An Improved Neural Baseline for Temporal Relation Extraction

Qiang Ning,1 Sanjay Subramanian,2 and Dan Roth1,2
1University of Illinois at Urbana-Champaign
2University of Pennsylvania
qning2@illinois.edu, {subs, danroth}@seas.upenn.edu

Abstract
Determining temporal relations (e.g., before or after) between events has been a challenging natural language understanding task, partly due to the difficulty to generate large amounts of high-quality training data. Consequently, neural approaches have not been widely used on it, or showed only moderate improvements. This paper proposes a new neural system that achieves about 10% absolute improvement in accuracy over the previous best system (25% error reduction) on two benchmark datasets. The proposed system is trained on the state-of-the-art MATRES dataset and applies contextualized word embeddings, a Siamese encoder of a temporal common sense knowledge base, and global inference via integer linear programming (ILP). We suggest that the new approach could serve as a strong baseline for future research in this area.

1 Introduction
Temporal relation (TempRel) extraction has been considered as a major component of understanding in natural language (Do et al., 2012; UzZaman et al., 2013; Minard et al., 2015; Llorens et al., 2015; Ning et al., 2018a). However, the annotation process for TempRels is known to be time consuming and difficult even for humans, and existing datasets are usually small and/or have low inter-annotator agreements (IAA); e.g., UzZaman et al. (2013); Chambers et al. (2014); O’Gorman et al. (2016) reported Kohens’ $\kappa$ and $F_1$ in the 60’s. Albeit the significant progress in deep learning nowadays, neural approaches have not been used extensively for this task, or showed only moderate improvements (Dligach et al., 2017; Lin et al., 2017; Meng and Rumshisky, 2018). We think it is important for to understand: is it because we missed a “magic” neural architecture, because the training dataset is small, or because the quality of the dataset should be improved?

Recently, Ning et al. (2018c) introduced a new dataset called Multi-Axis Temporal RELations for Start-points (MATRES). MATRES is still relatively small in its size (15K TempRels), but has a higher annotation quality from its improved task definition and annotation guideline. This paper uses MATRES to show that a long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) system can readily outperform the previous state-of-the-art system, CogCompTime (Ning et al., 2018d), by a large margin. The fact that a standard LSTM system can significantly improve over a feature-based system on MATRES indicates that neural approaches have been mainly dwarfed by the quality of annotation, instead of specific neural architectures or the small size of data.

To gain a better understanding of the standard LSTM method, we extensively compare the usage of various word embedding techniques, including word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), FastText (Bojanowski et al., 2016), ELMo (Peters et al., 2018), and BERT (Devlin et al., 2018), and show their impact on TempRel extraction. Moreover, we further improve the LSTM system by injecting knowledge from an updated version of TEMPROB, an automatically induced temporal common sense knowledge base that provides typical TempRels between events1 (Ning et al., 2018b). Altogether, these components improve over CogCompTime by about 10% in $F_1$ and accuracy. The proposed system is public2 and can serve as a strong baseline for future research.

2 Related Work
Early computational attempts to TempRel extraction include Mani et al. (2006); Chambers et al. (2007); Do et al. (2012); Wilks et al. (2012); Minard et al. (2015); Llorens et al. (2015); Ning et al. (2018a). For example, “explode” typically happens before “die”.

1Defining task as a challenge, we aim to improve the accuracy of TempRel extraction.
2For example, “explode” typically happens before “die”.

References
(2007); Bethard et al. (2007); Verhagen and Pustejovsky (2008), which aimed at building classic learning algorithms (e.g., perceptron, SVM, and logistic regression) using *hand-engineered features* extracted for each pair of events. The frontier was later pushed forward through continuous efforts in a series of SemEval workshops (Verhagen et al., 2007, 2010; UzZaman et al., 2013; Bethard et al., 2015, 2016, 2017), and significant progresses were made in terms of data annotation (Styler IV et al., 2014; Cassidy et al., 2014; Mostafazadeh et al., 2016; O’Gorman et al., 2016), structured inference (Chambers and Jurafsky, 2008a; Do et al., 2012; Chambers et al., 2014; Ning et al., 2018a), and structured machine learning (Yoshikawa et al., 2009; Leeuwenberg and Moens, 2017; Ning et al., 2017).

Since TempRel is a specific relation type, it is natural to borrow recent neural relation extraction approaches (Zeng et al., 2014; Zhang et al., 2015; Zhang and Wang, 2015; Xu et al., 2016). There have indeed been such attempts, e.g., in clinical narratives (Dligach et al., 2017; Lin et al., 2017; Tourille et al., 2017) and in newswire (Cheng and Miyao, 2017; Meng and Rumshisky, 2018; Leeuwenberg and Moens, 2018). However, their improvements over feature-based methods were moderate (Lin et al., 2017) even showed negative results. Given the low IAAs in those datasets, it was unclear whether it was simply due to the low data quality or neural methods inherently do not work well for this task.

A recent annotation scheme, Ning et al. (2018c), introduced the notion of multi-axis to represent the temporal structure of text, and identified that one of the sources of confusions in human annotation is asking annotators for TempRels across different axes. When annotating only same-axis TempRels, along with some other improvements to the annotation guidelines, MATRES was able to achieve much higher IAAs. This dataset opens up opportunities to study neural methods for this problem. In Sec. 3, we will explain our proposed LSTM system, and also highlight the major differences from previous neural attempts.

### 3 Neural TempRel Extraction

One major disadvantage of feature-based systems is that errors occurred in feature extraction proportionally to subsequent modules. Here we study the usage of LSTM networks on the TempRel extraction problem as an end-to-end approach that only takes a sequence of word embeddings as input (assuming that the position of events are known). Conceptually, we need to feed those word embeddings to LSTMs and obtain a vector representation for a particular pair of events, which is followed by a fully-connected, feed-forward neural network (FFNN) to generate confidence scores for each output label. Based on the confidence scores, global inference is performed via integer linear programming (ILP), which is a standard procedure used in many existing works to enforce the transitivity property of time (Chambers and Jurafsky, 2008b; Do et al., 2012; Ning et al., 2017). An overview of the proposed network structure and corresponding parameters can be found in Fig. 1. Below we also explain the main components.

#### 3.1 Handling Event Positions

Each TempRel is associated with two events, and for the same text, different pairs of events possess different relations, so it is critical to indicate the positions of those events when we train LSTMs for the task. The most straightforward way is to concatenate the hidden states from both time steps that correspond to the location of those events (Fig. 1b). Dligach et al. (2017) handled this issue differently, by adding XML tags immediately before and after each event (Fig. 1a). For example, in the sentence, *After eating dinner, he slept comfortably*, where the two events are bold-faced, they will convert the sequence into *After `<e1>` eating `<e2>` dinner; he `<e2>` slept `<e2>` comfortably*. The XML markups, which was initially proposed under the name of *position indicators* for relation extraction (Zhang and Wang, 2015), uniquely indicate the event positions to LSTM, such that the final output of LSTM can be used as a representation of those events and their context. We compare both methods in this paper, and as we show later, the straightforward concatenation method is already as good as XML tags for this task.

#### 3.2 Common Sense Encoder (CSE)

In naturally occurring text that expresses TempRels, connective words such as *since, 4We also tried convolutional neural networks but did not observe that CNNs improved performance significantly compared to the LSTMs. Comparison between LSTM and CNN is also not the focus of this paper.

---

3Between experts: Kohen’s $k \approx 0.84$. Among crowd-sourcers: accuracy 88%. More details in Ning et al. (2018c).
when, or until are often not explicit; nevertheless, humans can still infer the TempRels using common sense with respect to the events. For example, even without context, we know that die is typically after explode and schedule typically before attend. Ning et al. (2018b) was an initial attempt to acquire such knowledge, by aggregating automatically extracted TempRels from a large corpus. The resulting knowledge base, TEMPROB, contains observed frequencies of tuples \((v_1, v_2, r)\) representing the probability of verb 1 and verb 2 having relation \(r\) and it was shown a useful resource for TempRel extraction.

However, TEMPROB is a simple counting model and fails (or is unreliable) for unseen (or rare) tuples. For example, we may see (ambush, die) less frequently than (attack, die) in a corpus, and the observed frequency of (ambush, die) being before or after is thus less reliable. However, since “ambush” is semantically similar to “attack”, the statistics of (attack, die) can actually serve as an auxiliary signal to (ambush, die). Motivated by this idea, we introduce common sense encoder (CSE): We fit an updated version of TEMPROB via a Siamese network (Bromley et al., 1994) that generalizes to unseen tuples through the resulting embeddings for each verb (Fig. 1c). Note that the TEMPROB we use is reconstructed using the same method described in Ning et al. (2018b) with the base method changed to CogCompTime. Once trained, CSE will remain fixed when training the LSTM part (Fig. 1a or b) and the feedforward neural network part (Fig. 1d). We only use CSE for its output. In the beginning, we tried to directly use the output (i.e., a scalar) and the influence on performance was negligible. Therefore, here we discretize the CSE output, change it to categorical embeddings, concatenate them with the LSTM output, and then produce the confidence scores (Fig. 1d).

4 Experiments

4.1 Data

The MATRES dataset\(^5\) contains 275 news articles from the TempEval3 workshop (Uzzaman et al., 2013) with newly annotated events and TempRels. It has 3 sections: TimeBank (TB), AQUAINT (AQ), and Platinum (PT). We followed the official split (i.e., TB+AQ for training and PT for testing), and further set aside 20% of the training data as the development set to tune learning rates and epochs. We also show our performance on another dataset, \(^5\)http://cogcomp.org/page/publication_view/834
TCR\(^6\) (Ning et al., 2018a), which contains both temporal and causal relations, and which only need the temporal part. The label set for both datasets are before, after, equal, and vague.

<table>
<thead>
<tr>
<th>Purpose</th>
<th>#Doc</th>
<th>#Events</th>
<th>#TempRels</th>
</tr>
</thead>
<tbody>
<tr>
<td>TB+AQ</td>
<td>255</td>
<td>8K</td>
<td>13K</td>
</tr>
<tr>
<td>PT</td>
<td>Test</td>
<td>20</td>
<td>537</td>
</tr>
<tr>
<td>TCR</td>
<td>Test</td>
<td>25</td>
<td>1.3K</td>
</tr>
</tbody>
</table>

Table 1: TimeBank (TB), AQUAINT (AQ), and Platinum (PT) are from MATRES (Ning et al., 2018c) and TCR from Ning et al. (2018a).

4.2 Results and Discussion

We compare with the most recent version of CogCompTime, the state-of-the-art on MATRES.\(^7\)

Note that in Table 2, CogCompTime performed slightly different to Ning et al. (2018d): CogCompTime reportedly had \(F_1=65.9\) (Table 2 Line 3 therein) and here we obtained \(F_1=66.6\). In addition, Ning et al. (2018d) only reported \(F_1\) scores, while we also use another two metrics for a more thorough comparison: classification accuracy (acc.) and temporal awareness \(F_{\text{aware}}\), where the awareness score is for the graphs represented by a group of related TempRels (more details in the appendix). We also report the average of those three metrics in our experiments.

Table 2 compares the two different ways to handle event positions discussed in Sec. 3.1: position indicators (P.I.) and simple concatenation (Concat), both of which are followed by network (d) in Fig. 1 (i.e., without using Siamese yet). We extensively studied the usage of various pretrained word embeddings, including conventional embeddings (i.e., the medium versions of word2vec, GloVe, and FastText provided in the Magnitude package (Patel et al., 2018)) and contextualized embeddings (i.e., the original ELMo and large uncased BERT, respectively); except for the input embeddings, we kept all other parameters the same. We used cross-entropy loss and the StepLR optimizer in PyTorch that decays the learning rate by 0.5 every 10 epochs (performance not sensitive to it).

Comparing to the previously used P.I. (Diligach et al., 2017), we find that, with only two exceptions (underlined in Table 2), the Concat system saw consistent gains under various embeddings and metrics. In addition, contextualized embeddings (ELMo and BERT) expectedly improved over the conventional ones significantly, although no statistical significance were observed between using ELMo or BERT (more significance tests in Appendix).

Given the above two observations, we further incorporated our common sense encoder (CSE) into “Concat” with ELMo and BERT in Table 2. We split TEMPROB into train (80%) and validation (20%). The proposed Siamese network (Fig. 1c) was trained by minimizing the cross-entropy loss using Adam (Kingma and Ba, 2014) (learning rate 1e-4, 20 epochs, and batch size 500). We first see that CSE improved on top of Concat for both ELMo and BERT under all metrics, confirming the benefit of TEMPROB; second, as compared to CogCompTime, the proposed Concat+CSE achieved about 10% absolute gains in accuracy and \(F_1\), 5% in awareness score \(F_{\text{aware}}\), and 8% in the three-metric-average metric, with \(p<0.001\) per the McNemar’s test. Roughly speaking, the 8% gain is contributed by LSTMs for 2%, contextualized embeddings for 4%, and CSE for 2%. Again, no statistical significance were observed between using ELMo and BERT.

Table 3 furthermore applies CogCompTime and the proposed Concat+CSE system on a different test set called TCR (Ning et al., 2018a). Both systems achieved better scores (suggesting that TCR is easier than MATRES), while the proposed sys-
tem still outperformed CogCompTime by roughly 8% under the three-metric-average metric, consistent with our improvement on MATRES.

<table>
<thead>
<tr>
<th>System</th>
<th>Emb.</th>
<th>Acc.</th>
<th>$F_1$</th>
<th>$F_{macro}$</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CogCompTime</td>
<td>68.1</td>
<td>70.7</td>
<td>61.6</td>
<td>66.8</td>
<td></td>
</tr>
<tr>
<td>Concat+CSE</td>
<td>80.8</td>
<td>78.6</td>
<td>69.9</td>
<td>76.4</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Further evaluation of the proposed system, i.e., Concat (Table 3.1) plus CSE (Sec. 3.2), on the TCR dataset (Ning et al., 2018a).

5 Conclusion

Temporal relation extraction has long been an important yet challenging task in natural language processing. Lack of high-quality data and difficulty in the learning problem defined by previous annotation schemes inhibited performance of neural-based approaches. The discoveries that LSTMs readily improve the feature-based state-of-the-art CogCompTime on the MATRES and TCR datasets by a large margin not only give the community a strong baseline, but also indicate that the learning problem is probably better defined by MATRES and TCR. Therefore, we should move along that direction to collect more high-quality data, which can facilitate more advanced learning algorithms in the future.

Acknowledgements

This research is supported by a grant from the Allen Institute for Artificial Intelligence (allenai.org), the IBM-ILLINOIS Center for Cognitive Computing Systems Research (C3SR) – a research collaboration as part of the IBM AI Horizons Network, and contract HR0011-18-2-0052 with the US Defense Advanced Research Projects Agency (DARPA). Approved for Public Release, Distribution Unlimited. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

References


