**CNTRO Timeline Library Evaluation:**

CNTRO timeline library is a standalone set of Java APIs that reads a set of CNTRO Events and returns (asserted and inferred) timeline information - chronological sequence of events, duration of an event and duration between two events, for given set of events. One of the challenges was to device a way to determine the accuracy of the results returned by this library. We created a simple framework to evaluate the quality of the results by creating a set of files with ‘expected’ results - ‘gold’ standard, and then comparing the results of APIs against it. This process is explained below:

1. We included a set of annotated OWL files that will be used as input to create timeline for each note.
2. Program writes the ‘in-memory’ event details, asserted and inferred temporal relations and computed timeline information in a comma-separated file. This file has a particular format, which shows information in a 2-dimensional matrix, which can be easily reviewed by humans. This also has string tokens, which help computer program to understand the same information and compare it with other event timeline. And example event matrix in a comma-separated file is shown below:



Figure Event relation matrix to show asserted temporal relations

1. The annotators and curators review this set of files. The files are corrected manually if needed. This set of files becomes read-only set of files, which will be ‘programmatically’ compared with every run which program makes with the same set of files. Any mismatch is recorded and placed in a report file in a separate folder, which can be reviewed by user. The mismatch can occur either due to a bug in the program or incorrect/insufficient annotations.
2. Depending on where issue lies, either program is fixed or annotations are reviewed and corrected to generate a better run.
3. We discussed and found the following changes to the CNTRO ontology and program will be beneficial:
   1. Adding rules to the ontology that defines inherent order among normalized events. These rules might guide sequencing events during timeline computation of events, when there is no other information available. For example ‘Patient Death’ event must always occur at the end – after all other procedural/monitoring events.
   2. It was hard to compute and order two events solely on the base of ‘start’ or ‘finish’ temporal relation when start and end times both are missing in the annotation. Program can only compute and order the events especially when ‘start time’ (for ‘start’ relation) and ‘end time’ (for ‘finish’ relation) is annotated.
   3. ‘Duration’ calculation between two events, also needs to be refined based on granularity and the context provided by the annotators and curator. This implementation and testing is ‘in-progress’ at the time of writing this paper.
4. At the end of last successful run following statistics were computed:

Total Files = 183, Number of failures = 17

Out of these 17 instances following was observed:

1. Failure occurred due to annotation errors = 7
2. Failure occurred due to program errors = 10

Annotations were corrected manually and gold standard files were updated with these changes.

Annotation Errors:

* Event instance label was used to store relation information instead of its object properties.
* OWL “Same As” property was used instead of CNTRO “Equal” temporal association.
* ‘Time Instant’ is instantiated from a different OWL class than CNTRO ontology defined ‘Time Instant’ class. It might have been due to different Protégé or Ontology version, used by annotator. We need to investigate.
* Annotator relied on temporal information stored in patterns like “Next Day” and “3 days later”. Recognition and processing of these type of patterns have not been implemented yet and planned for future releases.
* Sometimes there is not enough information provided through correct annotation to sequence the events unambiguously on a time line. In that case, the computed timeline is agreed on its closest correctness. Annotators need to provide more information.

Program Errors:

* CNTRO Query processing was did not show consistency in processing OWL “Same As” association for all CNTRO Events. This needs to be investigated and fixed.
* Query API needs to process “START” and “FINISH” temporal relations correctly.
* Improve the event timeline computation. It failed for File 233 due to multiple temporal associations specified to compute sequence. This needs to be investigated and fixed.

So,

Mismatches (as compared to Gold Standard) =10

Accuracy (without resolving annotations) = (166 out of 183) = 90.7%

Accuracy (After fixing annotations) = (173 out of 183) = 94.5%

**How to read and interpret the comma separated file for CNTRO Timeline for a given annotation?**

For each note-annotation (OWL) file, CNTRO timeline library creates a comma-separated (CSV) file. This file when opened in MS Excel will show you the information extracted/inferred from annotation file.

Annotators and reviewers can review and edit the information and report about any information, which is different from expected.

* **Please note to open this csv file use MS Excel and when you save this file – choose “Save As…” csv file.**
* **Please review and make note of file name or which patient note, you reviewed and found errors.**

When you open the file it will have the following parts:

1. **Relationship Matrix:**

In this section events are assigned integer ID’s starting with zero (0).

Columns are temporal relations.



In this example (taken from TDS\_3.csv), We have 6 events from (EVENT 0 to Event 5). Their relation with other Events are shown in this matrix. For example “Event 3 is BEFORE Event 4”. An event can be related to multiple events, for example here “Event 1 is BEFORE Event 2, 3, and 4”. If one event is related to multiple events, then target events IDs delimited by a semicolon (“;”). Please note that this is a CSV file so we did not use comma “,” as delimiter.

Please review the relationships that are based on asserted and inferred relationships exported from the Pellet reasoner.

If you have to edit (for correction) the entries, please add or delete along with semicolons with it. For example, if we have add EVENTID-5 to EVENTID-2 for BEFORE relation:

From: “EVENTID-2” 🡪 BEFORE 🡪 “3;4”

To: “EVENTID-2” 🡪 BEFORE 🡪 “3;4;5”

If you have to delete a target item: Let’s say delete target Event 4 from BEFORE relation, then:

From: “EVENTID-2” 🡪 BEFORE 🡪 “3;4”

To: “EVENTID-2” 🡪 BEFORE 🡪 “3”

If there is no target event and you want to add one, then just put the Event ID in empty cell.

1. **Event Description:**

This section describes Events. This section is useful to map which event id is mapped to which event.

For example Events above has following description in the csv file:



You can also review their time information and please note down if something is missing or wrong.

1. Asserted and Inferred Relations:

This section lists out the asserted and inferred relationships between two events. Please note down that event descriptions text strings may be truncated to 50 characters (max), to make it align better.



1. Timeline Buckets:

This last section is display the sequence that timeline library exports. This is the computed sequence of events that library understand. If two events are equal or start at the same time, they will appear on same line and in same bucket (between curly brackets).

Please review and make notes of incorrect sequence entries.

