Context-Aware Neural Model for Temporal Information Extraction
Yuanliang Meng and Anna Rumshisky
Text Machine Lab, Department of Computer Science, University of Massachusetts Lowell

Overview

- "Can we mimic human ability to integrate information from previously processed context to understand the order and timing of events in a narrative?"
- We propose the Global Context Layer (GCL) to process and retrieve information about previously processed context.
- Inspired by the Neural Turing Machine (NTM), GCL has long-term memory and soft attention addressing, and thus can resolve long-distance dependencies.
- It has a uniform architecture for event-event, event-timex and timex-timex pairs, so there is no need to train separate models.

Our repository is available: https://github.com/text-machine-lab/TEA

Dataset

We used the Timebank-Dense data (https://www.usna.edu/Users/cs/nchamber/caevo/#corpus)
Training files are annotated with event tags, temporal expression tags (timexes) and temporal relation tags (TLINKs).

All the possible temporal relations within a sentence or cross consecutive sentences are labeled (so it is called "dense"). There are 22 training files, 5 validation files and 9 test files.

Experiment Setup

- Pair type: event-event, event-timex, or timex-timex
- Time value: for timex-timex pairs only. A tuple of real values indicating the difference.
- Words on syntactic dependency path:
  - Intra-sentence pairs: the shortest path between the two entities of interest.
  - Cross-sentence pairs: the paths between entities and their sentence roots, respectively.
- Words in context: The entity mentions and their surrounding words.

Input Features

- Words with syntactic dependency path:
  - Intra-sentence pairs: the shortest path between the two entities of interest.
  - Cross-sentence pairs: the paths between entities and their sentence roots, respectively.

Word tokens are initialized with glove.840B.300d word vectors (300 dimensional).

Results

Two models:
We trained a strictly pairwise model first, without Global Context Layer. Then concatenate the hidden layer of the trained model with the GCL and an output layer.

<table>
<thead>
<tr>
<th>Model</th>
<th>Micro-F1</th>
<th>Macro-F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAEVO (not NN model)</td>
<td>.507</td>
<td></td>
</tr>
<tr>
<td>CATENA (not NN model)</td>
<td>.511</td>
<td></td>
</tr>
<tr>
<td>Meng et al. 2017</td>
<td>.520</td>
<td></td>
</tr>
<tr>
<td>Meng et al. 2017 - pairwise model</td>
<td>.519</td>
<td></td>
</tr>
<tr>
<td>Two more hidden layers</td>
<td>.539</td>
<td>.532</td>
</tr>
<tr>
<td>GCL w/ state-tracking controller</td>
<td>.545</td>
<td>.538</td>
</tr>
<tr>
<td>GCL w/ stateless controller</td>
<td>.546</td>
<td>.538</td>
</tr>
<tr>
<td>GCL w/ pre-trained output layer</td>
<td>.541</td>
<td>.536</td>
</tr>
</tbody>
</table>

- The results in the lower blocks all use double-check.
  - Classify a pair in two ways and pick the result with higher score.
  - "Two more hidden layers" means adding two hidden layers on top of the pre-trained model without using GCL. It is used as a baseline model.
  - The last row corresponds to connecting the output layer (instead of hidden layer) of a pre-trained model to GCL layers with stateless controller.