# Cold Start Knowledge Base Population at TAC 2015

## Task Description[1]

Version 1.1 of July 14, 2015



[1] The TAC organizing committee welcomes comments on this Task Description, or on any aspect of the TAC evaluation. Please send comments to tac-kbp@nist.gov.

## What's New

These are the major changes in the Cold Start task for 2015:

- **All Variants:** Cold Start will have three variants in 2015: Knowledge Base (CSKB), Slot Filling (CSSF), and, new in 2015, Entity Discovery (CSED). Cold Start documents and query entities will be selected to increase the coverage of names that are ambiguous, confusable, or reflect entities that have multiple names.
- **Slot Filling Variant**: The Slot Filling variant of Cold Start, which ran for the first time in 2014, is superseding the Slot Filling track. Participants moving over from that track must add the ability to recognize the inverse of each of the required slots, and to provide type information for the slot fillers (the inventory of slots is not changing from 2014).
- **Entity Discovery Variant:** To foster research in scaling up EDL systems to larger number of documents, and to serve as a springboard into the full Knowledge Base variant, Cold Start is adding an Entity Discovery variant. This variant asks participants to produce a skeleton knowledge base from a large set of documents that includes only mentions and type information (i.e., it omits relations and canonical mentions). This variant is English-only in 2015, is restricted to named mentions of persons, organizations, and GPEs, and does not link to any existing TAC knowledge base (i.e., all links are NIL). The EDL task scorer will be used to score Entity Discovery variant entries. To make it easier for teams that would like to participate in both Trilingual Entity Discovery and Linking and CSED, a script will be provided to convert EDL output to the CSED format.

## Introduction

Since 2009, TAC has evaluated performance on two important aspects of knowledge base population: entity linking and slot filling. The goal of the Cold Start track is to exercise both of these areas, and evaluate the ability of a system to use these technologies to actually construct a knowledge base (KB) from the information provided in a text collection. Cold Start participants build a software system that processes a large text collection and creates a knowledge base that is consistent with and accurately represents the content of that collection. The knowledge base is then evaluated as a single connected resource, using queries that traverse entity nodes and relation (slot) links in the KB to determine if the KB contains correct relations between correct entities.

In 2015, Cold Start will have three variants. The first, the *Knowledge Base* variant (CSKB), is the same as the 2014 Cold Start Knowledge Base task; participants submit entire knowledge bases, without prior knowledge of the evaluation queries. The second, the *Slot Filling* variant (CSSF), supersedes the 2014 Slot Filling track. It is designed to make it easy for sites with slot filling systems to participate in Cold Start. In this variant, the evaluation queries are distributed at the start of the task. Participants do not have to submit entire knowledge bases. Rather, they apply their slot filling system twice, the first time on the entry point for each query, the second time on each of the results of the first round. The third variant, the Entity Discovery variant (CSED), asks participating systems to identify all named entities in the document collection, and to cluster all mentions that refer to the same entity.

The Entity Discovery and Linking task and the Slot Filling task have done a good job of evaluating key components of knowledge base population. They do not, however, evaluate every aspect of an automatically generated knowledge base. Things one might like to know about such a knowledge base include:

- Are the entities in the knowledge base correctly tied to real-world mentions of those entities? TAC Entity Discovery and Linking (EDL) tasks have measured this, as will TAC Cold Start ED.
- Are the facts and relations in the knowledge base accurate reflections of the facts and relations described in the source documents? The TAC Slot Filling tasks have measured this, as will TAC Cold Start SF.
- Are entity linking and slot filling correctly coordinated to produce a meaningful knowledge base? The TAC Cold Start KB task measures this.
- Can the knowledge base correctly perform inference over the extracted entities, such as temporal reasoning, confidence estimation, default reasoning, transitive closure, etc.? Cold Start is just beginning to measure this; it is designed to facilitate this kind of evaluation more thoroughly in future years.

We call the task *Cold Start* Knowledge Base Population to convey two features of the evaluation: it implies both that a knowledge base schema has been established at the start of the task, and that the knowledge base is initially unpopulated. Thus, we assume that a schema exists for the entities, facts, and relations that will compose the knowledge base; it is not part of the task to automatically identify and name facts and relationships present in the text collection. We will use the schema that was used for the TAC KBP 2014 English tasks for Entity Discovery and Linking, Slot Filling, and Cold Start Knowledge Base Population. Thus, the schema will include three named entity types (person, organization and geopolitical entity) and forty-one relation types and their inverses. For relations whose fills are themselves entities (such as `per:siblings` or `org:subsidiaries`), CSKB systems will be required to link that slot to the node in the submitted KB representing the correct entity. Slots whose fills are strings (such as `per:title` or `org:website`) will simply use strings to represent the information.

*Cold Start* also implies that the knowledge base is initially empty. To avoid solutions that rely on verifying content already present in Wikipedia or other large data sources about entities, the queries used in Cold Start will be dominated by entities that are not present in Wikipedia.

All participating systems will receive the following input:

1. a *document collection*;
2. a *knowledge base schema*

From these, systems participating in the Knowledge Base and Entity Discovery variants will produce a knowledge base. This KB will be submitted to NIST as a set of augmented triples. Participating systems must tie each entity mention in the document collection to a particular KB entity node; in this way, the knowledge base can be queried without first aligning it to a reference knowledge base. Knowledge bases submitted for the Entity Discovery variant will include only `mention` and `type` triples; those submitted for the Knowledge Base variant will also include the full range of slot fills, as well as `canonical_mention` triples (all triples are described more fully below).

Systems participating in the Slot Filling variant will also receive:

3. a set of Cold Start evaluation queries (each Cold Start query is a sequence of one or two slot filling queries to be applied in series).

For all three variants, the results will then be evaluated by NIST. Evaluation of the Knowledge Base variant will start by applying the evaluation queries to the submitted knowledge base. Each query will start at a named mention in a document (identified by the query's `<beg>` and `<end>` tags), identify the knowledge base entity that corresponds to that mention, follow a sequence of one or more relations within the knowledge base, and end in a slot fill. The resulting slot fills will be assessed and scored in much the same way as has been done in the Slot Filling task. For example, a

KB evaluation query might ask 'what are the ages of the siblings of the *Bart Simpson*[2] mentioned in Document 42?' A system that correctly identified descriptions of Bart's siblings in the document collection, linked them to the appropriate node in the KB, and also found evidence for and correctly represented the ages of those siblings would receive full credit.

| Relation | Inverse(s) |
|---|---|
| per:children | per:parents |
| per:other_family | per:other_family |
| per:parents | per:children |
| per:siblings | per:siblings |
| per:spouse | per:spouse |
| per:employee_or_member_of | {org,gpe}:employees_or_members* |
| per:schools_attended | org:students* |
| per:city_of_birth | gpe:births_in_city* |
| per:stateorprovince_of_birth | gpe:births_in_stateorprovince* |
| per:country_of_birth | gpe:births_in_country* |
| per:cities_of_residence | gpe:residents_of_city* |
| per:statesorprovinces_of_residence | gpe:residents_of_stateorprovince |
| per:countries_of_residence | gpe:residents_of_country* |
| per:city_of_death | gpe:deaths_in_city* |
| per:stateorprovince_of_death | gpe:deaths_in_stateorprovince* |
| per:country_of_death | gpe:deaths_in_country* |
| org:shareholders | {per,org,gpe}:holds_shares_in* |
| org:founded_by | {per,org,gpe}:organizations_founded* |
| org:top_members_employees | per:top_member_employee_of* |
| {org,gpe}:member_of | org:members |
| org:members | {org,gpe}:member_of |
| org:parents | {org,gpe}:subsidiaries |
| org:subsidiaries | org:parents |
| org:city_of_headquarters | gpe:headquarters_in_city* |
| org:stateorprovince_of_headquarters | gpe:headquarters_in_stateorprovince* |
| org:country_of_headquarters | gpe:headquarters_in_country* |

**Table 1. Entity-valued slots. Slots with asterisks represent inverse relations that will need to be added by participants from previous years Slot Filling task. The type qualifier of each relation (`per`, `org` or `gpe`) is the type of its subject, while the type qualifier for its inverse is the type of its object. A set of types means that any of those types is acceptable for that slot. All submitted slot names must use only a single type specification.**

[2] Many of the examples used to illustrate the Cold Start task are drawn from *The Simpsons* television show. Readers lacking a detailed working knowledge of genealogical relationships in the Bouvier/Simpson family need not agonize over what they have been doing with their lives for the past quarter century, but may simply visit http://simpsons.wikia.com/wiki/Simpson_Family.

| | |
|---|---|
| per:alternate_names | org:alternate_names |
| per:date_of_birth | org:political_religious_affiliation |
| per:age | org:number_of_employees_members |
| per:origin | org:date_founded |
| per:date_of_death | org:date_dissolved |
| per:cause_of_death | org:website |
| per:title | |
| per:religion | |
| per:charges | |

**Table 2. String-valued slots.**

## Schema

The schema for Cold Start 2015 is identical to the schema used for Cold Start 2014. The three entity types and their mentions are defined in *TAC KBP 2015 – Entity Discovery and Linking (ED&L) Guidelines*, and the inventory of slots is described thoroughly in *TAC KBP 2015 Slot Descriptions* and *TAC KBP 2015 Assessment Guidelines* available on the TAC Web site. Forty-one slots and their inverses are used for the evaluation. Twenty-six of these have fills that are themselves entities, as shown in Table 1. The remaining fifteen slots have string fills, as shown in Table 3. Each entity-valued slot will have an inverse.[3] All inverse relations must be explicitly identified in the submitted knowledge base. That is, if the KB asserts that relation R holds between entities A and B, then it must also assert that relation $R^{-1}$ holds between B and A. As a convenience, the Cold Start KB validation script can be used to introduce missing inverses into a KB. Please see the 2015 Slot Description and Slot Assessment documents (available from `http://www.nist.gov/tac/2015/KBP/ColdStart/guidelines.html`) for a complete description of and assessment criteria for each slot.

## Document Collection

The Cold Start evaluation document collection will comprise approximately 50,000 newswire, Web, and discussion forum documents formatted in the same way as the documents of the TAC 2014 KBP English Source Corpus. Cold Start evaluation documents will be new (previously unreleased) documents that will be distributed by NIST via Web download at the beginning of the Cold Start evaluation window. Provenance for submitted relations must be drawn from these documents for all variants of the Cold Start task.

## Evaluation Queries

Cold Start KB and Cold Start SF systems are evaluated by the same set of evaluation queries. Each Cold Start query describes a sequence of entities and slots to be filled. Participants in the Slot Filling variant of Cold Start will receive the Cold Start evaluation queries at the beginning of the

[3] Some slots, such as per:siblings, are symmetric. Others, such as per:parents, have inverses that were already in the 2014 English Slot Filling track (in this case, per:children). The remaining slots (e.g., org:founded_by) had no corresponding slot in the 2014 English Slot Filling track; Cold Start specifies new slot names for these inverses. All such slots are list-valued.

evaluation window, and will apply a script to incrementally convert the Cold Start queries to a form that looks similar to queries from the 2014 English Slot Filling task. Participants in the Knowledge Base variant will not receive the queries; rather, NIST will apply the evaluation queries to each submitted knowledge base and assess the results. An outline of the NIST assessment process for both Cold Start variants is given below.

All evaluation queries start with an *entry point* into the knowledge base being evaluated. The entry point is defined by an entity mention (name, docid, begin offset, and end offset), and is followed by the entity type and either one or two slots to be extracted for the entity.

Evaluation queries could take one of two forms: single-hop or multiple-hop. For example, here is a sample single-hop Cold Start evaluation query that asks "What is the age of the *June McCarthy* mentioned at offsets 16931-16943 in Document 42?":

```
<query id="CS11_ENG_211">
  <name>June McCarthy</name>
  <docid>42</docid>
  <beg>16931</beg>
  <end>16943</end>
  <enttype>PER</enttype>
  <slot>per:age</slot>
  <slot0>per:age</slot0>
</query>
```

This single-hop query looks very much like a query from the 2014 English Slot Filling task, except that each Cold Start evaluation query asks for a specific slot, rather than all slots for which there is information in the document collection. [4]

A more complex "two-hop" evaluation query might ask, "What are the ages of the children of the *June McCarthy* mentioned at offsets 16931-16943 in Document 42":

```
<query id="CS11_ENG_210">
  <name>June McCarthy</name>
  <docid>42</docid>
  <beg>16931</beg>
  <end>16943</end>
  <enttype>PER</enttype>
  <slot>per:children</slot>
  <slot0>per:children</slot0>
  <slot1>per:age</slot1>
</query>
```

In general, two-hop evaluation queries will start from an entry point (selecting the corresponding KB entity of a CSKB submission), follow a single entity-valued relation (from Table 1), then ask for a single slot value (from either Table 1 or Table 3). [5] Such queries will verify that the knowledge base is well-formed in a way that goes beyond basic entity linking and slot filling, without allowing combinations of errors to drive scores to zero.

---

[4] Participants in the Slot Filling variant should treat all other slots as if they appear in the <ignore> field of a Slot Filling query.

[5] In principle, multiple-hop queries could include more than two relations, but we limit ourselves to two in Cold Start 2015.

Because two-hop queries do not look like any slot filling queries from KBP 2009-2014, participants in the CS Slot Filling variant must process the Cold Start evaluation queries in two "rounds" using the `GenerateCSQueries.pl` script, which adds the `<slot>` entry to the NIST-distributed Cold Start queries. Participants in the Slot Filling variant must treat `<slot>` as the slot to be filled. During the first round, `<slot>` will be identical to `<slot0>`. The `GenerateCSQueries.pl` script will then convert a first round output file to a second round query file. Second round queries generated by this script will bear `<slot>` entries equivalent to `<slot1>`.

For the Knowledge Base variant, the following rules are applied to map from an evaluation query to a knowledge base entry: First, form a candidate set of all KB node mentions that have at least one character in common with the evaluation query mention and that have the same type. If this set is empty, the submission does not contain any answers for the evaluation query. Otherwise, for each mention K in the candidate set, calculate:

- COMMON, the number of characters in K that are also in the query mention Q.
- K_ONLY, the number of characters in K that are not in Q.

Execute each the following eliminations until the candidate set is size one, and select that candidate as the KB node that matches the query:

- Eliminate any candidate that does not have the maximal value of COMMON
- Eliminate any candidate that does not have the minimal value of K_ONLY
- Eliminate all but the candidate that appears first in the submission file

The proper specification of `mention` relations in a KB is therefore important for scoring well; CSKB participants should therefore take care to ensure that every entity mention in the evaluation collection serves as a `mention` relation for a node in the KB.

The NIST evaluation of a KB will proceed by finding all entries in the KB that fulfill an evaluation query. For example, if the evaluation query 'schools attended by the siblings of *Bart Simpson*' finds two siblings for the node specified by the entry point, and the KB indicates that those siblings attended two and one schools respectively, then three results would be assessed by NIST. These results will be converted to the same form as the output for the Slot Filling variant. Results will be pooled across all CSKB and CSSF submissions, and assessors will judge the validity of each result. Finally, a scoring script will report a variety of statistics for each submitted run.

To improve our ability to assess the relative importance of entity linking and slot filling, some of the evaluation queries will differ only in having different mentions for the same entity as their entry points. Participating systems are prohibited from using information about one evaluation query to inform the processing of another evaluation query.

In creating evaluation queries, LDC will strive to balance even distribution across slot types with productivity of those slots. Single hop queries, which are of greater interest for slot filling, will in many cases ask for multiple slots for a given entity regardless of whether fillers for those slots are attested in the document collection. Multiple hop queries will favor entities and slot sequences that are attested in the document collection (although here too, availability of answers is not guaranteed at any hop level).

## Task Output – Knowledge Base Variant

Systems must produce a knowledge base as output. The first line of the output file must contain a unique run ID, which is a single token that contains no white space and no pound sign, and that does not begin with a colon. The remainder of the KB is represented as a set of augmented triples. Assertions will appear, one-per-line, in tab-separated format. The output file will be automatically converted to RDF statements during evaluation. All output must be encoded in UTF-8.

Each triple appears in the output file in subject-predicate-object order. For example, to indicate that entity-4 has entity-7 as a sibling, the triple might be:

```
:e4    per:siblings  :e7
```

If entity-4 has siblings in addition to entity-7, these relations should be entered as separate triples.

### Entities

Each entity specification begins with a colon, followed by a sequence of letters, digits and underscores. Examples of legal entity specifications include `:Entity42`, `:EE74_R29`, and `:there_were_two_muffins_in_the_oven`. No meaning is ascribed to this sequence by the evaluation software; it is used only as a unique identifier. Any subsequent use of the same colon-preceded sequence will be taken as a reference to the same entity.

### Predicates

The legal predicates are those shown in Table 1 (including inverses) and Table 3, plus `type`, `mention`, and `canonical_mention`.

Predicates found in Table 1 must have entity specifications in both the subject and object positions. Predicates found in Table 3 must have an entity specification in the subject position, and a double quote-delimited string in the object position; the string in the object position will exactly correspond with the slot fill for that relation in the Slot Filling task. A backslash character must precede any occurrence of a double quote or a backslash in such a string.[6] At least one instance of each unique subject-predicate-object triple will be evaluated. If more than one instance of a given triple appears in the output (with each triple having different provenance), LDC will assess the instance with the highest confidence value (see below), and will assess additional instances if resources allow. If more than one such triple shares the same confidence value, the triple that appears earlier in the output will be considered to have higher confidence.

`type`

Each entity will be the subject of exactly one `type` triple. The object of that triple will be either `PER`, `ORG` or `GPE` depending on the type of the entity. It is up to submitting systems to correctly identify and report the type of each entity.

`mention`

Each entity will be the subject of one[7] or more `mention` triples. Together with the provenance information (see below), these triples indicate how the knowledge base is tied to the document collection. Each named entity mention in the collection must be submitted as the object of a `mention`

[6] Each backslash used to quote the following character doesn't itself have to be preceded by another backslash.

[7] While unmentioned but inferred entities may play a role in future TAC evaluations, Cold Start 2015 will work only with entities that have named mentions.

triple. For example, if an entity is mentioned by name five times in a document, five `mention` triples should be generated. The object of a `mention` triple is the double-quoted mention string; document ID and offset appear under provenance information (see below). The definition of what constitutes a named entity mention for Cold Start is the same as the definition of named entity mentions in the KBP 2014 English Entity Discovery and Linking task: a named entity mention is a mention that uniquely refers to an entity by its proper name, acronym, nickname, alias, abbreviation, or other alternate name, and includes post author names found in the metadata of discussion forum threads and web documents. The extent of the named entity mention is the entire string representing the name, excluding the preceding definite article and any other pre-posed or post-posed modifiers. Named entity mentions are allowed to nest or overlap; for example, the string "Philadelphia Eagles" might be a mention of an ORG (the football team), while the first word might simultaneously be a mention of a GPE (the city of Philadelphia).

`canonical_mention`

For each document that mentions an entity, one of the `mentions` must be identified as the *canonical mention* for that entity in that document; it is the string that will be seen by the assessor if that entity appears as a slot fill, supported by that document (in Slot Filling task terms, it is the content of Column 5 of a CSSF 2015 submission, and its provenance will serve as Column 7 of the CSSF submission).[8] Canonical mentions are expressed using a `canonical_mention` triple. The arguments for `canonical_mention` are the same as for `mention`. Note that there is no requirement that submissions select a single, global canonical mention for an entity. While such a name might be useful, here we require that a name be provided within each document for the assessor to use during assessment. Each `canonical_mention` is also a `mention`. As a convenience, if a submitted KB does not contain a `mention` triple for each `canonical_mention` triple, the missing relations will be inferred (albeit with a warning). This shortcut is provided to make submitted KBs easier to view, and does not relieve submitters from the requirement to provide each of the required `mentions` and `canonical_mentions`.

## Task Output – Entity Discovery Variant

Output for the Entity Discovery variant will look exactly like output for the Knowledge Base variant, with the exception that all submitted slot fills and `canonical_mention` triples will be ignored (thus, they need not be submitted). To make it easier for TAC EDL Task participants to submit to CSED, a script will be made available to convert the output of that task into the knowledge base format required by CSED. The main challenge faced by EDL systems is thus the ability to scale up to tens of thousands of documents.

[8] In the Slot Filling task of KBP 2009-2014 (and in the Slot Filling variant of Cold Start), all slot fills are strings. Assessors verify the validity of a slot fill by looking for that string in the specified document, using the provenance information provided in the system response. In a submitted KB, slots that are filled with entities hold not strings, but pointers to the KB structure for the appropriate entity. Thus, a canonical mention must be identified by the Cold Start KB for each entity in each document, so that the assessor can be presented with a string that represents the entity during assessment. A relation provenance (see below) entry may include more than one document, and at least one of those documents must contain a mention of the object of the relation; that document must therefore contain a canonical mention for the object. When selecting a canonical mention for presentation to the assessor, the first document appearing in the relation provenance that contains a mention of the object will be used for the canonical mention.

## Task Output – Slot Filling Variant

Output for the Slot Filling variant will be in the form of a tab-separated file. The columns of the submitted file are as follows:

| | |
|---|---|
| Column 1 | Query ID. For the first round, this is taken directly from the <query> XML tag. For the second round, this is drawn from the <query> tag of the query generated from the first round output. |
| Column 2 | The name of the slot being filled. |
| Column 3 | A unique run id for the submission. |
| Column 4 | Provenance for the relation between the query entity and slot filler, consisting of up to 4 docid:startoffset-endoffset triples separated by commas. Individual spans may comprise at most 150 UTF-8 characters. 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. Start offsets in these columns must be the index of the first character in the corresponding string, and end offsets must be the index of the last character of the string (therefore, the length of the corresponding mention string is endoffset – startoffset + 1). Note that <DOC> tags vary slightly across the different document genres included in the source corpus; it can be spelled with either upper case or lower case letters, and may optionally include additional attributes such as "id" (e.g., <doc id="doc_id_string">). Unlike the 2014 Slot Filling task, there is no requirement to generate NIL entries when no information about the target entity is available. |
| Column 5 | A slot filler (possibly normalized, e.g., for dates). This is used both to populate the <name> entry of the next round query, and by the assessor to judge the slot fill. The string should be extracted from the filler provenance in Column 7, except that any embedded tabs or newline characters should be converted to a space character and 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. |
| Column 6 | A filler type, selected from {PER, ORG, GPE, STRING}. The STRING filler is used for string-valued slots shown in Table 3. |
| Column 7 | Provenance for the slot filler string. This is either a single span (docid:startoffset-endoffset) from the document where the **canonical** slot filler string was extracted, or (in the case when the slot filler string in Column 5 has been normalized) a set of up to two comma-separated docid:startoffset-endoffset spans for the base strings that were used to |

| | |
|---|---|
| | generate the normalized slot filler string. The documents used for the slot filler string provenance must be a subset of the documents provided in Column 4. This column serves two purposes. First, LDC will judge Correct vs. Inexact with respect to the document(s) provided in the slot filler string provenance. Second, this column is used to fill the <docid>, <beg> and <end> entries in second round queries. If more than one provenance triple is provided here, the first one will be used to fill the second round query. |
| Column 8 | Confidence score. |

Note that Column 6, the type of the slot fill, is a new column in 2015.

The process for constructing a Slot Filling variant submission is as follows:

- Download the following from the NIST Web site:
  - The Cold Start evaluation documents
  - The evaluation queries
  - `CS-GenerateQueries.pl` script
  - `CS-PackageOutput.pl` script
- Configure your system to produce results only from the Cold Start evaluation documents.
- Run the `CS-GenerateQueries.pl` script on the evaluation queries to produce the first round queries your system will run on. Note that the raw evaluation queries might differ from the format given above, so you should not assume that you can use them as input to your system without running this script.
- Run your system, producing a slot-filling submission for the first round queries.
- Run the `CS-GenerateQueries.pl` script on the evaluation queries and your first round output to produce the second round queries.
- Run your system on the second round queries to produce a second output file.
- Run the `CS-PackageOutput.pl` script on the two output files to produce your submission.
- Upload the submission to NIST.

## Task Output – All Variants

### Provenance

Each triple in Knowledge Base or Entity Discovery variant submissions and each output line in Slot Filling variant submissions will include a set of augmentations (again using tabs as separators). Except for the `type` predicate (which does not require explicit support from a document) the first annotations will describe the provenance of the assertion. Provenance for submissions for the Slot Filling variant have already been described above; corresponding provenance for triples in KB and ED variant submissions are detailed here:

For predicates for relations from Table 1 or Table 2, up to four comma-separated justifications will be allowed for each entry, at the submitter's discretion. Justifications do not need to be explicitly associated with subject, relation or object. Each justification will include a document ID, followed by a colon, followed by two dash-separated offsets. The offsets that show the provenance of an extracted relation are used to narrow the assessor's focus within the documents when assessing the correctness of that relation. Provenance for a single relation may be drawn from more than one document. For the KB variant, when selecting a canonical mention for presentation to the assessor,

the first document appearing in the relation provenance that contains a mention of the object will be used for the canonical mention. (At least one of the documents in the KB's relation provenance must contain a mention of the object of the relation; that document must therefore contain a canonical mention for the object.) String-valued slots (from Table 3) whose values do not represent entities, place an additional constraint on provenance for Knowledge Base variant participants: the first justification must represent the document ID and offsets of the string fill. (Slot Filling variant participants are already providing this information in Column 7 of their submissions.) This requirement will allow assessors to quickly see the text from which the string fill was extracted.

Unlike entries for Slot Filling relations, the `mention` and `canonical_mention` predicates will have only a single justification, representing the exact location of the mention in the text. The `type` predicate requires no provenance.

### Confidence Measure

To promote research into probabilistic knowledge bases and confidence estimation, each triple or slot fill may have an associated confidence score. Confidence scores will not be used for any official TAC 2015 measure. However, the scoring system may produce additional measures if confidence scores are included. For these measures, confidence scores will be used to induce a total order over the facts being evaluated (ties are broken when two scores are equal by assuming that the assertion appearing earlier in the submission has a higher score). Any submitted confidence score must be a positive real number between 0.0 (exclusive, representing the lowest confidence) and 1.0 (inclusive, representing the highest confidence), and must include a decimal point (no commas, please) to clearly distinguish it from a document offset. Confidence scores, if present, will appear at the end of each output line, separated from the provenance information with a tab. Confidence scores may not be used to qualify two incompatible fills for a single slot; submitter systems must decide amongst such possibilities and submit only one. For example, if the system believes that Bart's only sibling is Lisa with confidence 0.7 and Milhouse with confidence 0.3, it should submit only one of these possibilities. If both are submitted, it will be interpreted as Bart having two siblings.

### Comments

Output files may contain comments, which begin at any occurrence of a pound sign (#) and continue through (but do not include) the end of the line. Comments and blank lines will be ignored. The first line of a KB variant or ED variant output file must contain the unique run ID (i.e., it may not be blank). Submitters may like to add a comment to this line giving further details about the run.

### Examples

The following four lines from a Knowledge Base variant submission[9] show examples of a triple without any annotations, one with only mention extent, one with only provenance annotation, and one with both provenance and confidence annotations.

```
:e4    type          PER
:e4    mention       "Bart Simpson"      Doc726:37-48
:e4    per:siblings  :e7    Doc124:283-288,Doc885:173-179,Doc885:274-281
:e4    per:age       "10"   Doc124:180-181,Doc885:173-179    0.9
```

[9] The first two entries might also be found in an Entity Discovery variant submission.

Here is an example line from a Slot Filling variant submission:

```
Q4 org:city_of_headquarters myrun doc42:3-8,doc8:3-11 Baltimore GPE doc8:3-11 1.0
```

## Differences between 2014 Slot Filling and the 2015 Cold Start Slot Filling Variant

Slot filling systems that participated in the 2014 Slot Filling task will need to handle the following differences to successfully participate in the 2015 CSSF task:

- Only the slot specified by the <slot> entry is to be filled; all other slots should be ignored. The <slot> entry is added to the queries received from NIST by running the CS-GenerateQueries.pl script.
- Participants will need to do one round of slot filling, run the CS-GenerateQueries.pl script to create the second round queries, then run slot filling again on the new queries. The results of rounds one and two are to be concatenated before submission using the CS-PackageOutput.pl script.
- CSSF requires that participants be able to fill all slots in both directions. For example, the 2014 Slot Filling task required detection of the per:cities_of_residence slot. CSSF also requires systems to detect the inverse of that slot, gpe:residents_of_city.
- Each slot filler must be assigned a type, selected from {PER, ORG, GPE, STRING}. This field represents an additional output column not found in the 2014 Slot Filling or CSSF tasks.
- NIL entries, indicating that no information about a particular slot is available, are not required in CSSF.

## Evaluation

### Assessment – Knowledge Base and Slot Filling Variants

Cold Start 2015 assessment and scoring will proceed as follows. The results for each assessment query (from both applicable task variants and from human-generated results created as the evaluation queries are created) will be pooled, and each response will be assessed by a person. The result of following the first relation will be assessed as if it were a Slot Filling query (the canonical name of the object entity in the first supporting document that mentions that entity will be used for the slot fill for Knowledge Base variant entries). The second relation in the query will also be assessed as a Slot Filling query, but only if the fill for the first relation is correct. ***If the fill for the first relation is not correct, each fill for the second relation is automatically counted as Wrong.*** For example, if the query asks for the ages of the siblings of "Bart Simpson," and the submitted knowledge base gives "Lisa age 8" and "Milhouse age 10" as siblings, then only the reported age of Lisa will be assessed (Milhouse is not Bart's sibling), and the reported age of Millhouse will automatically be counted as Wrong.

Cold Start uses *pseudo-slot* scoring to evaluate multi-hop queries, in which each evaluation query is treated as if it selects a single indivisible slot. For example, an evaluation query that asks for the children of the siblings of an entity will be scored as if it were a query about a virtual per:nieces_and_nephews slot.[10] The Slot Filling guidelines specify whether each of the component slots of a pseudo-slot is single-valued (*e.g.,* per:date_of_birth) or list-valued (*e.g.,*

[10] A pseudo-slot is similar to the concept of a *role chain,* which is supported by some knowledge representation systems based on description logic, including OWL 2.

per:employee_of, per:children). A pseudo slot is single-valued if each of its component slots is single-valued, and list-valued otherwise. In contrast to the Slot Filling task, Knowledge Base variant submissions may contain multiple fills for single-valued slots. If such are present in the submission, LDC will assess the slot fill with the highest confidence value, and will assess additional slot fills if resources allow. If more than one such slot fill shares the same confidence value, the slot fill that appears earlier in the output will be considered to have higher confidence.

As with the Slot Filling task, the object of each component relation that makes up a single evaluation query response is rated as correct, inexact, or wrong. Pseudo-slots will be scored just as slots in the Slot Filling task, with the additional constraint that both the slot fill and the path leading to that fill must be correct for the entirety to be judged correct. To receive credit for identifying Maggie Simpson as Patty Bouvier's niece, the knowledge base must not only include Maggie as the slot fill, but must also represent Maggie as Marge's child, and Marge as Patty's sibling:[11]

| | |
|---|---|
| **Evaluation query:** | Nieces and nephews of Patty Bouvier (per:siblings, per:children) |
| **Ground Truth:** | :PattyBouvier per:siblings :MargeSimpson |
| | :MargeSimpson per:children :MaggieSimpson |
| **Submission:** | :PattyBouvier per:siblings :MargeSimpson |
| | :MargeSimpson per:children :MaggieSimpson ⇒ **correct** |

A KB that indicated that Maggie was Patty's niece because she was Patty's sister Selma's child would be scored as incorrect:

| | |
|---|---|
| **Evaluation query:** | Nieces and nephews of Patty Bouvier (per:siblings, per:children) |
| **Ground Truth:** | :PattyBouvier per:siblings :MargeSimpson |
| | :MargeSimpson per:children :MaggieSimpson |
| **Submission:** | :PattyBouvier per:siblings :SelmaBouvier |
| | :SelmaBouvier per:children :MaggieSimpson ⇒ **incorrect** |

A response is inexact if it either includes only a part of the correct answer or includes the correct answer plus extraneous material. No credit is given for inexact answers:

| | |
|---|---|
| **Evaluation query:** | Titles of parents of Bart Simpson (per:parents, per:title) |
| **Ground Truth:** | :BartSimpson per:parents :HomerSimpson |
| | :HomerSimpson per:title "Attack-dog trainer" |
| **Submission:** | :BartSimpson per:parents :HomerSimpson |
| | :HomerSimpson per:title "dog trainer Pitiless Pup" ⇒ **inexact** |

In addition, the object of the *final* relation in a pseudo-slot may be rated as redundant if it is equivalent to another fill for the pseudo-slot. No credit is given for redundant answers:

| | |
|---|---|
| **Evaluation query:** | Nieces and nephews of Patty Bouvier (per:siblings, per:children) |
| **Ground Truth:** | :PattyBouvier per:siblings :MargeSimpson |
| | :MargeSimpson per:children :MaggieSimpson |
| | :MaggieSimpson per:alternate_names "Margaret Simpson" |
| **Submission:** | :PattyBouvier per:siblings :MargeSimpson |
| | :MargeSimpson per:children :MaggieSimpson ⇒ **correct** |
| | :MargeSimpson per:children :MargaretSimpson ⇒ **redundant** |

However, objects of relations other than the final relation will never be rated as redundant:

[11] In each of these examples, only the subject, predicate and object are shown, and only a subset of the relevant knowledge base is presented. Each entity is named after the mention that gave rise to it.

**Evaluation query:** Nieces and nephews of Patty Bouvier (`per:siblings`, `per:children`)

**Ground Truth:**
```
:PattyBouvier per:siblings :MargeSimpson
:MargeSimpson per:children :LisaSimpson
:MargeSimpson per:children :BartSimpson
:MargeSimpson per:alternate_names "Marjorie Simpson"
```

**Submission:**
```
:PattyBouvier per:siblings :MargeSimpson
:PattyBouvier per:siblings :MarjorieSimpson
:MargeSimpson per:children :LisaSimpson ⇒ correct
:MarjorieSimpson per:children :BartSimpson ⇒ correct
```

Here, Marge Simpson and Marjorie Simpson represent the same person in the ground truth, but two distinct entities in the KB. However, because the query is about Marge's children and not about Marge herself, both responses to the evaluation query are assessed as correct.

Since in Cold Start the facts being evaluated come from sequences of triples, confidence scores would need to be combined if we wanted to generate confidence scores for a derived pseudo-relation. The proper way to combine scores of course depends on the meaning of those scores, and for now, Cold Start is not mandating any particular meaning. Three general score combination functions are min, max and product; we welcome comments from the community on which combination methods to report.

### Scoring – Knowledge Base and Slot Filling Variants

Given the above approach to assessment, basic scoring for a given evaluation query proceeds as follows:

**Correct** = total number of system output pseudo-slots judged correct

**System** = total number of system output pseudo-slots

**Reference** = number of single-valued pseudo-slots with a correct response + number of equivalence classes[12] for all list-valued pseudo-slots

**Recall** = Correct / Reference

**Precision** = Correct / System

$\mathbf{F_1}$ = 2 * Precision * Recall / (Precision + Recall)

The $F_1$ score is the primary metric for the 2015 Cold Start Knowledge Base Population system evaluation.

As in 2013, each evaluation query in 2015 may have more than one instantiation (which will appear as separate queries in the query set). Each such instantiation will have the same relations, but will have different entry points. To give equal weight to each evaluation query, the 'Correct' score for a single query will be the average of the 'Correct' scores for each of its entry points. Put another way, evaluation queries will be macro-averaged across the variants.

### Scoring – Entity Discovery Variant

The scoring for the Entity Discovery variant will be identical to scoring for the 2015 TAC Trilingual Entity Discovery and Linking task, with the exception that no linking to an existing knowledge base

[12] See *TAC KBP 2015 Slot Descriptions* and *TAC KBP 2015 Assessment Guidelines* for further information on how and when two slot fills are treated as equivalent.

is required (that is, all mentions will be treated as NIL entries). Please see *TAC KBP2015 Entity Discovery and Linking Task Description* for complete details on scoring.

## Submissions

A two-week window from Monday August 3 to Monday August 17 will be available for downloading the Cold Start data, producing and submitting results. Systems should not be modified once the corpus has been downloaded. Participants may make up to five submissions, ranked in order of evaluation preference. The top-ranked submission must be made as a 'closed' system; in particular, it must not access the Web during the evaluation period. All submissions must obey the following external resource restrictions:

- Structured knowledge bases (e.g., Wikipedia infoboxes, DBPedia, Freebase) may not be used to directly fill slots or directly validate candidate slot fillers.
- Structured knowledge base entries for target entities may not be edited, either during, or after the evaluation.

In addition, because Cold Start focuses on the condition where the knowledge base is initially empty, we ask that each participating site submit at least one run that consults external entity knowledge bases only after entities and relations have been extracted from the document collection. The number of submissions actually judged will depend upon resources available to NIST. Details about submission procedures will be communicated to the track mailing list. Tools to validate formats will be available on the TAC Web site.

If there is a desire among participants in the Trilingual Entity Discovery and Linking task, we will add a second submission deadline in early October to allow participants' final EDL systems to be applied to the CSED task. Please indicate your interest in this option by sending mail to the TAC KBP mailing list (tac-kbp@nist.gov).

### Sample Collection

A sample Cold Start collection will be available from the NIST Web site (`http://www.nist.gov/tac/2015/KBP/ColdStart/`) shortly. Note that this is not a training collection; it serves only to illustrate the various facets of the task and the evaluation. The sample will include:

- A file describing the collection (`README.txt`).
- A document collection, comprising seventeen documents drawn from the domain of *The Simpsons* television show. Each `<DOC>` tag includes the original Web source, of which the text in the collection is a snippet.
- A KB created from the collection. Note that a reference KB will not be created for the actual Cold Start task.
- A set of sample submissions for the KB variant, including a variety of errors.
- A sample set of evaluation queries for the Slot Filling variant.
- A sample Slot Filling variant submission.
- A sample Entity Discovery variant submission.

## Change History

- Version 1.0
  - Original version, based on the 2014 specification
  - Added Entity Discovery variant
- Version 1.1
  - Rewritten for clarity
  - Corrected description of the evaluation document collection for 2015