TAC KBP 2014 Assessment

Version 0.1 – August 20, 2014

Linguistic Data Consortium

Created by: Joe Ellis (joellis@ldc.upenn.edu)

With contributions from: Jeremy Getman, Kira Griffitt, & Heather Simpson

http://projects.ldc.upenn.edu/kbp/

Changes from 2013:

1. Updated Section 6 to reflect changes to justification specification.

2. Removed special case guidelines referring to Temporal Slot Filling.

3. Expanded Section 5.1 to explain indirectly supported filler namestrings.

4. Expanded Section 7 to explain that using outside knowledge is acceptable for disambiguation of namestrings involved in justification.

Table of Contents

1 Introduction

Text Analysis Conference (TAC) is a series of workshops organized by the National Institute of Standards and Technology (NIST). TAC was developed to encourage research in natural language processing (NLP) and related applications by providing a large test collection, common evaluation procedures, and a forum for researchers to share their results. The Knowledge Base Population (KBP) track of TAC aims to develop systems that can determine whether or not entities have an existing Wikipedia page, extract information about those entities from web and newswire texts, and use the extracted information to populate an existing knowledge base.

In the Slot Filling task, the first KBP task for which these guidelines were developed, performing systems search a corpus for information about various entities and add any new information to respective infoboxes from a 2008 snapshot of Wikipedia. However, in 2014, there are two KBP tasks that require assessment of system output using these guidelines – regular Slot Filling and Cold Start.

There are two parts to the assessment task. Primarily, you will be judging the validity of the responses (**fillers**) and the justifications for them provided during the various KBP tasks (Slot Filling, Temporal Slot Filling, and Cold Start). Secondly, you will group together all of the correct, co-referring fillers into **equivalence classes** in order to arrive at a final number of correct and unique responses for each slot per entity, an essential component for scoring system output.

This document will guide you in the assessment of fillers and justifications and in the creation of equivalence classes. Note, however, that in order to correctly complete this task, you will also need a copy of *TAC KBP 2014 Slot Descriptions*, the document which details the subset of Wikipedia infobox slots that systems attempted to fill. Before beginning the assessment task, you must familiarize yourself with all of the 41 possible slots (25 for person (PER) entities and 16 for organization (ORG) entities) as they are described in *TAC KBP 2014 Slot Descriptions*.

However, while you are judging system responses, keep in mind that a filler is generally correct if it is supported by the document from which it was extracted and it meets the requirements for its respective slot as described in *TAC KBP 2014 Slot Descriptions*. While you are assessing system responses, it is also helpful to remember that the basic system task is to add information to a Wikipedia infobox. Keeping the basic task in mind is helpful because, if it is ever unclear whether a filler meets the description of its respective slot, you can ask yourself whether it would be appropriate for inclusion in a Wikipedia infobox.

Sections 2 - 7 of this document provide detailed guidance on how to use various pieces of information to assess system responses, the first stage of the assessment task. Section 8 provides guidelines on clustering correct responses into equivalence classes, the second stage of assessment.

2 Slot Content

Each of the TAC KBP 2014 Slot Descriptions are classified as name slots, value slots, or string slots based on the content of their fillers. In addition to classifying the slots, however, the content distinction also serves to guide the assessment of fillers, as detailed below.

2.1 Name slots

Fillers for name slots are required to be names, usually that of a person, organization, or geopolitical entity. Although adjectival forms of names are acceptable (e.g., “American”, “Christian”), you should reject any fillers that are clearly not names, for instance:

```
per:children = five            WRONG - not a name
per:spouse = his wife          WRONG - not a name
```

In some cases, systems may return extraneous or incomplete text strings as part of the names that constitute a filler. Following standard practice for Wikipedia infoboxes, fillers that include articles (the, a, & an), titles (Dr., Ms., etc.), or nominal modifiers are acceptable, though not preferred. However, as in the last example below, systems will sometimes include too much extraneous text for a filler, in which case the response should be marked ‘wrong’ or ‘inexact’ (see Section 2.4 for details on selecting the appropriate filler assessment category).

Text Excerpt	**Acceptable Filler?**
the Department of State	acceptable
Department of State	preferred
Rev. Al Sharpton	acceptable
Al Sharpton	preferred
Republican	acceptable
Republican Party	preferred
coach Joe Gibbs	acceptable
Joe Gibbs	preferred
city of Baltimore	acceptable
Baltimore	preferred
the singer-songwriter Hank Williams	acceptable
Hank Williams	preferred
the singer-songwriter Hank Williams who had a string of top hits	unacceptable

2.1.1 Normalizations of Name-Slot Fillers

Occasionally, systems will interpret and edit text strings to the most appropriate forms for Wikipedia pages, so some fillers might include normalizations of the text that must be assessed as ‘Correct’ or ‘Wrong’. For example, if your assigned entity was “John Doe” and you found a document containing the text "John Doe’s first wife, Ruth", then "Ruth Doe" could be assessed as a correct filler normalization for *per:spouse*, even though that exact string does not appear in the reference document. Although it is possible that Ruth might not have taken the last name of her husband at the time of their marriage, it is reasonable to assume that she did as long as there is no other information in the document indicating that this may not be the case.

Edited filler text may also be returned if an answer found in a document is correct but the form of the word is unnatural sounding as a knowledge base answer, as is often the case with the adjectival forms of GPE names (e.g., American, Texan, British, etc.). In the table of examples

below, note that the edited fillers for *per:country_of_birth* and *org:stateorprovince_of_headquarters* and the unedited text for *per:origin* are preferred. This is because the adjectival form of the word is more appropriate for the *per:origin* slot. However, both the adjectival and nominal responses should be assessed as correct for all of the slots

Also in the examples below, note that it would not be correct for “The Big Apple” to be edited to the correct name of the GPE it refers to, “New York City”. This is because, absent any additional information in the source document, making the connection between the two names for the city would be unsupported:

Slot	Document Context	Correct Answer from Document	Correct Edited Text
per:city_of_death	He passed away last year in his favorite city, the Big Apple	The Big Apple	n/a
per:origin	He is American-born	American	n/a
per:country_of_birth	He is American-born	American	America
org:stateorprovince_of_headquarters	The Texan band	Texan	Texas

2.2 Value Slots

Value slots are required to be filled by either a number or a date. Number fillers for these slots can be spelled out ("five thousand") or written as a number ("5000") but you should reject any answers that are not values or that cannot be resolved to a value, for instance:

```
org:date_founded = before he moved      WRONG - not a value
                   to this country
```

Keep in mind that valid date fillers will be provided in many different formats, not all of which look like numbers. For instance, if a document states that the assigned person entity was born on "New Year's Day 1985", the filler "New Year's Day 1985" would be acceptable for the *per:date_of_birth* slot.

2.2.1 Normalizations of Value-Slot Fillers

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).

Full date normalizations are usually calculated using document text and the date on which the document was published or posted. Publication dates for news articles and some web articles will be contained in the Doc ID; however, some web articles will only have a post date contained in the document within the POSTDATE tags. Consider the following examples,

each of which provides two different, correct fillers for *per:date_of_death* for the entity Wesley Posvar (the normalization on the right would be assessed as 'Correct'). Assume for each of these examples that the Doc ID is NYT_ENG_20010802.0034.LDC2007T07 (the first string of numbers '20010802' indicates that the document was published on August 2nd, 2001):

```
Wesley W. Posvar, former chancellor of the University of Pittsburgh,
died on July 27.
```

per:date_of_death – July 27 Normalization – 2001-07-27

Since the text above states that Posvar "died on July 27", the normalization "2001-07-27" would be assessed as correct. Even though the document does not explicitly state that Posvar died in 2001, the year can be reasonably inferred because the article was published on August 2nd of 2001.

```
Wesley W. Posvar, former chancellor of the University of Pittsburgh,
died on Thursday.
```

per:date_of_death – Thursday Normalization – 2001-08-02

Unless stated otherwise in the article, systems can infer that Posvar's death in the above example took place on the Thursday closest to, but not past, the article's publication date. You can check the resolution by referencing a 2001 calendar (either online or using the *cal* command in a UNIX terminal), which shows that 2001-08-02 was actually a Thursday itself, making it the Thursday closest to, but not past, the article's publication date.

```
Wesley W. Posvar, former chancellor of the University of Pittsburgh,
died last year.
```

per:date_of_death – last year Normalization – 2000-XX-XX

Since the above example states that Posvar died "last year" and the article was published in 2001, it is reasonable for systems to assume that the entity died in the year 2000 as long as no conflicting information is provided in the document. Also, note that, since no information is provided on the day or month of Posvar's death, only the year can be correctly inferred.

```
Wesley W. Posvar, former chancellor of the University of Pittsburgh,
died a few years ago.
```

per:date_of_death – n/a Normalization – n/a

In this last example, the text is not informative enough to determine when Posvar died. As a result, neither "a few years ago" nor any resolution drawn from it would be a correct filler.

2.3 String Slots

String slots are basically a "catch all", meaning that their fillers cannot be neatly classified as names or values. The text excerpts (or "strings") that make up these fillers can sometimes be just a name, but are often expected to be more than a name.

3 Slot Quantity

3.1 Single-value

Systems were only supposed to provide a single-filler for all single-value slots. However, since multiple teams participate in the slot filling task and contradictory information could be present in the corpus, it is possible to have more than one valid filler for any single-value slot. Be that as it may, the possibility of multiple, correct, single-value fillers is less likely for some slots (e.g., *per:date_of_birth*), and more likely for others (e.g., *per:age, per:religion* or *org:website*).

3.2 List Value

Systems were allowed to provide multiple fillers for list-value slots because, for instance, an assigned person entity might have multiple children or have been employed by multiple organizations. Note that list-value slots do not *require* multiple answers, but multiples are permitted.

4 TAC KBP 2014 Slot Descriptions Table

Although these guidelines do not include the slot descriptions, the following table of all 41 slots is provided for reference:

Type	Slot Name	Content	Quantity
PER	*per:alternate_names*	Name	List
PER	*per:children*	Name	List
PER	*per:cities_of_residence*	Name	List
PER	*per:city_of_birth*	Name	Single
PER	*per:city_of_death*	Name	Single
PER	*per:countries_of_residence*	Name	List
PER	*per:country_of_birth*	Name	Single
PER	*per:country_of_death*	Name	Single
PER	*per:employee_or_member_of*	Name	List
PER	*per:origin*	Name	List
PER	*per:other_family*	Name	List
PER	*per:parents*	Name	List
PER	*per:schools_attended*	Name	List
PER	*per:siblings*	Name	List
PER	*per:spouse*	Name	List
PER	*per:stateorprovince_of_birth*	Name	Single
PER	*per:stateorprovince_of_death*	Name	Single
PER	*per:statesorprovinces_of_residence*	Name	List
PER	*per:age*	Value	Single
PER	*per:date_of_birth*	Value	Single
PER	*per:date_of_death*	Value	Single
PER	*per:cause_of_death*	String	Single
PER	*per:charges*	String	List

PER	*per:religion*	String	Single
PER	*per:title*	String	List
ORG	*org:alternate_names*	Name	List
ORG	*org:city_of_headquarters*	Name	Single
ORG	*org:country_of_headquarters*	Name	Single
ORG	*org:founded_by*	Name	List
ORG	*org:member_of*	Name	List
ORG	*org:members*	Name	List
ORG	*org:parents*	Name	List
ORG	*org:political_religious_affiliation*	Name	List
ORG	*org:shareholders*	Name	List
ORG	*org:stateorprovince_of_headquarters*	Name	Single
ORG	*org:subsidiaries*	Name	List
ORG	*org:top_members_employees*	Name	List
ORG	*org:date_dissolved*	Value	Single
ORG	*org:date_founded*	Value	Single
ORG	*org:number_of_employees_members*	Value	Single
ORG	*org:website*	String	Single

5 Filler Assessment Categories

In the first task of slot filling assessment, you will mark each filler provided by systems as being ‘Correct’, ‘Wrong’, or ‘Inexact’.

5.1 Correct Fillers

Fillers must meet two requirements in order to be judged as correct. Primarily, all answers must meet the requirements of their respective slots as described in the document *TAC KBP 2014 Slot Descriptions*. Secondly, all fillers must be supported in the provided predicate justification strings or their surrounding context (1-2 sentences in either direction) (see section 6 for more information on justification strings). If a filler cannot be justified solely by the justification strings or their surrounding context, it should not be labeled as correct, even if you know it to be true because of an outside information source (for more information on the appropriate use of outside information sources in assessment, see section 7 – *Using Outside Knowledge Sources*).

Keep in mind that systems will attempt to return the fullest, most informative namestring available for each filler entity. In some cases the namestring returned as the filler may not be directly supported by the predicate justification strings. For instance, for the relation <Michelle Obama – *per:spouse*> the justification supporting a filler might be, simply:

```
He married Michelle in 1992.
```

However, the namestring returned as the filler is “Barack Obama”. In such a case it is your job to determine that “Barack Obama” is the entity to which “He” refers in the above justification. Assuming the two mentions refer to the same entity, “Barack Obama” would be a correct response.

5.2 Wrong Fillers

There are two ways in which slot fillers can be simply wrong. Primarily, all answers must meet the requirements of their respective slots as described in the document *TAC KBP 2014 Slot Descriptions*. As a result, any fillers that do not meet the requirements of their respective slots are wrong.

Secondly, all fillers must be supported in the provided predicate justification strings or their surrounding context. If a filler cannot be justified solely by the justification strings from which it was selected, it is wrong, even if you know it to be correct because of an outside information source. For example, if "William J. Clinton" was provided as a filler from a document that only contained mentions of "Bill Clinton", "William J. Clinton" would be marked 'Wrong'. (For more information on the appropriate use of outside information sources in assessment, see section 7 – *Using Outside Knowledge Sources*).

5.3 Inexact Fillers

A slot filler should be judged as inexact if it meets both of the standards for correct fillers (i.e., it is supported in its provided predicate justification and fulfills the requirements of its respective slot) but the string of text selected is incomplete, includes extraneous text, or is not the most informative text string in the document that refers to the filler entity. For example, given the entity 'Michelle Obama' and a source document with the text:

```
Barack Hussein Obama is the U.S. President. He was elected in November
2008. Obama and his wife, Michelle, have two daughters…
```

the correct filler for *per:spouse* would be "Barack Hussein Obama". The text excerpt "Obama" would be an inexact filler because "Barack Hussein Obama" would be the most informative answer in the document. Note that source documents from the web will occasionally contain entities' names in strange or informal formats. Such text excerpts are acceptable slot fillers and would only be inexact if another, more correct or informative version of the name appeared elsewhere in the document. Here are some additional examples:

Slot	**Document Text**	**System Answers**	**Assessment**
per:siblings	`His sister Emily…` (no other mention of Emily in document)	Emily	Correct
per:siblings	`Emily Smith, his adopted sister, was quick to support him. Emily has run a foundation for …`	Emily	Inexact ("Emily Smith" would be the correct filler)
org:founded_by	`Microsoft Founder Bill Gates is one of the world's most famous billionaires. Gates started his empire…`	Microsoft Founder Bill Gates	Inexact ("Bill Gates" would be the correct filler)
per:employee_or_member_of	`Blanton was a first-round pick for the A's in the 2002 entry draft. Oakland received the pick from the New`	Oakland	Inexact ("A's" would be the correct filler)

	York Yankees as compensation for the loss of Jason Giambi		

6 Justification Assessment Categories

Justification is the strings of text that prove a relation to be true. There is a minimum of one and as many as four “predicate strings” comprising the justification of a relation.

6.1 Correct Predicate Strings

Predicate strings are the strings of text that prove the relation itself. Ideally, these strings alone will give you all the information you need to connect the query entity to the filler via the respective slot.

Up to four predicate strings can be provided and each string can contain up to 150 characters. If multiple predicate strings are provided, they may be discontiguous in the source document and/or may be taken from different source documents.

Predicate strings must include enough information to link the query entity to the filler by the chosen slot while not containing too much extraneous text. If only one predicate justification string is provided, it alone must contain a mention of the query entity and a mention of the filler entity, as well as some string of text justifying the connecting slot:

Slot	**System Answer**	**Predicate String**
per:spouse	Bill Clinton	Hillary is married to Bill Clinton.

However, note that, in some cases a correct predicate justification string might not include all of this information. Informal source data, such as discussion forum documents, may not provide all of the entities involved in a relation outside of the document metadata. For example:

Slot	**Answer**	**Predicate Strings**
per:cities_of_residence	Minneapolis	Just moved to Minneapolis
		<post author="tango" datetime..>

Assuming the PER entity involved in the above relation can be determined (in this case, the author of the statement), then the above predicate string is correct, despite not containing a mention of the PER. An additional predicate string would simply need to be returned containing a mention of the PER entity implied by the above predicate string.

In many instances, concrete justification for a relation can be provided with multiple discontiguous predicate strings. For instance, given the query entity ‘Apple, Inc.’, slot *org:founded_by*, filler “Steve Jobs”, and the following text:

```
Apple, Inc. was founded on April 1, 1976. In its beginning, the company sold
computer kits hand-built by Steve Wozniak. Wozniak was one of three founders,
```

```
along with Steve Jobs and Ronald Wayne.
```

the two predicate strings provided would be the first sentence and the third sentence. Together these two strings support all aspects of the relation in question. The text intervening the two sentences is irrelevant to the *org:founded_by* relation and is thus excluded.

6.2 Wrong Predicate Strings

A predicate string is wrong if it does not provide any information necessary to link the query entity to the filler by the chosen slot.

6.3 Inexact-Short Predicate Strings

Predicate strings that contain part, but not all, of the information necessary to link the query entity to the filler by the chosen slot are considered inexact-short.

6.4 Inexact-Long Predicate Strings

Predicate strings that contain all of the information necessary to link the query entity to the filler by the chosen slot but also include an unacceptable amount of extraneous text are considered inexact-long.

6.5 Particular Cases

The following slots are given separate treatment with regard to justification.

per:alternate_names, org:alternate_names

It is possible to provide correct fillers for the *{per,org}:alternate_names* slots without any contextual information. While such contextual information may sometimes occur, it is not required. For example, “IBM” is a correct alternate name for “International Business Machines” solely based on the fact that the former is an acronym for the latter and they appear in the same document. In these situations, the justification must contain sentences that mention the names used to extract the filler. For the above example, the justification should contain two sentences, one mentioning “International Business Machines” and one mentioning “IBM”.

per:title

Titles that represent positions at different organizations are considered distinct fillers. For example, “Mitt Romney” has held three different “CEO” positions:

```
Mitt Romney, who was CEO of Bain & Company from 1991 to 1992; CEO of
the 2002 Winter Olympics Organizing Committee from 1999 to 2002; and
CEO of Bain Capital from 1984 to 2002, was the Republican Party’s
candidate in the 2012 presidential elections.
```

These positions are considered as three distinct, valid fillers since each refers to a position at a different organization. While they would not be included in the strings of text selected as the fillers, the corresponding organizations should be included in the predicate strings. So for the query entity “Mitt Romney”, a correct predicate string for the filler “CEO” would be, for example:

```
Mitt Romney, who was CEO of Bain & Company
```

instead of only (from the same text extent):

```
Mitt Romney, who was CEO
```

Even though the company name is not necessary to know that Mitt Romney had the title of CEO, the organization name is included to help disambiguate instances of “CEO” that are unique (though the strings are identical).

It is important to remember, however, that if no organization is mentioned in connection to a *per:title* filler, then none need be provided within the predicate string. In other words, this doesn’t apply to occupations that have no clear affiliation (e.g., “actor”, “star”) or to positions where the affiliation is missing.

7 Using Outside Knowledge Sources

Occasionally, you will learn or already know about a slot filler for the targeted entity through online searching, your own knowledge, and/or your entity’s reference document. Note that you **cannot** use this information to assess a filler without first checking whether the answer is supported in the context surrounding the justification. Remember, slot fillers are only correct if they can be justified solely by the surrounding context (1-2 sentences in either direction) of the provided justification strings.

For example, you might already be aware that Michelle Obama is President Obama’s wife and so, if “Barack Obama” were your assigned entity, and “Michelle Obama” was listed as a filler for *per:spouse*, you might be tempted to just mark it as correct. However, if the justification from which the text string “Michelle Obama” was extracted did not include any lexical clues to indicate that she was a spouse of the targeted entity (e.g., “marriage”, “wife”, “first lady”, etc.), the filler **must** be marked as ‘Wrong’.

Another common temptation for using world knowledge involves extension of geo-political entities for residence, birth, death, or headquarters slots. For example, if you had already labeled “Texas” as a correct filler for *per:stateorprovince_of_birth* for your assigned entity, “The United States” could not automatically be labeled as a correct filler for *per:country_of_birth* unless its justification clearly indicated that Texas was located within the United States or, more simply, that the entity resided in the United States.

Conversely, if you have to make a judgment on a filler that you know to be incorrect, **do not** mark it wrong without first checking to see if it is justified in the justification. For example, if “George Bush” was returned as a filler for *per:spouse* for Barack Obama and the predicate justification stated that “Barack Obama married George W. Bush in 2008” without any additional information in the surrounding context to indicate that the statement was figurative or untrue, the filler should be marked correct.

Although you cannot generally rely on outside knowledge to assess slot fillers, you can use it to clarify whether appropriate slots were selected for supported fillers. For example, imagine that the targeted entity “Abdurrahman Wahid” had “South Jakarta” provided as a filler for *per:cities_of_residence* and the predicate justification stated that “Abdurrahman returned to his house in Cilandak, South Jakarta, Indonesia”. While this sentence clearly supports one or more residence relations between Wahid and the named locations, you might not know the

geo-political level of Cilandak or South Jakarta. In such a case, it would be acceptable to perform an online search, which would reveal that South Jakarta is a city (and that the filler was correct) and that Cilandak is a subdistrict of the city (and, thereby, wrong for any city slot).

Along these same lines, consider the following case, where the relation in question is <Mary Todd Lincoln – *per:stateorprovince_of_death*>, the filler "Illinois" and the following two sentences returned as predicate justification:

"Mary Todd Lincoln died in Springfield."

"The Reisch Beer brewery was located in Springfield, Illinois."

Performing a quick online search reveals that the Springfield where Mary Todd Lincoln died is the same Springfield where the Reisch Beer brewery is located. With this knowledge, the above justification and filler can be judged as correct. The first predicate string alone would not be enough to prove that Illinois is Mary Todd Lincoln's state of death, because this would require the knowledge that Springfield exists within Illinois, which is not supported by the document. Adding the second predicate string, however, justifies there being a Springfield in Illinois. The online search simply gives you the knowledge that "Springfield" from the first predicate string and "Springfield, Illinois" from the second predicate string are the same entity. Using outside knowledge to disambiguate a reference like "Springfield" is acceptable, but using outside knowledge to add additional information not present in the justification (that Springfield is in Illinois, for instance) is not acceptable.

8 Creating Equivalence Classes

Throughout the corpus, all entities mentioned could be referred to by many different names (e.g. "Hillary Rodham Clinton" might be referred to as "Hillary", "Hillary Clinton", "Senator Clinton", "Secretary of State Clinton", etc.). As any of these names could have been marked correct during the first phase of assessment, your job in the second stage is to identify these coreferential fillers for each slot and cluster them together into equivalence classes (in the preceding example, all the different names for "Hillary Rodham Clinton" would be grouped together into a single entity equivalence class). This step is necessary because it provides a total number of correct **and** unique answers per slot for each entity and because it indicates whether systems returned any redundant fillers. Note that, in order for two fillers to be considered coreferential, they must refer to the same entity; they **cannot** be simply related. Consider the following *org:country_of_headquarters* examples for the entity FirstGroup:

"Britain's biggest bus firm FirstGroup..."

"UK's FirstGroup agrees to buy Laidlaw"

Given the text, both "Britain" and "UK" would be valid fillers for *org:country_of_headquarters*. Since the UK operates as a "country of countries", which includes Britain, it is likely that both of these fillers are referring to the same location of FirstGroup's headquarters. However, since the UK and Britain are not strictly the same entity, the two fillers should occupy separate equivalence classes. Note though that, following the guidelines for slots such as *per:origin*

and *per:country_of_birth*, there could be cases in which you would place adjectives and nouns into the same equivalence classes (e.g. “French” and “France”).

Because any slot could potentially have more than one correct answer (particularly list-value slots), you will likely have to create multiple equivalence classes, one for each unique entity. For instance, the entity “Michael Jackson” might contain the following correct fillers for the *per:children* slot:

```
Prince Jr.                          "Blanket" Jackson
Prince Michael Jackson, Jr.         Paris
Prince Michael "Blanket"            Prince Michael Jackson II
Jackson II                          Blanket
Paris Katherine Jackson             Prince
Paris Jackson
```

Each of the above names would be connected to a source document and so, after reading each in context and determining who was being referred to, you would be able to create three equivalence classes, one for each distinct entity mentioned:

Class 1	**Class 2**	**Class 3**
Prince Jr.	Paris Katherine Jackson	Prince Michael "Blanket" Jackson II
Prince Michael Jackson, Jr.	Paris Jackson	"Blanket" Jackson
Prince	Paris	Prince Michael Jackson II
		Blanket

While you should primarily rely on information contained in the documents when creating equivalence classes, you may utilize outside information sources to help make your determinations. For instance, if you found that "Blanket" was a nickname for "Prince Michael Jackson II", then you could cluster "Blanket" and "Prince Michael Jackson II" into the same equivalence class, even if the given source documents did not state the information explicitly. Note, however, that if the information contained in the source documents contradicts outside knowledge, you should cluster fillers based on information in the source documents.

8.1 Equivalence Classes for *per:title* Fillers

Fillers for *per:title* present a unique challenge to the process of equivalence class creation because, in addition to determining whether two titles are considered equivalent, assessors must also ascertain whether two or more equivalent titles were held in the same organization before grouping them together into a single equivalence class.
Primarily, you must adhere to the following rules when determining whether similar titles are equivalent:

- Exact or nearly-exact string matches are equivalent (e.g. “chief executive” & “chief executive officer”)
- Acronyms or common abbreviations should be considered equivalent (e.g. “CEO” & “chief executive officer”)

- Common word re-orderings are equivalent (e.g. “Finance Minister” and “Minister of Finance”)
- Nearly synonymous terms should be considered equivalent (e.g. “attorney” and “lawyer” as well as “Premier” and “Prime Minister”)
- Specified and unspecified positions should **not** be considered equivalent (e.g., “prosecutor”, “attorney”, and “U.S. Attorney” would all go into separate equivalence classes).

Once you’ve determined that a set of titles is equivalent, you must find out whether they all were held within the same organization before coreferencing them into a single equivalence class. For example, Mitt Romney has held three different “CEO” positions:

```
CEO, Bain Capital (1984–2002)
CEO, Bain & Company (1991–92)
CEO, 2002 Winter Olympics Organizing Committee (1999–2002)
```

Even though the three titles are exactly the same, each of these responses would be placed into separate equivalence classes because the titles were held in distinct organizations.

If you cannot determine the organization in which one or more equivalent *per:title* fillers were held or there simply is not a coupled organization (as is the case with most occupational references such as “actor”), you should group the unaffiliated responses into a separate equivalence class. For example, if an entity were described as “professor at NYU”, “professor at Berkeley” and simply as “professor”, you would place the three “professor” fillers into three separate equivalence classes – one for the position at NYU, one for the position at Berkeley, and a final one for the unaffiliated position.

9 Steps for Slot Filling Assessment

Sign in to the online assessment tool using the URL and username/password combination provided by your supervisor. Afterward, continue to repeat the following steps for each set of slot fillers for your assigned entity until you get a message stating that you have completed all assessment tasks for your assigned entity:

1. **Review entity reference document**
   When you first open the slot filling tool, the name of your assigned person or organization will appear in the upper-left corner of the tool and a reference document for the entity will be displayed in the right panel. You should review the reference document to get a sense of who or what your entity is before attempting to assess slot fillers for it.

2. **Review slot description**
   Assessment files are formatted so that you will review all of the fillers provided for a particular slot in one kit. Whenever you are given a set of fillers to assess, the relevant slot will be listed in the upper-left corner of the tool, next to the name of your assigned entity. Before attempting to assess the fillers, you should take a moment to review the relevant slot description in *TAC KBP 2014 Slot Descriptions* (see your supervisor if you do not have a copy of this document for reference during the slot filling assessment task).

3. Assess each slot filler and justification

The panel on the left side of the assessment tool contains all of the fillers and justifications provided by systems for your particular slot/entity kit. When you click on one of the fillers, the source document from which it was extracted appears in the center panel of the tool. Read the filler and its justification in the source document (both should be underlined), checking the description of the respective slot from *TAC KBP 2014 Slot Descriptions* if necessary to determine whether the filler is ‘Wrong’, ‘Inexact’, or ‘Correct’. Repeat this step until you have recorded a judgment for all of the fillers and justifications displayed.

After you have recorded a judgment for each filler in the left panel, proceed to the ‘Coreference’ tab.

4. Create equivalence classes

Click on the fillers in the panel on the left side of the tool and read the relevant sections of the source document to determine who or what each string refers to. If two or more fillers refer to the same entity, click the IDs next to their namestrings (they will turn red) and hit enter. This will move these entities into the center column (the 'DONE' column) under one header. In addition to simultaneously coreferencing multiple namestrings, you can also drag and drop a namestring from the 'UNDONE' column to an entity in the 'DONE' column to associate it with the already coreferenced entity.

All of the namestrings from the 'UNDONE' column that refer to the same entity must be coreferenced together in the 'DONE' column. Once all of the namestrings for one entity are coreferenced together, repeat the above process for all other distinct entities. The coreference task is completed when there are no remaining namestrings in the 'UNDONE' column, each entity in the 'DONE' column is associated with namestrings that refer only to that specific entity, and no two groupings of namestrings refer to the same entity.

Note that, if you did not mark any fillers ‘correct’ during assessment and the entity did not have any existing fillers from the knowledge base, there will be nothing for you to do in the coreference tab.