Overview of Temporal Module

This module is dedicated to processing electronic medical records for meaningful events, temporal expressions, and their relations on a timeline. Temporal relations are of prime importance in biomedicine as they are intrinsically linked to diseases, signs and symptoms, and treatments. The identification of temporal relations in medical text has drawn growing attention because of its potential to dramatically increase the understanding of many medical phenomena such as disease progression, longitudinal effects of medications, and a patient's clinical course, and its many clinical applications such as question answering[1, 2], clinical outcomes prediction[3], and the recognition of temporal patterns and timelines[4, 5].

Key components

Event annotator

A Begin-Inside-Outside (BIO) style sequence annotator for clinically meaningful events, i.e. anything that would show up on a detailed timeline of the patient's care or life.

Temporal expression annotators

A series of BIO style sequence annotators that employed forward and backward search algorithms and multiple learning methods (Support Vector Machine (SVM), Conditional Random Field (CRF) ) for annotating temporal expressions which would provide concrete temporal references throughout the document or section, e.g. “today”, “24 hours ago”, “postoperative”. More details can be found in [6].
**DocTimeRel annotator**

For every event, there is an SVM-based annotator that can automatically reason the temporal relation between the target event and the document creation time (DCT). This module provided a basic and stable temporal solution that can position all events into coarse temporal bins, e.g. “before the DCT”, “after the DCT”, or “overlap the DCT”. This annotator has proved helpful in solving real clinical temporal-sensitive tasks for multiple institutions [5].

**Temporal relation (TLINK) annotators**

SVM-based annotators for detecting within-sentence Event-Time relations and Event-Event relations. For i2b2 datasets there are also cross sentence Event-Time and Event-Event relation annotators. Multiple techniques have been implemented, including narrative container-based annotation concept [7], tree kernels [8] for syntactic similarity measurement, multi-layered temporal modeling [9], event expansion [10], and deep neural network methods [11, 12].

**Evaluation**

The SVM-based temporal relation annotators achieve an F-score of 0.589 which outperform the best system of Clinical TempEval 2016 [13], whose F-score was 0.573. State-of-the-art results for event-time relations have been achieved with our neural network approaches.

All the above annotators were trained and tested on colon cancer notes from the THYME data set [14].

**Evaluating within-sentence event-time temporal relations:**

The evaluation program is `org.apache.ctakes.temporal.eval.EvaluationOfEventTimeRelations.java`

`org.apache.ctakes.temporal.ae.EventTimeSelfRelationAnnotator` is used by `EvaluationOfEventTimeRelations`

**Required parameters:**

--text <path to the folder contains the THYME raw notes>

--format Anafora

--xml <path to the folder contains the THYME gold annotation files in xml format>

--xml <path to the target folder of xml file output>

If the xml folder is empty (e.g. the xml files have not already been produced), the code will run `getXMIWritingPreprocessorAggregateBuilder()` which writes xml files into the target folder.

--kernelParams "c 0.001953125 t 0.03125 d 3 g 1.0 S 1 C + L 0.5 T 0.1 N 3"

The parameters for SVM model

--patients 1-218

The patient IDs in colon cancer dataset

--useGoldAttributes

**Optional parameters:**

--closure

apply closure calculation to training instances

--test

train on the combination of training and development split, and test on the testing split. Without this parameter, the system will train on training split and test on development split

--skipTrain

If the model is already trained, we can use this to skip the training part and directly apply the model on predicting CONTAINS relations on the test set.

The model files can be found at `target/eval/thyme/train_and_test/event-time`. 
Evaluating within-sentence event-event temporal relations

The evaluation program is `org.apache.ctakes.temporal.eval.EvaluationOfEventEventThymeRelations.java`

`org.apache.ctakes.temporal.ae.EventEventRelationAnnotator` is used by `EvaluationOfEventEventThymeRelations`

The parameters are the same as event-time parameters described above.

The model files can be found at `target/eval/thyme/train_and_test/event-event`.

Evaluating DocTimeRel and Contextual Modality

The evaluation program is `org.apache.ctakes.temporal.eval.EvaluationOfEventProperties.java`


The parameters are the same as event-time parameters.

The model files for DocTimeRel can be found at `target/eval/event-properties/train_and_test/docTimeRel`

The model files for Contextual Modality can be found at `target/eval/event-properties/train_and_test/contextualModality`

Creating Temporal Output in Anafora XML Format

Within-sentence event-event and event-time temporal relations can be output in XML format that is used by Anafora using `org.apache.ctakes.temporal.eval.EvaluationOfBothEEAndETRelations.java`

That program uses the following two annotators:

`org.apache.ctakes.temporal.ae.EventTimeSelfRelationAnnotator`

`org.apache.ctakes.temporal.ae.EventEventRelationAnnotator`

Required parameters for `EvaluationOfBothEEAndETRelations`

`--text <path to the folder contains the THYME raw notes>`

`--format Anafora`

`--xml <path to the folder contains the THYME gold annotation files in xml format>`

`--xml <path to the target folder of xml file output>`

`--kernelParams "c 0.001953125 t 0.03125 d 3 g 1.0 S 1 C + L 0.5 T 0.1 N 3"`

`--patients 1-218`

`--useGoldAttributes`

`--test`

`--skipTrain`

`skipTrain` is now a required parameter. The system will use pretrained models in the `target/eval/thyme/train_and_test/` folder, and directly predict on the test split.

`--anaforaOutput <path to the target folder of xml file output>`

Potentially Improving Temporal Pipeline Results

When running the temporal module, you may want to consider using the newer BIO sentence detector instead of the original sentence detector - on the THYME data the BIO sentence detector achieved better results. However, the BIO sentence detector is not better in all cases.

Online Demo
A demo of the cTAKES Temporal Module can be found at http://54.68.117.30:8080/index.jsp.

Using the BIO Sentence Detector with the Temporal Pipeline

If you are comfortable with modifying Java code, in org.apache.ctakes.temporal.eval.Evaluation_ImplBase.java, modify the method getXMlWritingPreprocessorAggregateBuilder() as follows:

The default pipeline is using the old sentence detector, SentenceDetector. To switch to the new sentence detector, please comment out:

```java
//aggregateBuilder.add( AnalysisEngineFactory.
createEngineDescription( //SentenceDetector.class,
//SentenceDetector.SD_MODEL_FILE_PARAM,
"org/apache/ctakes/core/sentdetect/sd-
med-model.zip" ));
```

and add:

```java
aggregateBuilder.add(SentenceDetectorAnnotatorBIO.getDescription( FileLocator.locateFile("org/apache/ctakes/core/sentdetect/model.jar").getPath()));
```

References