KE.
- The set of mentions of the entity KE (or relation KE or event KE) is the union of the set of mentions (i.e., justifications) asserted for each `aida:Entity` (or `aida:Relation` or `aida:Event`) that is a member of the `aida:SameAsCluster` representing the KE.

This document uses the term "edge" to refer to an edge in a knowledge graph (as described above in this section). A knowledge graph contains an edge between an event (or relation) KE and an argument KE if and only if its representation in NIST-restricted AIF has an argument

assertion between some member of the `aida:SameAsCluster` representing the event (or relation) KE and some member of the `aida:SameAsCluster` representing the argument KE;

- The set of label(s) for the edge is the set of `rdf:predicate` (e.g., `ldcOnt:Conflict.Attack_Attacker`) in these AIF argument assertions.
- The set of justifications for the edge is the union of the set of justifications for these AIF argument assertions (so, the set of justifications for the edge is a set of `aida:CompoundJustification`).

4.1 Justifications

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 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, possibly from different modalities, as needed to establish the connection between the relation or event and its argument.

An ***informative mention*** (a.k.a. ***informative justification***) for an entity, relation, or event KE is defined to be a representative mention of the KE. The informative mention should be a mention that is highly likely to be coreferenced with the KE. If it is from text, the informative mention should be a name (preferred) or nominal mention rather than a pronominal mention; if it is from an image or video, the informative mention should (preferably) show a clear unobstructed image or audio mention of the KE.

An entity KE does ***not*** have to include ***all*** mentions of the entity in the document or document stream; however, if a document has any mentions of a query entry point entity, the entity KE (represented as an AIF cluster) that is linked to the query entry point must include exactly one informative mention per document, as these will be evaluated by TA1 class queries and TA2 zero-hop queries. For SM-KBP 2019, query entry point entities are {PER, ORG, GPE, FAC, LOC, WEA, VEH, LAW} and their finer-grained types. Furthermore, TA1, TA2, and TA3 must have exactly one informative mention for each `aida:Entity`, `aida:Event`, and `aida:Relation` that is an object of an AIF argument assertion, for each document that provides a justification for the argument assertion.

Given a source document element ID and document 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. In SM-KBP 2019, the keyframe and bounding box will be used only to help the assessor further localize the justification in the

shot.[5] 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 *[Speech files are not included in the 2019 evaluation]*
- PDF: page ID with one bounding box *[PDF files are not included in the 2019 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:
<sourcedocument> IC0011SZH </sourcedocument> # document ID
<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:
<sourcedocument> IC0011SZH </sourcedocument> # document ID
<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:
<sourcedocument> IC0011SZH </sourcedocument> # document ID
<source> IC000021A </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

[5] In future evaluations, when a query entry point descriptor may contain a video mention, the keyframe and bounding box will be used to determine whether a query entry point mention matches a mention of an entity/filler in the submitted knowledge graph.

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>.

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. To assert that a shot contains a mention of a KE, the system should return the keyframe for the shot (the keyframe identifies the entire shot, which is what will be assessed).

Audio is included in the 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 span(s) that, taken as a whole, are needed to justify the KE. A justification for an edge may include spans coming from multiple media, as long as the spans are in the same document. 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 (interpreted in the context of the document) provided sufficient evidence for the KE asserting that the BUK missile was the cause of the crash.

4.2 AIF

The **AIDA Interchange Format (AIF)** defines the format for a TA1, TA2, and TA3 knowledge graphs. All inter-TA communication between TA1, TA2, and TA3 must be via knowledge graphs expressed as reified RDF triples that conform to the AIF.

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:

- 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

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 entity types that do not allow HasName, NumericValue, and TextValue properties, will not be allowed to contain any strings from text.

4.3 Restricted AIF

For purposes of evaluation, NIST defines further restrictions on AIF, called **NIST restricted AIF,** which imposes particular semantics to AIF as well as additional syntactic restrictions.

A single real-world entity, event or relation is represented as a cluster in restricted AIF. A cluster for an entity has members (`aida:Entity`) that contain the mentions referring to the same real-world entity; the same is true with events and relation clusters, though the definition of coreference may be fuzzier than for entities. For entities, relations, and events, NIST follows LDC's determination of coreference as defined by what mentions they have linked to the same global kb_id in the *kb_linking.tab file of LDC's 2019 annotations (see, for example, data/R103/R103_kb_linking.tab in LDC2019E05); we assume a one-to-one mapping between LDC's kb_id and the clusters that participants should be producing.

NIST will probe TA1/TA2/TA3 output knowledge graphs using SPARQL queries that assume the following NIST restricted AIF format for knowledge graphs; all teams should ensure that their AIF knowledge graphs conform to these requirements, which adds constraints beyond what is generally applicable in AIF.[6]

1. An entity, event, or relation KE node must be represented by an AIF cluster (`aida:SameAsCluster`), even if there is only one member of the cluster.
2. Each entity/relation/event AIF cluster must have an IRI, which NIST will interpret to be a unique ID for the entity, relation, or event KE.
3. An entity/relation/event AIF cluster must **not** be hierarchical; rather, the KE node must be represented as a single-level AIF cluster whose members are not clusters themselves.
4. Each entity/relation/event type statement (i.e., a statement whose `rdf:predicate` is `rdf:type`) must have at least one justification. Each justification for an entity, event, or relation that will be examined by NIST SPARQL queries must be part of a type statement; if a mention is not a justification for a type statement, NIST will ignore that mention.
5. A KE edge must be represented by one or more AIF argument assertions, i.e., AIF assertions whose `rdf:predicate` is an event role or relation role (e.g., `ldcOnt:Conflict.Attack_Attacker`); the subject and object of an argument assertion should be an `aida:Entity`, `aida:Event`, or `aida:Relation` (i.e., members of clusters rather than clusters themselves).
6. A justification for an AIF argument assertion must have either one or 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 NIST's SPARQL query will take two spans at random. The spans in an `aida:CompoundJustification` must come from the same (parent) document, and `aida:CompoundJustification` must contain at least one span. The `aida:CompoundJustification` must be used only for justifications of argument

[6] In general, the AIF graph that is produce by TA1/TA2 is allowed to contain many things that will never be accessed by any of NIST's SPARQL queries; in the specification of restricted AIF, NIST is pointing out how information should be represented in the submitted AIF graphs so as to allow NIST's SPARQL queries to access the information that is needed and expected for the evaluation.

assertions, and not for justifications for entities, events, or relation KEs. (If an argument assertion in a TA1 or TA2 AIF graph does not have any justification spans, it will be ignored by NIST for the purposes of the TA1 and TA2 evaluation.)

7. Each video justification must be represented by `aida:KeyFrameVideoJustification`, with explicit keyframe ID and bounding box.
8. Each image justification must include an explicit bounding box.
9. Each confidence value must be between 0 and 1.
10. TA2 must link entities to the evaluation reference KB; `aida:link` with confidence must be used to assert that an `aida:Entity` or an `aida:SameAsCluster` can be coreferenced with an entity in the evaluation reference KB.
    a. `aida:link` must have one or more `aida:linkAssertion`; each `aida:linkAssertion` must have exactly one `aida:linkTarget` and exactly one `aida:confidence`
    b. TA2 zero-hop queries will look at the `aida:link` on the `aida:SameAsCluster` to determine which clusters (entity KEs) are asserted to be the same as the query entry point.
    c. TA2 graph queries will look at the `aida:link` on the `aida:Entity` to determine which AIF argument assertions have the query entry point as an argument.
11. AIF will allow each cluster and each `aida:Entity`, `aida:Event`, or `aida:Relation` to specify up to one informative mention per document.
    a. For TA1 and TA2, each entity KE (cluster) must have exactly one informative mention per document if and only if the entity appears in the document. The informative mention(s) for the entity KE will be evaluated in Task 1 class queries and Task 2 zero-hop queries.
    b. TA1, TA2, and TA3 must have exactly one informative mention for each `aida:Entity`, `aida:Event`, and `aida:Relation` that is an object of an AIF argument assertion, for each document that provides a justification for the argument assertion. For a given justification for an AIF argument assertion in a particular document, assessors will look at both the justification and the informative mention of the object in that document, to determine whether the AIF argument assertion is correct. If no informative mention is provided for the object for the document then, for the purposes of evaluation, it will be as if the justification for the argument assertion from that document did not exist in the knowledge graph.
12. The output of a NIST SPARQL query is a file where each line represents a connected subgraph in response to the query. Each line is tied to a particular set of justification spans that together constitute a possible assessment item that LDC could assess to determine if the subgraph is a correct response to the query, so there may be multiple lines for the same subgraph. For TA1 class-queries and TA2 zero-hop queries, a line will represent an entity KE (cluster) and one possible assessment item for that KE; for single-edge graph queries, the line will represent an edge KE (connecting an event or relation KE to its argument KE) and one possible assessment item for that edge KE. Pooling and scoring of responses to SPARQL queries requires a docker to compute aggregate confidence value by combining various confidence values in each line of the SPARQL output, to produce a ranking for the lines.

    a. Prior to the evaluation window, each performer may submit a docker to NIST for each task and type of query. The docker will rank the lines returned by the SPARQL query; for graph queries, the docker will also filter out some of the lines. The output of the docker will be used to determine which subgraphs and assessment items are pool, assessed, and scored.
    b. If a performer does not submit a docker that can be run efficiently and without error at NIST/NCC, NIST will apply a default docker that ranks SPARQL output lines by computing a default aggregation function that takes the product of specific confidence values in the line.
13. Each justification span must have exactly one `aida:source` (containing the document element ID) and one `aida:sourceDocument` (containing the document ID).
14. Each `aida:hasName`, `aida:textValue`, and `aida:numericValue` string is limited to 256 UTF-8 characters

Additional TA3 requirements:

1. Each hypothesis is submitted as a single, standalone file to NIST for evaluation (no more than 5MB each), containing all the KEs (and their justifications) that are included in the hypothesis; each file must contain exactly one `aida:Hypothesis` containing the KEs in the hypothesis (including `aida:clusterMembership` statements).
    a. The hypothesis file should include only KEs that are in the hypothesis.
    b. Each edge KE should have no more than 2 justifications (only two will be assessed; if more than two are provided, NIST will arbitrarily select two to assess and ignore the other justifications).
    c. Extraneous mentions should not be included for node KEs; ideally, include only informative mentions.
2. Each hypothesis graph has no disjunctions or alternative interpretations. All edges coming out of [members of] any event or relation cluster must be labeled with exactly the same event or relation type, subtype, and sub-subtype, and these must be the same type, subtype, and subsubtype for [all members of] the cluster itself.
3. Each hypothesis graph must have at least one event (cluster) or relation (cluster) with at least one edge.
4. Each hypothesis graph must have exactly one hypothesis importance value; NIST will rank the submitted hypotheses by importance value and LDC will assess the H most important hypotheses.
5. Each event or relation (cluster) in the hypothesis must have exactly one event/relation importance value; NIST will rank the events and relations in the assessed hypothesis by event/relation importance value and LDC will assess the N must important events/relations (and their arguments).
6. Each edge KE in the hypothesis graph must have exactly one edge importance value; NIST will rank the arguments in the assessed event/relation by the edge importance value and LDC will assess the E most important edges for each event. (Because edge importance is associated with an argument assertion in AIF, and because a single edge KE in a knowledge graph is allowed to be represented in AIF using multiple argument assertions, it is possible that a TA3 team might associate different importance values for

two argument assertions that are made for the same edge KE; in this case, NIST will arbitrarily pick one of the importance values to be the importance value of the edge.)

7. Each entity (cluster) in the hypothesis graph must have exactly one text description (called a "handle"), which will be displayed to the assessor to represent the entity when the hypothesis is being assessed for relevance, semantic coherence, and coverage of LDC's prevailing theories. The handle can be a generated string rather than a text span that is extracted from the corpus, and can be used to represent, for example, an entity that only appears in images or video. TA3 must coordinate with TA2 and TA1 to be able to get whatever information TA3 needs to provide NIST with a cluster-level textual "handle" for each entity for the TA3 evaluation.

5 Training Data

Training and evaluation data that is specific to the SM-KBP 2019 task will be listed on the SM-KBP data page (https://tac.nist.gov/2019/SM-KBP/data.html). Additional training/development data, as listed in the 2019 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 Streaming Multimedia KBP track, submit the Agreement Concerning Dissemination of TAC Results, and submit the 2019 TAC Evaluation License Agreement found on the NIST SM-KBP website.[7]

5.1 SM-KBP 2018 pilot data (seedling ontology and annotation format)

A set of 6 training topics (LDC2018E45) was released by the LDC for the 2018 pilot, consisting of annotations following the seedling ontology and annotation guidelines.

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 (LDC2018E63) was released by LDC for the 2018 pilot, including between 1200 and 1500 topic-relevant and/or scenario relevant documents; the remaining documents may or may not be relevant to the scenario.

The following packages contain seedling data from the SM-KBP 2018 pilot:

- LDC2018E01: AIDA Scenario 1 - Seedling Corpus V2.0
- LDC2018E45: AIDA Scenario 1 - Seedling Annotation V6.0
- LDC2018E52: AIDA Scenario 1 - Seedling Corpus Part 2 V2
- LDC2018E53: AIDA Scenario 1 - Seedling Background Corpus Non-Eval

[7] AIDA program performers do not need to submit the 2019 TAC Evaluation License Agreement if they are participating only in SM-KBP.

- LDC2018E63: AIDA Scenario 1 - Seedling Corpus Training Data Video Segmentation V2.0
- LDC2018E64: AIDA Scenario 1 - Seedling Background Corpus Non-Eval Video Segmentation V1.0

An evaluation corpus, LDC2018E62 (AIDA Month 9 Pilot Eval Corpus V1.0), was released by LDC for the 2018 pilot. All "evaluation" data from the SM-KBP 2018 pilot must be sequestered for potential repurposing for the SM-KBP 2019 evaluation. Participants are not allowed to examine or further process any data in LDC2018E62 or the list of core docs for LDC2018E62.

NIST selected a set of 79 documents from LDC2018E62 and their annotations to be unsequestered and released as LDC2018E76 (AIDA Month 9 Pilot Eval Annotation - Unsequestered V1.0). The annotations follow the seedling annotation guidelines used in the SM-KBP 2018 pilot evaluation and have not been re-annotated according to the new annotation ontology for SM-KBP 2019.

5.2 Reannotation of SM-KBP 2018 data

LDC will release three "practice" topics, consisting of re-annotation of SM-KBP 2018 training topics T103 (Who Started the Shooting at Maidan), T105 (Ukrainian War Ceasefire Violations in Battle of Debaltseve), and T107 (Donetsk and Luhansk Referendum, aka Donbas Status Referendum) according to the annotation ontology for SM-KBP 2019; reannotated topics will be renamed as R103, R105, and R107, respectively.

The following packages contain data for the practice topics:

- LDC2019E04_AIDA_Phase_1_Eval_Practice_Topic_Source_Data
- LDC2019E07_AIDA_Phase_1_Evaluation_Practice_Topic_Annotation

For any file contained in both the seedling data and the AIDA Phase 1 data, the AIDA Phase 1 version is definitive (including the video segmentation).

Along with the reannotated practice topics, LDC will release a partial reference KB consisting of the LORELEI Background KB (LDC2018E80), augmented with additional entities that are relevant to the scenario and reannotated practice topics; these additional entities will be in the same format as the rest of the LORELEI Background KB and will only have name strings associated with them; they will NOT have associated mention spans from the document collection.

6 Evaluation Data

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

6.1 Task 1 System Input File Format

Approximately 2,000 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.

6.1.1 Input multilingual multimedia source format

A multimedia document consists of one or more document elements, where each document element has a single modality (text, video, or image)[8]. 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. The same document element may be included in multiple documents (i.e., a document element ID does not uniquely identify the document that it's in). Because a mention is interpreted in the context of the document containing it, the `aida:sourceDocument` for a span must contain the document ID, and `aida:source` must contain the document element ID.

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.

6.1.2 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>
```

[8] Twitter and audio files are not included in SM-KBP 2019.

```
      ...
    </TEXT>
  </DOC>
</LCTL_TEXT>
```

6.1.3 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.[9] 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).

6.1.4 Input image source format

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

6.2 Reference KB

There will be a single reference KB for the scenario based on the LORELEI KB, augmented with additional entities specific to the scenario. This KB will be used for annotation as well as evaluation.

- The reference KB contains entities but no relations or events.
- There are four sources of entity nodes in the KB:
    - GPE and LOC entities from GeoNames (http://www.geonames.org/)
    - PER entities from the CIA World Leaders List (https://www.cia.gov/library/publications/world-leaders-1/)
    - ORG entities from Appendix B of the CIA World Factbook (https://www.cia.gov/library/publications/resources/the-worldfactbook/appendix/appendix-b.html)
    - Some number of additional entities relevant to the scenario

LDC will release three versions of the LORELEI-based reference KB:

1. An initial reference KB in the LORELEI KB format has been released as LDC2018E80 (LORELEI Background KB).

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

2. A ***partial*** reference KB will be released with practice topics prior to the evaluation windows, and will consist of the LORELEI Background KB, augmented with entities labeled for the practice topics.
3. At the beginning of the Task 1a evaluation window, a complete ***evaluation*** reference KB will be released as LDC2019E43 (AIDA Phase 1 Evaluation Reference Knowledge Base), consisting of the LORELEI Background KB, augmented with scenario-relevant entities released with the practice topics, plus additional entities that are relevant to the evaluation topics.

Task 2 participants must link entities in their KG to the ***evaluation*** reference KB when the entity is in the evaluation reference KB. Task 1 participants may also access and link to the evaluation reference KB but are not required to. Task 3 participants are allowed to access the evaluation reference KB because a Statement of Information Need may refer to entity IDs in the evaluation reference KB.

7 Task 1 and Task 2 Evaluation Specifications

7.1 Task 1 Definition

TA1 will process one document at a time and output a document-level knowledge graph for each input document from the document stream. 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, including TA1. TA1 may also perform some cross-document entity coreference by linking entities to the evaluation reference KB (this may be useful for TA2 downstream) but is not required to for the purposes of evaluation of TA1.

Task 1a requires extraction of knowledge elements conditioned upon generic background context, while Task 1b requires extraction of knowledge elements conditioned upon a specific hypothesis ("what if" analysis) in addition to the generic background context. The "what if" hypotheses will be the size of a topic-level hypothesis or a manually constructed prevailing theory from Task 3.

The "what if" hypothesis for Task 1b will be represented in restricted AIF and will include type information for each node and role labels for each edge in the hypothesis graph. Each entity cluster may include an `aida:handle` and `aida:link`; additionally, the `aida:Entity` members of the entity cluster may include `aida:hasName, aida:textValue, aida:numericalValue,` and `aida:link` properties.[10] Strings can come from any of the scenario languages (Ukrainian, Russian, English). For the 2019 evaluation, "what if" hypotheses are intended to represent user-constructed hypotheses with no provenance for any of the nodes or edges. In future evaluations, the "what if" hypothesis might be an automatically-generated hypothesis from Task 3 that is fed back to Task 1b, and thus may contain provenance from the data stream. For the 2019 evaluation, three "what if" hypotheses will be given as input for Task 1b.

[10] For 2019, the "what if" hypotheses will have `aida:handle` but will not include any `aida:hasName, aida:textValue,` or `aida:numericalValue,` because these annotations will not be available in time for the Task 1b evaluation window.

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 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.

7.2 Task 2 Definition

At the beginning of the Task 2 evaluation window, TA2 is given 1) a complete reference KB and 2) a stream of document-level knowledge graphs submitted to Task 1a, and must incrementally construct a knowledge graph 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 submission. (It is expected that participants will establish 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.*** **TA2 must link an entity in their KG to the evaluation reference KB if the entity is already in the reference KB.**

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 SM-KBP 2019 (as for the pilot evaluation), TA2 may process the entire stream of document-level knowledge graphs as a batch and output a single knowledge graph at the end of the stream; only this final knowledge graph 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 KG snapshots at those timepoints.

7.3 Task 2 Evaluation Queries

The goal of AIDA is to produce semantically coherent hypothesis graphs from a data stream. TA2 will be evaluated on whether they provided KEs needed to produce semantically coherent hypothesis generated from the data stream. SM-KBP 2019 will evaluate Task 2 using structured queries that probe the output graph(s) to search for a subgraph that is salient to a "prevailing theory" produced by LDC. 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 prevailing theories, but also correctly coreference the endpoints (nodes) of those edges to produce a larger subgraph corresponding to an event/relation and its arguments.

Task 2 queries may be one of two types: zero-hop queries and graph queries.[11] Each query will start with a descriptor for an entry point entity. Zero hop queries search for all informative mentions of the entity KE (where each document should have only one informative mention for

[11] Class queries will be used to evaluate Task 1 systems but not Task 2 systems, because it is expected that the interpretation of the type of a mention will come from Task 1 systems and would not be changed (or will be changed very little) by Task 2 systems.

the same entity KE), while graph queries search for events or relations in which that entity plays a particular role.

7.3.1 TA2 Query Entry Points

Each TA2 query starts with one point, where each **entry point** represents an entity and includes a **descriptor** that identifies the entry point entity and allows it to be aligned with an entity in a knowledge graph. In SM-KBP 2019, an entry point descriptor will be a node ID from the evaluation reference KB (LDC2019E43); in future evaluations, an entry point descriptor may be a mention of the entity in the document collection. Any named entity in the evaluation reference KB may be used as a query entry point entity. The entry point grounds the query to a specific entity (e.g., LDC2019E43:703448 "Kiev") and thus constrains responses so that they involve only those entities of interest (i.e., entities that are in LDC's prevailing theories).

7.3.2 Zero-hop queries

A TA2 **"zero-hop query"** measures how well TA2 is able to link an entity to its mentions in the document collection. A zero-hop query will provide a descriptor of an entity, and participants will be asked to return all informative mentions of **that particular entity**, where only a single informative mention can be returned for each entity KE (AIF cluster) for each document. The descriptor in SM-KBP 2019 will be a named entity ID from the evaluation reference KB. LDC will assess each informative mention for whether it correctly identifies the query entity; this indicates how well the system was able to localize a mention of the entity.

A zero-hop SPARQL query will search for any `aida:SameAsCluster` that has an `aida:link` to the query's evaluation reference KB ID, and return the cluster and its set of `aida:informativeJustification`, along with some confidence values.

Pooling and scoring require a docker that ranks all entity KEs (entity clusters, each with at most one informative mention per document). The ranking of entity KEs can be produced from the aggregate confidence for each entity KE. NIST will provide a default docker to compute aggregate confidence and rank entity KEs for zero-hop queries, but participants may optionally provide an alternative cluster ranking docker instead, to be run by NIST on the output of the zero-hop SPARQL query. For each output file of a SPARQL zero-hop query (one file per query), the docker must produce a file that contains a ranking of all the entity clusters in the SPARQL output file. The docker output file must contain two tab-delimited columns:

- Column 1: entity cluster ID
- Column 2: rank of entity cluster ID

For 2019, for each query, NIST will evaluate only the cluster with rank 1.

NIST will provide a default docker that ranks an entity cluster according to the confidence of the `aida:LinkAssertion` linking the cluster to the reference KB ID (see example zero-hop SQARQL query). If two clusters have the same `aida:LinkAssertion` confidence for the same reference KB ID, NIST will arbitrarily pick one cluster as having higher rank. Therefore, it is the responsibility of participants to define the confidence of the `aida:LinkAssertion` in such a

way that distinguishes between clusters (or else optionally provide an alternative docker to rank the entity clusters).

NIST will further rank the informative mentions for the cluster with rank 1, according to the confidence value of the mention (see example zero-hop SQARQL query). Ideally, LDC would assess all informative mentions for the highest-ranked AIF cluster (up to 2000 -- one per evaluation document). However, for each zero-hop query, NIST will instead pool and LDC will assess only the k highest confidence informative mentions for the cluster. LDC will assess the correctness of each pooled mention (a mention would be Correct if and only if it was a mention of the query entity), and also determine that R documents had some mention of the query entity.

For scoring, NIST will truncate each submission to min(k,R) highest confidence mentions and report AP for this truncated submission for this query; AP will consider an informative mention to have value of 1.0 if LDC assessed it as Correct, and 0.0 if LDC assessed it to be Wrong (even if it appeared in a document that contained some correct mention of the entity, since we are not evaluating document retrieval); if the submission contains n < min(k,R) mentions, then each of the min(k,R)-n missing items will be considered Wrong.

7.3.3 TA2 graph queries

TA2 graph queries will be composed of a combination of annotation ontology types and their instances and ask for a connected graph with at least one edge. An edge consists of a triple (cluster ID of subject, role, cluster ID of object), where the subject is an event or relation, and the object is an entity argument. Each graph query is generated from an underlying hypothesis graph (prevailing theory from LDC). A graph query asks for either a full relation or event frame from the prevailing theory (with some number of arguments), or a set of edges whose role and argument match a single edge from the hypothesis graph (but where there can be multiple events/relations with the same argument in the same role).

a. Single-edge query (for event or relation argument): Asks for an edge of a particular type, grounded in one entry point entity that is the object of the edge. For example, the query might ask for "Conflict.Attack events where Conflict.Attack_Target is the entity whose reference KB ID is 80000155", or "Physical.LocatedNear relations where Physical.LocatedNear_EntityOrFiller is the entity whose reference KB ID is 80000020". If the knowledge graph contains n event/relation clusters that are the subject of an edge that matches the query, then NIST will interpret this to mean that the knowledge graph is asserting that there are n different events/relations that have this particular entry point as an argument with this particular role.
b. Full graph query (for event or relation frame): Asks for a graph with a single event or relation and some of its arguments, connected by specific role labels; each argument will be grounded in an entry point entity. For example, "Conflict.Attack event where Conflict.Attack_Target is the entity whose reference KB ID is 80000155 and Conflict.Attack_Place is the entity whose reference KB ID is 80000020" or "Physical.LocatedNear relation where Physical.LocatedNear_EntityOrFiller is the entity whose reference KB ID is

80000020 and Physical.LocatedNear_Place is the entity whose reference KB ID is 703448". Full graph queries can be used to probe a knowledge graph for an entire event and all of its requested arguments (as found in a prevailing theory). At least one entry point is required to be in the response graph, but more credit will be given to a response graph that contains more entry points in the required roles. Full graph queries are implemented via pooling and assessment of single-edge queries.

NIST will apply executable SPARQL queries in a uniform way to all knowledge graphs from all teams. SPARQL queries will request an edge KE but not a full relation or event frame. Instead, the responses to a full graph query will be constructed by applying several single-edge SPARQL queries to the knowledge graph, and then gluing together edges (from the different SPARQL queries) that have the same event or relation cluster ID as its subject. Edge queries enable NIST to apply smaller single-edge SPARQL queries to a knowledge graph, with the aim of reconstructing most of the full event frame or relation using the results from SPARQL.

A graph response assessment item for LDC will be a triple (subject justification span, predicate justification spans, object justification span), where subject justification span is an informative mention for an `aida:Event`, or `aida:Relation` of a particular type t (e.g., Conflict.Attack), object justification span is an informative mention for an argument having a particular role r (e.g., Confict.Attack_Attacker), and predicate justification is an `aida:CompoundJustification` for an argument assertion with the given subject, object and predicate (role). (For TA2, the argument is always an entity, but for TA1 and TA3, the argument can be an entity, event, or relation). For any single assessment item, the spans may come from different document elements, but all document elements must come from the same document.

For a graph response assessment item (subject justification span, predicate justification spans, object justification span), 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.[12]
    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

[12] LDC will in fact ignore the subject justification during assessment, so each subject justification will simply have a placeholder that is always NIL.

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 object justification displayed to the assessor will be the informative mention in d for the `aida:Entity` (or, for TA1 and TA3, `aida:Event`, or `aida:Relation`) that is the object of the argument assertion. The subject justification displayed to the assessor will always be NIL for 2019, because the assessor will rely on the predicate justification and object justification to represent the event or relation that is the subject.

Each subject (event/relation) cluster that is returned in response to the SPARQL query will have some number of possible assessment items (corresponding to different justifications for an event/relation KE having the query entity in a particular role). NIST will use the ***aggregate edge justification confidence (AEJC)*** value to rank triples to determine which will be pooled and assessed by LDC. AEJC for TA2 is a joint probability of a (subject justification span, object justification span, predicate justification spans) triple tied to a particular subject event/relation cluster AND object AND informative justification for the object AND object's link to the reference KB ID.

Pooling and scoring require a docker to compute AEJC values. NIST will provide a default docker, but participants may optionally submit their own docker for TA2 graph queries. The docker must rank the subject clusters and (for each cluster) filter possible assessment items to select k=1 assessment items for the cluster. For each output file of a SPARQL graph query (one file per SPARQL query), the docker for TA2 graph queries must output a file containing the same tab-delimited columns as the SPARQL output fill, plus two additional columns (rank and aggregate edge justification confidence value) appended at the end of each line; the docker output file must filter the contents of the SPARQL output file such that for each subject cluster ID, there is at most k=1 line in the docker output file. Each line in the docker output file must have a rank that is unique across the entire output file of the docker.

There are two pooling strategies for TA2 graph queries.

1) Pooling Strategy 1 (pool by query): For each single-edge query, NIST will sort the subject cluster ID by its rank and pool, assess, and score the assessment items for only the C subject cluster IDs that have highest rank.
2) Pooling Strategy 2 (pool by relation/event frames): Find subjects (events or relations) that have the highest subject importance with respect to some query frame; the query frame is a set of queries that will ask for edges with particular arguments in particular roles, all for the same event/relation from a prevailing theory. The goal is to see if

participants are able to group edges together into a relation/event frame found in LDC's prevailing theories.

A ***query frame*** is a set of single-edge queries that have the same event or relation type; the query frame corresponds to a relation or event and its arguments, as found in a prevailing theory. To implement pooling strategy 2, we compute the ***subject importance*** of a TA2 event/relation S for a given query frame as follows: Given a query frame, select at most one edge coming out of S per query, such that the edge has highest aggregate edge justification confidence (AEJC) value. The subject importance of S is the sum of the AEJC of edges selected for S, given the query frame. We sort the events/relations by their subject importance and pool/assess up to K edges per query frame, starting from the event/relation with the highest subject importance. K is chosen to be 2* number of single-edge queries in the query frame.

Pooling strategy 2:

- Foreach submission
    - Foreach subject cluster ID S
        - Foreach query frame Q
            - Foreach query q in Q
                - Select at most one edge returned in response to q, where the edge goes out to the subject S and has maximal AEJC value, and add it to Edges[Q][S]
            - Calculate subject importance value of S for the query frame Q from Edges[Q][S]
    - Foreach query frame Q
        - Let K= 2 * sizeof(Q)
        - NumAdded=0
        - Sort subjects in submission by subject importance for query frame Q
            - For each subject S (ordered by the subject importance)
                - Sort Edges[Q][S] by the highest AEJC value for each edge.
                - Foreach edge in Edges[Q][S]
                    - If NumAdded < K
                        - Add one (subject justification, object justification, predicate justification) triple to the pool, where the triple has highest AEJC value
                    - NumAdded += 1

NIST will report scores at three levels (two for single-edge queries, and one for full graph queries requesting an entire relation or event frame).

Metrics for evaluating extraction of edges grounded in a particular argument:

1) Relation/Event Argument Extraction: For each single-edge query, how many unique (real-world – as determined by LDC's equivalence classes) edges are correct? For TA2, an assessment item for an edge will be considered correct for a given query only if the

predicate justification is correct and linkable to its object justification in the graph response assessment item AND the object justification is linkable to the query entity. For each TA2 single-edge query, NIST will pool and score edges for only C=3 event/relation KEs in the TA2 knowledge graph, and at most k=1 graph response assessment item will be assessed for each event/relation KE. If a submitted edge is correct, then it will be mappable to an edge equivalence class consisting of the triple (global KB ID of subject, role, global KB ID of object); if a submitted edge is not correct (i.e., it has no assessment items that are correct) it is counted as Wrong. After assessment, multiple subject cluster IDs may be given the same global KB ID by LDC. If a system returns multiple "edges" that have a correct assessment item but that map to the same edge equivalence class, then only one of those submitted "edges" will be counted as Correct (and the other submitted "edges" will be Ignored); the score per query is F1 of Precision and Recall, counting edge equivalence classes. The denominator of Recall will be max(C, number of different edge equivalence classes for the query, such that the object is the same as the query entity). The metric computes P/R/F1 of distinct events or relations that have the query entity as the argument in a particular role. The overall score is the mean F1 across all single-edge queries.

2) Recall of salient edges (for TA2): How many of the correct edges (across all TA2 single-edge graph queries) are salient to LDC's prevailing theories? A correct edge is salient if it has the same (global KB ID of subject, role, global KB ID of object) as an edge in LDC's prevailing theories. Score is Recall of salient edges.

Metric for evaluating graph connectivity in relation or event frames:

3) KE frame Recall (for TA2): How many edges were correctly connected for an event/relation from LDC's prevailing theory? NIST will generate a query frame from each event/relation in LDC's prevailing theories and pool responses using Pooling strategy 2. For a submitted edge to be correct, the predicate justification must be assessed as correct and linkable to its object justification in the graph response assessment item AND the object justification must be linkable to the query entity for TA2; a correct submitted edge will be mappable to an edge equivalence class consisting of the triple (global KB ID of subject, role, global KB ID of object). For each query frame, compute the Value for each subject event/relation KE that was submitted and pooled for the query frame; the Value is the maximum number of unique edges in the submitted event/relation KE that are correct AND that have the same global KB ID for the subject. KE frame Recall for the query frame is the (maximum Value of subject event/relation KEs submitted for the frame)/(number of queries in the query frame). The overall score is the mean KE frame Recall across all query frames.

A TA2 graph query requests an event or relation at a particular level of granularity from the annotation ontology, plus an argument with a particular role (specific to that entity or relation type and granularity). Because LDC annotates events and relation at two levels of granularity, there will be two levels of graph queries:

1. Query level 1: Type.Subtype.Subsubtype_Role
    a. Ex.: Government.Agreements.ViolateAgreement_OtherParticipant
2. Query level 2: Type.Subtype_Role
    a. Ex.: Conflict.Demonstrate_Demonstrator

Given a SPARQL query with event/relation type t1 (at some granularity) and role r1, an argument assertion whose `rdf:predicate` has type t2 and role r2 will match the query only if t2 is equal to or a subtype of t1, and r2 is exactly the same as r1.

For query level 1, the type and granularity in the query must match exactly with the type and granularity in the knowledge graph. For query level 2, the coarse-grained type in the query can match with the same type and subtype (and any – possibly empty -- subsubtype) in the knowledge graph, as long as the role also matches exactly. For most event/relation types, when role labels don't differ depending on the granularity of the type, this will allow a single event argument assertion to be made between an `aida:Event` or `aida:Relation` and its argument, at the finest type granularity that the knowledge graph can assert with any confidence, without separate argument assertions for the more coarse-grained type; for example, if the knowledge graph has high confidence that the event has a particular subsubtype (e.g., Conflict.Attack.Bombing), then it can have a single argument assertion for the target of the bombing (whose predicate is Conflict.Attack.Bombing_Target), and that same argument assertion could match at both query level 1 (Conflict.Attack.Bombing_Target) and query level 2 (Conflict.Attack_Target). However, for other event/relation types (such as Government.Agreements), the set of role labels differs depending on the granularity of the type; because the 2019 annotation ontology does not define a mapping between different role labels across different levels of granularity, the 2019 evaluation will not include any queries at query level 2 where the role for the coarse-grained event/relation type.subtype isn't explicitly defined for all of its finer-grained type.subtype.subtype. For example, there will be no queries asking for Government.Agreements_Participant, because the "Participant" role is not defined for Government.Agreements.ViolateAgreement; however, there may be queries asking for Government.Agreements_Place, because the "Place" role is defined for all of the subsubtypes of Government.Agreements.

For most of the TA1 and TA2 graph queries in the evaluation, the granularity of the event/relation type will be the same granularity that LDC provided in their prevailing theories. However, for some fine-grained queries, NIST will also generate a "back-off" query to evaluate the coarse-grained type, even if LDC's prevailing theories only included the finer-grained type. This is to ensure that the number of queries at the two levels of granularity is approximately balanced.

7.4 Task 1 Evaluation Queries

Task 1 evaluation queries will not be grounded in entry point entities but will instead sample the highest confidence responses from each "core" document for two types of queries: class queries and graph queries[13].

7.4.1 TA1 class queries

A TA1 class query will provide an entity type at a particular level of granularity from the annotation ontology, and participants will be asked to return exactly one informative mention of each entity in a document corresponding to the given type, *at that level of granularity or a finer-grained level*. For the SM-KBP 2019 evaluation, class level queries will ask for only the entity types that are valid for entry points (having top-level type PER, ORG, GPE, FAC, LOC, WEA, VEH, LAW). Entity KEs for both individuals and groups should be returned.

Only a "core" subset of the evaluation documents will be assessed and evaluated; these "core" documents will contain some topic-relevant documents that LDC has annotated for the topic, and some documents that may or may not be topic-relevant. Ideally LDC would assess all entity KEs in a document for TA1, where each entity KE has a single informative mention. However, given limited resources, NIST will pool and assess only the C=20 highest confidence entity KEs per query per document per submission; LDC will assess each informative mention for type correctness and cluster all the correctly typed mentions (across all submissions and teams) into R equivalence classes per query per document, where each equivalence class represents a real-world entity. When scoring, NIST will truncate the submission to the min(C,R) highest confidence entity KEs for this document and report average precision (AP) on this truncated submission. AP will consider a mention that is of the wrong type to be Wrong; for other mentions, AP will consider only one mention per equivalence class (the highest confidence mention) to be Correct, and all other lower confidence mentions in the same equivalence class will be considered Wrong; if the submission contains $n < \min(C,R)$ mentions, then each of the $\min(C,R)-n$ missing items will be considered Wrong. The metric evaluates only the C highest confidence entity KEs per document per submission. The metric for TA1 class queries will be MAP (average AP across queries and documents).

Pooling and scoring require, for a given document, a ranking of each entity KE (entity cluster, represented by a single informative mention); the ranking of entity KEs can be produced from the aggregate confidence for each entity KE. NIST will provide a default docker to compute aggregate confidence for class queries, described in the example SPARQL class query, but participants may optionally provide an alternative docker instead, to be run by NIST on the output of the SPARQL class query. By default, the aggregate confidence of a cluster will be the maximum For each output file of a SPARQL class query (one file per query per document in the evaluation source corpus), the docker must produce a file that contains a ranking of all the entity clusters in the SPARQL output file. The docker output file must contain two tab-delimited columns:

- Column 1: entity cluster ID
- Column 2: rank of entity cluster ID

[13] Zero-hop queries will be used to evaluate Task 2 systems but not Task 1 systems. Zero-hop queries evaluate cross-document linking of *topic-salient* entities to their KB ID in the evaluation reference KB. Class queries evaluate document-level entity detection of *all* entities of a particular type.

Because the annotation ontology contains 3 levels, class queries can specify the type at one of three levels:

1. Query level 1: type.subtype.subsubtype
    a. matches any type label that has the same type, subtype, and subsubtype
    b. E.g., query for "PER.Politician.Governer" matches: PER.Politician.Governer (does not match anything else.)
2. Query level 2: type.subtype
    a. Matches any type label that starts with the same type and subtype (possibly with more fine-grained subsubtype)
    b. E.g., query for "PER.Politician" matches: PER.Politician, PER.Politician.Governor, PER.Politician.HeadOfGovernment, PER.Politician.Mayor (does not match anything else)

3. Query level 3: type
    a. Matches any type label that starts with the same type (possibly with more fine-grained subtypes/ subsubtypes):
    b. E.g., query for "PER" matches: PER, PER.Politician, PER.Police,….., PER.Politician.Governor, PER.Politician.HeadOfGovernment, PER.Politician.Mayor, PER.Police.Chief of Police

It is expected that there will be up to 13 queries whose type and granularity come directly from the types and granularities found in LDC's prevailing theories. Additionally, because DARPA would like to give credit for types that are more coarse-grained than what may be specified in the prevailing theories, NIST will have up to 7 additional class queries at query level 3 (type).
It is expected that no more than 20 "core" documents will be assessed and evaluated.

7.4.2 TA1 graph queries

TA1 graph queries will be the same as TA2 single-edge graph queries, except that TA1 graph queries will not be grounded in any entry point entities. Instead, queries will ask for all events and relations of various types (and their arguments) in each document.

Pooling and scoring require a docker to compute AEJC values. NIST will provide a default docker, but participants may optionally submit their own docker for TA1 graph queries. For each output file of a SPARQL graph query (one file per SPARQL query), the docker for TA1 graph queries must output a file containing the same tab-delimited columns as the SPARQL output fill, plus two additional columns (rank and aggregate edge justification confidence value) appended at the end; the docker output file must filter the contents of the SPARQL output file such that for each unique edge (having unique combination of subject cluster ID, edge label, and object cluster ID), there is at most k=1 line in the docker output file. NIST will sort the lines for the edges by rank and pool, assess, and score only those edges that have highest rank. Each line must have a rank that is unique across the entire output file of the docker.

For each document, the C most confident edges for each query will be pooled and assessed. TA1 graph responses will be scored only at one level: Relation/Event Argument Extraction.

1) Relation/Event Argument Extraction: For each single-edge query, how many unique (real-world – as determined by LDC's equivalence classes) edges are correct. For TA1, an assessment item for an edge will be considered correct only if the predicate justification is correct and linkable to its object justification in the graph response assessment item. NIST will pool and score only C=10 submitted edges per document per TA1 single-edge graph query, and at most k=1 graph response assessment items will be assessed for each edge returned. If a submitted edge is correct, then it will be mappable to an edge equivalence class consisting of the triple (global KB ID of subject, role, global KB ID of object); if a submitted edge is not correct (i.e., it has no assessment items that are correct) it is counted as Wrong. If a system returns multiple "edges" that have a correct assessment item but that map to the same edge equivalence class, then only one of those submitted "edges" will be counted as Correct (and the other submitted "edges" will be Ignored); Score is F1 of Precision and Recall, counting edge equivalence classes. The denominator of Recall will be max(C, number of different edge equivalence classes for the query and document).

8 Task 3 Evaluation Specifications

In order to evaluate TA3 without penalizing TA3 for the mistakes made by TA1 and TA2, there will be three evaluation tasks for TA3, corresponding to different sources of input for TA3:

- Task 3a: input is the knowledge graph from an automatic TA2
- Task 3b: input is an AIF translation of LDC's annotations of the evaluation corpus
- Task 3c: input is the pooled correct responses from TA2 and LDC

TA3 performers are required to submit runs for all three tasks. Tasks 3a and 3b will be run concurrently during the same evaluation window. Task 3c is expected to be run in December, after all assessments of TA2 are complete.

8.1 Task Definition

Input to TA3 is a knowledge graph plus a statement of information need (SIN) for each evaluation topic, which requests information of interest for the topic. A topic might be something like "the shooting down of MH17" or "the assassination of person X". Given the TA2-produced knowledge graph and a SIN, TA3 systems should return the hypotheses relevant to the information need, focusing on information requested in the statement of information need, but also returning additional relevant KEs (up to a limit of *N* event and relations per hypothesis, and *E* arguments per event or relation). For example, SINs might not always explicitly request affiliations of entities, but the TA3 teams should presume that such affiliations are generally of interest.

TA3 output consists of a connected subgraph of the TA2 knowledge graph (with possible additional KEs derived from inference) with no KE-level disjunctions. TA3 hypotheses must be in the format defined by restricted AIF, with these additional TA3 requirements:

1. Each hypothesis is submitted as a single, standalone file to NIST for evaluation (no more than 5MB each), containing all the KEs (and their justifications) that are included in the hypothesis; each file must contain exactly one `aida:Hypothesis` containing the KEs in the hypothesis (including `aida:clusterMembership` statements).
    a. The hypothesis file should include only KEs that are in the hypothesis.
    b. Each edge KE should have no more than 2 justifications (only two will be assessed; if more than two are provided, NIST will arbitrarily select two to assess and ignore the other justifications).
    c. Extraneous mentions should not be included for node KEs; ideally, include only informative mentions.
2. Each hypothesis graph has no disjunctions or alternative interpretations. All edges coming out of [members of] any event or relation cluster must be labeled with exactly the same event or relation type, subtype, and sub-subtype, and these must be the same type, subtype, and subsubtype for [all members of] the cluster itself.
3. Each hypothesis graph must have at least one event (cluster) or relation (cluster) with at least one edge.
4. Each hypothesis graph must have exactly one hypothesis importance value; NIST will rank the submitted hypotheses by importance value and LDC will assess the H most important hypotheses.
5. Each event or relation (cluster) in the hypothesis must have exactly one event/relation importance value; NIST will rank the events and relations in the assessed hypothesis by event/relation importance value and LDC will assess the N must important events/relations (and their arguments).
6. Each edge KE in the hypothesis graph must have exactly one edge importance value; NIST will rank the arguments in the assessed event/relation by the edge importance value and LDC will assess the E most important edges for each event. (Because edge importance is associated with an argument assertion in AIF, and because a single edge KE in a knowledge graph is allowed to be represented in AIF using multiple argument assertions, it is possible that a TA3 team might associate different importance values for two argument assertions that are made for the same edge KE; in this case, NIST will arbitrarily pick one of the importance values to be the importance value of the edge.)
7. Each entity (cluster) in the hypothesis graph must have exactly one text description (called a "handle"), which will be displayed to the assessor to represent the entity when the hypothesis is being assessed for relevance, semantic coherence, and coverage of LDC's prevailing theories. The handle can be a generated string rather than a text span that is extracted from the corpus, and can be used to represent, for example, an entity that only appears in images or video. TA3 must coordinate with TA2 and TA1 to be able to get whatever information TA3 needs to provide NIST with a cluster-level textual "handle" for each entity for the TA3 evaluation.

The submitted .ttl file will contain RDF statements like those in a TA2 submission file, with the addition of 1) exactly one `aida:Hypothesis` and 2) one importance value per relation/event

cluster and 3) one importance value per edge. For the 2019 evaluation, NIST will assume that the set of all RDF statements in the submitted .ttl file, outside of the `aida:Hypothesis`, defines the contents of the hypothesis graph; NIST will use the `aida:Hypothesis` only to define the importance of the hypothesis. Therefore, for the 2019 evaluation, it is not crucial that the `aida:hypothesisContent` contain all RDF statements (or IRIs) that define the contents of the hypothesis graph.[14]

For the set of H hypotheses with highest importance:

1. Each hypothesis should be correct; that is, the knowledge elements have high confidence of being justified in the document collection. The real-world truth value of KEs and the credibility of information sources are both out of scope for the purposes of determining correctness of the hypothesis.
2. Each hypothesis should have events and relations that are relevant to the topic expressed in the statement of information need.
3. Each hypothesis should be semantically coherent.
4. The set of hypotheses as a whole must represent diverse hypotheses covering as many of the prevailing theories identified by LDC as possible.

Given the statement of information need for a topic, the output of the Task 3 system should be a set of hypotheses represented in AIF, where each hypothesis has an importance value associated with it. Information needs will be conveyed via a structured representation as in the SM-KBP 2018 pilot. The statement of information need will request (via variables inside an xml representation of a partially instantiated knowledge graph) information to answer questions such as "who, what, when, where, how?" regarding the topic, and a good hypothesis must contain the answers plus any additional relevant information. The SIN may also include temporal properties of relations/events to specify that the relations/events of interest hold or occur on specific dates.

8.1.1 Statement of Information Need (SIN)

An example statement of information need for training topic R103 is provided below. This statement of information need represents a question of "Who Started the Shooting at Maidan?"

```
<information_need id="AIDA_M18_TA3_R103">
  <frames>
    <frame id="AIDA_M18_TA3_R103_F1">
      <edges>

        <edge id="AIDA_M18_TA3_R103_F1_E1">
          <subject> ?FirearmattackEvent1 </subject>
          <predicate> Conflict.Attack.FirearmAttack_Place </predicate>
```

[14] In an end-to-end AIDA system, where a TA3 hypothesis might need to be fed back to TA1 or some other component of the system, it might be desirable that the `aida:Hypothesis` should contain all and only those RDF statements (or IRIs) that comprise the hypothesis graph, where IRIs are from a common repository that is shared by all components of the system. However, this is an engineering issue, which will not be enforced for the purposes of evaluating TA3 in 2019.

```
      <object> ?Location1 </object>
    </edge>

    <edge id="AIDA_M18_TA3_R103_F1_E2">
      <subject> ?FirearmattackEvent1 </subject>
      <predicate> Conflict.Attack.FirearmAttack_Attacker </predicate>
      <object> ?Attacker1 </object>
    </edge>

    <edge id="AIDA_M18_TA3_R103_F1_E3">
      <subject> ?FirearmattackEvent1 </subject>
      <predicate> Conflict.Attack.FirearmAttack_Target </predicate>
      <object> ?TargetOfAttack1 </object>
    </edge>

    <edge id="AIDA_M18_TA3_R103_F1_E4">
      <subject> ?LocatednearRelation1 </subject>
      <predicate> Physical.LocatedNear_EntityOrFiller </predicate>
      <object> ?Location1 </object>
    </edge>

    <edge id="AIDA_M18_TA3_R103_F1_E5">
      <subject> ?LocatednearRelation1 </subject>
      <predicate> Physical.LocatedNear_Place </predicate>
      <object> ?Place1 </object>
    </edge>

   </edges>
  </frame>
</frames>


<temporal_info_list>
  <temporal_info>
    <subject> ?FirearmattackEvent1 </subject>
    <start_time>
      <year>2014</year>
      <month>02</month>
      <day>20</day>
      <hour></hour>
      <minute></minute>
    </start_time>
    <end_time>
      <year>2014</year>
      <month>02</month>
      <day>20</day>
```

```
      <hour></hour>
      <minute></minute>
     </end_time>
  </temporal_info>
</temporal_info_list>

<entrypoints>

  <entrypoint>
    <node> ?Location1 </node>

    <typed_descriptors>
      <typed_descriptor>
        <enttype> FAC.Structure.Plaza </enttype>
        <text_descriptor>
          <doceid> HC00002ZU </doceid>
          <start> 1237 </start>
          <end> 1242 </end>
        </text_descriptor>
      </typed_descriptor>

      <typed_descriptor>
        <enttype> FAC.Structure.Plaza </enttype>
        <string_descriptor>
          <name_string>Maidan</name_string>
        </string_descriptor>
      </typed_descriptor>
    </typed_descriptors>

  </entrypoint>

 <entrypoint>
    <node> ?Place1 </node>

    <typed_descriptors>
      <typed_descriptor>
        <enttype> GPE.UrbanArea.City </enttype>
        <kb_descriptor>
          <kbid> LDC2019E43:703448 </kbid>
        </kb_descriptor>
      </typed_descriptor>
    </typed_descriptors>

  </entrypoint>

</entrypoints>
```

</information_need>

A SIN consists of frames, temporal information, and entry points. Each SIN will have at least one, but possibly several frames. A frame is a collection of triples consisting of “subject”, “predicate” and “object”. Each subject or object has an identifying variable name that starts with a “?”. The subject represents either an event or a relation, and the object represents an argument to the event or a relation. The predicate specifies the type of the event or the relation at a particular granularity, followed by an “_” and the role label. For example “Conflict.Attack_Target“, or “Conflict.Attack.FirearmAttack_Target”.

If several edges have the same variable name, then it means that the variable has the same value across the edges; however, if different variable names are used across edges, the variables may or may not have the same value. For example, if edges have the same subject variable, such as “?Conflict.Attack.FirearmAttack1”, then it means that these edges represent arguments that belong to the same event or relation. However, if there are edges in the SIN with different subject variables but the same type of event or relation, e.g. “?Conflict.Attack1 with an “Attacker” edge, and “?Conflict.Attack2 with a “Target” edge, then they likely represent different events, but could also represent the same event. The same logic also applies to object variables that are not grounded by an entrypoint.

Therefore, in their responses to the SIN, TA3 systems should not limit their response hypotheses to only the ones that match the first case of different event (or relation), but should also return the hypotheses that match the second case where these edges can be construed as belonging to the same event (or relation).

Some events and relations might have an entry in the temporal information list. Each temporal information specifies the event or relation to which it applies, and specifies the “start” and the “end” of a range that overlaps with the time during which the event took place or the relation was true. Each “start” / “end” component has a field for “year”, “month”, “day”, “hour”, “minute” and some or all of these fields may be blank. These start/end times mean that the duration of the event overlaps with the duration defined by the start time and the end time. Likewise, the relation was valid on or sometime after the start time and that the relation was valid on or sometime before the end time.

The object of some triples might have one or several entry point descriptors in the entry point list. Each entry point will specify the object to which it belongs, and the type of descriptor.

Please note that for those variables that are grounded by entry points, there is likely to be only a single entry point descriptor per variable. For some difficult variables, that are not grounded in the evaluation reference KB, NIST might choose to provide several descriptors. The participants should not expect the list of entry point descriptors to be very large, and should not rely on it to help them resolve their linking challenges.
Additionally, non-grounded variables in a given frame can potentially be further restricted by the type of entity or filler that can be used. In such cases, the edge will also include the object type of the variable. For example, “GPE.UrbanArea.City”.

While the format of the SIN is very similar to last year's format, there are several notable changes:

- We do not specify the number of hops anymore.
- While it is likely that each SIN will contain only a single frame, we retain the possibility of having multiple frames in the SIN in order to have the necessary flexibilities to address unforeseen intricacies of scenarios. If a SIN has multiple frames, each returned hypothesis is expected to match at least one frame.
- A new "kb_descriptor" which is a typed descriptor is added. It specifies the KBID from the evaluation reference KB as an entry point descriptor.
- New section provides temporal_info_list.

8.2 Evaluation

Before any assessment is done, NIST will prune all TA3 submitted hypothesis graphs as follows:

1. Hypotheses will be ordered by importance value in descending order and up to H most important hypotheses per SIN will be selected for assessment.
2. The event and relation KEs of each hypothesis will be sorted by their importance value, and only up to the top N most important event or relation KEs will be assessed for each hypothesis.
3. The arguments (edges) for each event or relation will be sorted by their importance value, and only the E most important edges will be assessed for each event; only the two most important edges will be assessed for each relation.

The values of H, N, and E will be selected by NIST prior to the assessment phase and are dependent on the number of returned KEs by all teams and the amount of annotator resources available. For the 2019 evaluation, H=14, N=25, and E=7.

TA3 will be assessed for:

1. Correctness: How many edges in the hypothesis graph are correct (i.e., have a correct justification in the source documents)?
2. Relevance: How many of the events and relations in the hypothesis are relevant to the topic as expressed in the statement of information need?
3. Semantic Coherence: How many edges and events/relations are compatible with each other (how many edges and events/relations need to be removed so that the remaining edges and events/relations are all compatible with each other)?
4. Coverage: How well does the set of H hypotheses returned cover the prevailing theories found by LDC?

Correctness is assessed first, and the hypotheses are then pruned so that the hypotheses that are assessed for relevance, semantic coherence, and coverage have only edges that are "correct". No pooling of hypotheses (either within a TA3 submission, or across submissions) is done for assessing relevance, semantic coherence, and coverage. Instead, each hypothesis is viewed as a unit for assessment of relevance and semantic coherence, and the set of H most important hypotheses returned for each topic will be viewed as a unit for assessment of coverage.

A TA3 hypothesis graph is allowed to contain **some** argument assertions that don't have any justifications, if justification for the assertion would require inference and cannot be represented by pointing to any spans in the source documents. If an edge does not contain any justification spans, then it will be ignored for the purposes of evaluating correctness but will be included in the hypothesis that LDC assesses for relevance, semantic coherence, and coverage.

Only a limited number of hypotheses, events/relations, and edges will be assessed, so the importance values from TA3 should be selected so as to maximize all 4 metrics.

Note that if a hypothesis graph erroneously splits the arguments of a single real-world event into multiple event nodes in the hypothesis, this would use up some of the allowance of event and relation KE to be assessed for the hypothesis, but would not otherwise be penalized directly.

8.2.1 Correctness

The justifications for *all* edges in the pruned hypothesis graphs will be assessed for correctness as described for Task 2 graph responses. For TA3, LDC will assess up to two justifications per edge. An edge is "correct" if it has at least one "correct" justification; a justification is "correct" if its predicate justification is correct and the object justification is linkable to the argument mentioned in the predicate justification.

TA3 correctness will be reported as Precision of Relation/Event Argument Extraction: For each truncated hypothesis, how many unique (real-world – as determined by LDC's equivalence classes) edges are correct; an assessment item for an edge will be considered correct only if the predicate justification is correct and linkable to its object justification in the graph response assessment item.

NIST will pool and score all submitted edges in the truncated hypothesis, and only two graph response assessment items will be assessed for each edge. If a system returns multiple graph response assessment items for an edge, the edge is considered correct if any one of its graph response assessment items is correct. If a submitted edge is correct, then it will be mappable to one (or two) edge equivalence class consisting of the triple (global KB ID of subject, role, global KB ID of object); if a submitted edge is not correct (i.e., it has no assessment items that are correct) it is counted as Wrong. If multiple graph response items for a submitted edge are assessed as correct but map to different edge equivalence classes, NIST will arbitrarily pick one of the two possible edge equivalence classes and discard the other for the purposes of scoring Correctness (however, both justifications will be associated with the edge when it is viewed by LDC during assessment of relevance, semantic coherence, and coverage, and may decrease the sematic coherence of the hypothesis). If a system returns multiple "edges" that have a correct assessment item but that map to the same edge equivalence class, then only one of those submitted "edges" will be counted as Correct (and the other submitted "edges" will be Ignored). The TA3 Correctness score is Precision, counting edge equivalence classes.

8.2.2 Relevance

An assessor will be presented with event/relation KEs including all arguments (edges) of each event/relation comprising a single hypothesis, and will judge the relevance of each event/relation within the displayed hypothesis. The assessor will judge whether a given event/relation KE is salient to the SIN and label each as "Fully relevant", "Partially relevant", or "Not relevant". Note that event/relation KEs are not marked 'partially relevant' because of inclusion of insignificant/non-salient arguments; instead, KEs which contain conflicting arguments, or which combine arguments of multiple real-world events, will be marked partially relevant if arguments of one event are relevant and arguments of the other are not. Furthermore, if LDC assesses an event/relation KE to be "Fully relevant" based on arguments that have correct justification, but the KE includes an argument that has no correct justifications, NIST will consider the KE to be only "Partially relevant" for the purposes of scoring relevance.

The assessors will attempt to make their judgement based on system-generated string ("handle") for each argument; however, if this is not informative enough, they have an option to view mentions/document provenance sorted by confidence.

The **relevance** metric is a precision-based metric:

$$Relevance\ (strict) = \frac{\#\ fully\ relevant\ KEs\ returned}{\#\ KEs\ returned}$$

$$Relevance\ (lenient) = \frac{\#\ fully/partially\ relevant\ KEs\ returned}{\#\ KEs\ returned}$$

"# KEs returned" includes both event/relation KEs that have some argument with some correct justification (and thus are assessed for relevance by LDC), and event/relation KEs that do not have any arguments with a correct justification (and thus are not assessed for relevance by LDC); an event/relation KE that does not have any arguments with a correct justification will not be assessed for relevance by LDC, but will automatically be counted by NIST as "Not relevant".

8.2.3 Semantic Coherence

The Assessor will judge:

- the semantic coherence of the event/relation arguments within the event/relation KEs comprising a hypothesis.
    - For each event/relation KE, can all of its arguments exist/coexist within a single event or relation?
        - Any arguments within an event/relation that do not "hang together" (e.g. are contradictory, an illogical/impossible combination, …) are marked incoherent
        - If an argument can't be disambiguated, is nonsensical, or is somehow itself illogical for that event or relation as a whole, it is marked incoherent

    - For each hypothesis, can all of its arguments (across all events/relations) exist/coexist within a single hypothesis?
        - Any arguments within a hypothesis that do not "hang together" are marked incoherent
- the semantic coherence of the events/relations comprising a hypothesis
    - For each hypothesis, can all of its events and relations exist/coexist within a single hypothesis?
        - Any events or relations within a hypothesis that do not "hang together" (e.g. are contradictory, an illogical/impossible combination, …) are marked incoherent
        - If an event/relation is somehow itself illogical for that hypothesis as a whole, it is marked incoherent

There are two semantic coherence metrics:

$$Argument\ Coherence = 1 - \frac{\#\ arguments\ marked\ incoherent}{\#\ arguments\ in\ hypothesis}$$

$$KE\ Coherence = 1 - \frac{\#events\ or\ relations\ marked\ incoherent}{\#\ events\ or\ relations\ in\ hypothesis}$$

An argument that has no correct justifications will automatically be marked as "incoherent", and "# arguments in hypothesis" includes arguments that have no correct justification.
"# events or relations in hypothesis" includes both event/relation KEs that have some argument with some correct justification (and thus are assessed for semantic coherence by LDC), and event/relation KEs that do not have any arguments with a correct justification (and thus are not assessed for semantic coherence by LDC); an event/relation KE that does not have any arguments with a correct justification will not be assessed for semantic coherence by LDC, but will automatically be counted by NIST as "Incoherent".

8.2.4 Coverage

Coverage of prevailing theories is assessed one topic at a time, for the entire set of hypotheses that the TA3 system returned for that topic.

For each system hypothesis and each prevailing theory for a given topic, assessors judge overlap/match between system KEs and KEs in the prevailing theory. (This could be thought of as similar to a KB linking task. Instead of linking to a KB, however, assessors link arguments in system hypotheses to arguments in LDC's prevailing theories.)

- For each argument of each event/relation in a system hypothesis, assessor compares against arguments (of the same event/relation type and argument role type) within LDC's prevailing theory. Is this system hypothesis argument equivalent/close enough to an argument of the prevailing theory?

- If a match exists, assessor links the hypothesis argument to the prevailing theory argument.
- System hypothesis arguments without a match in a prevailing theory are left unlinked.

The assessors will then judge overlap/match between system hypotheses and prevailing theories, at the hypothesis level. For each prevailing theory, if the assessor determined that a reasonable match to one of the system hypotheses exists, the assessor will indicate which prevailing theory is the best match and will label the coverage of the theory as either "fully covered" or "partially covered". Prevailing theories without a matching system hypothesis will be marked as such.

Coverage Metric:

For each system hypothesis h and prevailing theory p, we first compute the hypothesis' coverage (recall) of arguments in p:

$$ArgCoverage(p,h) = \frac{\#\ of\ arguments\ in\ p\ linked\ to\ h}{\#\ of\ arguments\ in\ p}$$

The coverage metric for H={hypotheses returned for a topic} is defined as:
Coverage(H) = $\sum_{i=1}^{n} ArgCoverage(p_i, h_i)$, such that h_i is the best matching hypothesis for the prevailing theory p_i.

After the official evaluation, LDC will perform an analysis on the additional hypotheses submitted by TA3 that weren't in any of the prevailing theories to see if the set of prevailing theories should be augmented in any way.

9 Submissions

TA1, TA2, and TA3 will submit their knowledge graphs in restricted AIF format to NIST at the end of the last date of their respective evaluation windows. NIST will apply SPARQL queries to all knowledge graphs and assume that they are in restricted AIF format.

In SM-KBP 2019, participants will not receive any evaluation queries; rather, participants will submit their output knowledge graphs to NIST, and NIST will apply evaluation queries to all the submissions. As in the TAC Cold Start track, TA1 and TA2 cannot construct knowledge graphs in response to queries; rather, they must construct their knowledge graphs based on the input for the task, and then NIST will apply the queries to their knowledge graphs to get responses that will then be assessed and evaluated.

For the SM-KBP 2019 evaluation, ***no*** manual intervention is allowed. Participants will have to freeze their systems before processing input.

Participants will be required to submit the following to NIST:

1. The entire knowledge graph(s) in restricted AIF format (to be queried and evaluated by NIST)
2. An optional docker to aggregate confidence values needed by the evaluation in the restricted AIF knowledge graph. If the participant does not submit an executable docker to NIST prior to the beginning of their evaluation window, NIST will apply default docker (implemented by NIST) to aggregate confidence values in a simple way.
3. Participants in Tasks 1a and Task 2 may optionally submit an additional version of their knowledge graph(s), in standard AIF (to be used as input to the next downstream task in the pipeline)

9.1 Submission procedure

Each team is allowed to submit up to 2 runs for each of Task 1a and Task 1b. For Task 2 and Task 3, each team is allowed to submit up to 2 runs plus an additional run for each additional combination of teams contributing input from the previous stages of the pipeline.

For AIDA performers, each TA1 team must send output to at least two TA2 teams, and each TA2 team must accept input from two TA1 teams and send output to two TA3 teams. Each TA3 team must accept input from at least two TA2 teams.

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

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.

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, and two subdirectories (one named "NIST" and one named "INTER-TA"), where each subdirectory contains one document-level KB file for each document in the evaluation source corpus; the document-level KB file must be named <document_id>.ttl. Each KB file in the "NIST" subdirectory must conform to the NIST

restrictions on AIF; if the INTER-TA subdirectory is non-empty, each of its KB files must conform with the standard definition of AIF.

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 (named "NIST"); the NIST directory must contain one subdirectory for each feedback hypothesis, named with the hypothesisID, with one document-level KB file for each document in the evaluation source corpus; the document-level KB file must be named <document_id>.ttl. Each KB file must conform to the NIST restrictions on AIF.

For each Task 2 run:

- The KB submission should be a compressed tarball (.tgz or .zip) of a single directory (named with the run ID) containing two subdirectories (one named "NIST" and one named "INTER-TA"); each subdirectory must contain one KB file in AIF for all documents in the evaluation source corpus. The KB file should be named <runID>.ttl (e.g., "BBN_1.Colorado_1.ttl"). The KB file in the "NIST" subdirectory must conform to the NIST restrictions on AIF; if the INTER-TA subdirectory is non-empty, its KB file must conform with the standard definition of AIF.

For each Task 3 run:

- The KB submission should be a compressed tarball (.tgz or .zip) of a single directory (named with the run ID). Each file in the directory should contain one hypothesis and should be named <run ID>.<SIN ID>.<SIN frame ID>.<H followed by three digits for the hypothesis number>.ttl (for example, "BBN_1.Colorado_1.UTexas_3.AIDA_M09_TA3_P103_Q002Q004Q005.F1.H001.ttl")

10 Procedure and Requirements for Participation

10.1 Registration for the TAC Streaming Multimedia KBP Track

Teams that would like to participate in any of the SM-KBP 2019 tasks must register for the SM-KBP track of TAC 2019, following instructions at:
https://tac.nist.gov/2019/registration.html

10.2 Submission Requirements

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.

10.3 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.
- There is no separate constrained vs unconstrained training condition for SM-KBP 2019. Participants may submit all runs under the unconstrained training condition, except that TA3 may not use Wikipedia or scenario-relevant resources released after November 26, 2013 (the date of the start of the scenario).
- The participant agrees to the rules governing the publication of the results.

10.4 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.

11 Schedule

Milestone	***Date***
Initial version of evaluation plan published	March 8, 2019
Registration deadline	June 15, 2019
Task 1a Evaluation (given data stream; output document-level TA1a KGs)	June 27 (7:00 AM EDT) – July 3 (11:59 PM EDT), 2019
Task 1b Evaluation (given data stream, feedback hypotheses; output document-level TA1b KGs)	July 8 (7:00 AM EDT) – July 14 (11:59 PM EDT), 2019
Task 2 Evaluation (given TA1a KG stream, reference KB; output corpus-level TA2 KB)	July 8 (7:00 AM EDT) – July 14 (11:59 PM EDT), 2019
Tasks 3a, 3b Evaluation (given corpus-level KB, statement of information need; output list of hypotheses with importance values)	August 5 (7:00 AM EDT) – August 11 (11:59 PM EDT), 2019
Deadline for short system descriptions	September 15, 2019
Deadline for workshop presentation proposals	September 15, 2019
Notification of acceptance of presentation proposals	Early October, 2019
Scores released to individual teams	Early October, 2019
Deadline for system reports (workshop notebook version)	November 1, 2019
TAC (and TRECVID) 2019 Workshop at NIST	November 12-13, 2019
Task 3c Evaluation (given corpus-level KB, statement of information need; output list of hypotheses with importance values)	December 2019 (TBA)
Deadline for system reports (final proceedings version)	February 15, 2020