Event Argument Extraction: Archived Version, Sept 4, 2014

Contents

Goal

The Event Argument Extraction task at NIST TAC KBP 2014 aims to extract information about entities (and times) and the role they play in an event. The extracted information should be suitable as input to a knowledge base. As such, participating systems will extract tuples that include *(EventType, Role, Argument)*. EventType and Role will be drawn from an externally specified ontology. Arguments will be strings from within a document representing the canonical (most-specific) name or description of the entity.

While this task does not require the reification of events (or linking the different arguments of an event), systems developed for this task will support KB-queries that link entities to participation in an event-- for example *"List ORGANIZATIONs with a PURCHASER role").* When combined with other KBP technologies (i.e. slot-filling and entity-linking), more complex, multi-hop queries are possible-- for example, *"List ORGANIZATIONs with MEMBERs who have served as an ATTACKER"*.

This is a new task in 2014 and will be evaluated in English only. In 2015, we expect to extend to additional TAC languages and add an evaluation of a system's ability to reify events and connect their arguments.

Task

The world's most powerful fine art auction houses, Sotheby's and Christie's, have agreed to pay 40 million dollars to settle an international price-fixing scam, Sotheby's said. The payment, if approved by the courts, would settle a slew of suits by clients over auctions held between 1993 and 2000....
In December 2001, a Manhattan jury convicted Sotheby's main shareholder and former chairman Alfred Taubman of entering into a price-fixing agreement with Christie's. Four months later, the billionaire was sentenced to one year in prison and a 7.5-million-dollar fine over the price-fixing, which prosecutors said had cost clients of both firms 400 million dollars.

EventType	Role	ArgumentString	Realis
Transfer-Money	Giver	The world 's most powerful fine art auction houses	OTHER
Transfer-Money	Giver	Sotheby's	OTHER
Transfer-Money	Giver	Christie's	OTHER
Transfer-Money	Money	40 million dollars	OTHER
Sue	Defendant	Sotheby's	ACTUAL
Sue	Plaintiff	clients	ACTUAL
Convict	Defendant	Alfred Taubman	ACTUAL
Convict	Place	Manhattan	ACTUAL
Convict	Adjudicator	a Manhattan jury	ACTUAL
Convict	Crime	entering into a price-fixing agreement	ACTUAL
Convict	Time	2001-12	ACTUAL
Sentence	Defendant	Alfred Taubman	ACTUAL
Sentence	Sentence	one year in prison	ACTUAL
Sentence	Sentence	7.5-million-dollar fine	ACTUAL
...	...	...	

FIGURE 1: DOCUMENT TEXT WITH TABLE OF EVENTTYPE, ARGUMENT EXTRACTIONS

Systems will be given a 500-1,000 document corpus and asked to extract instances of arguments that play a role in some event. Figure 1 illustrates the input and output for a short passage. The event-taxonomy, which specifies extractable event-types and argument-roles[1] appears in Table 1 Systems will need to identify resolved ArgumentStrings, i.e. if a mention can be resolved to a name, the ArgumentString should be the name, if the mention cannot be resolved to a name (e.g. "three police

[1] The taxonomy is based on the taxonomy developed for ACE 2005

officers"), systems should return a specific nominal phrase.

Event Taxonomy

A system will be assessed its performance at extracting event-arguments as described in the tables below. The event and event-specific roles (argument types) are listed in Table 1. All events can also have a Time and Place argument. Only certain entity types are valid for each role. Table 2 lists the valid entity-types per-role.

Event Type	ARG1	ARG2	ARG3	ARG4	ARG5
Business.Declare-Bankruptcy	Org				
Business.Merge-Org	Org	Org			
Conflict.Attack	Attacker	Target	Instrument		
Conflict.Demonstrate	Entity[2]				
Contact.Meet	Entity[3]	Entity[3]			
Contact.Phone-Write	Entity[3]	Entity[3]			
Life.Marry	Person	Person			
Life.Divorce	Person	Person			
Life.Injure	Agent[1]	Victim	Instrument		
Life.Die	Agent[1]	Victim	Instrument		
Movement.Transport	Agent[1]	Artifact[1]	Vehicle	Price[2]	Origin, Destination
Personnel.Start-Position	Person	Entity[1]	Position		
Personnel.End-Position	Person	Entity[1]	Position		
Personnel.Nominate	Agent[2]	Person	Position		
Personnel.Elect	Entity[3]	Person	Position		
Transaction.Transfer-Ownership	Seller	Buyer	Beneficiary	Price[1]	Artifact[2]
Transaction.Transfer-Money	Giver	Recipient	Beneficiary	Money[1]	
Justice.Arrest-Jail	Agent[1]	Person			Crime
Justice.Release-Parole	Entity[3]	Person			Crime
Justice.Trial-Hearing	Prosecutor	Adjudicator	Defendant		Crime
Justice.Sentence	Adjudicator	Defendant	Sentence		Crime
Justice.Fine	Adjudicator	Entity[3]		Money[2]	Crime
Justice.Charge-Indict	Prosecutor	Adjudicator	Defendant		Crime
Justice.Sue	Plaintiff	Adjudicator	Defendant		Crime
Justice.Extradite	Agent[1]	Person	Origin	Destination	Crime
Justice.Acquit	Adjudicator	Defendant			Crime
Justice.Convict	Adjudicator	Defendant			Crime
Justice.Appeal	Prosecutor	Adjudicator	Defendant		Crime
Justice.Execute	Agent[1]	Person			Crime
Justice.Pardon	Adjudicator	Defendant			Crime

TABLE 1: EVENT TYPES AND ARGUMENT ROLES. TIME AND PLACE ARE VALID ARGUMENT ROLES FOR ALL EVENT TYPES. NUMBERS IN [] DISTINGUISH BETWEEN ROLE LABELS FOR WHICH THE SAME ROLE-LABEL IS ASSIGNED DIFFERENT VALID ENTITY TYPES IN TABLE 2

Role	Valid Entity Types	Role	Valid Entity Types	Role	Valid Entity Types
Adjudicator	PER, ORG, GPE	**Entity[1]**	ORG, GPE	**Position**	JOB
Agent[1]	PER, ORG, GPE	**Entity[2]**	PER, ORG	**Price[1]**	MONEY

Agent[2]	PER, ORG, GPE, FAC	**Entity[3]**	PER, ORG, GPE	**Price[2]**	NUM
Artifact[1]	PER, WEA, VEH	**Giver**	PER, ORG, GPE	**Prosecutor**	PER, ORG, GPE
Artifact[2]	VEH, WEA, FAC, ORG	**Instrument**	WEA, VEH	**Recipient**	PER, ORG, GPE
Attacker	PER, ORG, GPE	**Money[1]**	MONEY	**Seller**	PER, ORG, GPE
Beneficiary	PER, ORG, GPE	**Money[2]**	NUM	**Sentence**	SENTENCE
Buyer	PER, ORG, GPE	**Org**	ORG	**Target**	PER, ORG, VEH, FAC, WEA
Crime	CRIME	**Origin**	GPE, LOC, FAC	**Vehicle**	VEH
Defendant	PER, ORG, GPE	**Person**	PER	**Victim**	PER
Destination	GPE, LOC, FAC	**Plaintiff**	PER, ORG, GPE		

TABLE 2: VALID ENTITY TYPES FOR EACH ROLE

System Output

Submissions should be in the form of a single .zip or .tar.gz archive containing exactly one file per input document (and nothing else). Each file's name should be exactly the document ID of the corresponding document, with no extension. All files should use the UTF-8 encoding.

Within each file, each response should be given on a single line using the tab-separated columns below. Completely blank lines and lines with '#' as the first character (comments) are allowable and will be ignored.

A sample response file can be found here:
https://drive.google.com/file/d/0Bxdmkxb6KWZnV0wwcU14cFBsTjQ/edit?usp=sharing

The values in this file were automatically transformed from LDC's ACE annotation of "APW_ENG_20030408.0090". Column 7 (PJ) only includes one offset pair per response line because in ACE event extraction was limited to within sentence event-mention detection. This limitation does not hold for the TAC task. Column 9 (AJ) is NIL because argument inference in the ACE task was limited to coreference. This limitation does not hold for the TAC task.

Column #	Source	Column Name/Description	Values
1	System	**Response ID**	32-bit signed integer (-2^31 to 2^31-1), unique within each file
2	System	**DocID**	
3	System	**EventType**	From ACE Taxonomy see Table 1, column

			1
4	System	**Role**	From ACE Taxonomy see Table 1
5	System	**Normalized/canonical argument string (CAS)**	String
6	System	**Offsets for the source of the CAS.**	Mention-length offset span
7	System	**Predicate Justification (PJ).** This is a list the offsets of text snippets which together establish (a) that an event of the specified type occurred, and (b) that there is some filler given in the document for the specified role. We will term the filler proven to fill this role the **base filler**. If the justifications prove there are multiple fillers (e.g. "John and Sally flew to New York"), which is to be regarded as the base filler for this response will be disambiguated by column 8. The provided justification strings should be sufficient to establish (a) and (b). "Justifications which include spans not needed to establish (a) and (b) will be marked inexact. However, if the number of unnecessary supporting sentences is extreme or inconvenient for annotation, the annotators will ignore this instance (causing it to be counted wrong for purposes of scoring)." Note that the task of the predicate justification is only to establish that there is a filler for the role, not that the CAS is the filler for the role	Set of unrestricted[2] offset spans
8	System	**Base Filler (BF).** This is the base filler referred to in 7.	Mention-length offset span
9	System	**Additional Argument Justification(AJ).** If the relationship between the base filler and the CAS is identity coreference, this must be the empty set. Otherwise, this must contain as many spans (but no more) as are necessary to establish that CAS filling the role of the event may be inferred from the base filler filling the role of the event. One example of such an inference will arguments derived through member-of/part-of relations.	Set of Unrestricted offsets
10	System	**Realis Label**	{ACTUAL, GENERIC, OTHER}
11	System	**Confidence Score.** In the range [0-1], with higher being more confident. In some scoring regimes, the confidence will be used to select between redundant system responses. If necessary due to the short-assessment time frame, confidence may also be used to select those responses to assess (e.g. assessing a system's top N responses).	[0-1]

[2] An unrestricted offset span may always be as long as a sentences without being assessed too long, even if a shorter span is available.

TABLE 3: COLUMNS IN SYSTEM OUTPUT

Offset Calculation and Formatting

As in TAC KBP SlotFilling, each document is represented as a UTF-8 character array and begins with the "<DOC>" tag, where the "<" character has index 0 for the document. Thus, offsets are counted before XML tags are removed. In general, start-end offset spans in columns 6 to 8 are inclusive on both ends: the start offset must be the index of the first character in the corresponding string, and end offset must be the index of the last character of the string (therefore, the length of the corresponding mention string is endoffset – startoffset + 1).

Start and end offsets should be separated by a dash ("-") with no surrounding spaces and pairs of start/end offsets for different mentions should be separated by comma (",") with no surrounding spaces. For example, for the above query, if "yesterday" appears at offset 200 in the document and the document date appears at offset 20, then a valid entry for Column 5 in this case would be: 200-208,20-32 (assuming the endoffset for the document date is 32).

Canonical Argument String

Canonical Argument Strings will be one of the following:

- A string that reflects the fullest/most informative name of a PER, ORG, GPE, FAC, WEA, VEH, LOC in the document
    - Assessments will follow the TAC KBP-Slot Filling guidelines for Name Slots (section 2.1 of http://surdeanu.info/kbp2013/TAC_KBP_2013_Assessment_Guidelines_V1.3.pdf)
- A string that reflects a nominal that cannot be resolved to a name for a PER, ORG, GPE, FAC, WEA, VEH, or LOC
- A normalized specific-date/time (in progress)
    - As in TAC KBP-SlotFilling, dates must be normalized. Systems have to normalize document text strings to standardized month, day, and/or year values, following the TIMEX2 format of yyyy-mm-dd (e.g., document text "New Year's Day 1985" would be normalized as "1985-01-01"). If a full date cannot be inferred using document text and metadata, partial date normalizations are allowed using "X" for the missing information. For example:
        - May 4th" would be normalized as "XXXX-05-04";
        - "1985" would be normalized as "1985-XX-XX";
        - "the early 1900s" would be normalized as "19XX-XX-XX" (note that there is no aspect of the normalization that captures the "early" part of the filler).

    - "the third week of June 2005" as "2005-06-XX"
    - "the third week of 2005" may be returned as **either** "2005-XX-XX" or "2005-01-XX".

- A string-fill for CRIME, SENTENCE, JOB, MONEY

Newlines and tabs in canonical argument strings

The following characters in canonical argument strings shall be replaced with a single space: Windows-style newlines ("\r\n"), Unix newlines ("\n"), and tabs ("\t").

Metadata in Source Documents

- <DATELINE>: For newswire documents, the date in the <DATELINE>… </DATELINE> is frequently important for resolving underspecified dates (e.g. yesterday).
- <post author="..." …. >: For discussion forum data, when accurate personal pronouns (I, you) should be resolved using the string in the author attribute
- <post … datetime="2011-09-01T09:38:00" ...>: For discussion forum data, when accurate, dates should be resolved using the datetime field. Textual context can overrule the datetime field.
- <quote>… </quote>: Answers derived from <quote>...</quote> will not be assessed in this task. The pooling process will automatically remove such answers. This process will remove response rows where either the base-filler (column 8) or canonical argument string offsets (column 6) are within <quote> tags.

Marking of Realis

Each (EventType, Role, ArgumentString) tuple should be augmented with a marker of Realis: ACTUAL, GENERIC, or OTHER.

ACTUAL will be used when the event actually happened with the ArgumentString playing the role as reported in the tuple. For this evaluation, ACTUAL will also include those tuples that are reported/attributed to some source (e.g. *Some sources said…..*, *Joe claimed that…..*)

GENERIC will be used for (EventType, Role, ArgumentString) tuples which refer to the event/argument in general and not a specific instance (e.g. *Weapon sales to terrorists are a problem*)

OTHER will be used for (EventType, Role, ArgumentString) tuples in which either the event itself or the argument did not actually occur. This will include failed events, denied participation, future events, and conditional statements.

If either GENERIC or OTHER could apply to an event (e.g. a negated generic), GENERIC should be used.

Inference and World Knowledge

In the KBP Event Argument Extraction task, assessors will be instructed to mark an answer as correct if a reasonable reader would interpret the document as evidence that the (EventType, Role, ArgumentString, Realis) tuple is correct. They will do this even if such a judgment is derived through inference rather than, for example, a direct linguistic connection between an event-trigger and an

argument. For purposes of this evaluation, systems should infer **argument participation** through links between events; however they should not infer the **occurrence** of one event from another.

Inferring Arguments

Inferences of arguments may include inferring casuality/part-of relations between the verbal-events in a passage, inferring locations through part-of relations, etc—for example inferring the Agent argument of Life.Injure from the Attacker argument of Conflict.Attack. While world-knowledge on its own is not a sufficient reason for a correct answer, such knowledge can contribute to a reasonable reader's assessment. For example, while every instance of a known terrorist group cannot be assumed to be an instance of (Conflict.Attack, Attacker), knowledge that the group has participated in terrorist activities can contribute to a reader's interpretation of vaguely worded text. In such cases, the assessor is instructed to judge "Does this document support the claim of the (EventType, Role, NormalizedArgumentString, Realis) tuple?" Inference about geographical locations (e.g. Cambridge in Massachusetts vs. Cambridge in England) will be assessed using similar guidance.

Invalid Inference of Events from Other Events

While events can in principle be inferred from other events, for purposes of this evaluation, systems should not infer such events. This does not preclude the same text from itself justifying multiple event types (e.g. *shot* in some contexts triggers both injury and attack). This principle applies to all event types. Some particularly common examples:

- Subtypes of Life (e.g. Life.Marry from Life.Divorce)
- Subtypes of Justice (e.g. Justice.Convict from Justice.Pardon)
- Subtypes of Personnel (e.g. Personnel.Start-Position from Personnel.End-Position)

Do not infer future events from current or past events, relations or states. For example, do not infer (Life.Die, Person, Bob Smith, Other) from statements about Bob Smith's marriage, employment, etc. .

Invalid Inference of Events from States

The distinction between a stative relation and the event this relation is a consequence of can be tricky. For most events, we rely on the annotator's judgment that an event is explicitly or implicitly described in the text. The following event types require heightened scrutiny: for these, either (a) a valid temporal argument for the event to be inferred must be available or (b) the event must be signaled by textual evidence of the event (and not only the state):

- Life.Marry
- Life.Divorce
- Personnel.Start-Position
- Personnel.End-Position
- Personnel.Nominate
- Personnel.Elect
- Transport.Movement

Examples of <u>blocked</u> events

- Personnel.Start-Position
    - *ACME spokesman John Smith.*
    - *John Smith works for ACME.*
- Life.Divorce
    - *Sue and her ex-husband John share custody of their children.*
- Transport.Movement
    - *John was born in Boston and went to school in California.*

Examples of <u>allowed</u> events

- Personnel.Start-Position events
    - *ACME hired John Smith. (explicit textual description)*
    - *John Smith has worked for ACME since 2005. (DATE)*
    - *ACME's spokesman since 2005 (DATE)*
- Life.Divorce
    - *Sue, John's ex since 2000…. (DATE)*
    - *John left his wife Sue. She retained ownership of the house. (textual evidence[3])*
- Movement.Transport
    - Bob went to the airport with no particular destination in mind, and the next day he found himself in Prague. (the event is described In the text itself)
- Justice-Arrest.Jail
    - *Bob, an inmate at the county jail…* (Justice.Arrest-Jail is not on the list of event types requiring heightened scrutiny. As such, the assessor will assess this in context without heightened scrutiny).

Departures from ACE 2005[4]

While the ACE 2005 event annotation is being provided to all participants, this task diverges from ACE in some cases. One example of divergence is the addition of correct answers derived through inference/world knowledge (see above). This evaluation will treat as correct some cases that were explicitly excluded in ACE 2005.

- EventType, Role, NormalizedArgumentString tuples that a reasonable reader considers correct but are not explicitly signaled in a single sentence. Some examples are as follows, but they are by no means exhaustive:
    - Inferable arguments (e.g. Agent, Place, Time, etc.), regardless of whether they appear in sentences where ACE would have marked an event-trigger.
    - Arguments that can be inferred through implicit or explicit causality (e.g. the ATTACKER

[3] This is an example of context being used to interpret what could be seen as ambiguous.

[4] ERE annotation will also be provided to participants. The ERE definition of event and event-argument differs in some cases from both the ACE and TAC KBP definitions, but participants may still find the annotation useful. Three notable differences are: (a) ERE allows arguments outside of the sentence in which a trigger is found; (b) ERE does not include certain entity types (e.g. VEHICLE, WEAPON); (c) ERE only marks 'actual' events and not generic, future, attempted etc.

of a Conflict.Attack event also being the AGENT of Life.Die event).

- This removes the "trumping" conditions between {ATTACK, INJURE, DIE} and {MEET, TRANSPORT, EXTRADITE}.

- Arguments which can be inferred through implicit or explicit relations present in the document. For example, PLACE arguments can be inferred through implicit (or explicit) LocatedIn relations in the document.

- For the most part, arguments will be considered valid even independently of the other event-arguments
  - The AGENT/VEHICLE/etc. arguments of a Movement.Transport event are correct even when the ARTIFACT is unspecified (or not a WEAPON, VEHICLE or PERSON); The AGENT/PRICE/etc. arguments of a Transaction.Transfer Ownership is correct even when the ARTIFACT is unspecified or not a WEAPON, VEHICLE or ORGANIZATION).
  - All valid Place arguments will be considered correct (e.g. a city, state, and country). ACE only marked a single Place per 'event-mention'.
- Temporal arguments
  - Temporal arguments should be normalized using the subset of timex2 that is valid in slot-filling. (see Normalized Argument Strings). Correct temporal arguments will capture a time during which the event happened/started/ended (i.e. from ACE: TIME-WITHIN, TIME-AT-BEGINNING, TIME-AT-ENDING, TIME-STARTING, TIME-ENDING, but not TIME-BEFORE or TIME-AFTER). Temporal arguments must be resolvable to a time period on the calendar (e.g. *September 2005* or the *first week of August*). Durations (*for three months*) or times marked by other events (*after his trip*) are not correct answers. Unlike ACE, we will not distinguish between different types of temporal roles, and all temporal arguments will be marked as Time.
  - In ACE, when a temporal argument might apply to multiple events, it is only marked on the most syntactically local. For this task, that restriction is removed, and temporal arguments are to be marked for all applicable events.
- Life.Injure and Life.Die. Life.Die events are frequently (perhaps always) preceded by a Life.Injure event. In ACE annotation, Life.Injure became a distinct event-mention if there was a distinct trigger "Bob was shot dead" → Life.Die and Life.Injure; "Assassins killed Bob" → only Life.Die. In this evaluation, for scoring purposes we assume Life.Die incorporates Life.Injure. If the assessed pool contains a correct Life.Die tuple, the scorer will ignore Life.Injure tuple(s) that are identical to the Life.Die tuple in CAS-id, role, and realis marker. Thus, if (Life.Die, Place, Springfield, Actual) is correct if (Life.Injure, Place, Springfield, Actual) will be ignored. This rule only applies when the Life.Die is assessed as correct.
  - Example 2: Bob was shot and killed.
    - Correct: (Life.Die, Victim, Bob, Actual) → rule applied

        - Ignore: (Life.Injure, Victim, Bob, Actual)
    - Example 2: Bob was beheaded, but miraculously they sewed his head back on and he survived.
        - Wrong: (Life.Die, Victim, Bob, Actual) à rule not applied
        - Correct: (Life.Injure, Victim, Bob, Actual)
    - Example 3: The friendship ended when Bob brutally destroyed Joe in a game of cards.
        - Wrong: (Life.Die, Victim, Bob, Actual) à rule not applied
        - Wrong: (Life.Injure, Victim, Bob, Actual)
- This evaluation also removes three event-types from the ACE taxonomy: Life.Born, Business.Start-Org, Business.End-Org[5].

Corpus

The corpus will be a mix of newswire and discussion forum documents. The total corpus size will be 500-1,000 documents. A discussion forum document may contain multiple posts. The corpus will be manually and automatically filtered to ensure at least a few instances of all event-types. The discussion-forum posts will be automatically filtered to identify those posts that are not simply reposts of newswire documents. Very long discussion-forum threads will be truncated.

Assessment and Evaluation

Systems will be evaluated on F-Score of (EventType, Role, NormalizedArgumentString, Realis) over the pooled assessments. The assessment pool for each document will include a human-participant. The human-participants submission will be assessed in the same manner as the system submissions.

In the event that the response pool is too large, NIST/LDC may select a subset of responses from each system to assess. For each system, tuples that are redundant in columns 2-6 and 10[6] will be collapsed to a single representative (selecting the highest confidence representative)[7]. Then the top-N answers from each system ranked by system confidence will be used to form the assessment pool.

Dimensions of Assessment

An LDC assessor will mark each system response for correctness. Assessment judgments evaluate the accuracy of the elements of the (EventType, Role, Cannoical Argument, Realis) tuple and of the justification for the EventType, Role, and CannonicalArgument.

Column #	Source	Column Name/Description	Values
12	Assessor	**Assessment of Event Type (AET).** Does the predicate justification (PJ) support the presence of an event of the type specified in	C – Correct W – Wrong

[5] These three events are well represented as KB-SlotFilling Slots.

[6] As described in System Output, the columns labeled docid, event type, event role, CAS, CAS offsets, realis

[7] This collapsing will happen before coreference of CAS and assessment, as answers that become redundant through CAS and/or the Life.Injure à Life.Die transformation (see Departures from ACE 2005) will not be collapsed.

[7]

		column 3? This may be marked NIL if either the PJs or AJs for a response are excessively large. Note that a response may be marked ignore for excessively long AJ, even though the AJ normally does not play a role in evaluating AET.	I - inexact –NIL
13	Assessor	**Assessment of Role (AR)**- Does the predicate justification (PJ) support the presence of who/what/when/where fills the role specified in column 4? (e.g. Can you tell from the predicate justification that some ATTACKER is mentioned for a Conflict.Attack event)? Note that this only has to prove that **some** filler is supported for this role; not necessarily the CAS (NIL if AET is WRONG or NIL)	C – Correct W – Wrong I - inexact NIL
14	Assessor	**Assessment of Normalized/Canonical Argument String**. Is the CAS a correct filler for the event-type/role tuple? (NIL if AET or AR is WRONG or NIL). Wrong (not inexact) will be used in the case of a CAS strings that are not canonical (e.g. a noun phrase when a name is available, an unnormalized date).	C – Correct W – Wrong I - inexact NIL
15	Assessor	**Assessment of base filler.** Is the base filler a correct filler for the event-type/role specified in columns 3 and 4? NIL if AET and/or AR are WRONG or NIL	C – Correct W – Wrong I - inexact NIL
16	Assessor	Coreference IDs for CAS. Singletons should be also be assigned coreference IDs. NIL if 14 is NIL or Wrong.	Unique ID (32-bit signed integer) NIL
17	Assessor	Realis Label	{ACTUAL, GENERIC, OTHER} NIL if AET or AR is NIL or WRONG
18	Assessor	Mention-type for CAS	{NAME, NOMINAL} NIL if AET or AR is NIL or WRONG

TABLE 4: QUESTIONS OF LDC ASSESSOR TASK

Scoring Metric

A package to automatically validate system output and score (given assessments) is available here: https://github.com/rgabbard-bbn/kbp-2014-event-arguments

The score is calculated as follows.

- Recall that certain responses will be removed from both the system output and the annotation

pool before scoring. For example, see the treatment of Life.Injure in "Departures from ACE".

- In all cases below, a 'response' is defined as columns 2-8 and 10 of the system output.
- Confidence is not considered part of the response, but confidences are used to define a confidence function *c(x)*.
- Recall the CAS stands for *canonical argument string* (column 5 of the system response).
- Recall that annotators will indicate coreference relationships between CASes within the same document (column 16 of the annotation). The transitive closure of this coreference relation defines an equivalence relation which we will call **C**.
- of the following are true
    - $x\text{document id}=y\text{document id}$
    - $x\text{event type}=y\text{event type}$
    - $x\text{argument role}=y\text{argument role}$
    - $x\text{cas}\sim y\text{cas}$
    - $x\text{realis}=y\text{realis}$
- Let **O** be the set of system responses. Define **O'** to be $\{\rho(X)|X\in O/\sim T\}$, where $\rho(X)$ is $\text{argmax}_{x\in X} c(x)$. If this maximum is not unique, ties will be broken 'randomly'.
- Let **P** be the union of the system responses of all systems, including the human 'system's'. Note that by definition $O'\subseteq P$.
- Let ***a*** be a function from system responses to their annotation tuples.
    - ***a*** will be completely defined on **P**.
    - Note that since ***a*** is a function, we assume there are not multiple inconsistent annotations for any response.
- We define an annotation tuple α to be a *good annotation for* a response *x* if none of columns 11-15 of α are WRONG or IGNORE and $\alpha\text{realis}=x\text{realis}$.
    - Informally, this means the response is correct except for possible errors regarding the extent of justifications.
- We define an annotation tuple α to be a *perfect annotation for* a response *x* if columns 11-15 of α are CORRECT and $\alpha\text{realis}=x\text{realis}$.
    - Informally, this means the response is exactly correct.
- We define an equivalence class **X** to be *good* if there exists an $x\in X$ such that $\alpha(x)$ is a good annotation for *x*.
- We define an equivalence class **X** to be *perfect* if there exists an $x\in X$ such that $\alpha(x)$ is a perfect annotation for *x*.
- We define an equivalence class **X** to be *semantically correct* with respect to **P** if **X** is non-empty and there exists $p\in P$ such that *p* is equivalent to the elements of **X** under **T** and $\alpha(p)$ is a good annotation for *p*.
    - Informally, this means we have evidence somewhere in the pool that the corresponding *docid-type-role-CAS-realis* tuple is correct, but there may or may not be a matching justification in the system output.
- Let $s\in\{\text{good, perfect, semantically correct}\}$. Define the *s-precision* of a system's output **O'** as the number of *s* equivalence classes of **O'** under **T** divided by the total number of equivalence classes of **O'** under **T.**
- Let $s\in\{\text{good, perfect, semantically correct}\}$. Define the *s-recall* of a system's output **O'** as the number of equivalence classes of **O'** under **T** which have property *s* divided by the total number of *s* equivalence classes of **P** under **T.**

    - Note the stipulation "which also has property *s"* is necessary because the responses which have a good or perfect annotation in an equivalence class of **P** may be missing from the matching equivalence class in **O'**.
- Let $s \in \{\text{good, perfect, semantically correct}\}$. Define the *s-F1* measure of a system's output **O** as $2prp+r$ where *p* is the *s*-precision of **O'** and *r* is the *s*-recall of **O'**.
- The first scoring measure for the task is good-F1, called the **standard** measure.
- The second scoring measure for the tasks is perfect-F1, called the **strict** measure.
- The third scoring measure for the tasks is semantically-correct-F1, called the **lax** measure.

Submissions and Schedule

Submission

Systems will have up to one week to process the evaluation documents. Submissions should be fully automatic and no changes should be made to the system once evaluation corpus has been downloaded. Up to five alternative system runs may be submitted per-team. Submitted runs should be ranked according to their expected overall score. Teams should submit at least one version of their system that does not access the web during evaluation. Any web-access of alternative systems should be documented in the system description.

Schedule

March 2014	Task definition released. ACE 2005 data released to participants as they sign up.
April 1, 2014	LDC releases sample documents (~50) to participants in preparation for pilot assessment
April 15 2014	Participants (optionally) submit pilot output to for pilot assessment. Pilot may be automatic, manual, or some combination. While participation is not required, participants are encouraged to submit output to better understand this new task (e.g. receive feedback on the assessment of inference, nominal extents, realis).
May 15th 2014	LDC releases the full set of pilot assessment to participants who submitted pilot output.
August 11-18 2014	Evaluation Period

Examples

Filtering of Redundant Responses during Pooling and Scoring

Text:

- s6: The trial started early in the morning and lasted for six hours before the ruling was announced by Judge Jones Chinyama.
- s7: Jones Chinyama said the prosecution team had failed to prove the case against Chiluba.
- s8: The judge said that an important attestor failed to show up during the trial, adding that "I find that the accused is not guilty on all counts.

(A) Responsesfrom System X

- R1: (Justice.Trial-Hearing, Adjudicator, Jones Chinyama, ACTUAL) CAS offsets: S6; Base Filler: Jones Chinyama, S6; conf: 0.9
- R2: (Justice.Trial-Hearing, Adjudicator, Jones Chinyama, ACTUAL) CAS offsets: S6; Base Filler: Jones Chinyama, S7; conf: 0.8
- R3: (Justice.Trial-Hearing, Adjudicator, Jones Chinyama, ACTUAL) CAS offsets: S7; Base Filler: Jones Chinyama, S7; conf: 0.7
- R4: (Justice.Trial-Hearing, Adjudicator, Jones Chinyama, ACTUAL) CAS offsets: S6; Base Filler: The judge, S8; conf: 0.6
- R5: (Justice.Trial-Hearing, Adjudicator, Jones Chinyama, ACTUAL) CAS offsets: S7; Base Filler: The judge, S8; conf: 0.5

(B) Automatically clustered responses from System X

- (R1 conf:0.9, R2 conf: 0.8, R4 conf: 0.6)
- (R3 conf: 0.7, R5 conf: 0.5)

(C) Responses from System X that are sent to assessment

- R1: (Justice.Trial-Hearing, Adjudicator, Jones Chinyama, ACTUAL) CAS offsets: S6; Base Filler: Jones Chinyama, S6; conf: 0.9
- R3: (Justice.Trial-Hearing, Adjudicator, Jones Chinyama, ACTUAL) CAS offsets: S7; Base Filler: Jones Chinyama, S7; conf: 0.7

(D) Assessment will add coreference IDs to CAS,leading to redundant responses:

(R1 conf: 0.9, R3 conf 0.7)

(E) Scoring will use R1 and ignore R3

References

ACE 2005 Training Data: http://catalog.ldc.upenn.edu/LDC2006T06

ACE Task Guidelines: https://www.ldc.upenn.edu/collaborations/past-projects/acehttp://catalog.ldc.upenn.edu/LDC2006T06

http://catalog.ldc.upenn.edu/LDC2006T06