# Extracting Temporal Event Relation with Syntax-guided Graph Transformer

**Anonymous ACL submission**


## Abstract

Extracting temporal relations (e.g., before, after, and simultaneous) among events is crucial to natural language understanding. One of the key challenges of this problem is that when the events of interest are far away in text, the context in-between often becomes complicated, making it challenging to resolve the temporal relationship between them. This paper thus proposes a new Syntax-guided Graph Transformer network (SGT) to mitigate this issue, by (1) explicitly exploiting the connection between two events based on their dependency parsing trees, and (2) automatically locating temporal cues between two events via a novel syntax-guided attention mechanism. Experiments on two benchmark datasets, MATRES and TB-DENSE, show that our approach significantly outperforms previous state-of-the-art methods on both end-to-end temporal relation extraction and temporal relation classification with up to 7.9% absolute F-score gain; this improvement also proves to be robust on the contrast set of MATRES. We will make all the programs publicly available once the paper is accepted.


## 1 Introduction

Temporal relationship, e.g., *Before*, *After*, and *Simultaneous*, is important for understanding the process of complex events and reasoning over them. Extracting temporal relationship automatically from text is thus an important component in many downstream applications, such as summarization (Jiang et al., 2011; Ng et al., 2014), dialog understanding and generation (Ritter et al., 2010; Shi et al., 2019), and reading comprehension (Harabagiu and Bejan, 2005; Sun et al., 2018; Ning et al., 2020). While event mentions can often be detected reasonably well (Lin et al., 2020; Huang and Ji, 2020), extracting event-event relationships, especially temporal relationship, still remains challenging (Chen et al., 2021).

**S1:** Now, Lockheed Martin which (***e1:* bought**) an early version of such a computer from the Canadian company D-Wave systems two years ago is confident enough in the technology to upgrade it to commercial scale, becoming the first company to (***e2:* use**) quantum computing as part of its business.

**bought** $\xrightarrow{relcl}$ Matin $\xrightarrow{nsubj}$ is $\xrightarrow{advcl}$ becoming $\xrightarrow{attr}$ company $\xrightarrow{relcl}$ **use**

Temporal Relation ($e_1 \rightarrow e_2$): ***Before***

**S2:** Mr. Erdogan' s office (***e1:* said**) he had (***e2:* accepted**) the apology , " In the name of the Turkish people ".

Temporal Relation ($e_1 \rightarrow e_2$): ***AFTER***

**S3:** "The desk thing really (***e1:* stuck**) with me ", Ms. Ayotte (***e2:* said**).

Temporal Relation ($e_1 \rightarrow e_2$): ***Before***

Figure 1: Examples of temporal relation annotations. Event mentions are boldfaced, the temporal relations between these events are listed below each sentence, and the temporal cues deciding those temporal relations are highlighted in red.

Recent studies (Han et al., 2019b; Ning et al., 2019a; Vashishtha et al., 2019; Wang et al., 2020a) have shown improved performance in temporal relation extraction by leveraging the contextual representations learned from pre-trained language models (Devlin et al., 2018; Liu et al., 2019). However, one remaining challenge of this task is that it requires accurate characterization of the connection between two event mentions and the cues indicating their temporal relationship, especially when the context is wide and complicated. For instance, by manually examining 200 examples of human annotated temporal relations from the MATRES (Ning et al., 2018) dataset, we find that about 52% of the temporal cues[1] come from the connection between two event mentions (e.g., S1 in Fig. 1), 39% from their surrounding contexts (S2 in Fig. 1) and the remaining 9% from others, e.g., event co-reference or subordinate clause structures (S3 in Fig. 1).

Syntactic features, such as dependency parsing trees, have proved to be effective in characterizing

[1]Temporal cues refer to the words of which the semantic meaning or related syntactic relations can determine the temporal relation of two event mentions.

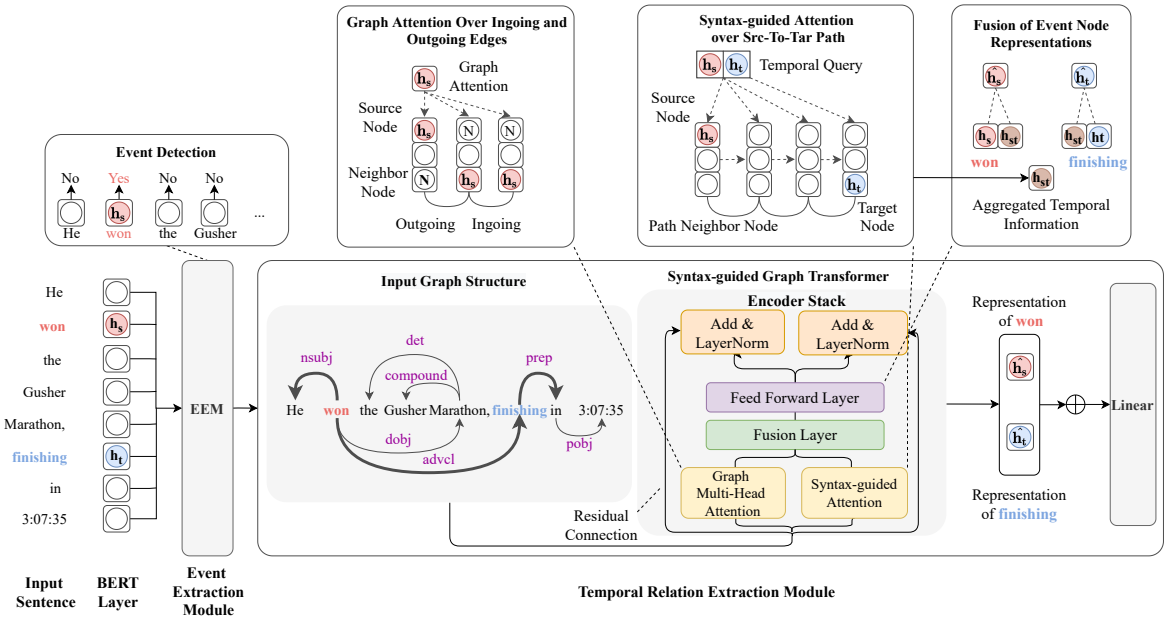


Figure 2: Architecture overview. The tokens highlighted with red and blue colors in the Input Sentence show the source and target events to be detected. The bold edges in the Input Graph Structure indicate the triples from the dependency path between the source and target event mentions as well as their surrounding context, and are considered by the syntax-guided attention.

the connection of two event mentions in pre-neural methods (Chambers, 2013; Chambers et al., 2014; Mirza and Tonelli, 2016). However, how to make use of these features has been under-explored since the adoption of neural methods in this field. This paper closes this gap with a novel Syntax-guided Graph Transformer (SGT) network – in addition to the attention heads in a typical Graph Transformer, we bring in a new attention mechanism that specifically looks at the path from a source node to a target node over dependency parsing trees. SGT thus not only learns event representations as in a typical Graph Transformer, but also provides a way to represent syntactic dependency information between a pair of events (for temporal relation extraction, this means attending to the aforementioned temporal cues). We conduct experiments on two benchmark datasets, MATRES (Ning et al., 2018) and TB-DENSE (Cassidy et al., 2014) on both end-to-end temporal relation extraction and classification, which demonstrate the effectiveness of SGT with up to 7.9% absolute F-score gain over previous state-of-the-art methods. Experiments on the contrast set (Gardner et al., 2020) of MATRES further proves the robustness of our approach.

## 2 Approach

Figure 2 shows the overview of our approach. Given an input sentence $\tilde{s} = [w_1, w_2, ..., w_n]$ with $n$ tokens, we aim to detect a set of event mentions $\{e_1, e_2, ...\}$ where each event mention $e_i$ may contain one or multiple tokens by leveraging the contextual representations learned from a pre-trained BERT (Devlin et al., 2018) encoder. Then, following previous studies (Ning et al., 2019b,a; Han et al., 2019b; Wang et al., 2020a), we consider each pair of event mentions that are detected from one or two continuous sentences, and predict their temporal relationship.

To effectively capture the temporal cues between two event mentions, we build a dependency graph from one or two input sentences and design a new Syntax-guided Graph Transformer network to automatically learn a new contextual representation for each event mention by considering the triples that they are locally involved as well as the triples along the dependency path of the two event mentions within the dependency graph. Finally, the two event mention representations are concatenated to predict their temporal relationship.

### 2.1 Sequence Encoder

Given an input sentence $\tilde{s} = [w_1, w_2, ..., w_n]$, we apply the same tokenizer as BERT (Devlin et al., 2018) to get all the subtokens. Then, we feed the sequence of subtokens as input to a pre-trained BERT model to get a contextual representation for each token $w_i$. If a token $w_i$ is split into multiple

subtokens, we use the contextual representation of the first subtoken to represent $w_i$. To enrich the contextualized representations, for each token, we create a one-hot Part-of-Speech (POS) tag vector and concatenate it with BERT contextual embeddings. In this way, we obtain a final representation $\boldsymbol{c}_i$[2] for each $w_i$. These representations will be later used for event mention detection and also as the initial representations to our syntax-guided graph transformer network.

### 2.2 Event Detection

To detect event mentions from the sentence, we take the contextual representation of each word as input to a binary linear classifier to determine whether it is an event mention or not, which is optimized by minimizing the following binary cross-entropy loss:

$$\tilde{\boldsymbol{y}}_i = \text{softmax}(\boldsymbol{W}_{eve}\boldsymbol{c}_i + \boldsymbol{b}_{eve})$$

$$\mathcal{L}_{eve} = -\sum_{\tilde{s}\in\mathcal{S}}\sum_{i=1}^{|\tilde{s}|}\sum_{\pi\in\{0,1\}} \alpha_\pi y_{i,\pi}\log(\tilde{y}_{i,\pi})$$

where $\mathcal{L}_{eve}$ denotes the cross-entropy loss for event detection. $\mathcal{S}$ is the set of sentences in the training dataset. $\alpha_\pi$ is a weight coefficient for each class (0 or 1) to mitigate the data imbalance problem and $\alpha_0 + \alpha_1 = 1$. $y_{i,\pi}$ is a binary indicator to show whether $\pi$ is the same as the groundtruth binary label ($y_{i,\pi} = 1$) or not ($y_{i,\pi} = 0$). $\tilde{y}_{i,\pi}$ denotes the probability of the $i$-th token in $s$ being predicted with a binary class label $\pi$. $\boldsymbol{W}_{eve}$ and $\boldsymbol{b}_{eve}$ are learnable parameters.

### 2.3 Syntax-guided Graph Transformer

From the example sentences in Fig. 1, the temporal cues for characterizing the temporal relationship between two event mentions mainly come from their surrounding contexts as well as their connections from their syntactic dependency path. However, a sequence encoder usually fails to capture such information, especially when the context between two event mentions is complicated, thus we further design a new Syntax-guided Graph Transformer (SGT) network.

Given a source event $e_s$ and a target event $e_t$ detected from one or two continuous sentences, we apply a public dependency parser[3] to parse each sentence into a tree-graph and connect the graphs of two continuous sentences with an arbitrary *cross-sentence* edge (Peng et al., 2017; Cheng and Miyao, 2017) pointing from the root node of the preceding sentence to the root node of the following one, and obtain a graph $G = (V, E)$. For each node $v_i$, we use $\mathcal{N}_i^{in} = \{(v_k, r_{ki}, v_i) \in E | v_k, v_i \in V\}$ and $\mathcal{N}_i^{out} = \{(v_i, r_{ij}, v_j) \in E | v_i, v_j \in V\}$ to denote all the neighbor triples of $v_i$ with in-going and out-going edges respectively, $r \in \Upsilon$ where $\Upsilon$ is the label set for syntactic dependency relation, and use $\mathcal{P}_{ij} = \{(v_i, r_{ig}, v_g), ..., (v_h, r_{hj}, v_j)\}$ as the triple set along the path from $v_i$ to $v_j$.

**Node Representation Initialization** For each node $v_i$ in graph $G$, we map it to a particular token $w_{i'}$ from the original sentence and obtain a contextual representation $\boldsymbol{c}_{i'}$ from the BERT encoder. Then, we learn an initial node representation for each node $v_i$ as:

$$\boldsymbol{h}_i^0 = \boldsymbol{W}_e\boldsymbol{c}_{i'} + \boldsymbol{b}_e$$

where $\boldsymbol{W}_e$ and $\boldsymbol{b}_e$ are learnable parameters.

**Graph Multi-head Self-attention** Following transformer model (Vaswani et al., 2017; Wang et al., 2020b), we adapt the multi-head self-attention to learn a contextual representation for each node in the graph $G$. Each node $v_i$ in graph $G$ is associated with a set of neighbor triples $\mathcal{N}_i^{in} \cup \mathcal{N}_i^{out}$ and a node representation $\boldsymbol{h}_i^{l-1}$ where $l$ is the index of a layer in our transformer architecture. To perform self-attention, we first apply a linear transformation to obtain a query vector based on each node $v_i$, and employ another two linear transformations to get the key and value vectors based on the node's neighbor triples:

$$\boldsymbol{Q}_i^l = \boldsymbol{W}_q^m\boldsymbol{h}_i^{l-1}$$

$$\boldsymbol{K}_{ij}^l = \boldsymbol{W}_k^m\boldsymbol{R}_{ij}^{l-1}$$

$$\boldsymbol{U}_{ij}^l = \boldsymbol{W}_u^m\boldsymbol{R}_{ij}^{l-1}$$

$$\boldsymbol{R}_{ij}^{l-1} = \boldsymbol{W}_r^m(\boldsymbol{h}_i^{l-1}\big|\big|\boldsymbol{r}_{ij}\big|\big|\boldsymbol{h}_j^{l-1}) + \boldsymbol{b}_r^m$$

where $m$ is the index of a particular head. $\boldsymbol{Q}_i^l$ denotes a query vector corresponding to node $v_i$, $\boldsymbol{K}_{ij}^l$ and $\boldsymbol{U}_{ij}^l$ is a key and value vector respectively, and both of them are learned from a triple $(v_i, r_{ij}, v_j) \in \mathcal{N}_i^{in} \cup \mathcal{N}_i^{out}$, which is represented as $\boldsymbol{R}_{ij}$. $m$ is the index of a particular head. $||$ denotes the concatenation operation. $\boldsymbol{r}_{ij}$ denotes the representation of a particular relation $r_{ij}$ between $v_i$ and $v_j$, which is randomly initialized and optimized

[2] We use bold lower case symbols to denote vectors.
[3] https://spacy.io/api/dependencyparser

by the model. $\boldsymbol{W}_q^m$, $\boldsymbol{W}_k^m$, $\boldsymbol{W}_u^m$, $\boldsymbol{W}_r^m$ and $\boldsymbol{b}_r^m$ are learnable parameters.

For each node $v_i$, we then perform self-attention over all the neighbor triples that it is involved, and compute a new context representation with multiple attention heads:

$$\boldsymbol{g}_i^l = (\Big\|_m^M \text{Head}_i^m)\boldsymbol{W}_o$$

$$\text{Head}_i^m = \text{softmax}(\frac{\boldsymbol{Q}_i^l(\boldsymbol{K}^l)^\top}{\sqrt{d_k}})\boldsymbol{U}^l$$

where $\boldsymbol{g}_i^l$ is the aggregated representation computed over all neighbor triples of node $v_i$ with $M$ attention heads at $l$-th layer. $\boldsymbol{g}_i^l$ will be later used to learn the updated representation of node $v_i$. $\sqrt{d_k}$ is the scaling factor denoting the dimension size of each key vector. $\boldsymbol{W}_o$ is a learnable parameter.

**Syntax-guided Attention** To automatically find the indicative temporal cues for two event mentions from their connection as well as surrounding contexts, we design a new syntax-guided attention mechanism. For two event nodes $v_s$ and $v_t$, we first extract the set of nodes from the dependency path between $v_s$ and $v_t$ (including $v_s$ and $v_t$), which is denoted as $\Theta_{st}$. We then get all the triples from the dependency path between $v_s$ and $v_t$ as well as the triples that any node from $\Theta_{st}$ is involved, which are denoted as $\Phi_{st} = \cup_{v_i \in \Theta_{st}}\{\mathcal{N}_i^{in} \cup \mathcal{N}_i^{out}\} \cup \mathcal{P}_{st}$. To compute the syntax-guided attention over all the triples from $\Phi_{st}$, we apply three linear transformations to get the query, key and value vectors where the query vector is obtained from the representation of two event mentions, and key and value vectors are computed from the triples in $\Phi_{st}$:

$$\tilde{\boldsymbol{Q}}_{st}^l = \tilde{\boldsymbol{W}}_q^m \cdot (\boldsymbol{h}_s^{l-1} \big\| \boldsymbol{h}_t^{l-1})^x$$

$$\tilde{\boldsymbol{K}}_{ij}^l = \tilde{\boldsymbol{W}}_k^m \tilde{\boldsymbol{R}}_{ij}^{l-1}$$

$$\tilde{\boldsymbol{U}}_{ij}^l = \tilde{\boldsymbol{W}}_u^m \tilde{\boldsymbol{R}}_{ij}^{l-1}$$

$$\tilde{\boldsymbol{R}}_{ij}^{l-1} = \tilde{\boldsymbol{W}}_r^m (\boldsymbol{h}_i^{l-1} \big\| \boldsymbol{r}_{ij} \big\| \boldsymbol{h}_j^{l-1}) + \tilde{\boldsymbol{b}}_r$$

where $m$ is the index of a particular head, $\tilde{\boldsymbol{Q}}_{st}^l, \tilde{\boldsymbol{K}}_{ij}^l, \tilde{\boldsymbol{U}}_{ij}^l$ denote the query, key and value vectors respectively. $\tilde{\boldsymbol{R}}_{ij}^{l-1}$ is the representation of a triple $(v_i, r_{ij}, v_j) \in \Phi_{st}$. $\tilde{\boldsymbol{W}}_q^m$, $\tilde{\boldsymbol{W}}_k^m$, $\tilde{\boldsymbol{W}}_v^m$ and $\tilde{\boldsymbol{W}}_r^m$ are learnable parameters.

Given the query vector, we then compute the attention distribution over all triples from $\Phi_{st}$ and get an aggregated representation to denote the meaningful temporal features captured from the connection between two event mentions and their surrounding contexts.

$$\tilde{\boldsymbol{g}}_{st}^l = (\Big\|_m^M \tilde{\text{Head}}_{st}^m) \cdot \tilde{\boldsymbol{W}}_p$$

$$\tilde{\text{Head}}_{st}^m = \text{softmax}(\frac{\tilde{\boldsymbol{Q}}_{st}^l(\tilde{\boldsymbol{K}}^l)^\top}{\sqrt{d_k}}) \cdot \tilde{\boldsymbol{U}}^l$$

where $\tilde{\boldsymbol{g}}_{st}^l$ is the aggregated temporal related information from all the triples in $\Phi_{st}$ based on the syntax-guided attention at $l$-th layer. $\boldsymbol{W}_p$ is a learnable parameter.

**Node Representation Fusion** Each event node in graph $G$ will receive two representations learned from the multi-head self-attention and syntax-guided attention, thus we further fuse the two representations for both the source node $v_s$ and the target node $v_t$:

$$\hat{\boldsymbol{h}}_s^l = \tilde{\boldsymbol{W}}_f(\boldsymbol{g}_s^l \big\| \tilde{\boldsymbol{g}}_{st}^l)\ , \quad \hat{\boldsymbol{h}}_t^l = \tilde{\boldsymbol{W}}_f(\tilde{\boldsymbol{g}}_{st}^l \big\| \boldsymbol{g}_t^l)$$

where $\boldsymbol{g}_s^l$ and $\boldsymbol{g}_t^l$ denote the context representations learned from the multi-head self-attention for $v_s$ and $v_t$. $\tilde{\boldsymbol{g}}_{st}^l$ denotes the representation learned from the triples from $\Phi_{st}$ using syntax-guided attention. $\hat{\boldsymbol{h}}_s^l$ and $\hat{\boldsymbol{h}}_t^l$ are the fused representations of $v_s$ and $v_t$, respectively. $\tilde{\boldsymbol{W}}_f$ is a learnable parameter.

For each non-event node $v_i$, which only receives a context representation $\boldsymbol{g}_i^l$ learned from the multi-head self-attention, we apply a linear projection and get a new node representation:

$$\hat{\boldsymbol{h}}_i^l = \boldsymbol{W}_t \boldsymbol{g}_i^l$$

Our Syntax-guided Graph Transformer encoder is composed of a stack of multiple layers, while each layer consists of the two attention mechanisms and the fusion sub-layer. We use residual connection followed by LayerNorm for each layer to get the final representations of all the nodes:

$$\boldsymbol{H}^l = \text{LayerNorm}(\hat{\boldsymbol{H}}^l + \boldsymbol{H}^{l-1})$$

### 2.4 Temporal Relation Prediction

To predict the temporal relation between two event mentions $e_s$ and $e_t$, we concatenate the final hidden states of $v_s$ and $v_t$ obtained from the Syntax-guided Graph Transformer network, and apply a Feedforward Neural Network (FNN) to predict their relationship

$$\tilde{\boldsymbol{y}}_{st} = \text{softmax}(\boldsymbol{W}_z(\boldsymbol{h}_s^L \big\| \boldsymbol{h}_t^L) + \boldsymbol{b}_t)$$

where $\tilde{\boldsymbol{y}}_{st}$ denotes the probabilities over all possible temporal relations between event mentions $e_s$ and $e_t$.

The training objective is to minimize the following cross-entropy loss function:

$$\mathcal{L}_{rel} = -\sum_{st \in \Delta} \sum_{x \in X} \beta_x y_{st,x} \log(\tilde{y}_{st,x}))$$

where $\Delta$ denotes the total set of event pairs for temporal relation prediction and $X$ denotes the whole set of relation labels. $y_{st,x}$ is a binary indicator (0 or 1) to show whether $x$ is the same as the groundtruth label ($y_{st,x} = 1$) or not ($y_{st,x} = 0$). We also assign a weight $\beta_x$ to each class to mitigate the label imbalance issue.

## 3 Experiment

### 3.1 Experimental Setup

We perform experiments on two public benchmark datasets for temporal relation extraction: (1) TB-DENSE (Cassidy et al., 2014), which is a densely annotated dataset with 6 types of relations: *Before*, *After*, *Simultaneous*, *Includes*, *Is_included* and *Vague*. (2) MATRES (Ning et al., 2018), which annotates verb event mentions along with 4 types of temporal relations: *Before*, *After*, *Simultaneous* and *Vague*. Additionally, we use POS tag information from MATRES provided by (Ning et al., 2019b). For TB-DENSE, we use spacy annotation for predicting POS tag information which is based on Universal POS tag set[4]. For both benchmark datasets, we use the same train/dev/test splits as previous studies (Ning et al., 2019b,a; Han et al., 2019a,b). Note that, for evaluation, similar as previous work, we disregard the *Vague* relation from both datasets (in the evaluation phase, we simply remove all ground truth *Vague* relation pairs). In addition, we will only consider event pairs from adjacent sentences due to the fact that it will require an exponential number of annotations if we also consider event pairs from non-adjacent sentences, which is beyond the scope of this study. Appendix A shows additional details of the two datasets and Appendix B provides implementation details.

### 3.2 Results

We evaluate SGT against two public benchmark datasets under two settings: (1) joint event and temporal relation extraction (Table 1); (2) temporal relation classification, where the gold event mentions are known beforehand (Table 2). Note in the "joint" setting, we adopt the same strategy proposed in (Han et al., 2019b): we first train the event extraction module, and then jointly optimize both event extraction and temporal relation extraction (using gold event pairs as input to ensure training quality) modules. Overall, we observe that our approach significantly outperforms baseline systems in both settings, with up to 7.9% absolute F-score gain on MATRES and 2.4% on TB-DENSE.

From Table 1, we see that our approach achieves better performance on event detection than baseline methods though they are based on the same BERT encoder. This is possibly because, during joint training, our approach leverages the dependency parsing trees, which improves the contextual representations of the BERT encoder. In Table 2, unlike other models which are based on larger contextualized embeddings such as RoBERTa, our approach with BERT base achieves comparable performance, and further surpasses the state-of-the-art baseline methods using BERT-large embeddings, which demonstrate the effectiveness of our Syntax-guided Graph Transformer network.

Some studies (Ning et al., 2019b; Han et al., 2019b; Wang et al., 2020a; Zhou et al., 2020) focus on resolving the inconsistency in terms of the symmetry and transitivity of the temporal relations. For example, if event A and event B are predicted as *Before*, event B and event C are predicted as *Before*, then if event A and event C are predicted as *Vague* or *After*, it will be considered as inconsistent. However, our approach shows consistent predictions with few inconsistent cases when *Simultaneous* relation is involved. This analysis also demonstrates that our approach can correctly capture the temporal cues between two event mentions.

We also examine the correctness and robustness of our approach on a contrast set of MATRES (Gardner et al., 2020), which is created with small manual perturbation based on the original test set of MATRES in a meaningful way, such as rephrasing the sentence or simply changing a word of the sentence to alter the relation type. The contrast set provides a local view of a model's decision boundary, thus it can be used to more accurately evaluate a model's true linguistic capabilities. Table 3 shows that our approach significantly outperforms the baseline model on both the original test set and the corresponding contrast set. The contrast consis-

[4] https://spacy.io/api/data-formats

| Dataset | Model | Pre-trained Model | Event Detection | Relation Extraction |
|---|---|---|---|---|
| TB-DENSE | HNP19 (Han et al., 2019b) | BERT Base | 90.9 | 49.4 |
| | Our Approach | BERT Base | **91.0** | **51.8** |
| MATRES | CogCompTime2.0 (Ning et al., 2019b) | BERT Base | 85.2 | 52.8 |
| | HNP19 (Han et al., 2019b) | BERT Base | 87.8 | 59.6 |
| | Our Approach | BERT Base | **90.5** | **62.3** |

Table 1: Comparison of various approaches on joint event and relation extraction with F-score (%). Note that HPN19 fixes BERT embeddings but relies on BiLSTM to capture the contextual features.

| Dataset | Model | Pre-trained Model | Relation Classification (F-score %) |
|---|---|---|---|
| TB-DENSE | LSTM (Cheng and Miyao, 2017) | BERT Base | 62.2 |
| | HNP19 (Han et al., 2019b) | BERT Base | 64.5 |
| | Our Approach | BERT Base | **66.7** |
| | PSL (Zhou et al., 2020) | RoBERTa Large | 65.2 |
| | DEER (Han et al., 2020) | RoBERTa Large | 66.8 |
| | Our Approach | BERT Large | **67.1** |
| MATRES | CogCompTime2.0 (Ning et al., 2019b) | BERT Base | 71.4 |
| | LSTM (Cheng and Miyao, 2017) | BERT Base | 73.4 |
| | HNP19 (Han et al., 2019b) | BERT Base | 75.5 |
| | Our Approach | BERT Base | **79.3** |
| | HMHD20 (Wang et al., 2020a) | RoBERTa Large | 78.8 |
| | DEER (Han et al., 2020) | RoBERTa Large | 79.3 |
| | Our Approach | BERT Large | **80.3** |

Table 2: Comparison of various approaches on temporal relation classification with gold event mentions as input.

| Model | Original Test | Contrast | Consistency |
|---|---|---|---|
| CogCompTime2.0 (Ning et al., 2019b) | 73.2 | 63.3 | 40.6 |
| Our Approach | **82.3** | **69.9** | **49.8** |

Table 3: Evaluation on the contrast set of MATRES. Original Test indicates the F-score on 100 examples sampled from the original MATRES test set. Contrast shows the F-score on 401 examples perturbed from the original 100 examples. Consistency is defined as the percentage of the original 100 examples for which the model's predictions of the perturbed examples are all correct in the contrast set.

tency in Table 3 also indicates how well a model's decision boundary aligns with the actual decision boundary of the test instances, based on which we can see that by explicitly capturing temporal cues, our approach is more accurate and robust than the baseline method.

**Ablation Study** We further conduct ablation studies to compare the performance of our approach with two ablated versions of our method: (1) BERT with Graph Transformer (BERT-GT), for which we remove the syntaxic-guided attention and only rely on the standard multi-head self-attention to obtain graph-based contextual representations of two event mentions and then predict their relation; (2) BERT, where we further remove the Graph Transformer, and only use the pre-trained BERT language model to encode the sentence and predict the temporal relationship of two event mentions based on their contextual representations.

| Ablation | F-score (%) | Gain (%) |
|---|---|---|
| BERT-SGT | 79.3 | 0 |
| BERT-GT | 77.5 | -2.0 |
| BERT | 75.5 | -3.8 |

Table 4: Ablation study on MATRES. We use BERT base as the comparison basis.

Table 4 also shows that by adding Graph Transformer, BERT-GT achieves 2.0% absolute F-score improvement over the BERT baseline model, demonstrating the benefit of dependency parsing trees to temporal relation prediction. By further adding the new syntax-guided attention into Graph Transformer, the absolute improvement on F-score (1.8%) shows the effectiveness of our new Syntax-guided Graph Transformer and the importance of capturing temporal cues from the connection of two event mentions as well as their surround contexts.

Figure 3 shows two examples as qualitative analysis. In S1, BERT mistakenly predicts the temporal relation as *Before* probably because it's confused by the context word *Before*. However, by incorporating the dependency graph, especially the triples {*worked*, *prep*, *Before*}, {*Before*, *pcomp*,

| Prediction | Example |
|---|---|
| ✗ BERT: ***Before***<br>✓ BERT-GT: ***After***<br>✓ BERT-SGT: ***After*** | **S1:** Before (***e₁*: retiring**) in 1984 , Mr. Lowe (***e₂*: worked**) as an inspector of schools with the department of education and sciences , and he leaves three sons from a previous marriage . |
| ✗ BERT: ***Before***<br>✗ BERT-GT: ***Before***<br>✓ BERT-SGT: ***After*** | **S2:** Mr. Erdogan has long (***e₁*: sought**) an apology for the raid in May 2010 on the Mavi Marmara , which was part of a Flotilla that (***e₂*: sought**) to break Israel's blockade of gaza. |

Figure 3: Comparison of the predictions from BERT, BERT-GT and our approach.

*retiring*} and the path between the two event mentions, *worked*→*prep*→*Before*→*pcomp*→*retiring*, both BERT-GT and our approach correctly determine the relation as *After*. In S2, both BERT and BERT-GT mistakenly predict the temporal relation as *Before* as the context between the two event mentions is very wide and complicated, and these two event mentions are not close within the dependency graph. However, by explicitly considering and understanding the connection between the two event mentions, $sought_{e_1}$→*on*→*Marmara*→*was*→*part*→*Flotilla* →$sought_{e_2}$, our approach correctly predicts the temporal relation between the two event mentions.

### 3.3 SGT on Temporal Cues

To analyze the source of temporal cues for relation prediction, we randomly sample 100 correct event relation predictions given gold event mentions from MATRES and select the triple that has the highest temporal attention weight from the last layer of the Syntax-guided Graph Transformer network as a temporal cue candidate. We manually evaluate the validity of each temporal cue candidate, and further analyze if the cue is from the dependency path between two event mentions, their surrounding context, or both. Our analysis shows that about 64% of the temporal cues are valid, 37% of them come from the dependency path, 17% are from local context, and the remaining 10% are from both. This verifies our initial observation that most of the temporal cues are from the dependency path between two event mentions as well as their surrounding context. It also demonstrates the effectiveness of our new syntax-guided attention mechanism.

### 3.4 Impact of Wide Context

We further illustrate the impact of context width to both baseline model and our approach. For fair comparison, we use three context width category, [context length $<$ 10, 10 $<$ context length $<$ 20, context length $>$ 20 ]. As we can see in Fig. 4, the first category has 267 pairs, the second category has 343 pairs and the third category has 817 pairs. From our results, we observe that the BERT baseline cannot predict the temporal relation of two event mentions with wide context but rather working well when the event mentions are close to each other. Our model overall performs slightly worse in the second category but in general is very good at predicting the temporal relationship for the event mentions with short and context width. This also proves the benefit of syntactic parsing trees to the prediction of temporal relationship. For the second category where the context length is within [10, 20], the performance of our approach slightly drops due to two reasons: (1) the training samples within this range are not as sufficient as the other two categories; (2) for most event pairs from this category, their dependency path is very long and there is no explicit temporal indicative features within their context or dependency path, making it more difficult for the model to predict their temporal relationship.

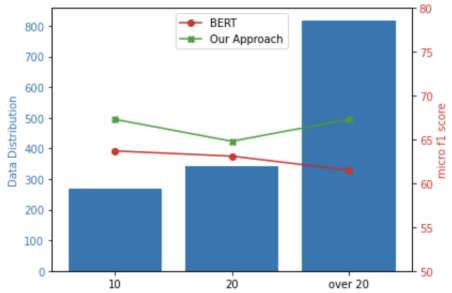


Figure 4: Context width analysis on TB-DENSE. The X axis shows the number of tokens between two events mentions. The left Y axis shows the data distribution of each width category indicating with blue bars. The right Y axis denotes the micro F-score for each width category.

### 3.5 Remaining Errors

We randomly sample 100 classification errors from the output of our approach and categorize them into four categories. As Figure 5 shows, the first category is due to the complex or ambiguous context (54% of the total errors). The second category is due to the complicated subordinate clause structure,

| Error Category (Percent) | Example |
|---|---|
| **Complex Context (54%)** | **S1:** "This is not a Lehman , " he (***e₁*: said**) to the disastrous chain reaction (***e₂*: touched)** off by the collapse of Lehman brothers in 2008 . (*After*) |
| **Subordinate Clause (22%)** | **S2:** "We were pleased that England and New Zealand knew about it, and we (***e₁*: thought**) that's where it would stop." He also (***e₂*: talked**) about his " second job " as the group's cameraman. (*Vague*) |
| **Hypothetical Events and Intentional Events (18%)** | **S3:** The day before Raymond Roth was (***e₁*: pulled**) over, his wife, Ivana, showed authorities emails she had discovered that (***e2*: appeared**) to detail a plan between him and his son to fake his death. (*Vague*) |
| **Imbalanced Labels (6%)** | **S4:** Microsoft (***e₁*: said)** it has identified three companies for the china program to (***e2*: run)** through June . (*Simultaneous*) |

Figure 5: Types of remaining errors

especially the clauses that are related to quote or reported speech, e.g., S2 in Figure 5. The third error category is that our approach cannot correctly differentiate the actual events from the hypothetical and intentional events, while in most cases, the temporal relation among hypothetical and intentional events is annotated as *Vague*. The last category is due to the lack of sufficient annotation. We observe that none of the *Simultaneous* relation can be correctly predicted for MATRES dataset as the percentage of *Simultaneous* (3.7%) is much lower than other relation types. In TB-DENSE dataset, labels are even more imbalanced as the percentage of *Vague* relation is over 50% while the percentage of *Includes*, *Is_Included* and *Simultaneous* are all less than 4%.

## 4 Related Work

Early studies on temporal relation extraction mainly model it as a pairwise classification problem (Mani et al., 2006; Verhagen et al., 2007; Verhagen and Pustejovsky, 2008; Verhagen et al., 2010; Bethard et al., 2016; MacAvaney et al., 2017) and rely on hand-crafted features and rules (Verhagen and Pustejovsky, 2008; Bethard et al., 2007) to extract temporal event relations. Recently, deep neural networks (Dligach et al., 2017; Tourille et al., 2017) and large-scale pre-trained language models (Han et al., 2019a, 2020; Wang et al., 2020a; Zhou et al., 2020) are further employed and show state-of-the-art performance.

Similar to our approach, several studies (Ling and Weld, 2010; Nikfarjam et al., 2013; Mirza and Tonelli, 2016; Meng et al., 2017; Cheng and Miyao, 2017) also explored syntactic path between two events for temporal relation extraction. Different from previous work, our approach considers three important sources of temporal cues: *local context*, denoting the neighbors of each event node within the dependency graph; *connection of two event mentions*, which is based on the dependency path between two event mentions; and *rich semantics of concepts and dependency relations*, for example, the dependency relation *nmod* between two event mentions usually indicates a *Before* relationship. All these indicative features are automatically selected and aggregated with the multi-head self-attention and our new syntax-guided attention mechanism.

Our work is also related to the variants of Graph Neural Networks (GNN) (Kipf and Welling, 2016; Veličković et al., 2017; Zhou et al., 2018), especially Graph Transformer (Yun et al., 2019; Chen et al., 2019; Hu et al., 2020; Wang et al., 2020b). Different from previous GNNs which aim to capture the context from neighbors of each node within the graph, in our task, we aim to select and capture the most meaningful temporal cues for two event mentions from their connections within the graph as well as their surrounding contexts.

## 5 Conclusion

Temporal relationship between events is important for understanding stories described in natural language text, and a main challenge is how to discover and make use of the connection between two event mentions, especially when the event pair is far apart in text. This paper proposes a novel Syntax-guided Graph Transformer (SGT) that represents the connection between an event pair via additional attention heads over dependency parsing trees. Experiments on benchmarking datasets, MATRES, TB-DENSE, and a contrast set of MATRES, show that our approach significantly outperforms previous state-of-the-art methods in a variety of settings, including event detection, temporal relation classification (where events are given), and temporal relation extraction (where events are predicted). In the future, we will investigate the potential of this approach to other relation extraction tasks.

## References


S. Bethard, J. H. Martin, and S. Klingenstein. 2007. Timelines from text: Identification of syntactic temporal relations. In *International Conference on Semantic Computing (ICSC 2007)*, pages 11–18.

Steven Bethard, Guergana Savova, Wei-Te Chen, Leon Derczynski, James Pustejovsky, and Marc Verhagen. 2016. Semeval-2016 task 12: Clinical tempeval. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1052–1062.

Taylor Cassidy, Bill McDowell, Nathanel Chambers, and Steven Bethard. 2014. An annotation framework for dense event ordering. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA.

Nathanael 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. *Transactions of the Association for Computational Linguistics*, 2:273–284.

Benson Chen, Regina Barzilay, and Tommi Jaakkola. 2019. Path-augmented graph transformer network. *arXiv preprint arXiv:1905.12712*.

Muhao Chen, Hongming Zhang, Qiang Ning, Manling Li, Heng Ji, Kathleen McKeown, and Dan Roth. 2021. Event-centric natural language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Tutorial Abstracts*, pages 6–14, Online. 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)*, pages 1–6.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dmitriy Dligach, Timothy Miller, Chen Lin, Steven Bethard, and Guergana Savova. 2017. Neural temporal relation extraction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 746–751.

Matt Gardner, Yoav Artzi, Victoria Basmova, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hanna Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. Evaluating nlp models via contrast sets. *arXiv preprint*.

Rujun Han, I Hsu, Mu Yang, Aram Galstyan, Ralph Weischedel, Nanyun Peng, et al. 2019a. Deep structured neural network for event temporal relation extraction. *arXiv preprint arXiv:1909.10094*.

Rujun Han, Qiang Ning, and Nanyun Peng. 2019b. Joint event and temporal relation extraction with shared representations and structured prediction. *arXiv preprint arXiv:1909.05360*.

Rujun Han, Xiang Ren, and Nanyun Peng. 2020. Deer: A data efficient language model for event temporal reasoning. *arXiv preprint arXiv:2012.15283*.

Sanda Harabagiu and Cosmin Adrian Bejan. 2005. Question answering based on temporal inference. In *Proceedings of the AAAI-2005 workshop on inference for textual question answering*, pages 27–34.

Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous graph transformer. In *Proceedings of The Web Conference 2020*, pages 2704–2710.

Lifu Huang and Heng Ji. 2020. Semi-supervised new event type induction and event detection. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 718–724.

Yexi Jiang, Chang-Shing Perng, and Tao Li. 2011. Natural event summarization. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 765–774.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009.

Xiao Ling and Daniel Weld. 2010. Temporal information extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Sean MacAvaney, Arman Cohan, and Nazli Goharian. 2017. Guir at semeval-2017 task 12: a framework for cross-domain clinical temporal information extraction. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1024–1029.

Inderjeet Mani, Marc Verhagen, Ben Wellner, Chungmin Lee, and James Pustejovsky. 2006. Machine learning of temporal relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 753–760.

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, Copenhagen, Denmark. Association for Computational Linguistics.

Paramita Mirza and Sara Tonelli. 2016. Catena: Causal and temporal relation extraction from natural language texts. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 64–75.

Jun Ping Ng, Yan Chen, Min-Yen Kan, and Zhoujun Li. 2014. Exploiting timelines to enhance multi-document summarization. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 923–933.

Azadeh Nikfarjam, Ehsan Emadzadeh, and Graciela Gonzalez. 2013. Towards generating a patient's timeline: extracting temporal relationships from clinical notes. *Journal of biomedical informatics*, 46:S40–S47.

Qiang Ning, Zhili Feng, and Dan Roth. 2019a. A structured learning approach to temporal relation extraction. *arXiv preprint arXiv:1906.04943*.

Qiang Ning, Sanjay Subramanian, and Dan Roth. 2019b. An improved neural baseline for temporal relation extraction. *arXiv preprint arXiv:1909.00429*.

Qiang Ning, Hao Wu, Rujun Han, Nanyun Peng, Matt Gardner, and Dan Roth. 2020. Torque: A reading comprehension dataset of temporal ordering questions. *arXiv preprint arXiv:2005.00242*.

Qiang Ning, Hao Wu, and Dan Roth. 2018. A multi-axis annotation scheme for event temporal relations. In *ACL*.

Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. 2017. Cross-sentence n-ary relation extraction with graph lstms. *Transactions of the Association for Computational Linguistics*, 5:101–115.

Alan Ritter, Colin Cherry, and William B Dolan. 2010. Unsupervised modeling of twitter conversations. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 172–180.

Weiyan Shi, Tiancheng Zhao, and Zhou Yu. 2019. Unsupervised dialog structure learning. *arXiv preprint arXiv:1904.03736*.

Yawei Sun, Gong Cheng, and Yuzhong Qu. 2018. Reading comprehension with graph-based temporal-casual reasoning. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 806–817.

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)*, pages 224–230.

Siddharth Vashishtha, Benjamin Van Durme, and Aaron Steven White. 2019. Fine-grained temporal relation extraction. *arXiv preprint arXiv:1902.01390*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

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 fourth international workshop on semantic evaluations (SemEval-2007)*, pages 75–80.

Marc Verhagen and James Pustejovsky. 2008. Temporal processing with the tarsqi toolkit. In *COLING 2008: Companion Volume: Demonstrations*, pages 189–192.

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*, pages 57–62.

Haoyu Wang, Muhao Chen, Hongming Zhang, and Dan Roth. 2020a. Joint constrained learning for event-event relation extraction. *arXiv preprint arXiv:2010.06727*.

Tianming Wang, Xiaojun Wan, and Hanqi Jin. 2020b. Amr-to-text generation with graph transformer. *Transactions of the Association for Computational Linguistics*, 8:19–33.

Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. 2019. Graph transformer networks. *arXiv preprint arXiv:1911.06455*.

Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2018. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434*.

Yichao Zhou, Yu Yan, Rujun Han, J Harry Caufield, Kai-Wei Chang, Yizhou Sun, Peipei Ping, and Wei Wang. 2020. Clinical temporal relation extraction with probabilistic soft logic regularization and global inference. *arXiv preprint arXiv:2012.08790*.

# Appendices

## A Details of the Datasets

Table 5 shows statistics of the two datasets and Table 6 shows the label distribution.

| Corpora | | Train | Dev | Test |
|---|---|---|---|---|
| TB-DENSE | # Documents | 22 | 5 | 9 |
| | # Relation Pairs | 4032 | 629 | 1427 |
| MATRES | # Documents | 255 | 20 | 25 |
| | # Relation Pairs | 13K | 2.6K | 837 |

Table 5: Data statistics for TB-DENSE and MATRES

| Labels | TB-DENSE | | MATRES | |
|---|---|---|---|---|
| Before | 384 | 26.9% | 417 | 49.8% |
| After | 274 | 19.2% | 266 | 31.8% |
| Includes | 56 | 3.9% | - | - |
| Is_Included | 53 | 3.7% | - | - |
| Simultaneous | 22 | 1.5% | 31 | 3.7% |
| Vague | 638 | 44.7% | 133 | 15.9% |

Table 6: Label distribution for TB-DENSE and MATRES. For each dataset, the first column shows the number of instances of each relation type while the second column shows the percentage.

## B Implementation Details

For fair comparisons with previous baseline approaches, we use the pre-trained bert-large-cased model[5] for fine-tuning and optimize our model with BertAdam. We optimize the parameters with grid search: training epoch 10, learning rate $\in \{3e\text{-}6, 1e\text{-}5\}$, training batch size $\in \{16, 32\}$, encoder layer size $\in \{4, 12\}$, number of heads $\in \{1, 8\}$. During training, we first optimize the event extraction module for 5 epochs to warm up, and then jointly optimize both event extraction and temporal relation extraction modules using gold event pairs for another 5 epochs.

[5] https://huggingface.co/transformers/pretrained_models.html