

# Joint Event and Temporal Relation Extraction with Shared Representations and Structured Prediction

Rujun Han,<sup>1,2</sup> Qiang Ning,<sup>3</sup> Nanyun Peng<sup>1,2</sup>

<sup>1</sup>Department of Computer Science, University of Southern California

<sup>2</sup>Information Sciences Institute, University of Southern California

<sup>3</sup>Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign  
 rujunhan@isi.edu, qning2@illinois.edu, npeng@isi.edu

## Abstract

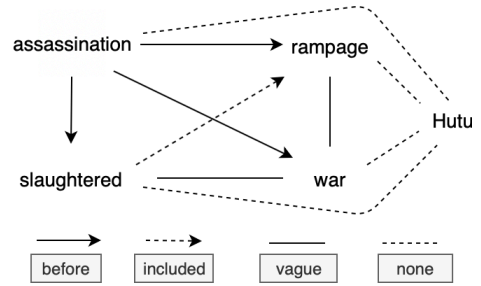
We propose a joint event and temporal relation extraction model with shared representation learning and structured prediction. The proposed method has two advantages over existing work. First, it improves event representation by allowing the event and relation modules to share the same contextualized embeddings and neural representation learner. Second, it avoids error propagation in the conventional pipeline systems by leveraging structured inference and learning methods to assign both the event labels and the temporal relation labels jointly. Experiments show that the proposed method can improve both event extraction and temporal relation extraction over state-of-the-art systems, with the end-to-end  $F_1$  improved by 10% and 6.8% on two benchmark datasets respectively.

## 1 Introduction

The extraction of temporal relations among events is an important natural language understanding (NLU) task that can benefit many downstream tasks such as question answering, information retrieval, and narrative generation. The task can be modeled as building a graph for a given text, whose nodes represent events and edges are labeled with temporal relations correspondingly. Figure 1a illustrates such a graph for the text shown therein. The nodes **assassination**, **slaughtered**, **rampage**, **war**, and **Hutu** are the candidate events, and different types of edges specify different temporal relations between them: **assassination** is *BEFORE* **rampage**, **rampage** *INCLUDES* **slaughtered**, and the relation between **slaughtered** and **war** is *VAGUE*. Since “Hutu” is actually not an event, a system is expected to annotate the relations between “Hutu” and all other nodes in the graph as *NONE* (i.e., no relation).

As far as we know, all existing systems treat this task as a pipeline of two separate subtasks,

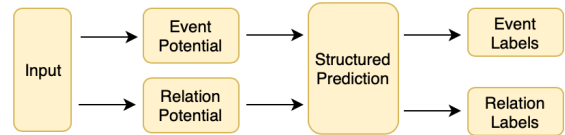
*The **assassination** touched off a murderous **rampage** by Hutu security forces and civilians, who **slaughtered** mainly Tutsis but also Hutus who favored reconciliation with the minority. It also reignited the civil **war**.*



(a) Temporal Relation Graph



(b) Pipeline Model



(c) Structured Joint Model

Figure 1: An illustration of event and relation models in our proposed joint framework. (a) is a (partial) graph of the output of the relation extraction model. “Hutu” is not an event and hence all relations including it should be annotated as *NONE*. (b) and (c) are comparisons between a pipeline model and our joint model.

i.e., event extraction and temporal relation classification, and they also assume that gold events are given when training the relation classifier (Verhagen et al., 2007, 2010; UzZaman et al., 2013; Chambers et al., 2014; Ning et al., 2017; Meng and Rumshisky, 2018). Specifically, they built end-to-end systems that extract events first and then predict temporal relations between them (Fig. 1b). In these pipeline models, event extraction errors will propagate to the relation classification step and cannot be corrected afterwards. **Our first contribution** is the proposal of a joint model that ex-

tracts both events and temporal relations simultaneously (see Fig. 1c). The motivation is that if we train the relation classifier with *NONE* relations between non-events, then it will potentially have the capability of correcting event extraction mistakes. For instance in Fig. 1a, if the relation classifier predicts *NONE* for (**Hutu**, **war**) with a high confidence, then this is a strong signal that can be used by the event classifier to infer that at least one of them is not an event.

**Our second contribution** is that we improve event representations by sharing the same contextualized embeddings and neural representation learner between the event extraction and temporal relation extraction modules for the first time. On top of the shared embeddings and neural representation learner, the proposed model produces a graph-structured output representing all the events and relations in the given sentences.

A valid graph prediction in this context should satisfy two structural constraints. First, the temporal relation should always be *NONE* between two non-events or between one event and one non-event. Second, for those temporal relations among events, no loops should exist due to the transitive property of time (e.g., if **A** is before **B** and **B** is before **C**, then **A** must be before **C**). The validity of a graph is guaranteed by solving an integer linear programming (ILP) optimization problem with those structural constraints, and our joint model is trained by structural support vector machines (SSVM) in an end-to-end fashion.

Results show that, according to the end-to-end  $F_1$  score for temporal relation extraction, the proposed method improves CAEVO (Chambers et al., 2014) by 10% on TB-Dense, and improves CogCompTime (Ning et al., 2018c) by 6.8% on MATRES. We further show ablation studies to confirm that the proposed joint model with shared representations and structured learning is very effective for this task.

## 2 Related Work

In this section we briefly summarize the existing work on event extraction and temporal relation extraction. To the best of our knowledge, there is no prior work on joint *event* and relation extraction, so we will review joint *entity* and relation extraction works instead.

Existing **event extraction** methods in the temporal relation domain, as in the TempEval3 work-

shop (UzZaman et al., 2013), all use conventional machine learning models (logistic regression, SVM, or Max-entropy) with hand-engineered features (e.g., ClearTK (Bethard, 2013) and NavyTime (Chambers, 2013)). While other domains have shown progress on event extraction using neural methods (Nguyen and Grishman, 2015; Nguyen et al., 2016; Feng et al., 2016), recent progress in the temporal relation domain is focused more on the setting where gold events are provided. Therefore, we first show the performance of a neural event extractor on this task, although it is not our main contribution.

Early attempts on **temporal relation extraction** use local pair-wise classification with hand-engineered features (Mani et al., 2006; Verhagen et al., 2007; Chambers et al., 2007; Verhagen and Pustejovsky, 2008). Later efforts, such as ClearTK (Bethard, 2013), UTTime (Laokulrat et al., 2013), NavyTime (Chambers, 2013), and CAEVO (Chambers et al., 2014) improve earlier work with better linguistic and syntactic rules. Yoshikawa et al. (2009); Ning et al. (2017); Leeuwenberg and Moens (2017) explore structured learning for this task, and more recently, neural methods have also been shown effective (Tourille et al., 2017; Cheng and Miyao, 2017; Meng et al., 2017; Meng and Rumshisky, 2018).

In practice, we need to extract both events and those temporal relations among them from raw text. All the works above treat this as two sub-tasks that are solved in a pipeline. To the best of our knowledge, there has been no existing work on **joint event-temporal relation extraction**. However, the idea of “joint” has been studied for *entity*-relation extraction in many works. Miwa and Sasaki (2014) frame their joint model as table filling tasks, map tabular representation into sequential predictions with heuristic rules, and construct global loss to compute the best joint predictions. Li and Ji (2014) define a global structure for joint entity and relation extraction, encode local and global features based on domain and linguistic knowledge, and leverage beam-search to find global optimal assignments for entities and relations. Miwa and Bansal (2016) leverage LSTM architectures to jointly predict both entity and relations, but fall short on ensuring prediction consistency. Zhang et al. (2017) combine the benefits of both neural net and global optimization with beam

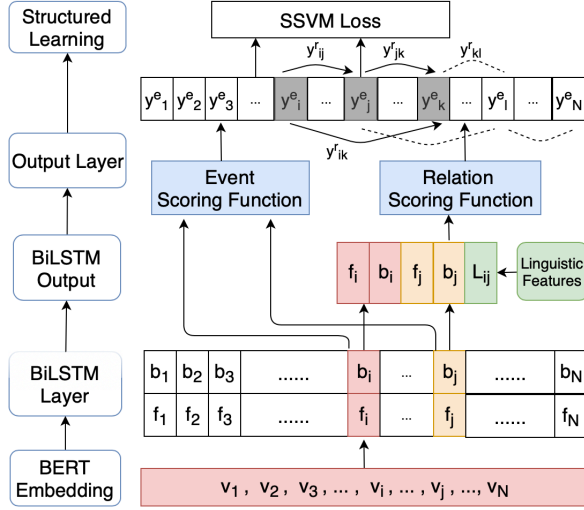


Figure 2: Deep neural network architecture for joint structured learning. Note that on the structured learning layer, grey bars denote tokens being predicted as events. Edge types between events follow the same notations as in 1a.  $y_l^e = 0$  (non-event), so all edges connecting to  $y_l^e$  are *NONE*.  $y_i^e = 1, y_j^e = 1, y_k^e = 1$  (events) and hence edges between them are forced to be the same ( $y_{ij}^r = y_{jk}^r = y_{ik}^r = \text{BEFORE}$  in this example) by transitivity. These global assignments are input to compute the SSVM loss.

search. Motivated by these works, we propose an end-to-end trainable neural structured support vector machine (neural SSVM) model to *simultaneously* extract events and their relations from text and ensure the global structure via ILP constraints. Next, we will describe in detail our proposed method.

### 3 Joint Event-Relation Extraction Model

In this section we first provide an overview of our neural SSVM model, and then describe each component in our framework in detail (i.e., the multi-tasking neural scoring module, and how inference and learning are performed). We denote the set of all possible relation labels (including *NONE*) as  $\mathcal{R}$ , all event candidates (both events and non-events) as  $\mathcal{E}$ , and all relation candidates as  $\mathcal{EE}$ .

#### 3.1 Neural SSVM

Our neural SSVM adapts the SSVM loss as:

$$\mathcal{L} = \sum_{n=1}^l \frac{C}{M^n} \left[ \max_{\hat{\mathbf{y}}^n \in \mathcal{Y}} (0, \Delta(\mathbf{y}^n, \hat{\mathbf{y}}^n) + \bar{S}_{\mathcal{R}}^n) + C_{\mathcal{E}} \bar{S}_{\mathcal{E}}^n \right] + \|\Phi\|^2, \quad (1)$$

where  $\bar{S}_{\mathcal{E}}^n = S(\hat{\mathbf{y}}_{\mathcal{E}}^n; \mathbf{x}^n) - S(\mathbf{y}_{\mathcal{E}}^n; \mathbf{x}^n)$  and  $\bar{S}_{\mathcal{R}}^n = S(\hat{\mathbf{y}}_{\mathcal{R}}^n; \mathbf{x}^n) - S(\mathbf{y}_{\mathcal{R}}^n; \mathbf{x}^n)$ ;  $\Phi$  denotes model parameters,  $n$  indexes instances,  $M^n = |\mathcal{E}|^n + |\mathcal{EE}|^n$  de-

notes the total number of relations  $|\mathcal{EE}|^n$  and events  $|\mathcal{E}|^n$  in instance  $n$ .  $\mathbf{y}^n, \hat{\mathbf{y}}^n$  denote the gold and predicted global assignments of events and relations for instance  $n$ —each of which consists of either one hot vector representing true and predicted relation labels  $\mathbf{y}_{\mathcal{R}}^n, \hat{\mathbf{y}}_{\mathcal{R}}^n \in \{0, 1\}^{|\mathcal{EE}|}$ , or entity labels  $\mathbf{y}_{\mathcal{E}}^n, \hat{\mathbf{y}}_{\mathcal{E}}^n \in \{0, 1\}^{|\mathcal{E}|}$ . A maximum a posteriori (MAP) inference is needed to find  $\hat{\mathbf{y}}^n$ , which we formulate as an interger linear programming (ILP) problem and describe more details in Section 3.3.  $\Delta(\mathbf{y}^n, \hat{\mathbf{y}}^n)$  is a distance measurement between the gold and the predicted assignments; we simply use the Hamming distance.  $C$  and  $C_{\mathcal{E}}$  are the hyper-parameters to balance the losses between event, relation and the regularizer, and  $S(\mathbf{y}_{\mathcal{E}}^n; \mathbf{x}^n), S(\mathbf{y}_{\mathcal{R}}^n; \mathbf{x}^n)$  are scoring functions, which we design a multi-tasking neural architecture to learn.

The intuition behind the SSVM loss is that it requires the score of gold output structure  $\mathbf{y}^n$  to be greater than the score of the best output structure under the current model  $\hat{\mathbf{y}}^n$  with a margin  $\Delta(\mathbf{y}^n, \hat{\mathbf{y}}^n)^1$  or else there will be some loss. The training objective is to minimize the loss.

The major difference between our neural-SSVM and the traditional SSVM model is the scoring function. Traditional SSVM uses a linear function over hand-crafted features to compute the scores, whereas we propose to use a recurrent neural network to estimate the scoring function and train the entire architecture end-to-end.

#### 3.2 Multi-Tasking Neural Scoring Function

The recurrent neural network (RNN) architecture has been widely adopted by prior temporal extraction work to encode context information (Tourille et al., 2017; Cheng and Miyao, 2017; Meng et al., 2017). Motivated by these works, we adopt a RNN-based scoring function for both event and relation prediction in order to learn features in a data driven way and capture long-term contexts in the input. In Fig. 2, we skip the input layer for simplicity.<sup>2</sup>

The bottom layer corresponds to contextualized word representations denoted as  $v_k$ . We use  $(i, j) \in \mathcal{EE}$  to denote a candidate relation and  $i \in \mathcal{E}$  to

<sup>1</sup>Note that if the best prediction is the same as the gold structure, the margin is zero; there will be no loss.

<sup>2</sup>Following the convention of event relation prediction literature (Chambers et al., 2014; Ning et al., 2018a,b), we only consider event pairs that occur in the same or neighboring sentences, but the architecture can be easily adapted to the case where inputs are longer than two sentences.

indicate a candidate event in the input sentences of length  $N$ . We fix word embeddings computed by a pre-trained BERT-base model (Devlin et al., 2018). They are then fed into a BiLSTM layer to further encode task-specific contextual information. Both event and relation tasks share this layer.

The event scorer is illustrated by the left two branches following the BiLSTM layer. We simply concatenate both forward and backward hidden vectors to encode the context of each token. As for the relation scorer shown in the right branches, for each pair  $(i, j)$  we take the forward and backward hidden vectors corresponding to them,  $f_i, b_i, f_j, b_j$ , and concatenate them with linguistic features as in previous event relation prediction research. We denote linguistic features as  $L_{i,j}$  and only use simple features provided in the original datasets: token distance, tense, and polarity of events.

Finally, all hidden vectors and linguistic features are concatenated to form the input to compute the probability of being an event or a softmax distribution over all possible relation labels—which we refer to as the RNN-based **scoring function** in the following sections.

### 3.3 MAP Inference

A MAP inference is needed both during training to obtain  $\hat{y}^n$  in the loss function (Equation 1), as well as during the test time to get globally coherent assignments. We formulate the inference problem as an ILP problem. The inference framework is established by constructing a global objective function using scores from local scorers and imposing several global constraints: 1) one-label assignment, 2) event-relation consistency, and 3) symmetry and transitivity as in Bramsen et al. (2006); Chambers and Jurafsky (2008); Denis and Muller (2011); Do et al. (2012); Ning et al. (2017).

#### 3.3.1 Objective Function

The objective function of the global inference is to find the global assignment that has the highest probability under the current model, as specified in Equation 2:

$$\begin{aligned} \hat{y} = \arg \max & \sum_{(i,j) \in \mathcal{EE}} \sum_{r \in \mathcal{R}} y_{i,j}^r S(y_{i,j}^r, \mathbf{x}) \\ & + C_{\mathcal{E}} \sum_{k \in \mathcal{E}} \sum_{e \in \{0,1\}} y_k^e S(y_k^e, \mathbf{x}) \\ \text{s.t. } & y_{i,j}^r, y_k^e \in \{0,1\}, \sum_{r \in \mathcal{R}} y_{i,j}^r = 1, \sum_{e \in \{0,1\}} y_k^e = 1, \end{aligned} \quad (2)$$

where  $y_k^e$  is a binary indicator of whether the  $k$ -th candidate is an event or not, and  $y_{i,j}^r$  is a binary indicator specifying whether the global prediction of the relation between  $(i, j)$  is  $r \in \mathcal{R}$ .  $S(y_k^e, \mathbf{x}), \forall e \in \{0,1\}$  and  $S(y_{i,j}^r, \mathbf{x}), \forall r \in \mathcal{R}$  are the scoring functions obtained from the event and relation scoring functions, respectively. The output of the global inference  $\hat{y}$  is a collection of optimal label assignments for all events and relation candidates in a fixed context.  $C_{\mathcal{E}}$  is a hyper-parameter controlling weights between relation and event. The constraint that follows immediately from the objective function is that the global inference should only assign one label for all entities and relations.

#### 3.3.2 Constraints

We introduce several additional constraints to ensure the resulting optimal output graph forms a valid and plausible event graph.

**Event-Relation Consistency.** Event and relation prediction consistency is defined with the following property: **a pair of input tokens have a positive temporal relation if and only if both tokens are events**. The following global constraints will satisfy this property,

$$\begin{aligned} \forall (i, j) \in \mathcal{EE}, e_i^P \geq r_{i,j}^P, e_j^P \geq r_{i,j}^P \\ \text{and } e_i^N + e_j^N \geq r_{i,j}^N \end{aligned}$$

where  $e_i^P$  denotes an event and  $e_i^N$  denotes a non-event token.  $r_{i,j}^P$  indicates positive relations: *BEFORE*, *AFTER*, *SIMULTANEOUS*, *INCLUDES*, *IS\_INCLUDED*, *VAGUE* and  $r_{i,j}^N$  indicate a negative relation, i.e., *NONE*. A formal proof of this property can be found in Appendix A.

**Symmetry and Transitivity Constraint.** We also explore the symmetry and transitivity constraints of relations. They are specified as follows:

$$\forall (i, j), (j, k) \in \mathcal{EE}, y_{i,j}^r = y_{j,i}^{\bar{r}}, \text{ (symmetry)}$$

$$y_{i,j}^{r_1} + y_{j,k}^{r_2} - \sum_{r_3 \in \text{Trans}(r_1, r_2)} y_{i,k}^{r_3} \leq 1, \text{ (transitivity)}$$

Intuitively, the symmetry constraint forces two pairs of events with flipping orders to have reversed relations. For example, if  $r_{i,j} = \text{BEFORE}$ , then  $r_{j,i} = \text{AFTER}$ . The transitivity constraint rules that if  $(i, j)$ ,  $(j, k)$  and  $(i, k)$  pairs exist in the graph, the label (relation) prediction of  $(i, k)$  pair



has to fall into the transitivity set specified by  $(i, j)$  and  $(j, k)$  pairs. The full transitivity table can be found in Ning et al. (2018a).

### 3.4 Learning

We begin by experimenting with optimizing SSVM loss directly, but model performance degrades.<sup>3</sup> Therefore, we develop a two-state learning approach which first trains a pipeline version of the joint model without feedback from global constraints. In other words, the local neural scoring functions are optimized with cross-entropy loss using gold events and relation candidates that are constructed directly from the outputs of the event model. During the second stage, we switch to the global SSVM loss function in Equation 1 and re-optimize the network to adjust for global properties. We will provide more details in Section 4.

## 4 Implementation Details

In this section we describe implementation details of the baselines and our four models to build an end-to-end event temporal relation extraction system with an emphasis on the structured joint model. In Section 6 we will compare and contrast them and show why our proposed structured joint model works the best.

### 4.1 Baselines

We run two event and relation extraction systems, CAEVO<sup>4</sup> (Chambers et al., 2014) and CogCompTime<sup>5</sup> (Ning et al., 2018c), on TB-Dense and MATRES, respectively. These two methods both leverage conventional learning algorithms (i.e., MaxEnt and averaged perceptron, respectively) based on manually designed features to obtain separate models for events and temporal relations, and conduct end-to-end relation extraction as a pipeline. Note Chambers et al. (2014) does not report event and end-to-end temporal relation extraction performances, so we calculate the scores per our implementation.

<sup>3</sup>We leave further investigation for future work.

<sup>4</sup><https://www.usna.edu/Users/cs/nchamber/caevo/>

<sup>5</sup>[http://cogcomp.org/page/publication\\_view/844](http://cogcomp.org/page/publication_view/844)

### 4.2 End-to-End Event Temporal Relation Extraction

**Single-Task Model.** The most basic way to build an end-to-end system is to train separate event detection and relation prediction models with gold labels, as we mentioned in our introduction. In other words, the BiLSTM layer is not shared as in Fig. 2. During evaluation and test time, we use the outputs from the event detection model to construct relation candidates and apply the relation prediction model to make the final prediction.

**Multi-Task Model.** This is the same as the single-task model except that the BiLSTM layer is now shared for both event and relation tasks. Note that both single-task and multi-task models are not trained to tackle the *NONE* relation directly. They both rely on the predictions of the event model to annotate relations as either positive pairs or *NONE*.

**Pipeline Joint Model.** This shares the same architecture as the multi-task model, except that during training, we use the predictions of the event model to construct relation candidates to train the relation model. This strategy will generate *NONE* pairs during training if one argument of the relation candidate is not an event. These *NONE* pairs will help the relation model to distinguish negative relations from positive ones, and thus become more robust to event prediction errors. We train this model with gold events and relation candidates during the first several epochs in order to obtain a relatively accurate event model and switch to a pipeline version afterwards inspired by Miwa and Bansal (2016).

**Structured Joint Model.** This is described in detail in Section 3. However, we experience difficulties in training the model with SSVM loss from scratch. This is due to large amounts of non-event tokens, and the model is not capable of distinguishing them in the beginning. We thus adopt a two-stage learning procedure where we take the best pipeline joint model and re-optimize it with the SSVM loss.

To restrict the search space for events in the ILP inference of the SSVM loss, we use the predicted probabilities from the event detection model to filter out non-events since the event model has a strong performance, as shown in Section 6. Note that this is very different from the pipeline

model where events are first predicted and relations are constructed with *predicted* events. Here, we only leverage an additional hyper-parameter  $T_{evt}$  to filter out highly unlikely event candidates. Both event and relation labels are assigned *simultaneously* during the global inference with ILP, as specified in Section 3.3. We also filter out tokens with POS tags that do not appear in the training set as most of the events are either nouns or verbs in TB-Dense, and all events are verbs in MATRES.

**Hyper-Parameters.** All single-task, multi-task and pipeline joint models are trained by minimizing cross-entropy loss. We observe that model performances vary significantly with dropout ratio, hidden layer dimensions of the BiLSTM model and entity weight in the loss function (with relation weight fixed at 1.0). We leverage a pre-trained BERT model to compute word embedding<sup>6</sup> and all MLP scoring functions have one hidden layer.<sup>7</sup> In the SSVM loss function, we fix the value of  $C = 1$ , but fine-tune  $C_{\mathcal{E}}$  in the objective function in Equation 2. Hyper-parameters are chosen using a standard development set for TB-Dense and a random holdout-set based on an 80/20 split of training data for MATRES. To solve ILP in the inference process, we leverage an off-the-shelf solver provided by Gurobi optimizer; i.e. the best solutions from the Gurobi optimizer are inputs to the global training. The best combination of hyper-parameters can be found in Table 9 in our appendix.<sup>8</sup>

## 5 Experimental Setup

In this section we first provide a brief overview of temporal relation data and describe the specific datasets used in this paper. We also explain the evaluation metrics at the end.

### 5.1 Temporal Relation Data

Temporal relation corpora such as TimeBank (Pustejovsky et al., 2003) and RED (O’Gorman et al., 2016) facilitate the research in temporal relation extraction. The common issue in these corpora is missing annotations. Collecting densely

<sup>6</sup>We use a pre-trained BERT-Base model with 768 hidden size, 12 layers, 12 heads implemented by <https://github.com/huggingface/pytorch-pretrained-BERT>

<sup>7</sup>Let  $H, K$  denotes the dimension of (concatenated) vector from BiLSTM and number of output classes. MLP layer consists of  $|H| * |K| + |K| * |K|$  parameters

<sup>8</sup>PyTorch code will be made available upon acceptance.

	TB-Dense	MATRES
<b># of Documents</b>		
Train	22	183
Dev	5	-
Test	9	20
<b># of Pairs</b>		
Train	4032	6332
Dev	629	-
Test	1427	827

Table 1: Data overview. Note that the numbers reported for MATRES do not include the AQUAINT section.

annotated temporal relation corpora with all events and relations fully annotated is reported to be a challenging task as annotators could easily overlook some facts (Bethard et al., 2007; Cassidy et al., 2014; Chambers et al., 2014; Ning et al., 2017), which made both modeling and evaluation extremely difficult in previous event temporal relation research.

The TB-Dense dataset mitigates this issue by forcing annotators to examine all pairs of events within the same or neighboring sentences, and it has been widely evaluated on this task (Chambers et al., 2014; Ning et al., 2017; Cheng and Miyao, 2017; Meng and Rumshisky, 2018). Recent data construction efforts such as MATRES (Ning et al., 2018a) further enhance the data quality by using a multi-axis annotation scheme and adopting a start-point of events to improve inter-annotator agreements. We use TB-Dense and MATRES in our experiments and briefly summarize the data statistics in Table 1.

### 5.2 Evaluation Metrics

To be consistent with previous research, we adopt two different evaluation metrics. The first one is the standard **micro-average scores**. For densely annotated data, the micro-average metric should share the same precision, recall and F1 scores. However, since our joint model includes *NONE* pairs, we follow the convention of IE tasks and exclude them from evaluation. The second one is similar except that we exclude both *NONE* and *VAGUE* pairs following (Ning et al., 2018c). Please refer to Figure 4 in the appendix for a visualizations of the two metrics.

## 6 Results and Analysis

The main results of this paper can be found in Table 2. All best-recall and F1 scores are achieved by our structured joint model, and the results outperform the baseline systems by 10.0% and 6.8% on end-to-end relation extraction per F1 scores and

Corpus	Models	Event			Relation		
		P	R	F1	P	R	F1
TB-Dense	Structrued Joint Model (Ours)	89.2	<b>92.6</b>	<b>90.9</b>	<b>52.6</b>	<b>46.5</b>	<b>49.4</b>
	Chambers et al. (2014)	<b>97.2</b>	79.4	87.4	43.8	35.7	39.4
MATRES	Structrued Joint Model (Ours)	<b>87.1</b>	<b>88.5</b>	<b>87.8</b>	<b>59.0</b>	<b>60.2</b>	<b>59.6</b>
	Ning et al. (2018c)	83.5	87.0	85.2	48.4	58.0	52.8

Table 2: Event and Relation Extraction Results on TB-Dense and MATRES

Micro-average F1 (%)	TB-Dense			MATRES		
	Event	Relation (G)	Relation (E)	Event	Relation (G)	Relation (E)
Baselines	87.4 <sup>†</sup>	57.0 <sup>‡</sup>	39.4 <sup>†</sup>	85.2	65.9	52.8
Single-task	88.6	61.9	44.3	86.9	75.3	57.2
Multi-task	89.2	<b>64.5</b>	48.4	86.4	<b>75.5</b>	58.7
Pipeline Joint	90.3	-	48.5	87.2	-	58.5
Structured Joint	<b>90.9</b>	-	<b>49.4</b>	<b>87.8</b>	-	<b>59.6</b>

Table 3: Further ablation studies on event and relation extractions. Relation (G) denotes train and evaluate using gold events to compose relation candidates, whereas Relation (E) means end-to-end relation extraction. <sup>†</sup> is the event extraction and pipeline relation extraction F1 scores for CAEVO (Chambers et al., 2014). 57.0<sup>‡</sup> is the best previously reported micro-average score for temporal relation extraction based on gold events by Meng and Rumshisky (2018). All MATRES baseline results are provided by Ning et al. (2018c).

3.5% and 2.6% on event extraction per F1 scores. The best precision score for the TB-Dense dataset is achieved by CAEVO, which indicates that the linguistic rule-based system can make highly precise predictions by being conservative.

Table 3 shows a more detailed analysis, in which we can see that our single-task models with BERT embeddings and a BiLSTM encoder already outperform the baseline systems on end-to-end relation extraction tasks by 4.9% and 4.4% respectively. In the following sections we discuss step-by-step improvement by adopting multi-task, pipeline joint, and structured joint models on end-to-end relation extraction, event extraction, and relation extraction on gold event pairs.

## 6.1 End-to-End Relation Extraction

**TB-Dense.** The improvements over the single-task model per F1 score are 4.1% and 4.2% for the multi-task and pipeline joint model respectively. This indicates that the pipeline joint model is helpful only marginally. Table 4 shows that the structured joint model improves both precision and recall scores for *BEFORE* and *AFTER* and achieves the best end-to-end relation extraction performance at 49.4%—which outperforms the baseline system by 10.0% and the single-task model by 5.1%.

**MATRES.** Compared to the single-task model, the multi-task model improves F1 scores by 1.5%, while the pipeline joint model improves F1 scores by 1.3%—which means that pipeline joint training does not bring any gains for MATRES. The structured joint model reaches the best end-to-end F1 score at 59.6%, which outperforms the baseline system by 6.8% and the single-task model by 2.4%. We speculate that the gains come from the joint model’s ability to help deal with *NONE* pairs, since recall scores for *BEFORE* and *AFTER* increase by 1.5% and 1.1% respectively (Table 10 in our appendix).

## 6.2 Event Extraction

**TB-Dense.** Our structured joint model outperforms the CAEVO baseline by 3.5% and the single-task model by 1.3%. Improvements on event extraction can be difficult because our single-task model already works quite well with a close-to 89% F1 score, while the inter-annotator agreement for events in TimeBank documents is merely 87% (UzZaman et al., 2013).

**MATRES.** The structured model outperforms the the baseline model and the single-task model by 2.6% and 0.9% respectively. However, we observe that the multi-task model has a slight drop in event extraction performance over the single-task model (86.4% vs. 86.9%). This indicates

	CAEVO			Pipeline Joint			Structure Joint		
	P	R	F1	P	R	F1	P	R	F1
<b>B</b>	41.4	19.5	26.5	59.0	46.9	52.3	59.8	46.9	52.6
<b>A</b>	42.1	17.5	24.7	69.3	45.3	54.8	71.9	46.7	56.6
<b>I</b>	50.0	3.6	6.7	-	-	-	-	-	-
<b>II</b>	38.5	9.4	15.2	-	-	-	-	-	-
<b>S</b>	14.3	4.5	6.9	-	-	-	-	-	-
<b>V</b>	44.9	59.4	51.1	45.1	55.0	49.5	45.9	55.8	50.4
<b>Avg</b>	43.8	35.7	39.4	51.5	45.9	48.5	52.6	46.5	49.4

Table 4: Model performance breakdown for TB-Dense. “-” indicates no predictions were made for that particular label, probably due to the small size of the training sample. *BEFORE* (B), *AFTER* (A), *INCLUDES* (I), *IS\_INCLUDED* (II), *SIMULTANEOUS* (S), *VAGUE* (V)

that incorporating relation signals are not particularly helpful for event extraction on MATRES. We speculate that one of the reasons could be the unique event characteristics in MATRES. As we described in Section 5.1, all events in MATRES are verbs. It is possible that a more concentrated single-task model works better when events are homogeneous, whereas a multi-task model is more powerful when we have a mixture of event types, e.g., both verbs and nouns as in TB-Dense.

### 6.3 Relation Extraction with Gold Events

**TB-Dense.** There is much prior work on relation extraction based on gold events in TB-Dense. Meng and Rumshisky (2018) proposed a neural model with global information that achieved the best results as far as we know. The improvement of our single-task model over that baseline is mostly attributable to the adoption of BERT embedding. We show that sharing the LSTM layer for both events and relations can help further improve performance of the relation classification task by 2.6%. For the joint models, since we do not train them on gold events, the evaluation would be meaningless. We simply skip this evaluation.

**MATRES.** Both single-task and multi-task models outperform the baseline by nearly 10%, while the improvement of multi-task over single task is marginal. In MATRES, a relation pair is equivalent to a verb pair, and thus the event prediction task probably does not provide much more information for relation extraction.

In Table 4 we further show the breakdown performances for each positive relation on TB-Dense. The breakdown on MATRES is shown in Table 10 in the appendix. *BEFORE*, *AFTER* and *VAGUE* are the three dominant label classes in TB-Dense. We observe that the linguistic rule-based model, CAEVO, tends to have a more evenly spread-out

Labels	TB-Dense	MATRES
<i>BEFORE</i>	384	417
<i>AFTER</i>	274	266
<i>INCLUDES</i>	56	-
<i>IS_INCLUDED</i>	53	-
<i>SIMULTANEOUS</i>	22	31
<i>VAGUE</i>	638	113

Table 5: Label Size Breakdown in the Test Data

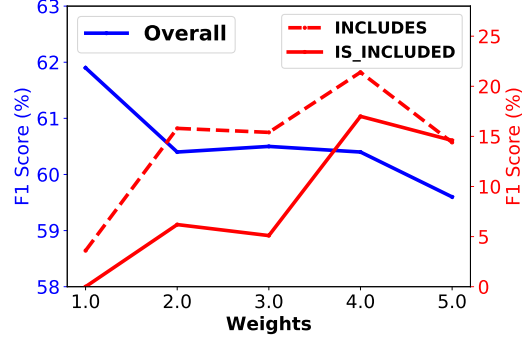


Figure 3: Performances from a single-task relation model under different class weights. Left-axis: overall model; right-axis: two minority relations.

performance, whereas our neural network-based models are more likely to have concentrated predictions due to the imbalance of the training sample across different label classes.

### 6.4 Discussion

**Label Imbalance.** One way to mitigate the label imbalance issue is to increase the sample weights for small classes during model training. We investigate the impact of class weights by refitting our single-task model with larger weights on *INCLUDES*, *IS\_INCLUDED* and *VAGUE* in the cross-entropy loss.

Figure 3 shows that increasing class weights up to 4 times can significantly improve the F1 scores of *INCLUDES* and *IS\_INCLUDED* classes with a decrease less than 2% for the overall F1 score. Performance of *INCLUDES* and *IS\_INCLUDED* eventually degrades when class weights are too large. These results seem to suggest that more labels are needed in order to improve the performance on both of these two classes and the overall model. For *SIMULTANEOUS*, our model does not make any correct predictions for both TB-DENSE and MATRES by increasing class weight up to 10 times, which implies that *SIMULTANEOUS* could be a hard temporal relation to predict in general.

**Global Constraints.** In Table 6 we conduct an ablation study to understand the contributions from the event-relation prediction consis-



Micro-average	TB-Dense	MATRES
No Structure	48.5	58.5
+ $\mathcal{ER}$ Consistency	49.4	59.5
+ Transitivity	49.4	59.6

Table 6: Ablation Study on Global Constraints

tency constraint and the temporal relation transitivity constraint for the structured joint model. As we can see, the event-relation consistency helps improve the F1 scores by 0.9% and 1% for TB-Dense and MATRES, respectively, but the gain by using transitivity is either non-existing or marginal. We hypothesize two potential reasons: 1) We leveraged BERT contextualized embedding as word representation, which could tackle transitivity in the input context; 2) *NONE* pairs could make transitivity rule less useful, as positive pairs can be predicted as *NONE* and transitivity rule does not apply to *NONE* pairs.

**Error Analysis.** By comparing gold and predicted labels for events and temporal relations and examining predicted probabilities for events, we identified three major sources of mistakes made by our structured model, as illustrated in Table 7 with examples.

**Type 1.** Both events in Ex 1 are assigned low scores by the event module ( $<< 0.01$ ). Although the structured joint model is designed to predict events and relations jointly, we leverage the event module to filter out tokens with scores lower than a threshold. Consequently, some true events can be mistakenly predicted as non-events, and the relation pairs including them are automatically assigned *NONE*.

**Type 2.** In Ex 2 the event module assigns high scores to tokens **happened** (0.97) and **according** (0.89), but **according** is not an event. When the structured model makes inference jointly, the decision will weigh heavily towards assigning **1** (event) to both tokens. With the event-relation consistency constraint, this pair is highly likely to be predicted as having a positive temporal relation. Nearly all mistakes made in this category follow the same pattern illustrated by this example.

**Type 3.** The existence of *VAGUE* makes temporal relation prediction challenging as it can be easily confused with other temporal relations, as shown in Ex 3. This challenge is compounded with *NONE* in our end-to-end extraction task.

Type 1 and Type 2 errors suggest that building a stronger event detection module will be helpful for

<b>Type 1:</b> Event predicted as non-event	189 pairs
<b>Ex 1.</b> What Microsoft gets are developers around the world <b>working</b> on ideas that could potentially <b>open</b> up Kinect for Windows ...	
<b>Type 2:</b> <i>NONE</i> predicted as true relation	135 pairs
<b>Ex 2.</b> Mr. Netanyahu told Mr. Erdogan that what <b>happened</b> on board the Mavi Marmara was “unintentional” ... , <b>according</b> to the statement.	
<b>Type 3:</b> <i>VAGUE</i> relation	87 pairs
<b>Ex 3.</b> Microsoft <b>said</b> it has identified 3 companies for the China program to run through June. The company <b>gives</b> each participating startup \$ 20,000 to create ...	

Table 7: Error Types Based on MATRES Test Data

both event and temporal relation extraction tasks. To improve the performance on *VAGUE* pairs, we could either build a stronger model that incorporates both contextual information and common-sense knowledge or create datasets with annotations that better separate *VAGUE* from other positive temporal relations.

## 7 Conclusion

In this paper we investigate building an end-to-end event temporal relation extraction system. We propose a novel neural structured prediction model with joint representation learning to make predictions on events and relations simultaneously; this can avoid error propagation in previous pipeline systems. Experiments and comparative studies on two benchmark datasets show that the proposed model is effective for end-to-end event temporal relation extraction. Specifically, we improve the performances of previously published systems by 10% and 6.8% on the TB-Dense and MATRES datasets, respectively.

Future research can focus on creating more robust structured constraints between events and relations, especially considering event types, to improve the quality of global assignments using ILP. Since a better event model is generally helpful for relation extraction, another promising direction would be to incorporate multiple datasets to enhance the performance of our event extraction systems.

## Acknowledgements

This work is supported in part by Contracts W911NF-15-1-0543 and 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

- Steven Bethard. 2013. [Cleartk-TimeML: A minimalist approach to TempEval 2013](#). In *Second Joint Conference on Lexical and Computational Semantics (\*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 10–14. Association for Computational Linguistics.
- Steven Bethard, James H. Martin, and Sara Klingsstein. 2007. [Timelines from text: Identification of syntactic temporal relations](#). In *Proceedings of the International Conference on Semantic Computing, ICSC '07*, pages 11–18, Washington, DC, USA. IEEE Computer Society.
- Philip Bramsen, Pawan Deshpande, Yoong Keok Lee, and Regina Barzilay. 2006. [Inducing temporal graphs](#). In *EMNLP*, Sydney, Australia.
- Taylor Cassidy, Bill McDowell, Nathanael Chambers, and Steven Bethard. 2014. [An annotation framework for dense event ordering](#). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 501–506. Association for Computational Linguistics.
- Nate Chambers. 2013. [Navytime: Event and time ordering from raw text](#). In *Second Joint Conference on Lexical and Computational Semantics (\*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 73–77, Atlanta, Georgia, USA. Association for Computational Linguistics.
- Nathanael Chambers, Taylor Cassidy, Bill McDowell, and Steven Bethard. 2014. [Dense event ordering with a multi-pass architecture](#). In *ACL*.
- Nathanael Chambers and Dan Jurafsky. 2008. [Jointly combining implicit constraints improves temporal ordering](#). In *EMNLP*, Honolulu, United States.
- Nathanael Chambers, Shan Wang, and Dan Jurafsky. 2007. [Classifying temporal relations between events](#). In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, ACL '07*, pages 173–176, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Fei Cheng and Yusuke Miyao. 2017. Classifying temporal relations by bidirectional LSTM over dependency paths. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 1–6.
- Pascal Denis and Philippe Muller. 2011. [Predicting globally-coherent temporal structures from texts via endpoint inference and graph decomposition](#). In *IJCAI*, Barcelone, Spain.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.
- Quang Xuan Do, Wei Lu, and Dan Roth. 2012. [Joint inference for event timeline construction](#). In *EMNLP*, Jeju, Korea.
- Xiaocheng Feng, Lifu Huang, Duyu Tang, Heng Ji, Bing Qin, and Ting Liu. 2016. [A language-independent neural network for event detection](#). pages 66–71.
- Natsuda Laokulrat, Makoto Miwa, Yoshimasa Tsuruoka, and Takashi Chikayama. 2013. [UTTime: Temporal relation classification using deep syntactic features](#). In *Second Joint Conference on Lexical and Computational Semantics (\*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 88–92, Atlanta, Georgia, USA. Association for Computational Linguistics.
- Artuur Leeuwenberg and Marie-Francine Moens. 2017. Structured learning for temporal relation extraction from clinical records. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 1150–1158.
- Qi Li and Heng Ji. 2014. [Incremental joint extraction of entity mentions and relations](#). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 402–412, Baltimore, Maryland. Association for Computational Linguistics.
- Inderjeet Mani, Marc Verhagen, Ben Wellner, Chong Min Lee, and James Pustejovsky. 2006. [Machine learning of temporal relations](#). In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics, ACL-44*, pages 753–760, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Yuanliang Meng and Anna Rumshisky. 2018. Context-Aware neural model for temporal information extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.
- Yuanliang Meng, Anna Rumshisky, and Alexey Romanov. 2017. Temporal information extraction for question answering using syntactic dependencies in an lstm-based architecture. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 887–896.
- Makoto Miwa and Mohit Bansal. 2016. [End-to-end relation extraction using LSTMs on sequences and tree structures](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Berlin,

- Germany. Association for Computational Linguistics.
- Makoto Miwa and Yutaka Sasaki. 2014. [Modeling joint entity and relation extraction with table representation](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1858–1869, Doha, Qatar. Association for Computational Linguistics.
- Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. [Joint event extraction via recurrent neural networks](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 300–309, San Diego, California. Association for Computational Linguistics.
- Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with convolutional neural networks. In *ACL*.
- Qiang Ning, Zhili Feng, and Dan Roth. 2017. [A structured learning approach to temporal relation extraction](#). In *EMNLP*.
- Qiang Ning, Zhili Feng, Hao Wu, and Dan Roth. 2018a. [Joint reasoning for temporal and causal relations](#). In *ACL*.
- Qiang Ning, Hao Wu, and Dan Roth. 2018b. [A Multi-Axis annotation scheme for event temporal relations](#). In *ACL*.
- Qiang Ning, Ben Zhou, Zhili Feng, Haoruo Peng, and Dan Roth. 2018c. CogCompTime: A tool for understanding time in natural language. In *EMNLP*.
- Tim O’Gorman, Kristin Wright-Bettner, and Martha Palmer. 2016. [Richer Event Description: Integrating event coreference with temporal, causal and bridging annotation](#). In *Proceedings of 2nd Workshop on Computing News Storylines*, pages 47–56. Association for Computational Linguistics.
- James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, and Lisa Ferro. 2003. The TIMEBANK corpus. In *Corpus linguistics*, pages 647–656.
- Julien Tourille, Olivier Ferret, Aurelie Neveol, and Xavier Tannier. 2017. Neural architecture for temporal relation extraction: a bi-lstm approach for detecting narrative containers. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 224–230.
- Naushad UzZaman, Hector Llorens, Leon Derczynski, James Allen, Marc Verhagen, and James Pustejovsky. 2013. [SemEval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations](#). In *Second Joint Conference on Lexical and Computational Semantics (\*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 1–9. Association for Computational Linguistics.
- Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz, and James Pustejovsky. 2007. [SemEval-2007 task 15: TempEval temporal relation identification](#). In *Proceedings of the 4th International Workshop on Semantic Evaluations, SemEval ’07*, pages 75–80, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Marc Verhagen and James Pustejovsky. 2008. [Temporal processing with the TARSQI toolkit](#). In *22Nd International Conference on Computational Linguistics: Demonstration Papers, COLING ’08*, pages 189–192, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Marc Verhagen, Roser Sauri, Tommaso Caselli, and James Pustejovsky. 2010. [SemEval-2010 task 13: Tempeval-2](#). In *Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval ’10*, pages 57–62, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Katsumasa Yoshikawa, Sebastian Riedel, Masayuki Asahara, and Yuji Matsumoto. 2009. Jointly identifying temporal relations with Markov logic. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 405–413. Association for Computational Linguistics.
- Meishan Zhang, Yue Zhang, and Guohong Fu. 2017. [End-to-end neural relation extraction with global optimization](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1730–1740, Copenhagen, Denmark. Association for Computational Linguistics.