Event Time Extraction and Propagation via Graph Attention Networks

Haoyang Wen[1], **Yanru Qu**[1], **Heng Ji**[1], **Qiang Ning**[2]*,
Jiawei Han[1], **Avirup Sil**[3], **Hanghang Tong**[1], **Dan Roth**[4]
[1] University of Illinois at Urbana-Champaign [2]Amazon
[3] IBM Research AI [4]University of Pennsylvania
{wen17, yanruqu2, hengji, hanj, htong}@illinois.edu
qning@amazon.com avi@us.ibm.com danroth@seas.upenn.edu

Abstract

Grounding events into a precise timeline is important for natural language understanding but has received limited attention in recent work. This problem is challenging due to the inherent ambiguity of language and the requirement for information propagation over inter-related events. This paper first formulates this problem based on a 4-tuple temporal representation used in entity slot filling, which allows us to represent fuzzy time spans more conveniently. We then propose a graph attention network-based approach to propagate temporal information over document-level event graphs constructed by shared entity arguments and temporal relations. To better evaluate our approach, we present a challenging new benchmark on the ACE2005 corpus, where more than 78% of events do not have time spans mentioned explicitly in their local contexts. The proposed approach yields an absolute gain of 7.0% in match rate over contextualized embedding approaches, and 16.3% higher match rate compared to sentence-level manual event time argument annotation.[1]

1 Introduction

Understanding and reasoning about *time* is a crucial component for comprehensive understanding of evolving situations, events, trends and forecasting event abstractions for the long-term. Event time extraction is also useful for many downstream Natural Language Processing (NLP) applications such as event timeline generation (Huang and Huang, 2013; Wang et al., 2015; Ge et al., 2015; Steen and Markert, 2019), temporal event tracking and prediction (Ji et al., 2009; Minard et al., 2015), and temporal question answering (Llorens et al., 2015; Meng et al., 2017).

In order to ground events into a timeline we need to determine the start time and end time of each event as precisely as possible (Reimers et al., 2016). However, the start and end time of an event is often not explicitly expressed in a document. For example, among 5,271 annotated event mentions in the Automatic Content Extraction (ACE2005) corpus[2], only 1,100 of them have explicit time argument annotations. To solve the temporal event grounding (TEG) problem, previous efforts focus on its subtasks such as temporal event ordering (Bramsen et al., 2006; Chambers and Jurafsky, 2008; Yoshikawa et al., 2009; Do et al., 2012; Meng et al., 2017; Meng and Rumshisky, 2018; Ning et al., 2017, 2018, 2019; Han et al., 2019) and duration prediction (Pan et al., 2006, 2011; Vempala et al., 2018; Gusev et al., 2011; Vashishtha et al., 2019; Zhou et al., 2019). In this paper we aim to solve TEG directly using the following novel approaches.

To capture fuzzy time spans expressed in text, we adopt a 4-tuple temporal representation proposed in the TAC-KBP temporal slot filling task (Ji et al., 2011, 2013) to predict an event's earliest possible start date, latest possible start date, earliest possible end date and latest possible end date, given the entire document. We choose to work at the day-level and leave time scales smaller than that for future work since, for example, only 0.6% of the time expressions in the newswire documents in ACE contain smaller granularities (e.g., hours or minutes).

Fortunately, the uncertain time boundaries of an event can often be inferred from its related events in the global context of a document. For example, in Table 1, there are no explicit time expressions or clear linguistic clues in the local context to infer the time of the *appeal* event. But the earliest possible date of the *refuse* event is explicitly expressed as 2003-04-18. Since the *appeal* event must happen before the *refuse* event, we can infer

*Work done prior to joining Amazon.

[1] The resource for this paper is available at https://github.com/wenhycs/NAACL2021-Event-Time-Extraction-and-Propagation-via-Graph-Attention-Networks.

[2] https://catalog.ldc.upenn.edu/LDC2006T06

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 62–73
June 6–11, 2021. ©2021 Association for Computational Linguistics

Malaysia' s Appeal Court Friday$_{[2003\text{-}04\text{-}18]}$ refused to overturn the conviction and nine-year jail sentence imposed on ex-deputy prime minister Anwar Ibrahim. Anwar now faces an earliest possible release date of April 14, 2009$_{[2009\text{-}04\text{-}14]}$. The former heir says he was framed for political reasons, after his appeal was rejected ... Mahathir's sacking of Anwar in September 1998$_{[1998\text{-}09]}$ rocked Malaysian politics ... Within weeks he was arrested and charged with ... Anwar was told Monday$_{[2003\text{-}04\text{-}14]}$ that he had been granted a standard one-third remission of a six-year corruption sentence for good behavior, and immediately began to serve the nine-year sentence ...

	Event	Earliest Start Date	Latest Start Date	Earliest End Date	Latest End Date	Evidence
Local Context	sentence	2003-04-14	2003-04-14	-inf	+inf	
	appeal	-inf	+inf	-inf	+inf	
+ Sharing Arguments	sentence	2003-04-14	2003-04-14	**2009-04-14**	+inf	release→Anwar→sentence
	appeal	-inf	+inf	**2003-04-18**	**2003-04-18**	refuse→Anwar→appeal
+ Temporal Relation	sentence	2003-04-14	2003-04-14	2009-04-14	+inf	
	appeal	**1998-09-01**	+inf	2003-04-18	2003-04-18	sack→arrest→appeal

Table 1: Examples of temporal propagation via related events for two target events, *sentence* and *appeal*. By leveraging related events with temporal relations and shared arguments, some infinite dates can be refined with temporal boundaries. *Note*: The event triggers that we are focusing are highlighted in orange, time expressions in blue, and normalized TIMEX dates in subscripts. Related events are underlined.

the earliest start and the latest end date of *appeal* as 2003-04-18. However, there are usually many other irrelevant events that are in the same document, which requires us to develop an effective approach to select related events and perform temporal information propagation. We first use event-event relations to construct a document-level event graph for each input document, as illustrated in Figure 1. We leverage two types of event-event relations: (1) if two events share the same entity as their arguments, then they are implicitly connected; (2) automatic event-event temporal relation extraction methods such as (Ning et al., 2019) provide important clues about which element in the 4-tuple of an event can be propagated to which 4-tuple element of another event. We propose a novel time-aware graph propagation framework based on graph attention networks (GAT, Velickovic et al., 2018) to propagate temporal information across events in the constructed event graphs.

Experimental results on a benchmark, newly created on top of ACE2005 annotations, show that our proposed cross-event time propagation framework significantly outperforms state-of-the-art event time extraction methods using contextualized embedding features.

Our contributions can be summarized as follows.

- This is the first work taking advantage of the flexibility of 4-tuple representation to formulate absolute event timeline construction.
- We propose a GAT based approach for timeline construction which effectively propagates temporal information over document-level event graphs without solving large constrained optimization problems (e.g., Integer Linear Programming (ILP)) as previous work did. We propose two effective methods to construct the event graphs, based on shared arguments and temporal relations, which allow the time information to be propagated across the entire document.
- We build a new benchmark with over 6,000 human annotated non-infinite time elements, which implements the 4-tuple representation for the first time as a timeline dataset, and is intended to be used for future research on absolute timeline construction.

2 A New Benchmark

2.1 4-tuple Event Time Representation

Grounding events into a timeline necessitates the extraction of the start and end time of each event. However, the start and end time of most events is not explicitly expressed in a document. To capture such uncertainty, we adopt the 4-tuple representation introduced by the TAC-KBP2011 temporal slot filling task (Ji et al., 2011, 2013). We define **4-tuple event time** as four time elements for an event $e \rightarrow \langle \tau^-_{\text{start}}, \tau^+_{\text{start}}, \tau^-_{\text{end}}, \tau^+_{\text{end}} \rangle$,[3] which indicate *earliest possible start date*, *latest possible start date*, *earliest possible end date* and *latest possible end date*, respectively. These four dates follow hard constraints:

$$\begin{cases} \tau^-_{\text{start}} \leq \tau^+_{\text{start}} \\ \tau^-_{\text{end}} \leq \tau^+_{\text{end}} \end{cases}, \quad \begin{cases} \tau^-_{\text{start}} \leq \tau^-_{\text{end}} \\ \tau^+_{\text{start}} \leq \tau^+_{\text{end}} \end{cases}. \qquad (1)$$

[3] We use subscripts "start" and "end" to denote start and end time, and superscripts "−" and "+" to represent earliest and latest possible values.

The enemy have *now* been **flown out** and we're treating them including a man who is almost dead with a **gunshot wound** to the chest after we (Royal Marines) **sent** in one of our companies of about 100 men in here (Umm Kiou) *this morning*.

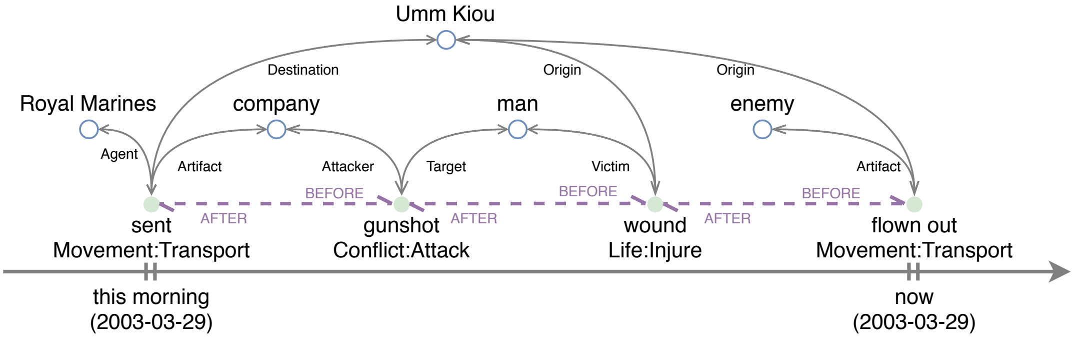

Figure 1: The example event graph. The graph with solid lines is constructed from event arguments. The graph with dash lines is constructed from temporal relations. Entities in the text are underlined and events in the text are in boldface.

Category	#
# documents	182
usenet	1
broadcast conversations	5
broadcast news	63
webblogs	26
newswire	87
# train/dev/test	92/39/51
# event mentions	2,084
# average tokens/document	436
# non-infinite elements	6,058
# infinite elements	2,278

Table 2: Data Statistics

Symbol	Explanation
w_i	the i-th word of document D
D	a document, $D = [w_1, \ldots, w_n]$
e_i	an event trigger in D
E	the event mention set of D, $E = \{e_1, \ldots, e_m\}$
τ_i	a time element of event i, can be $\{\tau^-_{i,\text{start}}, \tau^+_{i,\text{start}}, \tau^-_{i,\text{end}}, \tau^+_{i,\text{end}}\}$
t_i	a time expression in D
T	the time set of D, $T = \{t_1, \ldots, t_l\}$
r_i	a relation, either event argument roles or event temporal relations
R	relation set, $R = \{r_1, \ldots, r_q\}$

Table 3: Notations

The above temporal representation was originally designed for entity slot filling, and we regard it as an expressive way for describing events too as: (1) it allows for flexible representation of fuzzy time spans and thus, for those events that we cannot determine the accurate dates, they can also be grounded into a timeline; and (2) it allows for a unified treatment of various types of temporal information and thus makes it convenient to propagate over multiple events.

2.2 Annotation

We choose the Automatic Content Extraction (ACE) 2005 dataset because it includes rich annotations of event types, entity/time/value argument roles, time expressions and their normalization results. In our annotation interface, each document is highlighted with event triggers and time expressions. The annotators are required to read the whole document and provide as precise information as possible for each element of the 4-tuple of each event. If there is no possible information for a specific time, the annotators are asked to provide +/-infinite labels.

Overall, we have annotated 182 documents from this dataset. Most of the documents are from broadcast news or newswire genres. Detailed data statistics and data splits are shown in Table 2. We annotated all the documents with two independent passes. Two experts led the final adjudication based on independent annotations and discussions with annotators since single annotation pass is likely to miss important clues, especially when the event and its associated time expression appear in different paragraphs.

3 Approach

3.1 Overview

The input is a document $D = [w_1, \ldots, w_n]$, containing event triggers $E = [e_1, \ldots, e_m]$ and time expressions $T = [t_1, \ldots, t_l]$, and we use gold-standard annotation for event triggers and time expressions. Our goal is to connect the event triggers E and time expressions T scattered in a document, and estimate their association scores to select the most possible values for the 4-tuple elements. At a high-level, our approach is composed of: (1) a text encoder to capture semantic and narrative information in local context, (2) a document-level event graph to facilitate global knowledge, (3) a graph-based time propagation model to propagate time along event-event relations, and (4) an extraction algorithm to generate 4-tuple output. Among these four components, (1) and (4) build up the minimal requirements of an extractor, which serve as our baseline model and will be described in Section 3.2. We will detail how we utilize event arguments and temporal ordering to construct the document-level event graph, namely component (2), in Section 3.3. We will present our graph-based time propagation model in Section 3.4, and wrap up our model with training objective and other details in Section 3.5.

We list notations in Table 3, which will be explained when encountered.

3.2 Baseline Extraction Model

Our baseline extraction model is an event-time pair classifier based on a pre-trained language model (Devlin et al., 2019; Liu et al., 2019; Beltagy et al., 2020) encoder. The pre-trained language models allow us to have contextualized representation for every token in a given text. We directly derive the local representation for event triggers and time expressions from the contextualized representation. The representations are denoted as $\boldsymbol{h}_{e_i}$ for event trigger e_i and $\boldsymbol{h}_{t_j}$ for time expression t_j. For events or time expressions containing multiple tokens, we take the average of token representations. Thus, all $\boldsymbol{h}_{e_i}$ and $\boldsymbol{h}_{t_j}$ are of the same dimensions.

We pair each event and time in the document, i.e., $\{(e_i, t_j) \mid e_i \in E, t_j \in T\}$, to form the training examples. After obtaining event and time representations, we concatenate them and feed them into a 2-layer feed-forward neural classifier. The classifier estimates the probability of filling t_j in e_i's 4-tuple time elements, i.e., $\langle \tau^-_{i,\text{start}}, \tau^+_{i,\text{start}}, \tau^-_{i,\text{end}}, \tau^+_{i,\text{end}} \rangle$. The probabilities are:

$$p_{i,j,k} = \sigma(\boldsymbol{w}_{2,k}\text{ReLU}(\boldsymbol{W}_1[\boldsymbol{h}_{e_i};\boldsymbol{h}_{t_j}] + \boldsymbol{b}_1) + b_{2,k}) \tag{2}$$

where $\sigma(\cdot)$ is sigmoid function, and $\boldsymbol{W}_{1,2}$ and $\boldsymbol{b}_{1,2}$ are learnable parameters. In short, we use $\tau_{i,k}$ to represent the k^{th} element in τ_i ($k \in \{1, 2, 3, 4\}$) and $p_{i,j,k}$ represents a probability that t_j fills in the k^{th} element of 4-tuple τ_i. The baseline model consists of 4 binary classifiers, one for each element of the 4-tuple.

When determining the 4-tuple for each event e_i, we estimate the probability of t_1 through t_l. For each element, we take the time expression with the highest probability to fill in this element. A practical issue is that the same time is often expressed by different granularity levels, such as `2020-01-01` and `2020-W1`, following the most common TIMEX format (Ferro et al., 2005). To uniformly represent all the time expressions and allow certain degree of uncertainty, we introduce the following 2-tuple normalized form for time expressions, which indicates the time range of t_j by two dates,

$$t_i \rightarrow \langle t_i^-, t_i^+ \rangle \tag{3}$$

where t^-_* represents the earliest possible dates and t^+_* represents the latest possible dates.

We also make a simplification that the earliest possible values can only fill in earliest possible dates, i.e., $T^- = \{t_1^-, \ldots, t_l^-\} \mapsto \tau^-_{\text{start}}, \tau^-_{\text{end}}$, similarly for the latest dates, $T^+ = \{t_1^+, \ldots, t_l^+\} \mapsto \tau^+_{\text{start}}, \tau^+_{\text{end}}$. This constraint can be relaxed in future work. Here is an example of how we determine the binary labels for event-time pairs. If the 4-tuple time for an event is ⟨`2020-01-01, 2020-01-03, 2020-01-01, 2020-01-07`⟩ and the 2-tuple for time expression `2020-W1` is ⟨`2020-01-01, 2020-01-07`⟩, then the classification labels of this event-time pair will be ⟨`True,False,True,True`⟩.

3.3 Event Graph Construction

Before we conduct the global time propagation, we first construct document-level event graphs. In this paper, we focus on two types of event-event relations: (1) shared entity arguments, and (2) temporal relations.

Event Argument Graph. Event argument roles provide local information about events and two events can be connected via their shared arguments.

We denote the event-argument graph as $G_{\text{arg}} = \{(e_i, v_j, r_{i,j})\}$, where e_i represents an event, v_j represents an entity or a time expression, and $r_{i,j}$ represents the bi-directed edge between e_i and v_j, namely the argument role. For example, in Figure 1, there will be two edges between the "sent" event (e_1) and the entity "Royal Marines" (v_1), namely (e_1, v_1, AGENT) and (v_1, e_1, AGENT). In addition, we add a self-loop for each node in this graph. The graph can be constructed by Information Extraction (IE) techniques and we use gold-standard event annotation from ACE 2005 dataset in our experiments.

Event Temporal Graph. Event-event temporal relations provide explicit directions to propagate time information. If we know that an attack event happened before an injury event, the lower-bound end date of the attack can possibly be the start date of the injury. We denote the event temporal graph as $G_{\text{temp}} = \{(e_i, e_j, \gamma_{i,j})\}$, where e_i and e_j denote events, and $\gamma_{i,j}$ denotes the temporal order between e_i and e_j. Similar to G_{arg}, we also add a self-loop in G_{temp} and edges for two directions. For example, for a BEFORE relation from e_1 to e_2, we will add two edges, $(e_1, e_2, \text{BEFORE})$ and (e_2, e_1, AFTER). We only consider BEFORE and AFTER relations when constructing the event temporal graph. To propagate time information, we also use local time arguments as in event argument graphs.

We apply the state-of-the-art event temporal relation extraction model (Ning et al., 2019) to extract temporal relations for event pairs that appear in the same sentence or two consecutive sentences, and we only keep the relations whose confidence score is over 90%.

3.4 Event Graph-based Time Propagation

After obtaining the document-level graphs G_{arg} and G_{temp}, we design a novel time-aware graph neural network to perform document-level 4-tuple propagation.

Graph neural networks (Dai et al., 2016; Kipf and Welling, 2017; Hamilton et al., 2017; Schlichtkrull et al., 2018; Velickovic et al., 2018) have shown effective for relational reasoning (Zhang et al., 2018; Marcheggiani et al., 2018). We adopt graph attention networks (GAT, Velickovic et al., 2018) to propagate time through event-argument or event-event relations. GAT are proposed to aggregate and update information for each node from its neighbors through attention mechanism. Compared to the original GAT, we further include relational embedding for edge labels when performing attention to capture various types of relations between each event and its neighboring events.

The graphs G_{arg} and G_{temp} together with the GAT model are placed in the intermediate layer of our baseline extraction model (Section 3.2), i.e., between the pre-trained language model encoder and the 2-layer feed-forward neural classifier (Eq. (2)). For clarity, we denote all events and entities as nodes $V = \{v_1, \ldots, v_n\}$, and we use $r_{i,j}$ to denote their relation types. More specifically, we stack several layers of GAT on top of the contextualized representations of nodes $\boldsymbol{h}_{v_i}$. And we follow Vaswani et al. (2017) to use multi-head attention for each layer. We use the simplified notation $\boldsymbol{h}_{v_i}$ to describe one of the attention heads for $\boldsymbol{h}^k_{v_i}$.

$$\alpha_{ij} = \frac{\exp(a_{ij})}{\sum_{k \in \mathcal{N}(i)} \exp(a_{ik})} \tag{4}$$

$$\boldsymbol{h}'_{v_i} = \text{ELU}\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} \boldsymbol{W}_5 \boldsymbol{h}_{v_j}\right) \tag{5}$$

where ELU is exponential linear unit (Clevert et al., 2016), a_{ij} is the attention coefficient of node v_i and v_j, α_{ij} is the attention weight after softmax, and $\boldsymbol{h}_{v_i}$ and $\boldsymbol{h}'_{v_i}$ are the hidden states of node v_i before and after one GAT layer, respectively. We use $\mathcal{N}(i)$ to denote the neighborhood of v_i. The attention coefficients are calculated through

$$a_{ij} = \sigma\left(\boldsymbol{w}_4 \left[\boldsymbol{W}_3 \boldsymbol{h}_{v_i}; \boldsymbol{W}_3 \boldsymbol{h}_{v_j}; \boldsymbol{\phi}_{r_{i,j}}\right]\right) \tag{6}$$

where σ is LeakyReLU (Clevert et al., 2016) activation function. $\boldsymbol{\phi}_{r_{i,j}}$ is the learnable relational embedding for relation type of $r_{i,j}$ that we further add compared to the original GAT.

We concatenate m different attention heads to compute the representation of v_i for the next layer after performing attention for each head,

$$\boldsymbol{h}'_{v_i} = \bigg\Vert_{k=1}^{m} \boldsymbol{h}'^k_{v_i}. \tag{7}$$

We stack n_l GAT layers to obtain the final representations for events and time. These representations are fed into the 2-layer feed-forward neural classifier in Eq. (2) to generate the corresponding probabilities.

3.5 Training Objective

Since we model the 4-tuple extraction task by four binary classifiers, we adopt the log loss as our model objective:

$$\begin{aligned} \mathcal{L}(\tau_{i,k}, t_j) \quad = \quad & \mathbb{1}(\tau_{i,k} = t_j) \log p_{i,j,k} \\ & + \mathbb{1}(\tau_{i,k} \neq t_j) \log(1 - p_{i,j,k}) \end{aligned} \tag{8}$$

Since the 4-tuple elements are extracted from time expressions, the model cannot generate `+/-inf` (infinite) output. To address this issue, we adopt another hyperparameter, `inf` threshold, and convert those predicted time values with scores lower than the threshold into `+/-inf` values. That is, we regard the probability $p_{i,j,k}$ also as a confidence score. A low score indicates the model cannot determine the results for some 4-tuple elements. Thus it is natural to set those elements as `inf`. When this case happens in τ^-_{start} or τ^-_{end}, we correct the value to be `-inf`, and when it is τ^+_{start} or τ^+_{end}, we set the value to be `+inf`. This threshold and its searching will be applied to both baseline extract and GAT-based extraction systems. The extraction model may generate 4-tuples that do not follow the constraints on Eq. (1) and we leave enforcing the constraints for future work.

4 Experiments

4.1 Data and Experiment Setting

We conduct our experiments on previously introduced annotated data. Statistics of the dataset and splits are shown in Table 2.

Experiment Setup. We compare our proposed graph-based time propagation model with the following baselines:

- Local gold-standard time argument: The gold-standard time argument annotation provides the upperbound of the performance that a local time extraction system can achieve in our document 4-tuple time extraction task. We map gold-standard time argument roles to our 4-tuple representation scheme and report its performance for comparison. Specifically, if the argument role indicates the start time of an event (e.g., TIME-AFTER, TIME-AT-BEGINNING) we will map the date to τ^-_{start} and τ^+_{start}; if the argument role indicates the end time of an event (e.g., TIME-BEFORE) we will map the date to τ^-_{end} and τ^+_{end}; if the argument role is TIME-WITHIN, we will map the date to all elements. And we will leave all other elements as infinite.

- Document creation time: Document creation time plays an important role in previous absolute timeline construction (Chambers et al., 2014; Reimers et al., 2018). We build a baseline that uses document creation time as τ^+_{start} and τ^-_{end} for all events.

- Rule-based time propagation: We also build rule-based time propagation method on top of local gold-standard time arguments. One strategy is to set 4-tuple time for all events that do not have time arguments as document creation time. Another strategy is to set 4-tuple time for events that do not have time arguments as 4-tuple time for their previous events in context.

- Baseline extraction model: We compare our model with the baseline extraction model using contextualized embedding introduced in Section 3.2. We use two contextualized embedding methods, RoBERTa (Liu et al., 2019) and Longformer (Beltagy et al., 2020), which provide sentence-level[4] and document-level contextualized embeddings respectively.

For our proposed graph-based time propagation model, we use contextualized embedding from Longformer and consider two types of event graphs: (1) constructed event arguments, and (2) constructed temporal relations and time arguments.

We optimize our model with Adam (Kingma and Ba, 2015) for up to 500 epochs with a learning rate of 1e-4. We use dropout with a rate of 0.5 for each layer. The hidden size of two-layer feed-forward neural networks and GAT heads for all models is 384. The size of relation embeddings is 50. We use 4 different heads for GAT. The number of layers n_l is 2 for all GAT models. And we use a fixed pretrained model[5] to obtain contextualized representation for each sentence or document. We use 10 different random seeds for our experiments and report the averaged scores. We evaluate our model at each epoch, and search the best threshold for infinite dates on the development set. We use all predicted scores from the development set as candidate thresholds. We choose the model with the best performance on accuracy based on the development set and report the performance on test set using the best searched threshold on the development set.

Evaluation Metrics. We evaluate the performance of models based on two different metrics, exact match rate and approximate match rate proposed in TAC-KBP2011 temporal slot filling evaluation (Ji et al., 2011). For exact match

[4] We use RoBERTa to encode sentences instead of the entire documents because many documents exceed its maximal input length.

[5] We use roberta-base and longformer-base-4096 for RoBERTa and Longformer, respectively.

Model	EM	AM
Document Creation Time (DCT)	26.90	27.58
Time Argument Annotation	39.21	39.55
Rule-based Time Propagation		
DCT as Default	40.63	41.54
From Previous Event	46.20	48.15
Baseline Extraction Model		
RoBERTa	45.70*	49.92
Longformer	48.84*	52.41*
Temporal Relation based Propagation		
GAT	53.55*	56.60*
GAT w/ relation embedding	55.56*	58.63*
Argument based Propagation		
GAT	55.50*	58.79*
GAT w/ relation embedding	55.84	59.18

Table 4: System performance (%) on 4-tuple representation extraction on test set, averaged over 10 different runs. All standard deviation values are $\leq 2\%$. Scores with standard deviation values $\leq 1\%$ are marked with *. EM: exact match rate; AM: approximate match rate (see Eq. (9)).

rate, credits will only be assigned when the extracted date for a 4-tuple element exactly matches the ground truth date. The approximate match rate $Q(\cdot)$ compares the predicted 4-tuple $\hat{\tau}_i = \langle \hat{\tau}^-_{i,\text{start}}, \hat{\tau}^+_{i,\text{start}}, \hat{\tau}^-_{i,\text{end}}, \hat{\tau}^+_{i,\text{end}} \rangle$ with ground truth $\tau_i = \langle \tau^-_{i,\text{start}}, \tau^+_{i,\text{start}}, \tau^-_{i,\text{end}}, \tau^+_{i,\text{end}} \rangle$ by the averaged absolute difference between the corresponding dates,

$$Q(\hat{\tau}_i, \tau_i) = \frac{1}{4} \sum_{\substack{s\in\{+,-\},\\ p\in\text{start,end}}} \frac{1}{1 + |\hat{\tau}^s_{i,\text{p}} - \tau^s_{i,\text{p}}|}. \qquad (9)$$

In this way, partial credits will be assigned based on how close the extracted date is to the ground truth. For example, if a gold standard date is `2001-01-01` and the corresponding extracted date is `2001-01-02`, the credit will be $\frac{1}{1+|\texttt{2001-01-01}-\texttt{2001-01-02}|} = \frac{1}{2}$. If a gold standard date is `inf` and the corresponding extracted date is `2001-01-02`, the credit will be $\frac{1}{1+|\texttt{inf}-\texttt{2001-01-02}|} = 0$.

4.2 Results

Our experiment results are shown in Table 4. From the results of directly converting sentence-level time arguments to 4-tuple representation, we can find that local time information is not sufficient for our document-level 4-tuple event time extraction. And the document creation time baseline does not perform well because a large portion of document-level 4-tuple event time information does not coincide with document creation time, which is widely used in previous absolute timeline construction. By comparing the performance of basic extraction framework that uses sentence-level and document-level contextualized embedding, we can also find that involving document-level information from embeddings can already improve the system performance. Similarly, we can also see performance improvement by involving rule-based time propagation rules, which again indicates the importance of document-level information for this task.

Our GAT based time propagation methods significantly outperform those baselines, both when using temporal relations and when using arguments to construct those event graphs. Specifically, we find that using relation embedding significantly improves the temporal relation based propagation, by 2.01% on exact match rate and 2.03% on approximate match rate. This is because temporal labels between events, for example, BEFORE and AFTER, are more informative than argument roles in tasks related to time. Although our argument-based propagation model does not explicitly resolve conflict, the violation rate of 4-tuple constraints is about 4% in the output.

Our time propagation framework has also been integrated into the state-of-the-art multimedia multilingual knowledge extraction system GAIA (Li et al., 2020a,b) for NIST SM-KBP 2020 evaluation and achieves top performance at intrinsic temporal evaluation.

4.3 Qualitative Analysis

Table 5 shows some cases of comparison of various methods. In the first example, our argument based time propagation can successfully propagate "Wednesday", which is attached to the event "arrive", to "talk" event, through the shared argument "Blair". In the second example, "Negotiation" and "meeting" share arguments "Washington" and "Pyongyang". So the time information for "Negotiation" can be propagated to "meeting". In contrast, for these two cases, the basic extraction framework extracts wrong dates.

The third example shows the effectiveness of temporal relation based propagation. We use the extracted temporal relation that "rumble" happens before "secured" to propagate time information. The basic extraction model does not know the temporal relation between these two events and thus makes mistakes.

4.4 Remaining Challenges

Some temporal boundaries may require knowledge synthesis of multiple temporal clues in the docu-

... Meanwhile Blair arrived in Washington late Wednesday$_{\text{[2003-03-26]}}$ for two days of talks with Bush at the Camp David presidential retreat. ...
Element: Latest Start Date | **Baseline Extraction:** 2003-03-27 | **Argument based GAT:** 2003-03-26
Propagation Path: Wednesday ⟶ arrive ⟶ Blair ⟶ talks

... Negotiations between Washington and Pyongyang on their nuclear dispute have been set for April 23$_{\text{[2003-04-23]}}$ in Beijing and are widely seen here as a blow to Moscow efforts to stamp authority on the region by organizing such a meeting. ...
Element: Latest Start Date | **Baseline Extraction:** +inf | **Argument based GAT:** 2003-04-23
Propagation Path: April 23 ⟶ Negotiations ⟶ Pyongyang ⟶ meeting

... Saturday morning$_{\text{[2003-03-22]}}$, American Marines and British troops rumbled along the main road from the Kuwaiti border to Basra, Highway 80, nicknamed the "Highway of Death" during the 1991 Gulf War , when U. S. airstrikes wiped out an Iraqi military convoy along it. American units advancing west of Basra have already secured the Rumeila oil field, whose daily output of 1.3 million barrels makes it Iraq's most productive. ...
Element: Earliest Start Date | **Baseline Extraction:** 2003-03-21 | **Temporal based GAT w/ rel:** 2003-03-22
Propagation Path: Saturday morning ⟶ rumbled $\xrightarrow{\text{BEFORE}}$ secured

Table 5: Comparison of different system outputs. The first two examples demonstrate the effectiveness of argument based propagation. The third example demonstrates the effectiveness of temporal relation based propagation.

ment. For example, in Table 1, the latest end date of the "sentence" event (2012-04-14) needs to be inferred by aggregating two temporal clues in the document, namely its duration as nine-year, and its start date as 2003-04-14.

Temporal information for many events, especially major events, may be incomplete in a single document. Taking Iraq war as an example, one document may mention its start date and another may mention its end date. To tackle this challenge, we need to extend document-level extraction to corpus-level and then aggregate temporal information for coreferential events in multiple documents.

It is also challenging for the current 4-tuple representation to represent temporal information for recurring events such as paying monthly bills. Currently we consider recurring events as different events and fill in slots separately. Besides, this work does not capture more fine-grained information such as hours and minutes, but it is straightforward to extend the 4-tuple representation to these time scales in future work.

Our current annotations are done by linguistic experts and thus they are expensive to acquire. It is worth exploring crowd-sourcing methods in the future to make it more scalable and less costly.

5 Related Work

Event Temporal Anchoring. Event temporal anchoring is first introduced by Setzer (2002) using temporal links (TLINKS) to specify the relation among events and time. However, the TimeBank Corpus and TimeBank Dense Corpus using TimeML scheme (Pustejovsky et al., 2003b,a; Cassidy et al., 2014) is either too vague and sparse or is dense only with limited scope. Recently, Reimers et al. (2016) annotate the start and end time of each event on TimeBank. We have made several extensions by adding event types, capturing uncertainty by 4-tuple representation instead of TLINKS so that indirect time can also be considered, and extending event-event relations to document-level.

Models trained on TimeBank often formulate the problem as a pair-wise classification for TLINKS. Efforts have been made to use Markov logical networks or ILP to propagate relations (Bramsen et al., 2006; Chambers and Jurafsky, 2008; Yoshikawa et al., 2009; Do et al., 2012), sieve-based classification (Chambers et al., 2014), and neural networks based methods (Meng et al., 2017; Meng and Rumshisky, 2018; Cheng et al., 2020). There are also efforts on event-event temporal relations (Ning et al., 2017, 2018, 2019; Han et al., 2019).

Especially, Reimers et al. (2018) propose a decision tree that uses a neural network based classifier to find start and end time on Reimers et al. (2016). Leeuwenberg and Moens (2018) use event time to construct relative timeline.

Temporal Slot Filling. Earlier work on extracting 4-tuple representation focuses on temporal slot-filling (TSF, Ji et al., 2011, 2013) to collect 4-tuple dates as temporal boundaries for entity attributes. Attempts on TSF include pattern matching (Byrne and Dunnion, 2011) and distant supervision (Li et al., 2012; Ji et al., 2013; Surdeanu et al., 2011; Sil and Cucerzan, 2014; Reinanda et al., 2013; Reinanda and de Rijke, 2014). In our work, we directly adopt 4-tuple as a fine-grained temporal representation for events instead of entity attributes.

Temporal Reasoning. Some early efforts attempt to incorporate event-event relations to perform temporal reasoning (Tatu and Srikanth, 2008) and propagate time information (Gupta and Ji,

2009) based on hard constraints learned from annotated data. Our work is largely inspired from Talukdar et al. (2012) on graph-based label propagation for acquiring temporal constraints for event temporal ordering. We extend the idea by constructing rich event graphs, and proposing a novel GAT based method to assign weights for propagation.

The idea of constructing event graph based on sharing arguments is also motivated from Centering Theory (Grosz et al., 1995), which has been applied to many NLP tasks such as modeling local coherence (Barzilay and Lapata, 2008) and event schema induction (Chambers and Jurafsky, 2009).

6 Conclusions and Future Work

In this paper, we have created a new benchmark for document-level event time extraction based on 4-tuple representation, which provides rich representation to handle uncertainty. We propose a graph-based time propagation and use event-event relations to construct document-level event graphs. Our experiments and analyses show the effectiveness of our model. In the future, we will focus on improving the fundamental pretraining model for time to represent more fine-grained time information and cross-document temporal aggregation.

Acknowledgement

This research is based upon work supported in part by U.S. DARPA KAIROS Program No. FA8750-19-2-1004, U.S. DARPA AIDA Program No. FA8750-18-2-0014, and Air Force No. FA8650-17-C-7715. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Regina Barzilay and Mirella Lapata. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *CoRR*, abs/2004.05150.

Philip Bramsen, Pawan Deshpande, Yoong Keok Lee, and Regina Barzilay. 2006. Inducing temporal graphs. In *EMNLP 2006, Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, 22-23 July 2006, Sydney, Australia*, pages 189–198. ACL.

Lorna Byrne and John Dunnion. 2011. UCD IIRG at TAC 2011. In *Proceedings of Text Analysis Conference (TAC2011)*.

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, Baltimore, Maryland. 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.

Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. 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*, pages 602–610, Suntec, Singapore. Association for Computational Linguistics.

Nathanael Chambers and Daniel Jurafsky. 2008. Jointly combining implicit constraints improves temporal ordering. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 698–706, Honolulu, Hawaii. Association for Computational Linguistics.

Fei Cheng, Masayuki Asahara, Ichiro Kobayashi, and Sadao Kurohashi. 2020. Dynamically updating event representations for temporal relation classification with multi-category learning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1352–1357, Online. Association for Computational Linguistics.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2016. Fast and accurate deep network learning by exponential linear units (elus). In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Hanjun Dai, Bo Dai, and Le Song. 2016. Discriminative embeddings of latent variable models for structured data. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2702–2711. JMLR.org.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference* *of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Quang Do, Wei Lu, and Dan Roth. 2012. Joint inference for event timeline construction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 677–687, Jeju Island, Korea. Association for Computational Linguistics.

Lisa Ferro, Laurie Gerber, Inderjeet Mani, Beth Sundheim, and George Wilson. 2005. TIDES2005 standard for the annotation of temporal expressions. *MITRE Corporation Technical Report*.

Tao Ge, Wenzhe Pei, Heng Ji, Sujian Li, Baobao Chang, and Zhifang Sui. 2015. Bring you to the past: Automatic generation of topically relevant event chronicles. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 575–585, Beijing, China. Association for Computational Linguistics.

Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. 1995. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2):203–225.

Prashant Gupta and Heng Ji. 2009. Predicting unknown time arguments based on cross-event propagation. In *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore, Short Papers*, pages 369–372. The Association for Computer Linguistics.

Andrey Gusev, Nathanael Chambers, Divye Raj Khilnani, Pranav Khaitan, Steven Bethard, and Dan Jurafsky. 2011. Using query patterns to learn the duration of events. In *Proceedings of the Ninth International Conference on Computational Semantics (IWCS 2011)*.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 1024–1034.

Rujun Han, Qiang Ning, and Nanyun Peng. 2019. Joint event and temporal relation extraction with shared representations and structured prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 434–444. Association for Computational Linguistics.

Lifu Huang and Lian'en Huang. 2013. Optimized event storyline generation based on mixture-event-aspect model. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 726–735, Seattle, Washington, USA. Association for Computational Linguistics.

Heng Ji, Taylor Cassidy, Qi Li, and Suzanne Tamang. 2013. Tackling representation, annotation and classification challenges for temporal knowledge base population. *Knowledge and Information Systems*, 41(3):611–646.

Heng Ji, Ralph Grishman, Zheng Chen, and Prashant Gupta. 2009. Cross-document event extraction and tracking: Task, evaluation, techniques and challenges. In *Proceedings of the International Conference RANLP-2009*, pages 166–172, Borovets, Bulgaria. Association for Computational Linguistics.

Heng Ji, Ralph Grishman, and Hoa Trang Dang. 2011. An overview of the TAC2011 knowledge base population track. In *Proceedings of Text Analysis Conference (TAC2011)*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Artuur Leeuwenberg and Marie-Francine Moens. 2018. Temporal information extraction by predicting relative time-lines. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1237–1246, Brussels, Belgium. Association for Computational Linguistics.

Manling Li, Ying Lin, Tuan Manh Lai, Xiaoman Pan, Haoyang Wen, Sha Li, Zhenhailong Wang, Pengfei Yu, Lifu Huang, Di Lu, Qingyun Wang, Haoran Zhang, Qi Zeng, Chi Han, Zixuan Zhang, Yujia Qin, Xiaodan Hu, Nikolaus Parulian, Daniel Campos, Heng Ji, Brian Chen, Xudong Lin, Alireza Zareian, Amith Ananthram, Emily Allaway, Shih-Fu Chang, Kathleen McKeown, Yixiang Yao, Michael Spector, Mitchell DeHaven, Daniel Napierski, Marjorie Freedman, Pedro Szekely, Haidong Zhu, Ram Nevatia, Yang Bai, Yifan Wang, Ali Sadeghian, Haodi Ma, and Daisy Zhe Wang. 2020a. GAIA at SM-KBP 2020 - a dockerlized multi-media multi-lingual knowledge extraction, clustering, temporal tracking and hypothesis generation system. In *Proceedings of Thirteenth Text Analysis Conference (TAC 2020)*.

Manling Li, Alireza Zareian, Ying Lin, Xiaoman Pan, Spencer Whitehead, Brian Chen, Bo Wu, Heng Ji, Shih-Fu Chang, Clare Voss, Daniel Napierski, and Marjorie Freedman. 2020b. GAIA: A fine-grained multimedia knowledge extraction system. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 77–86, Online. Association for Computational Linguistics.

Qi Li, Javier Artiles, Taylor Cassidy, and Heng Ji. 2012. Combining flat and structured approaches for temporal slot filling or: How much to compress? In *Computational Linguistics and Intelligent Text Processing - 13th International Conference, CICLing 2012, New Delhi, India, March 11-17, 2012, Proceedings, Part II*, volume 7182 of *Lecture Notes in Computer Science*, pages 194–205. Springer.

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. *CoRR*, abs/1907.11692.

Hector Llorens, Nathanael Chambers, Naushad UzZaman, Nasrin Mostafazadeh, James Allen, and James Pustejovsky. 2015. SemEval-2015 task 5: QA TempEval - evaluating temporal information understanding with question answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 792–800, Denver, Colorado. Association for Computational Linguistics.

Diego Marcheggiani, Jasmijn Bastings, and Ivan Titov. 2018. Exploiting semantics in neural machine translation with graph convolutional networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 486–492, New Orleans, Louisiana. 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, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 527–536. Association for Computational Linguistics.

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, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 887–896. Association for Computational Linguistics.

Anne-Lyse Minard, Manuela Speranza, Eneko Agirre, Itziar Aldabe, Marieke van Erp, Bernardo Magnini, German Rigau, and Rubén Urizar. 2015. SemEval-2015 task 4: TimeLine: Cross-document event ordering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 778–786, Denver, Colorado. Association for Computational Linguistics.

Qiang Ning, Zhili Feng, and Dan Roth. 2017. A structured learning approach to temporal relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1027–1037, Copenhagen, Denmark. Association for Computational Linguistics.

Qiang Ning, Zhili Feng, Hao Wu, and Dan Roth. 2018. Joint reasoning for temporal and causal relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2278–2288. Association for Computational Linguistics.

Qiang Ning, Sanjay Subramanian, and Dan Roth. 2019. An improved neural baseline for temporal relation extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 6202–6208. Association for Computational Linguistics.

Feng Pan, Rutu Mulkar, and Jerry R. Hobbs. 2006. Learning event durations from event descriptions. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 393–400, Sydney, Australia. Association for Computational Linguistics.

Feng Pan, Rutu Mulkar-Mehta, and Jerry R. Hobbs. 2011. Annotating and learning event durations in text. *Computational Linguistics*, 37(4):727–752.

James Pustejovsky, José M. Castaño, Robert Ingria, Roser Saurí, Robert J. Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R. Radev. 2003a. Timeml: Robust specification of event and temporal expressions in text. In *New Directions in Question Answering, Papers from 2003 AAAI Spring Symposium, Stanford University, Stanford, CA, USA*, pages 28–34. AAAI Press.

James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, et al. 2003b. The timebank corpus. In *Corpus linguistics*, volume 2003, page 40. Lancaster, UK.

Nils Reimers, Nazanin Dehghani, and Iryna Gurevych. 2016. Temporal anchoring of events for the TimeBank corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2195–2204, Berlin, Germany. Association for Computational Linguistics.

Nils Reimers, Nazanin Dehghani, and Iryna Gurevych. 2018. Event time extraction with a decision tree of neural classifiers. *Transactions of the Association for Computational Linguistics*, 6:77–89.

Ridho Reinanda and Maarten de Rijke. 2014. Prior-informed distant supervision for temporal evidence classification. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 996–1006, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Ridho Reinanda, Daan Odijk, and de M Rijke. 2013. Exploring entity associations over time. In *SIGIR2013; Workshop on time-awareInformation access*. TAIA'13.

Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, volume 10843 of *Lecture Notes in Computer Science*, pages 593–607. Springer.

Andrea Setzer. 2002. *Temporal information in newswire articles: an annotation scheme and corpus study.* Ph.D. thesis, University of Sheffield.

Avirup Sil and Silviu-Petru Cucerzan. 2014. Towards temporal scoping of relational facts based on Wikipedia data. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 109–118, Ann Arbor, Michigan. Association for Computational Linguistics.

Julius Steen and Katja Markert. 2019. Abstractive timeline summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 21–31, Hong Kong, China. Association for Computational Linguistics.

Mihai Surdeanu, Sonal Gupta, John Bauer, David McClosky, Angel X. Chang, Valentin I. Spitkovsky, and Christopher D. Manning. 2011. Stanford's distantly-supervised slot-filling system. In *Proceedings of the Fourth Text Analysis Conference, TAC 2011, Gaithersburg, Maryland, USA, November 14-15, 2011*. NIST.

Partha Pratim Talukdar, Derry Wijaya, and Tom M. Mitchell. 2012. Acquiring temporal constraints between relations. In *21st ACM International Conference on Information and Knowledge Management, CIKM'12, Maui, HI, USA, October 29 - November 02, 2012*, pages 992–1001. ACM.

Marta Tatu and Munirathnam Srikanth. 2008. Experiments with reasoning for temporal relations between events. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 857–864, Manchester, UK. Coling 2008 Organizing Committee.

Siddharth Vashishtha, Benjamin Van Durme, and Aaron Steven White. 2019. Fine-grained temporal relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2906–2919, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Alakananda Vempala, Eduardo Blanco, and Alexis Palmer. 2018. Determining event durations: Models and error analysis. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 164–168, New Orleans, Louisiana. Association for Computational Linguistics.

Lu Wang, Claire Cardie, and Galen Marchetti. 2015. Socially-informed timeline generation for complex events. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1055–1065, Denver, Colorado. 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*, pages 405–413, Suntec, Singapore. Association for Computational Linguistics.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215, Brussels, Belgium. Association for Computational Linguistics.

Ben Zhou, Daniel Khashabi, Qiang Ning, and Dan Roth. 2019. "going on a vacation" takes longer than "going for a walk": A study of temporal commonsense understanding. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3363–3369, Hong Kong, China. Association for Computational Linguistics.