EVENT TIME REPRESENTATION, PROPAGATION AND PREDICTION IN TEMPORAL INFORMATION EXTRACTION

BY

HAOYANG WEN

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois Urbana-Champaign, 2021

Urbana, Illinois

Adviser:

Professor Heng Ji

ABSTRACT

Temporal information extraction is a challenging task due to the inherent ambiguity of language. Event time plays an important role in temporal information extraction, which can help ground events into a timeline, and can help other temporal information extraction tasks such as temporal relation extraction. However, explicit information for event time are not often expressed in a document. In this thesis, we first focus on a new event time representation that adopts the 4-tuple temporal representation proposed in the TAC-KBP temporal slot filling to resolve the uncertainty and sparsity problem. We then propose a graph neural network-based method to propagate local time information over constructed event graphs. We also study the event time in temporal relation extraction. We predict relative timestamps for events from event-event relation annotations and use those timestamps as additional features for training a temporal relation extraction system. We use the Stack-Propagation framework to jointly train the timestamps prediction and temporal relation extraction task. Finally, we demonstrate two knowledge extraction systems that has integrated the temporal information extraction models and show their effectiveness.

To my parents, for their love and support.

ACKNOWLEDGMENTS

First, I would like to express my sincerely gratitude to my advisor, Professor Heng Ji, for her dedicated support and selfless guidance. I always admire her brilliant thoughts and her endless energy and passion for research. It is my most memorable experience to work with her during my study at Urbana-Champaign.

Besides my advisor, I would like to thank my collaborators: Yanru Qu, Professor Jiawei Han, Dr. Qiang Ning, Professor Hanghang Tong, Dr. Avirup Sil and Professor Dan Roth. It has been a great honor to work with them and they have given so many insightful comments and suggestions on my research. During my study, I also had a wonderful summer internship working remotely with Dr. Avirup Sil, Dr. Radu Florian and Anthony Ferritto. I am particularly grateful to Professor Jiawei Han and Dr. Radu Florian for their support during my Ph.D. applications.

I feel proud to work with so many excellent peers in our Blender Lab, including Professor Lifu Huang, Dr. Xiaoman Pan, Dr. Spencer Whitehead, Manling Li, Qingyun Wang, Qi Zeng, Pengfei Yu, Tuan Lai, Yi Fung, Haoran Zhang, Meha Kumar, Zhenhailong Wang and Revanth Reddy. I will always remember the time that we argued with each other, and the moments that we gathered for food, movie and play.

Finally, I am so fortunate to have the unconditioned love and support from my family and girlfriend, especially during the pandemic.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

There is an increasing demand of Information Extraction (IE) technology that can extract structured knowledge from unstructured data such as news articles. Normally, the extracted knowledge graph consists of events, entities, and relations. However, since events are highly dynamic and intercorrelated, it is also crucial to extract the temporal information of events such as their start/end time and temporal orders. Those temporal information are important for applications such as future event prediction and event timeline generation.

For example, if there are two events "*police arrived at the location*" and "*someone died*", we may have different stories by altering the timeline. If "*someone died*" happens before "*police arrived at the location*", the story may be that there was a homicide event and police came to investigate it. If "*police arrived at the location*" happens before "*someone died*", then the story may be that the police came and the murderer was shot dead there and then. A good extraction system for those temporal information will significantly contribute to the comprehensive understanding of scenarios that consist of multiple events.

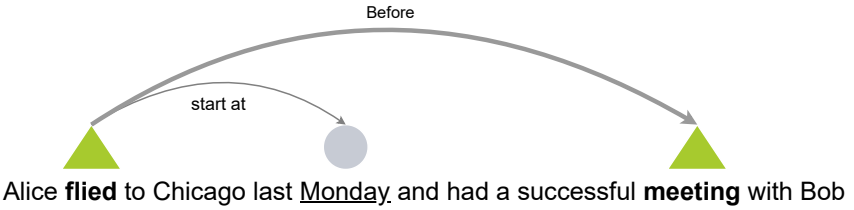


Figure 1.1: Illustration of an example sentence and its event time and temporal relation extraction results.

Specifically, in this thesis we focus on two types of temporal information: event time extraction and event-event temporal relation extraction. Explicit event time extraction is to extract the start time and end time for a given event in a document. Event-event temporal relation extraction is to extract the temporal order between a event pair. As an example, we depict the event time extraction and event-event temporal relation extraction of the sentence "*Alice flied to Chicago last Monday and had a successful meeting with Bob.*" as shown in Figure 1.1. In this sentence, *"Monday"* is an explicit timestamp and it indicates the start time of the event *"flied"*. In practice, we often do one step further that normalizes the time information to a standard format, such as normalizing *"Monday"* to `2001-01-01T08:00:00`. From the narrative, we can also conclude that event *"flied"* happened before event *"meeting"*. We focus on formulating better event time representation and extraction as well as utilizing event time information for relation extraction.

1.1 CHALLENGES

The characteristic of time information makes extracting temporal information different from other natural language understanding tasks. One uniqueness is that natural language expresses time in different levels of granularity, such as *"last year"*, *"today"*, *"an hour ago"*. However, for computers we always hope that time can be normalized with the same precision so that we can easily perform computation on top of those numerical representations. On the other hand, in many domains such as news, the duration of an atomic event is normally very short. But it is also very rare that we can obtain very precise time information from news articles.

Different levels of granularity for time representation also bring uncertainty. For example, if someone says *"I will fly to Chicago this week."*, we can only know that the event *"fly"* will happens in a specific range of time (*"now"* to *"the end of this week"*). However, since a flight usually only lasts for several hours, it is still very hard to precisely ground the *"fly"* events onto the timeline. Another source of the uncertainty comes from the sparcity of explicit time arguments. For example, in Figure 1.1, there is no specific information for event *"meeting"*. We have to infer its time information from related events *"flied"* and know that the start time for event *"meeting"* may be on or after *"Monday"*. The uncertainty and sparcity also make it difficult to acquire a large and clean event time dataset. In ACE 2005 dataset, only about one third of the event mentions have explicit time arguments in their local sentences.

Additionally, event-event temporal relation extraction is also extremely challenging. Ideally, there always exists a temporal relation between two events that have already happened. However it is still very difficult to extract event-event temporal relations from text, even for human. One reason is that the narrative orders of most domains do not strictly follow their temporal orders. For example, in news domain, it is very common that we have a paragraph in the middle of a main story describing some background events. Another reason is that sometimes we need to find out the temporal relations between two events with long distances, such as several sentences or even paragraphs. Meanwhile, if we extract the temporal relations in a document separately, the extracted results may conflict with each other. For example, if we have already know that event A happened before event B, and event B happened before event C, we can conclude that event A happened before event C.

1.2 THESIS OUTLINE

In Chapter 2, we introduce the basic concepts about event time and event-event temporal relation extraction. We will also briefly introduce the existing framework and resources for

these two tasks.

In Chapter 3, 4, 5, we present our work related to event time and event-event temporal relation extraction.

1.2.1 Chapter 3: Event Time Extraction and Propagation via Graph Attention Networks

In this chapter, we present our new task formulation toward event time extraction. Instead of extracting the start and end time, we adopt a 4-tuple temporal representation proposed in the TAC-KBP temporal slot filling task [1, 2] 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 further construct event graphs based on event arguments and temporal orders and propagate local time information within the constructed event graph using graph attention networks (GAT) [3].

1.2.2 Chapter 4: Incorporating Relative Timestamps into Event-Event Temporal Relation Extraction

In this chapter, we present our novel model on event-event temporal relation extraction. Compared to previous work, our model will first ground events onto a relative timeline and then use this feature for temporal relation classification. Specifically, we use an auxiliary task, which focuses on predicting relative event timestamps using pairwise temporal relation annotation. We further use the Stack-Propagation framework to incorporate the predicted event timestamps for temporal relation classification while keep the differentiability to jointly train these tasks. Our experiments show that our model can achieve better performance compared to many strong baselines, and can achieve similar performance compared to model using additional data.

1.2.3 Chapter 5: Applications of Temporal Information Extraction

In this chapter, we will briefly introduce two typical applications of event time extraction and event-event temporal relation extraction system. We first discuss our state-of-the-art multi-media multi-lingual knowledge extraction system and the performance of event time extraction in an end-to-end testing. We then discuss our schema-guided cross-document cross-lingual cross-media information extraction and event tracking system and show how our temporal relation extraction module contribute to the event tracking and prediction.

1.3 CONTRIBUTIONS

In this thesis, we make the following contributions:

1. We formulated document-level event time as 4-tuple representation, following TAC-KBP 2011 Temporal Slot Filling [1] to resolve the scarcity of current explicit event argument annotation and provided a new benchmark on ACE2005 documents.

2. We proposed a new event time extraction system based on time propagation over constructed event graphs from event arguments and event-event temporal relations.

3. We use relative timestamp prediction trained from event-event temporal relation annotations and incorporate them as features into event-event temporal relation extraction model via Stack-Propagation based joint training.

4. We integrated our algorithms and models into knowledge extraction systems and can provide the state-of-the-art performance for downstream applications.

1.4 PUBLICATIONS

[1] H. Wen, Y. Qu, H. Ji, Q. Ning, J. Han, A. Sil, H. Tong, and D. Roth, "Event time extraction and propagation via graph attention networks," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT2021)*, 2021

[2] H. Wen, Y. Lin, T. M. Lai, X. Pan, S. Li, X. Lin, B. Zhou, M. Li, H. Wang, H. Zhang, X. Yu, A. Dong, Z. Wang, Y. R. Fung, P. Mishra, Q. Lyu, D. Surís, B. Chen, S. W. Brown, M. Palmer, C. Callison-Burch, C. Vondrick, J. Han, D. Roth, S.-F. Chang, and H. Ji, "Resin: A dockerized schema-guided cross-document cross-lingual cross-media information extraction and event tracking system," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT2021) Demo Track*, 2021

[3] M. Li, Y. Lin, T. M. Lai, X. Pan, H. Wen, S. Li, Z. Wang, P. Yu, L. Huang, D. Lu, Q. Wang, H. Zhang, Q. Zeng, C. Han, Z. Zhang, Y. Qin, X. Hu, N. Parulian, D. Campos, H. Ji, B. Chen, X. Lin, A. Zareian, A. Ananthram, E. Allaway, S.-F. Chang, K. McKeown, Y. Yao, M. Spector, M. DeHaven, D. Napierski, M. Freedman, P. Szekely, H. Zhu, R. Nevatia, Y. Bai, Y. Wang, A. Sadeghian, H. Ma, and D. Z. Wang, "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)*, 2020

CHAPTER 2: BACKGROUND

In this section, we will first explain basic concepts in temporal information extraction. Then we will introduce existing resources and methods related to temporal information extraction.

2.1 BASIC CONCEPTS

Definition 2.1 (Time Expression)**.** According to [7], time expressions can "reference calendar dates, times of day, or durations".

For example, following the example sentence that has shown in Figure 1.1, "Alice flied to Chicago last Monday and had a successful meeting with Bob", "Monday" refers to a calender day that can be inferred from document creation time.

Definition 2.2 (Entity)**.** According to [8], an entity is "an object or set of objects in the world".

For example, in Figure 1.1, "Alice" and "Bob" are entities. Entities can be categorized into different types, such as person, organization and location. In this example, "Alice" and "Bob" are two person entities.

Definition 2.3 (Event)**.** According to [9], an event is "something that happens". It usually indicates a change of state.

Following the example sentence in Figure 1.1, it describe two different events, "Alice flied to Chicago last Monday" and "Alice had a successful meeting with Bob". Events with similar patterns can by categorized into specific types. For example, following the ontology defined in [9], "Alice flied to Chicago last Monday" is a MOVEMENT.TRANSPORT-PERSON event.

Definition 2.4 (Event Mention)**.** An event mention indicates the provenance of an event from text.

An event can be described multiple times in a given context. Following the example in Figure 1.1, if there is another sentence, "The flight that Alice took to Chicago last Monday was delayed", in the same document as the example sentence, then both two sentences contain the event mention for the same MOVEMENT.TRANSPORT-PERSON event.

Definition 2.5 (Event Trigger)**.** According to [9], an event's trigger is "the word that most clearly expresses its occurrence".

Following the example in Figure 1.1, word "flied" is the trigger of the MOVEMENT.TRANSPORT-PERSON event. In most cases, event triggers are verbs, but sometimes they can also be nouns, pronouns and even adjectives. For example, "meeting", a noun word in the same sentence in Figure 1.1, is also an event trigger.

Definition 2.6 (Event Argument)**.** Event arguments are participants or attributes of an event. Participants are entities that are involved in an event. Attributes are properties of an event.

Following the example in Figure 1.1, "Alice" is the participant of the MOVEMENT.TRANSPORT-PERSON event and "Monday" is an attribute of the MOVEMENT.TRANSPORT-PERSON indicating its start time. Participants can be categorized into specific semantic roles. In the above example, "Alice" can be categorized to PERSON. We normally refer all time attributes as time arguments.

Definition 2.7 (Event-Event Temporal Relation)**.** The temporal relation between two events references to the relative order of these two events based on their start time and end time.

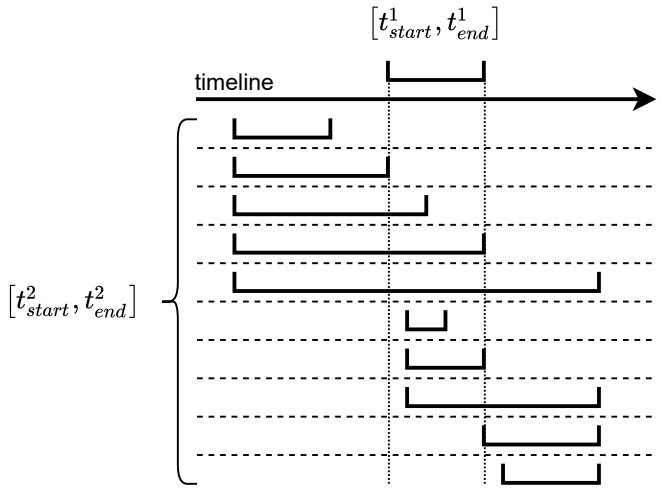


Figure 2.1: Different temporal relation types and their corresponding relative order for their start time and end time according to [10].

Following the example in Figure 1.1, we can clearly conclude from the sentence that "Alice flied to Chicago last Monday" happened before "Alice had a successful meeting with Bob". Temporal relation type, such as BEFORE, AFTER and SIMULTANEOUS, are commonly used when annotating event-event temporal relations. Thirteen types and the corresponding relative order for their start time and end time are are shown in Figure 2.1, following [10]. Please note that different event-event temporal relation datasets may used slightly different

annotation scheme. For example, in [11] they only consider start time when determining the order.

2.2 EXISTING RESOURCES

2.2.1 ACE2005

The Automatic Content Extraction (ACE) 2005 Multilingual Training Corpus[1] is a dataset annotated for entities, relations and events on multiple genre text. ACE 2005 dataset contains 7 entity types, covering named entity mentions, nominal mentions, pronominal mentions and nested mentions. It also contains 6 relation types and 17 subtypes. The annotation for events covering event triggers, event arguments and their modality, tense, polarity and genericity. It provides annotations for 8 event types and 33 subtypes.

2.2.2 TimeBank and Its Series

The original TimeBank Corpus [12, 13] contains document-level temporal relation annotation on 183 news articles. The major differences for events compared to ACE 2005 is that TimeBank series data only consider event triggers when annotating events, and it doesn't contain event type annotations. TimeBank dataset uses TLINK to annotate the temporal relationship on event-time, time-time, and event-event pairs following TimeML specification [14]. It only contains annotations that are critical to understand the documents.

One important variant of TimeBank Corpus, TimeBank-Dense Corpus [15], provides a dense annotation that covering all pairs in a given window size. And it contains annotation for label VAGUE, referring to the pairs that are unable to be categorized into types. There are also annotations that focus on refining other temporal information. For example, [16] focus on annotating the start and end time of each event from document-level information on TimeBank.

2.2.3 MATRES

One major limitation of previous temporal annotation is the low inter-annotator agreement. MATRES provides a new annotation scheme, that focuses on main-axis annotation in a given context window, providing relatively dense annotation with reliable inter-annotator

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

agreement. MATRES dataset use the same documents from TimeBank and TempEval-3 [17] while providing its own annotations.

CHAPTER 3: EVENT TIME EXTRACTION AND PROPAGATION VIA GRAPH ATTENTION NETWORKS

3.1 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 [18, 19, 20, 21], temporal event tracking and prediction [22, 23], and temporal question answering [24, 25].

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 [16]. However, the start and end time of an event are often not explicitly expressed in a document. For example, among 5,271 annotated event mentions in the Automatic Content Extraction (ACE2005) corpus[1], 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 [25, 26, 27, 28, 29, 30, 31, 32, 33, 34] and duration prediction [35, 36, 37, 38, 39, 40]. In this chapter 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 [1, 2] 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 3.1, 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 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 3.1. We leverage two types of event-event relations: (1) if

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

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 3.1: 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.

two events share the same entity as their arguments, then they are implicitly connected; (2) automatic event-event temporal relation extraction methods such as [33] 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 [3]) 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 allows the time information to be propagated across the

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*.

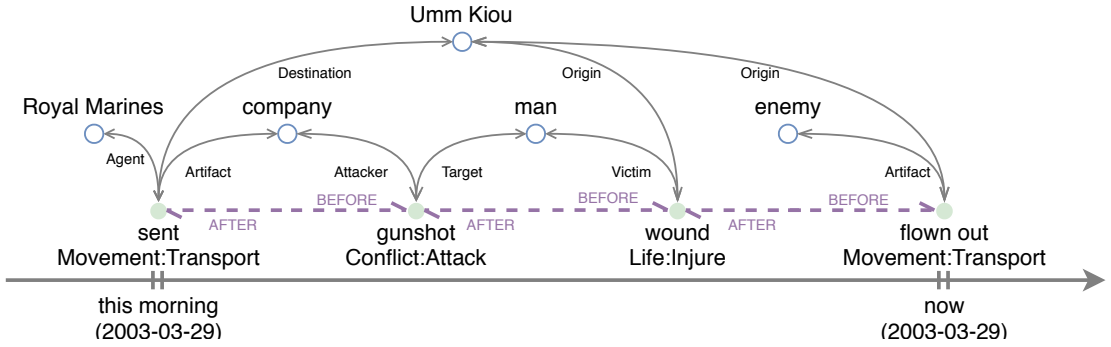


Figure 3.1: The example event graph. The graph using solid lines is constructed from event arguments. The graph using dash lines is constructed from temporal relations.

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.

3.2 A NEW BENCHMARK

3.2.1 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 are 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 [1, 2]. We define **4-tuple event time** as four time elements for an event $e \rightarrow \langle \tau^-_{\text{start}}, \tau^+_{\text{start}}, \tau^-_{\text{end}}, \tau^+_{\text{end}} \rangle$,[2] 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:

$$\begin{cases} \tau^-_{\text{start}} \leq \tau^+_{\text{start}} \\ \tau^-_{\text{end}} \leq \tau^+_{\text{end}} \end{cases}, \quad \begin{cases} \tau^-_{\text{start}} \leq \tau^-_{\text{end}} \\ \tau^+_{\text{start}} \leq \tau^+_{\text{end}} \end{cases}. \tag{3.1}$$

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

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 3.2: Data Statistics

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 more 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.

3.2.2 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 3.2. 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 the time expression appear in different paragraphs.

3.3 APPROACH

3.3.1 Overview

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 3.3: Notations

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 3.3.2. We will detail how we utilize event arguments and temporal ordering to construct the document-level event graph, namely component (2), in Section 3.3.3. We will present our graph-based time propagation model in Section 3.3.4, and wrap up our model with training objective and other details in Section 3.3.5.

We list notations in Table 3.3, which will be explained when encountered.

3.3.2 Baseline Extraction Model

Our baseline extraction model is an event-time pair classifier based on a pre-trained language model [41, 42, 43] 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:

$$p_{i,j,k} = \sigma(\boldsymbol{w}_{2,k}\text{ReLU}(\boldsymbol{W}_1[\boldsymbol{h}_{e_i}; \boldsymbol{h}_{t_j}] + \boldsymbol{b}_1) + b_{2,k}) \tag{3.2}$$

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 [44]. 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,

$$t_i \rightarrow \langle t^-_i, t^+_i \rangle \tag{3.3}$$

where t^- represents the earliest possible date and t^+ represents the latest possible date.

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 $\langle$`2020-01-01`, `2020-01-03`, `2020-01-01`, `2020-01-07`$\rangle$ 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⟩.

3.3.3 Event Graph Construction

Before we conduct the global time propagation, we first construct document-level event graphs. In this chapter, 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 $r_{i,j}$ denotes the bi-directed edge between e_i and v_j, namely the argument role. For example, in Figure 3.1, 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 self-loops 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 self-loops 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 [33] to extract temporal relations for event pairs that appear in the same sentence or two consecutive sentences, and we only keep relations with over 90% confidence score.

3.3.4 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 [3, 45, 46, 47, 48] have shown effective for relational reasoning [49, 50]. We adopt graph attention networks (GAT [3]) 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 3.3.2), i.e., between the pre-trained language model encoder and the 2-layer feed-forward neural classifier (Eq. (3.2)). 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 [51] 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 $\boldsymbol{h}^k_{v_i}$.

$$\alpha_{ij} = \frac{\exp(a_{ij})}{\sum_{k \in \mathcal{N}(i)} \exp(a_{ik})} \tag{3.4}$$

$$\boldsymbol{h}'_{v_i} = \textsc{Elu}\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} \boldsymbol{W}_5 \boldsymbol{h}_{v_j} \right) \tag{3.5}$$

where Elu is exponential linear unit [52], 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

$$a_{ij} = \sigma\left(\boldsymbol{w}_4 \left[\boldsymbol{W}_3 \boldsymbol{h}_{v_i}; \boldsymbol{W}_3 \boldsymbol{h}_{v_j}; \boldsymbol{\phi}_{r_{i,j}} \right] \right) \tag{3.6}$$

where σ is LeakyReLU [52] 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,

$$\boldsymbol{h}'_{v_i} = \Big\Vert_{k=1}^{m} \boldsymbol{h}'^{k}_{v_i}. \tag{3.7}$$

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. (3.2) to generate the corresponding probabilities.

3.3.5 Training Objective

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

$$\begin{aligned}\mathcal{L}(\tau_{i,k}, t_j) \quad &= \quad \mathbb{1}(\tau_{i,k} = t_j) \log p_{i,j,k} \\ &\quad + \mathbb{1}(\tau_{i,k} \neq t_j) \log(1 - p_{i,j,k}) \end{aligned} \tag{3.8}$$

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. (3.1) and we leave enforcing the constraints for future work.

3.4 EXPERIMENTS

3.4.1 Data and Experiment Setting

We conduct our experiments on previously introduced annotated data. Statistics of the dataset and splits are shown in Table 3.2.

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 providesthe 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-AFTER, 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 [53, 54]. 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 3.3.2. We use two contextualized embedding methods, RoBERTa [42] and Longformer [43], which provide sentence-level[3] 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 [55] 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[4] 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

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

[4]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 3.4: 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. (3.9)).

slot filling evaluation [1]. For exact match 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,

$$Q(\hat{\tau}_i, \tau_i) = \frac{1}{4} \sum_{\substack{s \in \{+,-\}, \\ p \in \text{start,end}}} \frac{1}{1 + |\hat{\tau}^s_{i,\text{p}} - \tau^s_{i,\text{p}}|}. \tag{3.9}$$

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 $\frac{1}{1+|\texttt{inf}-\texttt{2001-01-02}|} = 0$.

3.4.2 Results

Our experiment results are shown in Table 3.4. From the results of directly converting sentence-level time arguments to 4-tuple representation, we can find that local time informa-

tion 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 [6, 56] for NIST SM-KBP 2020 evaluation and achieves top performance at intrinsic temporal evaluation.

3.4.3 Qualitative Analysis

Table 3.5 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.

... Meanwhile Blair arrived in Washington late Wednesday[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[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[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 $\overset{\text{BEFORE}}{\longrightarrow}$ secured

Table 3.5: 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.

3.5 RELATED WORK

Event Temporal Anchoring. Event temporal anchoring is first introduced by [14] using temporal links (TLINKS) to specify the relation among events and time. However, the TimeBank Corpus and TimeBank Dense Corpus using TimeML scheme [12, 13, 15] is either too vague and sparse or is dense only with limited scope. Recently, [16] 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 tried to use Markov logical networks or ILP to propagate relations [26, 27, 28, 29], sieve-based classification [53], and neural networks based methods [25, 30, 57]. There are also efforts on event-event temporal relations [31, 32, 33, 34].

Especially, [54] propose a decision tree that uses a neural network based classifier to find start and end time on [16]. [58] use event time to construct relative timeline.

Temporal Slot Filling. Earlier work on extracting 4-tuple representation focuses on temporal slot-filling (TSF, 1, 2) to collect 4-tuple dates as temporal boundaries for entity attributes. The attempts on TSF include pattern matching [59] and distant supervision [2, 60, 61, 62, 63, 64]. 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 [65] and propagate time information [66] based on hard constraints learned from annotated data. Our work is largely inspired from [67] 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 [68], which has been applied to many NLP tasks such as modeling local coherence [69] and event schema induction [70].

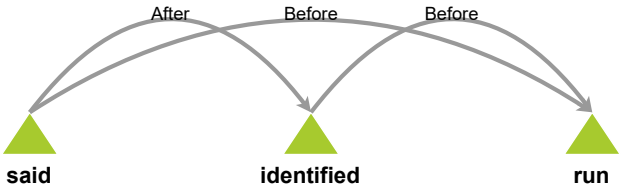
3.6 SUMMARY

In this chapter, 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.

CHAPTER 4: INCORPORATING RELATIVE TIMESTAMPS INTO EVENT-EVENT TEMPORAL RELATION EXTRACTION

4.1 INTRODUCTION

Temporal ordering is an important task to understand the evolving of events. Event temporal relation extraction is to automatic extract the temporal order given a pair of events and to further build a temporal graph. Figure 4.1 illustrates an example sentence and their temporal graph. There are three events in the sentence, **said**, **identified** and **run**. The temporal relation between **said** and **identified** is AFTER, and the temporal relations between **said** and **run** and between **identified** and **run** are BEFORE.



Microsoft **(e1, said)** it has **(e2, identified)** three companies for the China program to **(e3, run)** through June .

Figure 4.1: An example sentence and their temporal relations. In this example, there three different events, and the final extracted graph shows the pair-wise temporal relation extraction results.

Neural network based methods have achieved promising improvement for temporal relation extraction [25, 30, 57, 71]. They mostly consider the task as pairwise classification. There are also efforts focusing on the global structures, including Markov logical networks and Integer Linear Programming (ILP) based methods [26, 27, 28, 29, 31, 32, 33, 34]. Though achieving great success, event time, an important feature, is often overlooked by previous work. Ideally, if we know the exact time information for all events, their temporal relations can by naturally derived. For example, if we know that event A happened on Monday while event B happened on Tuesday in the same week. It is obvious that A happened BEFORE B. However, explicit time arguments can be rarely found in text, especially in news articles. For example, in ACE 2005 dataset, only about one third of the events in news articles can find explicit time arguments, which makes it difficult to find and compare the event time between two events.

In this chapter, we use an auxiliary task that try to predict the relative timestamps [58] for events from temporal relation data to address the sparsity of time arguments in local context. Given the temporal relation between two events, we adapt margin-based optimization to

constrain the distance between two relative event time. Instead of directly comparing relative timestamps as in [58], we them as additional features for temporal relation classification so that the classification can benefits from both representations of event pairs as well as the predicted relative event time. We further use the stack-propagation framework [72] that allows us to jointly train relative timestamp prediction and temporal relation classification.

Our experiments show that the relative timestamp prediction can significantly help learn better temporal relation classification, compared to vanilla RoBERTa-based [42] baseline. We have also achieved the similar performance compared to the state-of-the-art temporal relation classification system that uses additional data [71].

4.2 APPROACH

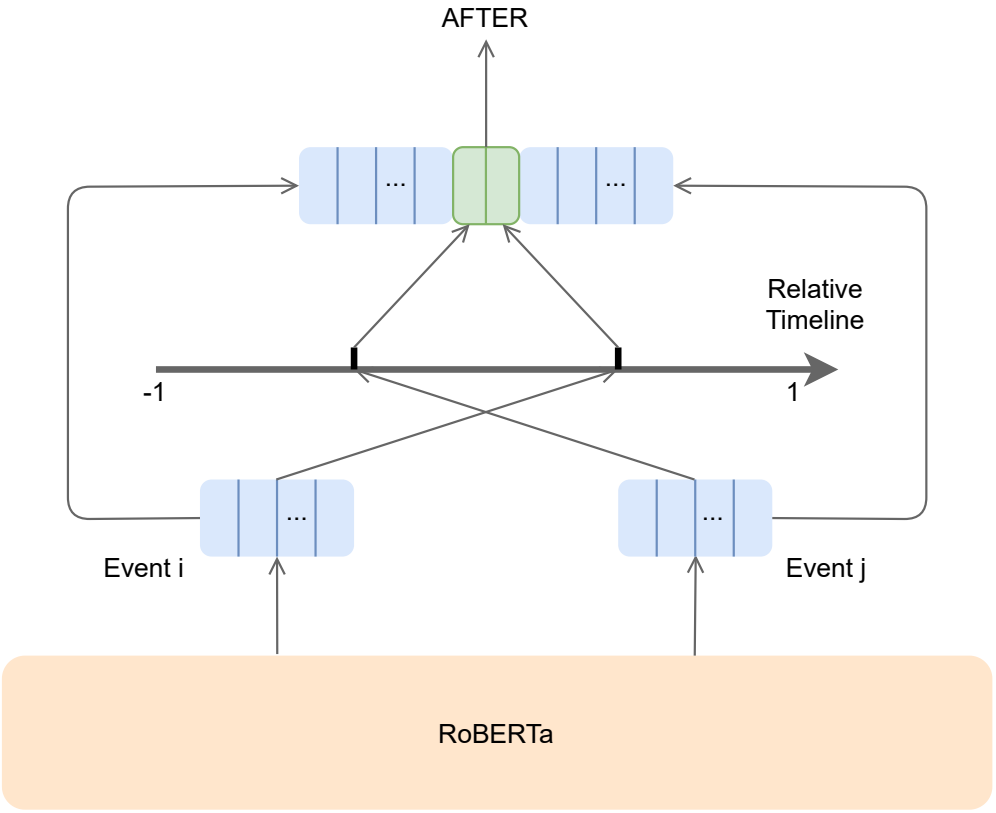


Figure 4.2: The overall architecture.

In this section, we will discuss the relative event timestamp prediction for temporal relation extraction. We will first discuss our pretrained language model based baseline method. Then we will discuss the relative timestamp prediction and incorporate it to temporal relation extraction in a stack-propagation framework. The overall approach is illustrated in Figure 4.2.

4.2.1 Our Baseline Model

Our baseline is a pretrained language model based pairwise classification model. It takes a sequence of tokens X with length n as input after subword tokenization [41] or byte pair encoding (BPE) [42]. The input also includes the positions of two events (e_i, e_j) in the text. For simplicity, we only use the start positions from the corresponding event spans. We denote the start position of an event e_i as p_i. The baseline model is to predict the temporal relations between the given two events.

The model first computes the contextualized representation for each input token using pretrained language model [41, 42]. We denote the contextualized representation as $\boldsymbol{H}$, where $\boldsymbol{h}_i$ is the contextualized representation for token at position i.

Then we concatenate the representations of the given two events using the representation at their corresponding start positions,

$$\boldsymbol{c}_{i,j} = \left[\boldsymbol{h}_{p_i}; \boldsymbol{h}_{p_j}\right]. \tag{4.1}$$

We use a feed-forward neural network (FFN) layer with a softmax layer to convert the representation into a probability distribution,

$$P(r \mid e_i, e_j) = \text{softmax}\left(\text{FFN}_1(\boldsymbol{c}_{i,j})\right) = \text{softmax}\left(\boldsymbol{W}_1\boldsymbol{c}_{i,j} + \boldsymbol{b}_1\right). \tag{4.2}$$

4.2.2 Relative Timestamp Prediction

To better utilize the contextual information for events, we use an auxiliary task, relative timestamp prediction, to predict event time for all events given its context, similar to [58]. Contrary to the above baseline method that takes a pair of representations and predict the pair-wise relation, event time information is only related to its event itself. Therefore, we predict the relative event time information by mapping the representation of an event e_i from pretrained language model to a numerical logit, $t_i \in (-1, 1)$. We use a linear mapping with a tanh activation,

$$t_i = \tanh(\text{FFN}_2(\boldsymbol{h}_{p_i})). \tag{4.3}$$

Although we may not have explicit time information in the given context, the gold standard pair-wise temporal relations can be considered as an incidental supervision to constrain the predicted timestamps. For example, given two events e_i and e_j, and their temporal relation e_i BEFORE e_j, then their predicted timestamp t_i and t_j should follow $t_i < t_j$. Similarly, if their relation is EQUAL, then the distance of their predicted timestamp should be as close as

possible.

Motivated from the above connection between event time and temporal relations, we use a margin-based optimization method to constrain our predicted event timestamp. We use different margins based on different temporal relations,

$$\begin{aligned}\mathcal{L}_t \quad = \quad & \mathbb{1}[r_{(e_i,e_j)} = \textsc{Before}] \max(0, 1 - (t_j - t_i)) \\ & + \mathbb{1}[r_{(e_i,e_j)} = \textsc{After}] \max(0, 1 - (t_i - t_j)) \\ & + \mathbb{1}[r_{(e_i,e_j)} = \textsc{Equal}] |t_i - t_j|. \end{aligned} \tag{4.4}$$

If e_i is Before e_j, the above optimization will maximize the distance $(t_j - t_i)$ unless it is equal or larger than 1, which follows the constraint $t_i < t_j$. On the contrary, If e_i is After e_j, it will maximize the distance $(t_i - t_j)$, which follows the constraint $t_i > t_j$. If e_i is Equal e_j, then it instead minimize the distance $|t_i - t_j|$.

4.2.3 Stack-Propagation on Relative Timestamp

After obtaining relative timestamp for each events, we would like to further incorporate this predicted feature into temporal relation extraction. Since both relative timestamp prediction and temporal relation extraction are based on contextualized representation from the pretrained language model, we adopt Stack-Propagation framework to connect this two task while keep the differentiability.

Specifically, besides event-pair contextualized representation that the baseline method used for pair-wise temporal relation classification, we further incorporate their predicted relative timestamps into classification,

$$P(r \mid e_i, e_j) = \text{softmax}\left(\text{FFN}_1([\boldsymbol{c}_{i,j}; t_i; t_j])\right). \tag{4.5}$$

During training, we use the cross entropy for classification,

$$\mathcal{L}_r = -\log P(r = r_{(e_i,e_j)} \mid e_i, e_j). \tag{4.6}$$

The final training objective will be the interpolation of the classification task and timestamp prediction task,

$$\mathcal{L} = \alpha \mathcal{L}_t + \mathcal{L}_r. \tag{4.7}$$

Since we keep the differentiability for classification, the gradient from cross entropy can be propagated to timestamps and their following calculation.

	Train	Development	Test
Docs	260	21	20
Relations	10,888	1,852	840

Table 4.1: Data splits and statistics on MATRES.

4.3 EXPERIMENTS

4.3.1 Dataset

We conduct our experiments on MATRES [11]. This dataset contains refined annotation on TimeBank and TempEval documents. We follow the previous work that use TimeBANK and AQUAINT as the training and development set. We randomly select 21 documents as development set. We use Platinum as the test set. The detailed statistics can be found in Table 4.1.

4.3.2 Experimental Setup

In our experiments, we use RoBERTa-large[1] as our pretrained language model. Our best model is optimized using AdamW for 30 epochs with learning rate between {1e-5, 2e-5} for both pretrained model and other parameters. We use linear scheduler with warmup proportion at 0.1. We set weight decay to 0.01 and set dropout rate to 0.1 for all parameters. The training batch size is 16. We use 5 different random seed for our experiments, and we choose the learning rate and model with the best averaged performance on development set for comparison on test set.

We use F_1 to evaluate our system performance, following [33], where we consider VAGUE as "no relation" to calculate the Precision and Recall. We compare our model with existing systems including:

- A BiLSTM based joint event and temporal relation extraction model with MAP inference [34].
- LSTM-based method incorporating pretrained language model embedding, commensense prior (TEMPROB) and ILP [33].
- A constrained learning based optimization for joint temporal and hierarchy relation extraction [73].

[1] `https://huggingface.co/transformers/pretrained_models.html`

- Multi-task self-training on temporal relation extraction using additional time annotation from ACE2005 and original TimeBank [71].

Model	Precision	Recall	F_1
BiLSTM+MAP [34]	-	-	75.5
LSTM+TemProb+ILP [73]	71.3	82.1	76.3
Constrained Learning [71]	73.4	85.0	78.8
Self-Training [71]	-	-	81.6
Our Model	**78.4**	**85.2**	**81.7**

Table 4.2: Temporal relation extraction results on MATRES. Precision and recall are not reported by [34, 71]. We report our averaged test performance on all metrics.

4.3.3 Overall Performance

Our overall performance are shown in Table 4.2. Among those baseline systems, the multi-task self-training method [71] have achieved the best performance. While our proposed method can achieve slightly better performance against their system, without introducing additional annotated and raw data, which demonstrates the effectiveness of our the relative timestamp prediction objective, and the stack-propagation based method to incorporate predicted timestamps.

4.3.4 Ablation Study

Model	Precision	Recall	F_1
RoBERTa baseline	78.1	82.5	80.2
RoBERTa+Time Prediction	76.5	**85.2**	80.6
Our Model	**78.4**	**85.2**	**81.7**

Table 4.3: Ablation study on our proposed method. We report our averaged test performance on all metrics.

We also conduct ablation study for relative timestamp prediction objective and stack-propagation to demonstrate the effectiveness of each component in Table 4.3. Compared to the model that only uses relative timestamp prediction objective via vanilla multitask training, we can find a significant performance drop on precision and the final F_1, which indicates the importance of explicitly incorporating predicted timestamp for relation extraction. While

the model that only uses relative timestamp prediction objective can still outperform vanilla RoBERTa baseline relation extraction model.

4.4 RELATED WORK

Earlier efforts on temporal relation extraction focus on using Markov logical networks or ILP to propagate relations [26, 27, 28, 29]. [53] then proposed a sieve-based classification method. Neural network-based methods have also achieved promising improvement [25, 30, 57]. Efforts on temporal relations between events also include incorporating contextual and syntactic information and commonsense database [33] via joint learning [32, 34, 71, 73] and structural learning [31]. Especially, [58] proposed a similar relative timestamp prediction objective and directly use the comparison of relative timestamps as temporal relations. While our work focuses on jointly train relative timestamp prediction and temporal relation extraction via a Stack-Propagation framework.

4.5 SUMMARY

In this section, we used relative event timestamp prediction that can ground event onto a relative timeline to help event-event temporal relation extraction. We use Stack-Propagation framework to further incorporate predicted timestamp explicit for relation classification. Our experiment results demonstrate the effectiveness of our proposed method.

CHAPTER 5: APPLICATIONS OF TEMPORAL INFORMATION EXTRACTION

Our temporal information extraction models and algorithms can be further integrated into applications related to knowledge extraction. In this section, we will introduce two typical knowledge extraction systems that have successfully integrated models in this thesis and have achieved top performance at NIST evaluations.

5.1 GAIA AT SMP 2020: MULTI-MEDIA MULTI-LINGUAL KNOWLEDGE EXTRACTION AND TEMPORAL TRACKING SYSTEM

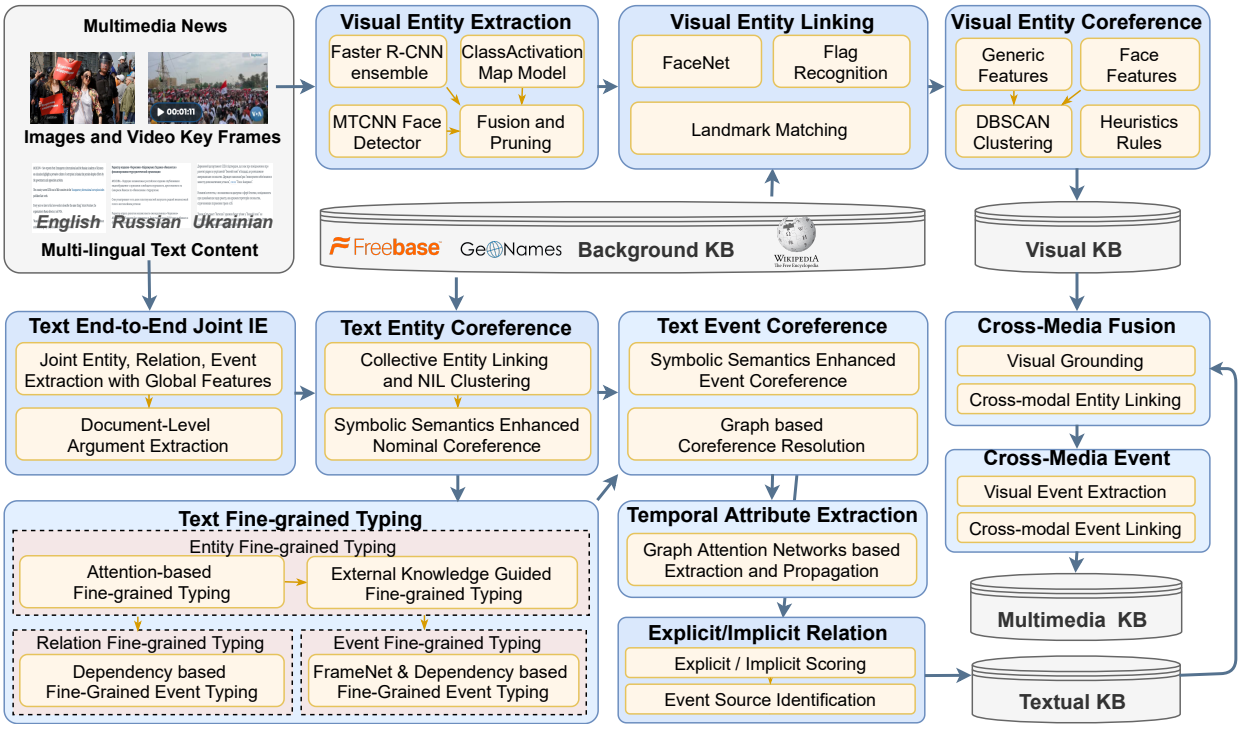


Figure 5.1: Overall architecture of GAIA at SMP 2020.

GAIA is an open-source multi-media multi-lingual knowledge extraction and temporal tracking system. GAIA can extract entities, relations and events from English, Spanish and Russian documents and multimedia input such as images and videos, and link those knowledge elements to external knowledge bases. The extracted knowledge graph can be further used for hypothesis generation. We integrate our 4-tuple document-level event time extraction algorithm into GAIA system for SMP 2020 evaluation. Our system has achieved

the best performance on overall knowledge extraction as well as on event time extraction. The overall architecture is illustrated in Figure 5.1.

5.2 RESIN: A SCHEMA-GUIDED CROSS-DOCUMENT CROSS-LINGUAL CROSS-MEDIA INFORMATION EXTRACTION AND EVENT TRACKING SYSTEM

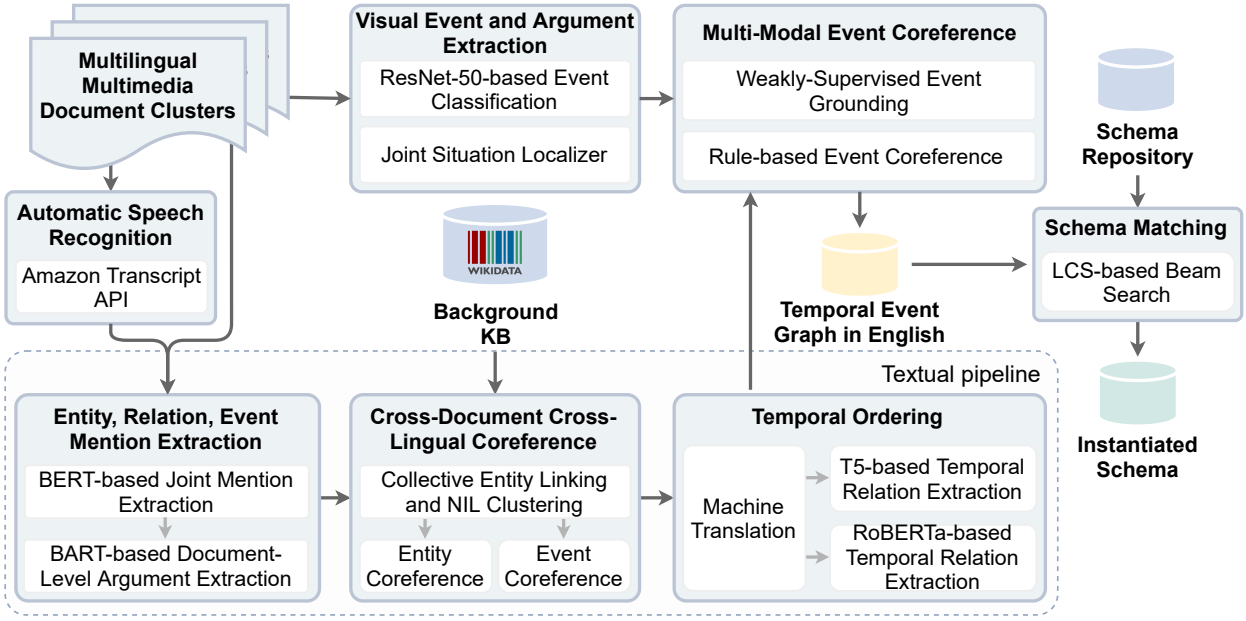


Figure 5.2: Overall architecture of RESIN.

RESIN is a schema-guided cross-document cross-lingual cross-media information extraction and event tracking system. Compared to GAIA, RESIN takes a batch of multi-media English and Spanish documents and extracts cross-document knowledge elements. The extracted knowledge elements will be ordered by the event-event temporal relation extraction system. We will further find coreferential events and arguments extracted from videos and images sources and ground them to our text-based event graph. The final multi-modal event graph will be aligned with a schema from human curated schema repository. To improve the alignment performance, we use two different temporal relation extraction systems to construct two different graphs, and we choose the graph that matches best with a schema as the final extracted graph.The final schema-augmented output can be further used in applications such as event prediction. The overall architecture is illustrated in Figure 5.2.

Figure 5.3 illustrates a subset of examples for the best matched results from our end-to-end system. We can see that our system can extract events, entities and relations and align them

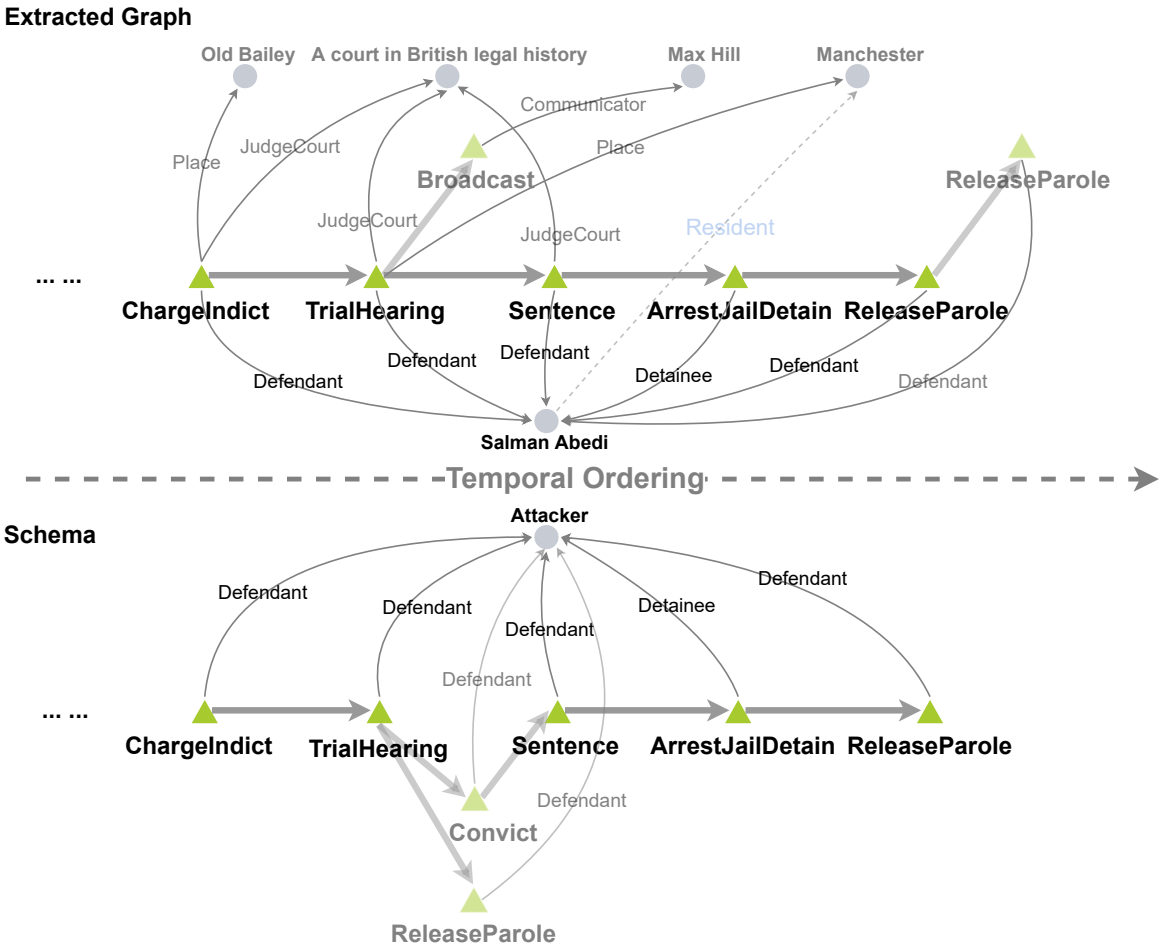


Figure 5.3: An example of extracted knowledge graph and schema matching. The unmatched knowledge elements are blurred.

well with the selected schema. The final instantiated schema is the hybrid of two graphs from merging the matched elements. The DARPA program's phrase 1 human assessment on about 25% of our system output shows that about 70% of events are correctly extracted.

CHAPTER 6: CONCLUSION AND FUTURE DIRECTIONS

6.1 CONCLUSIONS

In this thesis, we first explored new representation for event time in temporal information extraction. In Chapter 2, to address the uncertainty and scarcity of explicit event time arguments, we proposed to adopt 4-tuple representation from TAC-KBP 2011 temporal slot filling to represent event time. This representation is more flexible than event time arguments, especially for those events that we cannot determine their accurate time. We provided a new benchmark on 4-tuple representation, and further proposed a time propagation framework based on constructed event graphs.

In Chapter 3, we focused on incorporating even time information into temporal relation extraction. Instead of extracting absolute time from document, we used an auxiliary task to predict relative timestamps for events from event-event temporal relation annotations. One advantage of using relative timestamp is that we can perform time prediction for every events in the given context, while absolute time is very sparse and hard to extraction. Besides, relative timestamps can be directly used as features for temporal relation extraction, while text-based time needs further normalization. We then used a Stack-Propagation framework to jointly those two tasks to promote each other.

In Chapter 4, we demonstrated two knowledge extraction systems that have integrated the temporal information extraction models discussed in this thesis. The integrated systems have achieved the state-of-the-art performance in end-to-end evaluations.

6.2 FUTURE DIRECTIONS

Although we have witnessed the great improvement for temporal information extraction, there are still many remaining challenges and open questions. Event time in natural language conveys textual information as well as the absolute time or duration information. For example, if we mention "night", it may indicate time after 7pm, while it also implies events such as "sleep". It is important to discover a hybrid textual and numeric representation for event time, especially for cases that involves multiple temporal clues.

Besides, It is still very challenging to create large and clean annotation for many temporal information extraction tasks, such as document-level temporal relations and our proposed 4-tuple event time extraction. We usually need experts to annotate those resources, and we need to further adjudication and cleaning, which is expensive, time-consuming and in

a small scale. However, we are still suffering low inter-annotation agreement for those tasks. Therefore, it is important to investigate a crowd-sourcing approach to annotate those resources in a large scale with high confidence.

Although we have made many efforts on document-level temporal information extraction, it is still possible that many events' temporal information are missing, especially for major events. For example, the exact dates are often missing for Iraq War in ACE2005 corpus. We still need to further support temporal information extraction from document level to corpus level.

REFERENCES

[1] H. Ji, R. Grishman, and H. T. Dang, "An overview of the tac2011 knowledge base population track," in *Proceedings of Text Analysis Conference (TAC2011)*, 2011.

[2] H. Ji, T. Cassidy, Q. Li, and S. Tamang, "Tackling representation, annotation and classification challenges for temporal knowledge base population," *Knowledge and Information Systems*, vol. 41, no. 3, pp. 611–646, 2013.

[3] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "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, 2018. [Online]. Available: https://openreview.net/forum?id=rJXMpikCZ

[4] H. Wen, Y. Qu, H. Ji, Q. Ning, J. Han, A. Sil, H. Tong, and D. Roth, "Event time extraction and propagation via graph attention networks," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT2021)*, 2021.

[5] H. Wen, Y. Lin, T. M. Lai, X. Pan, S. Li, X. Lin, B. Zhou, M. Li, H. Wang, H. Zhang, X. Yu, A. Dong, Z. Wang, Y. R. Fung, P. Mishra, Q. Lyu, D. Surís, B. Chen, S. W. Brown, M. Palmer, C. Callison-Burch, C. Vondrick, J. Han, D. Roth, S.-F. Chang, and H. Ji, "Resin: A dockerized schema-guided cross-document cross-lingual cross-media information extraction and event tracking system," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT2021) Demo Track*, 2021.

[6] M. Li, Y. Lin, T. M. Lai, X. Pan, H. Wen, S. Li, Z. Wang, P. Yu, L. Huang, D. Lu, Q. Wang, H. Zhang, Q. Zeng, C. Han, Z. Zhang, Y. Qin, X. Hu, N. Parulian, D. Campos, H. Ji, B. Chen, X. Lin, A. Zareian, A. Ananthram, E. Allaway, S.-F. Chang, K. McKeown, Y. Yao, M. Spector, M. DeHaven, D. Napierski, M. Freedman, P. Szekely, H. Zhu, R. Nevatia, Y. Bai, Y. Wang, A. Sadeghian, H. Ma, and D. Z. Wang, "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)*, 2020.

[7] L. Ferro, L. Gerber, I. Mani, S. Beth, and G. Wilson, "TIDES 2003 standard for the annotation of temporal expressions," 2005.

[8] L. D. Consortium, "ACE (Automatic Content Extraction) English annotation guidelines for entities, version6.6 2008.06.13 edition," 2008.

[9] L. D. Consortium, "ACE (Automatic Content Extraction) English annotation guidelines for events, version 5.4.3 2005.07.01 edition," 2005.

[10] J. F. Allen, "Towards a general theory of action and time," *Artif. Intell.*, vol. 23, no. 2, pp. 123–154, 1984. [Online]. Available: https://doi.org/10.1016/0004-3702(84)90008-0

[11] Q. Ning, H. Wu, and D. Roth, "A multi-axis annotation scheme for event temporal 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*, I. Gurevych and Y. Miyao, Eds. Association for Computational Linguistics, 2018. [Online]. Available: https://www.aclweb.org/anthology/P18-1122/ pp. 1318–1328.

[12] J. Pustejovsky, P. Hanks, R. Sauri, A. See, R. Gaizauskas, A. Setzer, D. Radev, B. Sundheim, D. Day, L. Ferro et al., "The timebank corpus," in *Corpus linguistics*, vol. 2003. Lancaster, UK., 2003, p. 40.

[13] J. Pustejovsky, J. M. Castaño, R. Ingria, R. Saurí, R. J. Gaizauskas, A. Setzer, G. Katz, and D. R. Radev, "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*, M. T. Maybury, Ed. AAAI Press, 2003, pp. 28–34.

[14] A. Setzer, "Temporal information in newswire articles: an annotation scheme and corpus study." Ph.D. dissertation, University of Sheffield, 2002.

[15] T. Cassidy, B. McDowell, N. Chambers, and S. Bethard, "An annotation framework for dense event ordering," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Baltimore, Maryland: Association for Computational Linguistics, June 2014. [Online]. Available: https://www.aclweb.org/anthology/P14-2082 pp. 501–506.

[16] N. Reimers, N. Dehghani, and I. Gurevych, "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)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016. [Online]. Available: https://www.aclweb.org/anthology/P16-1207 pp. 2195–2204.

[17] N. UzZaman, H. Llorens, L. Derczynski, J. Allen, M. Verhagen, and J. Pustejovsky, "SemEval-2013 task 1: TempEval-3: Evaluating time expressions, events, and temporal relations," in *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. Atlanta, Georgia, USA: Association for Computational Linguistics, June 2013. [Online]. Available: https://www.aclweb.org/anthology/S13-2001 pp. 1–9.

[18] L. Huang and L. Huang, "Optimized event storyline generation based on mixture-event-aspect model," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, Oct. 2013. [Online]. Available: https://www.aclweb.org/anthology/D13-1068 pp. 726–735.

[19] L. Wang, C. Cardie, and G. Marchetti, "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*. Denver, Colorado: Association for Computational Linguistics, May–June 2015. [Online]. Available: https://www.aclweb.org/anthology/N15-1112 pp. 1055–1065.

[20] T. Ge, W. Pei, H. Ji, S. Li, B. Chang, and Z. Sui, "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)*. Beijing, China: Association for Computational Linguistics, July 2015. [Online]. Available: https://www.aclweb.org/anthology/P15-1056 pp. 575–585.

[21] J. Steen and K. Markert, "Abstractive timeline summarization," in *Proceedings of the 2nd Workshop on New Frontiers in Summarization*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019. [Online]. Available: https://www.aclweb.org/anthology/D19-5403 pp. 21–31.

[22] H. Ji, R. Grishman, Z. Chen, and P. Gupta, "Cross-document event extraction and tracking: Task, evaluation, techniques and challenges," in *Proceedings of the International Conference RANLP-2009*. Borovets, Bulgaria: Association for Computational Linguistics, Sep. 2009. [Online]. Available: https://www.aclweb.org/anthology/R09-1032 pp. 166–172.

[23] A.-L. Minard, M. Speranza, E. Agirre, I. Aldabe, M. van Erp, B. Magnini, G. Rigau, and R. Urizar, "SemEval-2015 task 4: TimeLine: Cross-document event ordering," in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Denver, Colorado: Association for Computational Linguistics, June 2015. [Online]. Available: https://www.aclweb.org/anthology/S15-2132 pp. 778–786.

[24] H. Llorens, N. Chambers, N. UzZaman, N. Mostafazadeh, J. Allen, and J. Pustejovsky, "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)*. Denver, Colorado: Association for Computational Linguistics, June 2015. [Online]. Available: https://www.aclweb.org/anthology/S15-2134 pp. 792–800.

[25] Y. Meng, A. Rumshisky, and A. Romanov, "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*, M. Palmer, R. Hwa, and S. Riedel, Eds. Association for Computational Linguistics, 2017. [Online]. Available: https://doi.org/10.18653/v1/d17-1092 pp. 887–896.

[26] P. Bramsen, P. Deshpande, Y. K. Lee, and R. Barzilay, "Inducing temporal graphs," in *EMNLP 2006, Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, 22-23 July 2006, Sydney, Australia*, D. Jurafsky and É. Gaussier, Eds. ACL, 2006. [Online]. Available: https://www.aclweb.org/anthology/W06-1623/ pp. 189–198.

[27] N. Chambers and D. Jurafsky, "Jointly combining implicit constraints improves temporal ordering," in *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. Honolulu, Hawaii: Association for Computational Linguistics, Oct. 2008. [Online]. Available: https://www.aclweb.org/anthology/D08-1073 pp. 698–706.

[28] K. Yoshikawa, S. Riedel, M. Asahara, and Y. Matsumoto, "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*. Suntec, Singapore: Association for Computational Linguistics, Aug. 2009. [Online]. Available: https://www.aclweb.org/anthology/P09-1046 pp. 405–413.

[29] Q. Do, W. Lu, and D. Roth, "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*. Jeju Island, Korea: Association for Computational Linguistics, July 2012. [Online]. Available: https://www.aclweb.org/anthology/D12-1062 pp. 677–687.

[30] Y. Meng and A. Rumshisky, "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*, I. Gurevych and Y. Miyao, Eds. Association for Computational Linguistics, 2018. [Online]. Available: https://www.aclweb.org/anthology/P18-1049/ pp. 527–536.

[31] Q. Ning, Z. Feng, and D. Roth, "A structured learning approach to temporal relation extraction," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017. [Online]. Available: https://www.aclweb.org/anthology/D17-1108 pp. 1027–1037.

[32] Q. Ning, Z. Feng, H. Wu, and D. Roth, "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*, I. Gurevych and Y. Miyao, Eds. Association for Computational Linguistics, 2018. [Online]. Available: https://www.aclweb.org/anthology/P18-1212/ pp. 2278–2288.

[33] Q. Ning, S. Subramanian, and D. Roth, "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*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Association for Computational Linguistics, 2019. [Online]. Available: https://doi.org/10.18653/v1/D19-1642 pp. 6202–6208.

[34] R. Han, Q. Ning, and N. Peng, "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*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Association for Computational Linguistics, 2019. [Online]. Available: https://doi.org/10.18653/v1/D19-1041 pp. 434–444.

[35] F. Pan, R. Mulkar, and J. R. Hobbs, "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*. Sydney, Australia: Association for Computational Linguistics, July 2006. [Online]. Available: https://www.aclweb.org/anthology/P06-1050 pp. 393–400.

[36] F. Pan, R. Mulkar-Mehta, and J. R. Hobbs, "Annotating and learning event durations in text," *Computational Linguistics*, vol. 37, no. 4, pp. 727–752, 2011. [Online]. Available: https://www.aclweb.org/anthology/J11-4005

[37] A. Vempala, E. Blanco, and A. Palmer, "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)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018. [Online]. Available: https://www.aclweb.org/anthology/N18-2026 pp. 164–168.

[38] A. Gusev, N. Chambers, D. R. Khilnani, P. Khaitan, S. Bethard, and D. Jurafsky, "Using query patterns to learn the duration of events," in *Proceedings of the Ninth International Conference on Computational Semantics (IWCS 2011)*, 2011. [Online]. Available: https://www.aclweb.org/anthology/W11-0116

[39] S. Vashishtha, B. Van Durme, and A. S. White, "Fine-grained temporal relation extraction," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019. [Online]. Available: https://www.aclweb.org/anthology/P19-1280 pp. 2906–2919.

[40] B. Zhou, D. Khashabi, Q. Ning, and D. Roth, ""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)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019. [Online]. Available: https://www.aclweb.org/anthology/D19-1332 pp. 3363–3369.

[41] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "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)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019. [Online]. Available: https://www.aclweb.org/anthology/N19-1423 pp. 4171–4186.

[42] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019. [Online]. Available: http://arxiv.org/abs/1907.11692

[43] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," *CoRR*, vol. abs/2004.05150, 2020. [Online]. Available: https://arxiv.org/abs/2004.05150

[44] L. Ferro, L. Gerber, I. Mani, B. Sundheim, and G. Wilson, "TIDES2005 standard for the annotation of temporal expressions," *MITRE Corporation Technical Report*, 2005.

[45] H. Dai, B. Dai, and L. Song, "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*, ser. JMLR Workshop and Conference Proceedings, M. Balcan and K. Q. Weinberger, Eds., vol. 48. JMLR.org, 2016. [Online]. Available: http://proceedings.mlr.press/v48/daib16.html pp. 2702–2711.

[46] T. N. Kipf and M. Welling, "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, 2017. [Online]. Available: https://openreview.net/forum?id=SJU4ayYgl

[47] W. L. Hamilton, Z. Ying, and J. Leskovec, "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*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017. [Online]. Available: http://papers.nips.cc/paper/6703-inductive-representation-learning-on-large-graphs pp. 1024–1034.

[48] M. S. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, ser. Lecture Notes in Computer Science, A. Gangemi, R. Navigli, M. Vidal, P. Hitzler, R. Troncy, L. Hollink, A. Tordai, and M. Alam, Eds., vol. 10843. Springer, 2018. [Online]. Available: https://doi.org/10.1007/978-3-319-93417-4_38 pp. 593–607.

[49] Y. Zhang, P. Qi, and C. D. Manning, "Graph convolution over pruned dependency trees improves relation extraction," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018. [Online]. Available: https://www.aclweb.org/anthology/D18-1244 pp. 2205–2215.

[50] D. Marcheggiani, J. Bastings, and I. Titov, "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)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018. [Online]. Available: https://www.aclweb.org/anthology/N18-2078 pp. 486–492.

[51] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "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*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017. [Online]. Available: http://papers.nips.cc/paper/7181-attention-is-all-you-need pp. 5998–6008.

[52] D. Clevert, T. Unterthiner, and S. Hochreiter, "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*, Y. Bengio and Y. LeCun, Eds., 2016. [Online]. Available: http://arxiv.org/abs/1511.07289

[53] N. Chambers, T. Cassidy, B. McDowell, and S. Bethard, "Dense event ordering with a multi-pass architecture," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 273–284, 2014. [Online]. Available: https://www.aclweb.org/anthology/Q14-1022

[54] N. Reimers, N. Dehghani, and I. Gurevych, "Event time extraction with a decision tree of neural classifiers," *Transactions of the Association for Computational Linguistics*, vol. 6, pp. 77–89, 2018. [Online]. Available: https://www.aclweb.org/anthology/Q18-1006

[55] D. P. Kingma and J. Ba, "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*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[56] M. Li, A. Zareian, Y. Lin, X. Pan, S. Whitehead, B. Chen, B. Wu, H. Ji, S.-F. Chang, C. Voss, D. Napierski, and M. Freedman, "GAIA: A fine-grained multimedia knowledge extraction system," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL2020) Demo Track*, 2020.

[57] F. Cheng, M. Asahara, I. Kobayashi, and S. Kurohashi, "Dynamically updating event representations for temporal relation classification with multi-category learning," in *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020. [Online]. Available: https://www.aclweb.org/anthology/2020.findings-emnlp.121 pp. 1352–1357.

[58] A. Leeuwenberg and M.-F. Moens, "Temporal information extraction by predicting relative time-lines," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018. [Online]. Available: https://www.aclweb.org/anthology/D18-1155 pp. 1237–1246.

[59] L. Byrne and J. Dunnion, "UCD IIRG at TAC 2011." in *Proceedings of Text Analysis Conference (TAC2011)*, 2011.

[60] Q. Li, J. Artiles, T. Cassidy, and H. Ji, "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*, ser. Lecture Notes in Computer Science, A. F. Gelbukh, Ed., vol. 7182. Springer, 2012. [Online]. Available: https://doi.org/10.1007/978-3-642-28601-8_17 pp. 194–205.

[61] M. Surdeanu, S. Gupta, J. Bauer, D. McClosky, A. X. Chang, V. I. Spitkovsky, and C. D. Manning, "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, 2011. [Online]. Available: https://tac.nist.gov/publications/2011/participant.papers/Stanford1.proceedings.pdf

[62] A. Sil and S.-P. Cucerzan, "Towards temporal scoping of relational facts based on Wikipedia data," in *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*. Ann Arbor, Michigan: Association for Computational Linguistics, June 2014. [Online]. Available: https://www.aclweb.org/anthology/W14-1612 pp. 109–118.

[63] R. Reinanda, D. Odijk, and d. M. Rijke, "Exploring entity associations over time," in *SIGIR2013; Workshop on time-awareiInformation access*. TAIA'13, 2013.

[64] R. Reinanda and M. de Rijke, "Prior-informed distant supervision for temporal evidence classification," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin, Ireland: Dublin City University and Association for Computational Linguistics, Aug. 2014. [Online]. Available: https://www.aclweb.org/anthology/C14-1094 pp. 996–1006.

[65] M. Tatu and M. Srikanth, "Experiments with reasoning for temporal relations between events," in *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*. Manchester, UK: Coling 2008 Organizing Committee, Aug. 2008. [Online]. Available: https://www.aclweb.org/anthology/C08-1108 pp. 857–864.

[66] P. Gupta and H. Ji, "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*. The Association for Computer Linguistics, 2009. [Online]. Available: https://www.aclweb.org/anthology/P09-2093/ pp. 369–372.

[67] P. P. Talukdar, D. Wijaya, and T. M. Mitchell, "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*, X. Chen, G. Lebanon, H. Wang, and M. J. Zaki, Eds. ACM, 2012. [Online]. Available: https://doi.org/10.1145/2396761.2396886 pp. 992–1001.

[68] B. J. Grosz, A. K. Joshi, and S. Weinstein, "Centering: A framework for modeling the local coherence of discourse," *Computational Linguistics*, vol. 21, no. 2, pp. 203–225, 1995. [Online]. Available: https://www.aclweb.org/anthology/J95-2003

[69] R. Barzilay and M. Lapata, "Modeling local coherence: An entity-based approach," *Computational Linguistics*, vol. 34, no. 1, pp. 1–34, 2008. [Online]. Available: https://www.aclweb.org/anthology/J08-1001

[70] N. Chambers and D. Jurafsky, "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*. Suntec, Singapore: Association for Computational Linguistics, Aug. 2009. [Online]. Available: https://www.aclweb.org/anthology/P09-1068 pp. 602–610.

[71] M. Ballesteros, R. Anubhai, S. Wang, N. Pourdamghani, Y. Vyas, J. Ma, P. Bhatia, K. McKeown, and Y. Al-Onaizan, "Severing the edge between before and after: Neural architectures for temporal ordering of events," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-main.436 pp. 5412–5417.

[72] Y. Zhang and D. Weiss, "Stack-propagation: Improved representation learning for syntax," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016. [Online]. Available: https://www.aclweb.org/anthology/P16-1147 pp. 1557–1566.

[73] H. Wang, M. Chen, H. Zhang, and D. Roth, "Joint constrained learning for event-event relation extraction," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-main.51 pp. 696–706.

[74] M. Gardner, Y. Artzi, V. Basmov, J. Berant, B. Bogin, S. Chen, P. Dasigi, D. Dua, Y. Elazar, A. Gottumukkala, N. Gupta, H. Hajishirzi, G. Ilharco, D. Khashabi, K. Lin, J. Liu, N. F. Liu, P. Mulcaire, Q. Ning, S. Singh, N. A. Smith, S. Subramanian, R. Tsarfaty, E. Wallace, A. Zhang, and B. Zhou, "Evaluating models' local decision boundaries via contrast sets," in *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020. [Online]. Available: https://www.aclweb.org/anthology/2020.findings-emnlp.117 pp. 1307–1323.