Understanding Time in Natural Language

Ph.D. Thesis Proposal

Qiang Ning

Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
201 N Goodwin Ave
Urbana, IL 61801
qning2@illinois.edu

Ph.D. Thesis Committee:

Prof. Dan Roth (Chair)
Prof. Mark Hasegawa-Johnson
Prof. Julia Hockenmaier
Prof. Wen-mei Hwu
Prof. Martha Palmer

Time: Thursday, October 11th, 2018 at 1pm (CDT)
Location: Siebel Center Room 2102

Abstract

Time is an important dimension when we describe the world because the world is evolving over time and many facts are time-sensitive. Understanding time is thus an important aspect of natural language understanding and many applications may rely on it, e.g., information retrieval, summarization, causality, and question answering.

My thesis research has so far focused on a key component of it, temporal relation extraction. That is, given a pair of events or actions, determine which of them occurred first (or other temporal relations between them, e.g., simultaneous or overlapping). The task is challenging because the actual timestamps of those events or actions are rarely expressed explicitly and their temporal order has to be inferred, from lexical cues, between the lines, and often based on strong background knowledge. In addition, collecting enough and high-quality annotations to facilitate machine learning algorithms for the task is also difficult, which makes it even more challenging to investigate the task.

My contribution has been along three directions. One is the *machine learning direction*, where I propose to better exploit the structure induced by the transitivity property of temporal relations in the structured learning framework [1-2]; two is to inject *human prior knowledge* of the typical temporal ordering of events to help resolving cases lack of contextual information [3-4]; three is the *natural language perspective*, where I investigate how temporal relations are expressed (from the authors' standpoint) and perceived (from the readers' standpoint) [5]. *Integrating these improvements, my current system significantly improves the state-of-the-art temporal relation extraction performance by approximately 20% in* F_1. My current system is made public through this online demo `http://groupspaceuiuc.com/temporal/` described by [6].

In my future thesis work, I propose to further investigate the problem of understanding time from the three perspectives above, and also apply the technique on timeline construction. Specifically, I propose to study partial annotations from the standpoint of incidental supervision (Sec. 6.1), to represent and exploit common sense knowledge of time (Sec. 6.2), and to extend the new dataset I collected (Sec. 6.3); with these additional components, I plan to move forward to multi-document timeline construction (Sec. 6.4). This will be practically more useful, but also technically more interesting/challenging due to its requirement for a synergy of entity coreference, event coreference, and temporal relation extraction methods.

Publications

Parts of the work in this proposal have appeared in the following publications:

1. Qiang Ning, Zhili Feng, and Dan Roth, "A Structured Learning Approach to Temporal Relation Extraction," EMNLP, 2017. [link]

2. Qiang Ning, Zhongzhi Yu, Chuchu Fan, and Dan Roth, "Exploiting Partially Annotated Data in Temporal Relation Extraction," *SEM (short paper), 2018. [link]

3. Qiang Ning, Hao Wu, Haoruo Peng, and Dan Roth, "Improving Temporal Relation Extraction with a Globally Acquired Statistical Resource," NAACL, 2018. [link]

4. Qiang Ning, Zhili Feng, and Dan Roth, "Joint Reasoning for Temporal and Causal Relations," ACL, 2018. [link]

5. Qiang Ning, Hao Wu, and Dan Roth, "A Multi-Axis Annotation Scheme for Event Temporal Relations," ACL, 2018. [link]

6. Qiang Ning, Ben Zhou, Zhili Feng, Haoruo Peng, and Dan Roth, "CogCompTime: A Tool for Understanding Time in Natural Language," EMNLP (demo track), 2018. [link]

Contents

1 Introduction

Time is an important dimension when we describe the world and has been involved ubiquitously in many techniques. For instance, in discrete signal processing, the sampling time is needed for spectral analysis; in causality analysis, two time-series or random processes are involved for testing the Granger causality, or for measuring the transfer entropy; in data-mining, many analyses also rely on the timestamps associated with database entries (e.g., purchase records or real estate prices). Similarly, in natural language understanding (NLU), time is also a crucial dimension and the temporal ordering of events is often critical for understanding them. For instance, let $A \rightarrow B$ denote that A is temporally before B. If *people were angry*$\rightarrow$*police suppressed people*, then it leaves an impression that people got angry and perhaps ended up with a violent confrontation with the police, and then the police restored order by suppressing them. Instead, if *police suppressed people*$\rightarrow$*people were angry*, then it means that people got angry only due to the suppression. Similarly, *found a tumor*$\rightarrow$*did a surgery* tells us that the purpose of the surgery is to remove the tumor (so there is probably no tumor anymore), while *did a surgery* $\rightarrow$*found a tumor* sounds like a(nother) tumor is found after the surgery (so there is still a tumor).

Nowadays, with the evergrowing natural language data available in the form of news articles, narratives, books, product reviews, and social network posts, people have realized that it is increasingly desirable to make use of these data to understand how things are evolving along time and hopefully, make decisions based on this understanding. As a result, there have been many studies towards this goal, such as timeline construction (Do et al., 2012, Minard et al., 2015), clinical record analysis (Jindal and Roth, 2013, Bethard et al., 2015), temporal question answering (Llorens et al., 2015), and causality inference (Ning et al., 2018a).

A key difference between the time dimension in NLU and in other techniques is the availability of gold timestamps: In techniques such as signal processing, the gold timestamps often naturally come along with the data, while in natural language text, gold timestamps rarely exist. For instance, a sentence like *people were angry and then the police suppressed people* sounds more natural than a sentence like *people were angry at 8:00 am, 2013-01-02; the police suppressed people at 8:05am, 2013-01-02*. Therefore, to facilitate other applications (e.g., timeline construction and question answering), temporal understanding is necessary.

Temporal understanding in NLU requires two basic components (Verhagen et al., 2007, 2010, UzZaman et al., 2013). The first, also known as the Timex component, requires to understand those explicit time expressions (i.e., "Timex") so that these Timexes can serve as the timestamps we are expecting. In Example 1, *t1:February 27, 1998* is such a Timex. The second basic component of temporal understanding is the temporal relation (i.e., "TempRel") component, which conceptually aims at determining which event or action happens earlier temporally. While Timexes often act as absolute time anchors which carry temporal information *explicitly*, TempRels provide another type of *implicit* temporal information, i.e., the relative order of events, which is important especially in absence of Timexes. In Example 1, there are two events: *e1:exploded* and *e2:died*. The text tells us that *e1* was on *1998-02-27* but does not tell us when *e2* happened exactly. Nevertheless, we

know that there is a TempRel between them, i.e., *e1:exploded* happened *before e2:died*. The Timex component and the TempRel component together provide a more complete picture of the temporal aspect of a story, either explicitly or implicitly, so they are naturally the most important building blocks towards temporal understanding.

Example 1:
A car (*e1:exploded*) in the middle of a group of men playing volleyball on (*t1:February 27, 1998*) and more than 10 people have (*e2:died*).

While the Timex task has been well handled by state-of-the-art systems (Strötgen and Gertz, 2010, Chang and Manning, 2012, Zhao et al., 2012, Chambers, 2013a, Lee et al., 2014) with end-to-end F_1 scores around 80%, the TempRel task is still quite challenging. Even the top systems only achieved F_1 scores of around 35% in the TempEval3 workshop (UzZaman et al., 2013)–the most recent competition on the TempRel task. ***My thesis research has been focused on this TempRel task.*** The difficulty of it is two-fold. First, temporal ordering is often not expressed explicitly in text, so TempRels often have to be inferred, from lexical cues, between the lines, and sometimes even purely based on background knowledge. In Example 2, knowing that people usually become friends before getting married, we understand that *e3* is before *e4*. For a computer, however, identifying this TempRel is very difficult, since it is unclear when "they were in college"; let alone the fact that the narrative order of the two events can even be altered without changing the meaning of the text (as shown by *e5* and *e6*). How to build machine learning algorithms and how to inject human prior knowledge for TempRel extraction is thus the first challenge.

Example 2:
They (*e3:became*) friends in college. They got (*e4:married*) in 2015.
They got (*e5:married*) in 2015. They (*e6:became*) friends in college.

Second, collecting enough, high-quality TempRel annotations is also very challenging. On one hand, annotating the TempRels among n events requires $\mathcal{O}(n^2)$ individual annotations, which makes data annotation difficult to scale up to large datasets. Although existing studies have tried to annotate only between events that are close-by in text, the annotation remains to be so time consuming that existing datasets are still relatively small. On the other hand, for each individual TempRel, the annotation not only requires a TempRel label (e.g., *before* or *after*), but also involves whether the TempRel actually exists[1] and what are the events[2]. These complications do not have a straightforward answer and, despite of the various efforts people have made, still cause low inter-annotation agreements.

To address the aforementioned difficulties of the TempRel task, my contribution has been on three aspects. First, *machine learning* (Sec. 3). TempRels are inherently structured, i.e., one TempRel may be affected by other TempRels. I propose to better exploit the structure

[1]For instance, what if the temporal order is *vague*.
[2]For instance, in "I wanted to leave this place", should we consider "leave" as an event?

induced by the transitivity property of temporal relations in the structured learning framework (Ning et al., 2017, 2018d). Second, *injection of common sense knowledge* (Sec. 4). I have collected a probabilistic knowledge base called TEMPROB to encode humans' prior knowledge of the typical ordering of events, which is also proved to be a useful resource for TempRel extraction Ning et al. (2018b). Third, *data annotation* (Sec. 5). I have investigated into the existing TempRel datasets and proposed a multi-axis modeling for the temporal structure of stories Ning et al. (2018c). *Integrating these components, my current system significantly improves the state-of-the-art temporal relation extraction performance by approximately 20% in* F_1. On top of the progress I have made so far, I propose in Sec. 6 to further advance temporal understanding from two key directions: One is to continue improving the TempRel component (Secs. 6.1-6.3), and the other one is to move forward to multi-document timeline construction (Sec. 6.4). Advice and critiques are very welcome.

2 Preliminaries

2.1 Temporal Graphs

The TempRel task is to determine which event happens temporally earlier or later than the other (or other more fine-grained temporal relations such as *overlapped with*). Technically, when all the events from a document or multiple documents are considered, the TempRel task can be modeled as a graph extraction problem, where the nodes represent events and the edges TempRels. Figure 1 is such a graph representation of the TempRels in Example 3, and we hereafter call such graphs the *temporal graphs*. Valid temporal graphs need to satisfy the following two properties:

1. *Symmetry*. For example, if A is *before* B, then B must be *after* A.
2. *Transitivity*. For example, if A is *before* B and B is *before* C, then A must be *before* C.

Given the symmetry property, we can use a single and directed edge to represent the TempRel between two nodes; as in Example 3, *e9:hurt* is *after e8:ripping*, but in Fig. 1, we are safe to use a single "*before*" relation edge pointing from *e8:ripping* to *e9:hurt*.

Example 3:
. . . tons of earth (*e7:cascaded*) down a hillside, (*e8:ripping*) two houses from their foundations. No one was (*e9:hurt*), but firefighters (*e10:ordered*) the evacuation of nearby homes and said they'll (*e11:monitor*) the shifting ground. . . .

The transitivity property, however, needs to be treated with caution. Specifically, it interrelates all the nodes in a graph, so the decision of one individual TempRel depends on, or even dictated by, TempRels among other nodes. This property has two further implications. First, the TempRel *annotation process* is very challenging even for humans due to this global consideration since it potentially requires an annotation decision to be based on the entire article. The data annotation task hence needs to be designed very carefully. The second

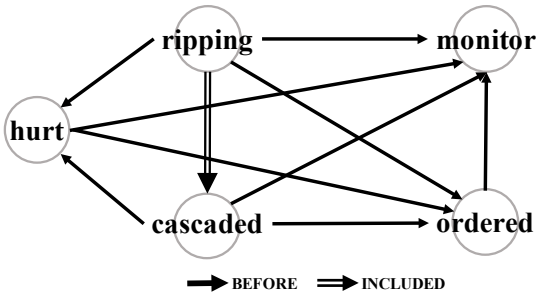


Figure 1: The temporal graph of Example 3, where the nodes represent events, and the edges TempRels between those events.

implication is that TempRel systems also need to produce temporal graphs that respect this transitivity property. There have been many existing work incorporating the global consideration in the *inference* phase, and I have also investigated ways to incorporating the global consideration in the *learning* phase in Ning et al. (2017).

In addition to the two properties, another important issue of temporal graph extraction is the definition of its nodes (or events). Generally speaking, an event is considered to be an action associated with corresponding participants involved in this action. In the literature, however, different definitions are adopted by different applications. In Song et al. (2015) and Mitamura et al. (2015), for example, the definition of events is limited to a set of predefined types, such as "Business", "Conflict", and "Justice". In my thesis work so far, I have been following the event definition of TempEval3 UzZaman et al. (2013), in which the limitation to predefined types is lifted; instead, all terms for situations that happen or occur are considered events, except for generic and temporally static ones. If not stated otherwise, this thesis typically assumes gold events and only the TempRels need to be inferred.

2.2 Temporal Relation Labels

The TempRel between two time points is often straightforward: *before*, *after*, and *equal*. Following Allen (1984), existing work often represent the time scope of an event by an interval, $[t_{start}, t_{end}]$ (i.e., the start- and end-point of an event). Generally speaking, comparing two time intervals is the same as comparing the four time points, which results in 13 different TempRel labels Fig. 2. However, existing works usually use a reduced set of these 13 labels. For instance, in Bethard et al. (2007) and Do et al. (2012), only 4 specific relations were considered: *before*, *after*, *overlap* and *equal*; TimeBank-Dense (Cassidy et al., 2014) uses 5 relations: *before*, *after*, *includes*, *is included* and *equal*. People use a reduced set instead of the original 13 mainly due to the following two reasons.

1. The non-uniform distribution of all the 13 labels makes it difficult to separate low-frequency ones from the others (see Table 1 in Mani et al. (2006)). For example, labels such as *immediately before* or *immediately after*, albeit possible, rarely exist in practice. As a result, intentionally omitting the rarely existing labels in the label set of a system often leads to better performances.

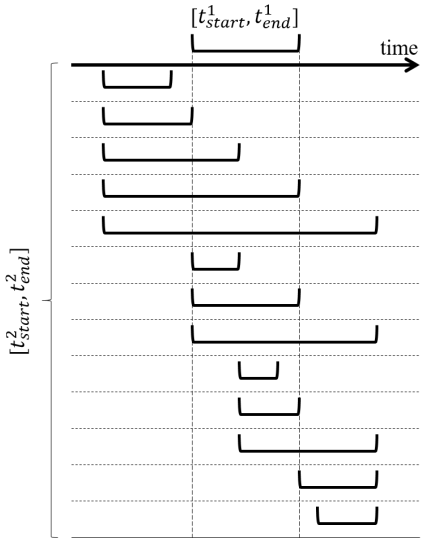


Figure 2: Thirteen possible relations between two events whose timescopes are $[t^1_{start}, t^1_{end}]$ and $[t^2_{start}, t^2_{end}]$ (from top to bottom): *after, immediately after, after and overlap, ends, included, started by, equal, starts, includes, ended by, before and overlap, immediately before* and *before*.

2. Due to the ambiguity in natural language, the subtask of differentiating between *before* and *immediately before* may be not well-defined (Chambers et al., 2014). In Example 4, it is disputable whether *e12:locked* is *before* or *immediately before* due to the ambiguity between $t^{''locked''}_{end}$ and $t^{''left''}_{start}$. In addition, the granularity that the user cares about also affects the decision here. As a result, intentionally reducing these confusing labels often leads to better annotation agreement levels.

Example 4: I (*e12:locked*) the door and (*e13:left*) the place.

In my thesis work, I follow the reduced set used by TimeBank-Dense (i.e., *before*, *after*, *includes*, *is included* and *equal*). However, in Sec. 5, where I introduce my newly collected TempRel dataset, I switch to only focus on the TempRel between the start-points of events, and on that particular dataset, the labels are changed to *before*, *after*, and *equal*. Additionally, in both the aforementioned Bethard et al. (2007), Do et al. (2012), Cassidy et al. (2014), Chambers et al. (2014) and my thesis work, an extra label called *vague* or *none* is also included as another relation type when the TempRel is not clear and no single relations can be convincingly assigned to it. In Example 5, it is unclear whether *e14:ate* or *e15:drank* happened first. Vagueness has long been an issue for the TempRel task: For human annotators, it leads to confusion and lowers the inter-annotator agreement (IAA) levels; for systems, it is very difficult to tell if a TempRel is *vague* or not.

Example 5: I (*e14:ate*) a burger and (*e15:drank*) a bottle of juice for lunch today.

3 Structured Learning for TempRel Extraction

Given the transitivity property that valid temporal graphs possess, the TempRel extraction problem is a structured prediction problem. In this section, I briefly review the existing methods and then explain my approach to both learning and inference.

3.1 Existing Methods

Early attempts by Mani et al. (2006), Chambers et al. (2007), Bethard et al. (2007), Verhagen and Pustejovsky (2008) studied *local* methods. That is, look at each pair of nodes and make decisions irrespective of edges between other pairs, during which both the learning and inference are *local*. Some of the state-of-the-art methods, including ClearTK (Bethard, 2013), UTTime (Laokulrat et al., 2013), and NavyTime (Chambers, 2013b), use better designed rules or more advanced features such as syntactic tree paths, but remain to be *local*. An apparent disadvantage is that the decisions made by local models may be globally inconsistent (i.e., the symmetry and/or transitivity constraints are not satisfied for the entire temporal graph).

Integer linear programming (ILP) methods Roth and Yih (2004) were used in this domain to enforce global consistency by several authors including Bramsen et al. (2006), Chambers and Jurafsky (2008), Do et al. (2012), which formulated temporal graph extraction as an ILP and showed that it improves over local methods for densely connected graphs. Since these methods perform inference ("I") on top of pre-trained local classifiers ("L"), they are often referred to as L+I (Punyakanok et al., 2005). In another state-of-the-art method, CAscading EVent Ordering (CAEVO) Chambers et al. (2014), some hand-crafted rules and machine learned classifiers (called sieves therein) form a pipeline. The global consistency is enforced by inferring all possible relations before passing the graph to the next sieve. This best-first architecture is conceptually similar to L+I but the inference is greedy, similar to Mani et al. (2007), Verhagen and Pustejovsky (2008). In other words, these methods have all successfully incorporated the transitivity structure in the inference phase.

While it is clear that the transitivity structure of temporal graphs requires global considerations, all the aforementioned methods depend on classifiers that are *learned* locally without taking structural information into account. Although L+I methods impose global constraints in the *inference* phase, we argue that global considerations are necessary in the *learning* phase as well, which falls into the category of structured learning.

3.2 Inference via Integer Linear Programming

Since inference is usually an important step in structured learning schemes, we first introduce the standard inference algorithm via ILP. In a temporal graph with n edges, let $\phi_i \in \mathcal{X} \subseteq \mathbb{R}^d$ be the extracted d-dimensional feature and $y_i \in \mathcal{Y}$ be the temporal relation for the i-th edge, $i = 1, 2, \ldots, n$, where $\mathcal{Y} = \{r_j\}_{j=1}^{6}$ is the label set for the six temporal relations we use, i.e., *before*, *after*, *includes*, *is included*, *equal*, and *vague*. Moreover, let $\mathbf{x} = \{\phi_1, \ldots, \phi_n\} \in \mathcal{X}^n$ and $\mathbf{y} = \{y_1, \ldots, y_n\} \in \mathcal{Y}^n$ be more compact representations of all the features and labels

in this temporal graph. Given the weight vector $\mathbf{w}_r$ of a linear classifier trained for relation $r \in \mathcal{Y}$ (i.e., using the one-vs-all scheme), the global inference step is to solve the following constrained optimization problem:

$$\hat{\mathbf{y}} = \arg \max_{\mathbf{y} \in \mathcal{C}(\mathcal{Y}^n)} f(\mathbf{x}, \mathbf{y}), \tag{1}$$

where $\mathcal{C}(\mathcal{Y}^n) \subseteq \mathcal{Y}^n$ constrains the temporal graph to be symmetrically and transitively consistent, and $f(\mathbf{x}, \mathbf{y})$ is the soft-max scoring function:

$$f(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} f_{y_i}(\phi_i) = \sum_{i=1}^{n} \frac{e^{\mathbf{w}_{y_i}^T \phi_i}}{\sum_{r \in \mathcal{Y}} e^{\mathbf{w}_r^T \phi_i}}.$$

Specifically, $f_{y_i}(\phi_i)$ is the probability of the i-th edge having relation y_i. $f(\mathbf{x}, \mathbf{y})$ is simply the sum of these probabilities over all the edges in a temporal graph, which we think of as the confidence of assigning $\mathbf{y} = \{y_1, ..., y_n\}$ to this document and therefore, it needs to be maximized in Eq. (1).

Note that when $\mathcal{C}(\mathcal{Y}^n) = \mathcal{Y}^n$, Eq. (1) can be solved for each $\hat{y}_i$ independently, which is what the so-called local methods do. When $\mathcal{C}(\mathcal{Y}^n) \neq \mathcal{Y}^n$, Eq. (1) cannot be decoupled for each $\hat{y}_i$ and is usually formulated as an ILP problem (Roth and Yih, 2004, Chambers and Jurafsky, 2008, Do et al., 2012). Specifically, let $\mathcal{I}_r(ij) \in \{0, 1\}$ be the indicator function of relation r for node i and node j and $f_r(ij) \in [0, 1]$ be the corresponding soft-max score. Then the ILP objective for global inference is formulated as follows.

$$\begin{gathered} \hat{\mathcal{I}} = \operatorname*{argmax}_{\mathcal{I}} \textstyle\sum_{i<j} \sum_{r \in \mathcal{Y}} f_r(ij) \mathcal{I}_r(ij) \\ \text{s.t.} \quad \underset{\text{(uniqueness)}}{\Sigma_r \mathcal{I}_r(ij) = 1}, \\ \underset{\text{(transitivity)}}{\mathcal{I}_{r_1}(ij) + \mathcal{I}_{r_2}(jk) - \Sigma_{m=1}^{N} \mathcal{I}_{r_3^m}(ik) \leq 1}, \end{gathered} \tag{2}$$

for all distinct nodes i, j, and k, where N is the number of possible relations for r_3 when r_1 and r_2 are true. Our formulation in Eq. (2) is different from previous work (Chambers and Jurafsky, 2008, Do et al., 2012). Previously, transitivity constraints were formulated as $\mathcal{I}_{r_1}(ij) + \mathcal{I}_{r_2}(jk) - \mathcal{I}_{r_3}(ik) \leq 1$, which is a special case when $N = 1$ and can be understood as "r_1 and r_2 determine a single r_3". However, it was overlooked that, although some r_1 and r_2 cannot uniquely determine r_3, they can still constrain the set of labels that r_3 can take. For example, as shown in Fig. 3, when r_1=*before* and r_2=*is_included*, r_3 is not determined but we know that $r_3 \in \{$*before*, *is_included*$\}$. This information can be easily exploited by allowing $N > 1$. Despite this difference, this optimization problem (2) can still be solved using off-the-shelf ILP packages such as GUROBI (Gurobi Optimization, Inc., 2012).

3.3 Learning via Structured Perceptron

With the inference solver defined above, we propose to use the structured perceptron (Collins, 2002) as a representative structured learning algorithm for TempRel extraction. Specifically,

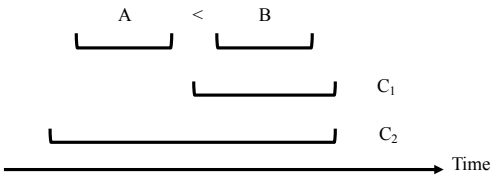


Figure 3: When *A is before B* and *B is_included in C*, *A* can either be *before* C_1 or *is_included in* C_2. We propose to incorporate this via the transitivity constraints for Eq. (2).

let $\mathcal{L} = \{\mathbf{x}_k, \mathbf{y}_k\}_{k=1}^K$ be the labeled training set of K instances (usually documents). The structured perceptron training algorithm for this problem is shown in Algorithm 1. The Illinois-SL package (Chang et al., 2010) was used in our experiments for its structured perceptron component. In terms of the features used in this work, we adopt the same set of features designed for E-E TLINKs in Sec. 3.1 of Do et al. (2012).

In Algorithm 1, Line 6 is the inference step as in Eq. (1) or (2). If there was only one pair of events in each instance (thus no structure to take advantage of), Algorithm 1 would reduce to the conventional perceptron algorithm and Line 6 simply chooses the top scoring label. With a structured instance instead, Line 6 becomes slower to solve, but it can provide valuable information so that the perceptron learner is able to look further at other labels rather than an isolated pair. For example in Example 3 and Fig. 1, the fact that (ripping,ordered)=*before* is established through two other relations: 1) *ripping* is an adverbial participle and thus *included* in *cascaded* and 2) *cascaded* is *before ordered*. If (ripping,ordered)=*before* is presented to a local learning algorithm without knowing its predictions on (ripping,cascaded) and (cascaded,ordered), then the model either cannot support it or overfits it. In structured perceptron, however, if the classifier was correct in deciding (ripping,cascaded) and (cascaded,ordered), then (ripping,ordered) would be correct automatically due to structural constraints, and would not contribute to updating the classifier.

Algorithm 1: Structured perceptron algorithm for temporal relations

Input: Training set $\mathcal{L} = \{\mathbf{x}_k, \mathbf{y}_k\}_{k=1}^K$, learning rate λ

1 Perform graph closure on each $\mathbf{y}_k$
2 Initialize $\mathbf{w}_r = \mathbf{0}$, $\forall r \in \mathcal{Y}$
3 **while** *convergence criteria not satisfied* **do**
4 Shuffle the examples in $\mathcal{L}$
5 **foreach** $(\mathbf{x}, \mathbf{y}) \in \mathcal{L}$ **do**
6 $\hat{\mathbf{y}} = \arg\max_{\mathbf{y} \in \mathcal{C}} f(\mathbf{x}, \mathbf{y})$
7 **if** $\hat{\mathbf{y}} \neq \mathbf{y}$ **then**
8 $\mathbf{w}_r = \mathbf{w}_r + \lambda(\sum_{i:\mathbf{y}_i=r} \phi_i - \sum_{i:\hat{\mathbf{y}}_i=r} \phi_i)$, $\forall r \in \mathcal{Y}$
9 **return** $\{\mathbf{w}_r\}_{r \in \mathcal{Y}}$

3.4 Experiments

The TempEval3 (TE3) workshop UzZaman et al. (2013) provided the TimeBank (TB) Pustejovsky et al. (2003), AQUAINT (AQ) Graff (2002), Silver (TE3-SV), and Platinum (TE3-PT) datasets, where TB and AQ are usually for training, and TE3-PT is usually for testing. The TE3-SV dataset is a much larger, machine-annotated and automatically-merged dataset based on multiple systems, with the intention to see if these "silver" standard data can help when included in training (although almost all participating systems saw performance drop with TE3-SV included in training).

Two popular augmentations on TB are the Verb-Clause temporal relation dataset (VC) and TimebankDense dataset (TD). The VC dataset has specially annotated event pairs that follow the so-called Verb-Clause structure (Bethard et al., 2007), which is usually beneficial to be included in training (UzZaman et al., 2013). The TD dataset contains 36 documents from TB which were re-annotated using the dense event ordering framework proposed in Cassidy et al. (2014). The experiments included in this paper will involve the TE3 datasets as well as these augmentations. Therefore, some statistics on them are shown in Table 1.

Table 1: Facts about the datasets used in this section. Note that the column of TLINKs only counts the non-vague TLINKs. The TLINK annotations in TE3-SV is not used in this paper and its number is thus not shown.

Dataset	Doc	Tokens	Event	TLINK	Note
TB+AQ	256	100K	12K	12K	Training
VC	132	-	1.6K	0.9K	Training
TD	36	-	1.6K	5.7K	Training
TD-Train	22	-	1K	3.8K	Training
TD-Dev	5	-	0.2K	0.6K	Dev
TD-Test	9	-	0.4K	1.3K	Eval
TE3-PT	20	6K	0.7K	0.9K	Eval
TE3-SV	2.5K	666K	81K	-	Unlabeled

In addition to the state-of-the-art systems, another two baseline methods were also implemented for a better understanding of the proposed. The first is the regularized averaged perceptron (AP) Freund and Schapire (1998) implemented in the LBJava package (Rizzolo and Roth, 2010) and is a local method. On top of the first baseline, we performed global inference in Eq.(2), referred to as the L+I baseline (AP+ILP). Both of them used the same feature set (i.e., as designed in Do et al. (2012)) as in the proposed structured perceptron (SP) for fair comparisons. To clarify, SP is a training algorithm and its immediate outputs are the weight vectors $\{\mathbf{w}_r\}_{r\in\mathcal{Y}}$ for local classifiers. An ILP inference was performed on top of it to yield the final output, and we refer to it as "S+I" (i.e., structured learning+inference) methods.

To show the benefit of using structured learning, we tested the scenario where the gold pairs of events that have a non-vague TLINK were known priori. This setup was a standard task presented in TE3 (Task C – Relation Only). UTTime Laokulrat et al. (2013) was the

Table 2: Temporal awareness scores on TE3-PT given gold event pairs. Systems that are significantly better (per McNemar's test with $p < 0.0005$) than the previous rows are underlined. The last column shows the relative improvement in F1 score over AP-1, which identifies the source of improvement: 5.2% from additional training data, 9.3% (14.5%-5.2%) from constraints, and 10.4% from structured learning.

System	Method	P	R	F1	%
UTTime	Local	55.6	57.4	56.5	+5.0
AP-1	Local	56.3	51.5	53.8	0
AP-2	Local	58.0	55.3	56.6	+5.2
AP+ILP	L+I	62.2	61.1	61.6	+14.5
SP+ILP	S+I	**69.1**	**65.5**	**67.2**	**+24.9**

top system in this task in TE3. Since UTTime is not available to us, and its performance was reported in TE3 in terms of both EE and ET TLINKs together, we locally trained an ET classifier based on Do et al. (2012) and included its prediction only for fair comparisons.

UTTime is a local method and was trained on TB+AQ and tested on TE3-PT. We used the same datasets for our local baseline and its performance is shown in Table 2 under the name "AP-1". Note that the reported numbers below are the temporal awareness scores obtained from the official evaluation script provided in TE3. We can see that UTTime is about 3% better than AP-1 in the absolute value of F_1, which is expected since UTTime included more advanced features derived from syntactic parse trees. By adding the VC and TD datasets into the training set, we retrained our local baseline and achieved comparable performance to UTTime ("AP-2" in Table 2). On top of AP-2, a global inference step enforcing symmetry and transitivity constraints ("AP+ILP") can further improve the F_1 score by 9.3%, which is consistent with previous observations (Chambers and Jurafsky, 2008, Do et al., 2012). SP+ILP further improved the performance in precision, recall, and F_1 significantly (per the McNemar's test Everitt (1992), Dietterich (1998) with $p <$0.0005), reaching an F_1 score of 67.2%. This meets our expectation that structured learning can be better when the local problem is difficult (Punyakanok et al., 2005).

4 Injection of Human Prior Knowledge

As in many NLP tasks, one of the challenges in temporal relation extraction is that it requires high-level prior knowledge; in this case, we care about the temporal order that events *usually* follow. In Example 6, we have deleted events from the original sentence. Rich temporal information is encoded in the events' names, and this often plays an indispensable role in making our decisions. As a result, it is very difficult even for humans to figure out the TempRels between those events. In the first paragraph of Example 6, it is difficult to understand what really happened without the actual event verbs; let alone the TempRels between them. In the second paragraph, things are even more interesting: if we had *e18:dislike* and *e19:stop*, then we would know easily that "I dislike" occurs *after* "they stop the column". However, if we had *e18:ask* and *e19:help*, then the relation between *e18* and *e19* is now reversed and

e18 is *before e19*. We are in urgent need of the event names to determine the TempRels. In Example 7, where we show the complete sentences, the task has become much easier for humans due to our prior knowledge. Motivated by these examples (which are in fact very common), we believe in the importance of such a prior knowledge in determining TempRels.

Example 6: Difficulty in understanding TempRels when event content is missing. Note that *e16* and *e17* have the same tense, and *e18* and *e19* have the same tense.
More than 10 people have (*e16:*), police said. A car (*e17:*) on Friday in the middle of a group of men playing volleyball.
The first thing I (*e18:*) is that they (*e19:*) writing this column.

However, most existing systems only make use of rather local features of these events, which cannot represent the prior knowledge humans have about these events and their "typical" order. As a result, existing systems almost always attempt to solve the situations shown in Example 6, even when they are actually presented with input as in Example 7. In this section, we propose such a resource in the form of a probabilistic knowledge base, constructed from a large New York Times (NYT) corpus. We hereafter name our resource *TEMporal relation PRObabilistic knowledge Base* (TemProb), which can potentially benefit many time-aware tasks. A few example entries of TemProb are shown in Table 3.

Example Pairs		Before (%)	After (%)
accept	determine	42	26
ask	help	86	9
attend	schedule	1	82
accept	propose	10	77
die	explode	14	83
...			

Table 3: **TemProb is a unique source of information of the temporal order that events *usually* follow.** The probabilities below do not add up to 100% because less frequent relations are omitted. The word sense numbers are not shown here for convenience.

Example 7: The original sentences in Example 6.
More than 10 people have (*e2:died*), police said. A car (*e1:exploded*) on Friday in the middle of a group of men playing volleyball.
The first thing I (*e18:ask*) is that they (*e19:help*) writing this column.

4.1 TemProb: A Probabilistic Resource for TempRels

To construct the desired resource, we need to extract events and TempRels from a large, unannotated corpus. First, we consider semantic-frame based events, which can be directly detected via off-the-shelf semantic role labeling (SRL) tools. We also only look at *verb* semantic frames as a starting point. Then we apply our existing TempRel extractor on top

of the events we extracted. We perform this procedure on more than 1 million NYT articles, spanning 20 years (1987-2007)[3]. In total, we discovered 51K unique verb semantic frames and 80M relations among them in the NYT corpus (15K of the verb frames had more than 20 relations extracted and 9K had more than 100 relations).

We denote the set of all verb semantic frames by V. Let $D_i, i = 1, \ldots, N$ be the i-th document in our corpus, where N is the total number of documents. Let $G_i = (V_i, E_i)$ be the temporal graph inferred from D_i using the approach described above, where $V_i \subseteq V$ is the set of verbs/events extracted in D_i and $E_i = \{(v_m, v_n, r_{mn})\}_{m<n} \subseteq V_i \times V_i \times R$ is the edge set of D_i, which is composed of TempRel triplets; specifically, a TempRel triplet $(v_m, v_n, r_{mn}) \in E_i$ represents that in document D_i, the TempRel between v_m and v_n is r_{mn}. Due to the symmetry in TempRels, we only keep the triplets with $m < n$ in E_i. Assuming that the verbs in V_i are ordered by their appearance order in text, then $m < n$ means that in the i-th document, v_m appears earlier in text than v_n does.

Given the usual confusion between that one event is *temporally before* another and that one event is *physically appearing* before another in text, we will refer to temporally before as **T-Before** and physically before as **P-Before**. Using this language, for example, E_i only keeps the triplets that v_m is P-Before v_n in D_i.

4.1.1 Extreme cases

We first show extreme cases that some events are *almost always* labeled as T-Before or T-After in the corpus. Specifically, for each pair of verbs $v_i, v_j \in V$, we define the following ratios:

$$\eta_b = \frac{C(v_i, v_j, before)}{C(v_i, v_j, before) + C(v_i, v_j, after)}, \eta_a = 1 - \eta_b, \tag{3}$$

where $C(v_i, v_j, r)$ is the count of v_i P-Before v_j with TempRel $r \in R$:

$$C(v_i, v_j, r) = \sum_{i=1}^{N} \sum_{(v_m, v_n, r_{mn}) \in E_i} \mathcal{I}_{\{v_m = v_i \& v_n = v_j \& r_{mn} = r\}}, \tag{4}$$

where $\mathcal{I}_{\{\cdot\}}$ is the indicator function. In Table 4, we show some event pairs with either $\eta_b > 0.9$ (upper part) or $\eta_a > 0.9$ (lower part).

4.1.2 Distribution of Following Events

For each verb v, we define the marginal count of v being P-Before to arbitrary verbs with TempRel $r \in R$ as $C(v, r) = \sum_{v_i \in V} C(v, v_i, r)$. Then for every other verb v', we define

$$P(v \text{ T-Before } v' | v \text{ T-Before}) \triangleq \frac{C(v, v', before)}{C(v, before)}, \tag{5}$$

which is the probability of v T-Before v', conditioned on v T-Before anything. Similarly, we define

$$P(v \text{ T-After } v' | v \text{ T-After}) \triangleq \frac{C(v, v', after)}{C(v, after)}. \tag{6}$$

[3]https://catalog.ldc.upenn.edu/LDC2008T19

Example Pairs		#T-Before	#T-After
chop.01	taste.01	133	8
concern.01	protect.01	110	10
conspire.01	kill.01	113	6
debate.01	vote.01	48	5
dedicate.01	promote.02	67	7
fight.01	overthrow.01	98	8
achieve.01	desire.01	7	104
admire.01	respect.01	7	121
clean.02	contaminate.01	3	82
defend.01	accuse.01	13	160
die.01	crash.01	8	223
overthrow.01	elect.01	3	100

Table 4: **Several extreme cases from TemProb**, where some event is almost always labeled to be T-Before or T-After throughout the NYT corpus. By "extreme", we mean that either the probability of T-Before or T-After is larger than 90%. The upper part of the table shows the pairs that are both P-Before and T-Before, while the lower part shows the pairs that are P-Before but T-After. In TemProb, there are about 7K event pairs being extreme cases.

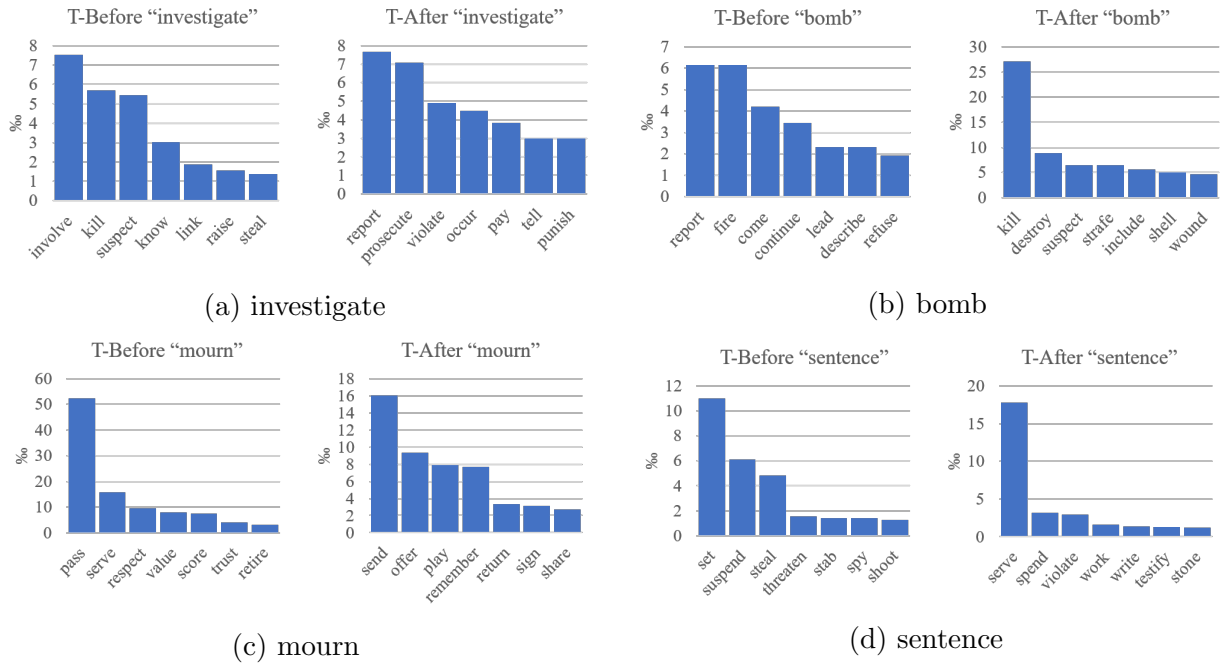
For a specific verb, e.g., v=*investigate*, each verb $v' \in V$ is sorted by the two conditional probabilities above. Then the most probable verbs that temporally precede or follow v are shown in Fig. 4, where the y-axes are the corresponding conditional probabilities. We can see reasonable event sequences like {*involve*, *kill*, *suspect*, *steal*}→*investigate*→{*report*, *prosecute*, *pay*, *punish*}, which indicates the possibility of using TemProb for event sequence predictions or story cloze tasks. There are also suspicious pairs like *know* in the T-Before list of *investigate* (Fig. 4a), *report* in the T-Before list of *bomb* (Fig. 4b), and *play* in the T-After list of *mourn* (Fig. 4c). Since the arguments of these verb frames are not considered here, whether these few seemingly counter-intuitive pairs come from system error or from a special context needs further investigation.

4.2 Experiments

The prior distributions from TemProb can be used to regularize the conventional ILP formulation. Then the ILP objective for global inference is formulated as follows.

$$\begin{gathered}\hat{\mathcal{I}} = \operatorname*{argmax}_{\mathcal{I}} \textstyle\sum_{i<j} \sum_{r\in\mathcal{Y}} (f_r(ij) + \underline{\lambda h_r(ij)}) \mathcal{I}_r(ij) \qquad (7)\\ \text{s.t.} \quad \underset{\text{(uniqueness)}}{\Sigma_r \mathcal{I}_r(ij) = 1},\\ \underset{\text{(transitivity)}}{\mathcal{I}_{r_1}(ij) + \mathcal{I}_{r_2}(jk) - \Sigma_{m=1}^{M} \mathcal{I}_{r_3^m}(ik) \leq 1},\end{gathered}$$

for all distinct events i, j, and k, where λ adjusts the regularization term and was heuristically set to 0.5 in this work. Comparing to Eq. (2), the difference is the underlined regularization



(a) investigate (b) bomb

(c) mourn (d) sentence

Figure 4: **Top events that most frequently precede or follow “investigate”, “bomb”, “mourn”, or “sentence” in time**, sorted by their conditional probabilities in ‰. Word senses have been disambiguated and the “bomb” and “sentence” here are their verb meanings. There are some possible errors (e.g., *report* is T-Before *bomb*) and some unclear pairs (e.g., *know* is T-Before *investigate* and *play* is T-After *mourn*), but overall the event sequences discovered here are reasonable. More examples can be found in the appendix.

term $h_r(ij)$, which is the prior distributions of all labels obtained from TemProb:

$$h_r(v_i, v_j) = \frac{C(v_i, v_j, r)}{\sum_{r' \in R} C(v_i, v_j, r')}, \; r \in \mathcal{Y}. \tag{8}$$

No.	System	P	R	F_1	F_{aware}
1	Baseline	48.1	44.4	46.2	42.5
2	+Feature: $\{h_r\}_{r \in \mathcal{Y}}$	50.6	52.0	51.3	49.1
3	+Regularization	**51.3**	**53.0**	**52.1**	**49.6**

Table 5: **Regularizing global methods by the prior distribution derived from TemProb.** The “+” means adding a component on top of its preceding line. F_{aware} is the temporal awareness F-score, another evaluation metric used in TempEval3. The baseline system is to use (unregularized) ILP. System 3 is the proposed. Per the McNemar’s test, System 3 is significantly better than System 1 with $p<0.0005$.

We present our results on the test split of TimeBank-Dense in Table 5, which is an ablation study showing improvements in two F-metrics. In Table 5, the baseline used the same feature set we used earlier and applied global ILP inference with transitivity constraints (i.e., L+I). System 2, “+Feature: $\{h_r\}_{r \in \mathcal{Y}}$”, is to add prior distributions as features when

training the local classifiers. Technically, this means that the scores in Eq. (7) used by baseline were changed. On top of this, System 3 sets $\lambda = 0.5$ in Eq. (7) to add regularizations to the conventional ILP formulation. The sum of these regularization terms represents a confidence score of how coherent the predicted temporal graph is to our TemProb, which we also want to maximize. Even though a considerable amount of information from TemProb had already been encoded as features (as shown by the large improvements by System 2), these regularizations were still able to further improve the precision, recall and awareness scores. To sum up, the total improvement over the baseline system brought by TemProb is 5.9% in F_1 and 7.1% in awareness F_1, both with a notable margin.

(*e20*, *e21*), (*e22*, *e23*), and (*e24*, *e25*): TempRels that are difficult even for humans. Note that only relevant events are highlighted here.
Example 8: Serbian police tried to eliminate the pro-independence Kosovo Liberation Army and (*e20:restore*) order. At least 51 people were (*e21:killed*) in clashes between Serb police and ethnic Albanians in the troubled region.
Example 9: Service industries (*e22:showed*) solid job gains, as did manufacturers, two areas expected to be hardest (*e23:hit*) when the effects of the Asian crisis hit the American economy.
Example 10: We will act again if we have evidence he is (*e24:rebuilding*) his weapons of mass destruction capabilities, senior officials say. In a bit of television diplomacy, Iraq's deputy foreign minister (*e25:responded*) from Baghdad in less than one hour, saying that ...

5 Data Annotation for TempRels

5.1 Existing Datasets

Initiated by TimeBank (TB) Pustejovsky et al. (2003), a number of TempRel datasets have been collected, including but not limited to those in Table 1. These datasets were annotated by experts, but most still suffered from low inter-annotator agreements (IAA). For instance, the IAAs of TimeBank-Dense, RED O'Gorman et al. (2016) and THYME-TimeML Styler IV et al. (2014) were only below or near 60% (given that events are already annotated). Since a low IAA usually indicates that the task is difficult even for humans (see Examples 8-10), the community has been looking into ways to simplify the task, by reducing the label set, and by breaking up the overall, complex task into subtasks (e.g., getting agreement on which event pairs should have a relation, and then what that relation should be) Mostafazadeh et al. (2016), O'Gorman et al. (2016). In contrast to other existing datasets, Bethard et al. (2007) achieved an agreement as high as 90%, but the scope of its annotation was narrowed down to a very special verb-clause structure.

5.2 Multi-Axis Modeling

In contrast to the annotation difficulties, humans can easily grasp the meaning of news articles, implying a potential gap between the difficulty of the annotation task and the one

of understanding the actual meaning of the text. In Examples 8-10, the writers did not intend to explain the TempRels between those pairs, and the original annotators of TimeBank did not label relations between those pairs either, which indicates that both writers and readers did not think the TempRels between these pairs were crucial. Instead, what is crucial in these examples is that "Serbian police *tried* to restore order but *killed* 51 people", that "two areas were *expected* to be hit but *showed* gains", and that "*if* he rebuilds weapons *then* we will act." To "*restore* order", to be "hardest *hit*", and "if he was *rebuilding*" were only the intention of police, the opinion of economists, and the condition to *act*, respectively, and whether or not they actually happen is not the focus of those writers.

Example 11: Dense Annotation Scheme.
Serbian police (*e26:tried*) to (*e27:eliminate*) the pro-independence Kosovo Liberation Army and (*e20:restore*) order. At least 51 people were (*e21:killed*) in clashes between Serb police and ethnic Albanians in the troubled region.
Given 4 Non-Generic events above, the dense scheme presents 6 pairs to annotators one by one: (*e26*, *e27*), (*e26*, *e20*), (*e26*, *e21*), (*e27*, *e20*), (*e27*, *e21*), and (*e20*, *e21*). Apparently, not all pairs are well-defined, e.g., (*e27*, *e21*) and (*e20*, *e21*), but annotators are forced to label all of them.

Given a set of events, one important question in designing the TempRel annotation task is: which pairs of events should have a relation? The answer to it depends on the modeling of the overall temporal structure of events. In TimeBank, the annotators were allowed to label TempRels between any pairs of events. This setup models the overall structure of events using *a general graph*, which made annotators inadvertently overlook some pairs, resulting in low IAAs and many false negatives. To address this issue, Cassidy et al. (2014) proposed a dense annotation scheme, TimeBank-Dense, which annotates all event pairs within a sliding, two-sentence window (see Example 11). It conceptually models the overall structure by a single time axis. This implies that we need a modeling which is more restrictive than a general graph so that annotators can focus on relation annotation (rather than looking for pairs first), but also more flexible than a single axis so that ill-defined relations are not forcibly annotated. Specifically, we need axes for intentions, opinions, hypotheses, etc. in addition to the main axis of an article. We thus argue for *multi-axis modeling*, as defined in Table 6. Following the proposed modeling, Examples 8-10 can be represented as in Fig. 5. This modeling aims at capturing what the author has explicitly expressed and it only asks annotators to look at comparable pairs, rather than forcing them to make decisions on often vaguely defined pairs.

To summarize, our annotation scheme is two-step. First, we mark every event candidate as being temporally *Anchorable* or not (based on the time axis we are working on). Second, we adopt the dense annotation scheme to label TempRels only between *Anchorable* events.

5.3 Ambiguity of End-Points

During our pilot annotation, the annotation quality dropped significantly when the annotators needed to reason about relations involving end-points of events. Table 7 shows four met-

Event Type	Category
Intention, Opinion	On an orthogonal axis
Hypothesis, Generic	On a parallel axis
Negation	Not on any axis
Static, Recurrent	Other

Table 6: The interpretation of various event types that are not on the main axis in the proposed multi-axis modeling.

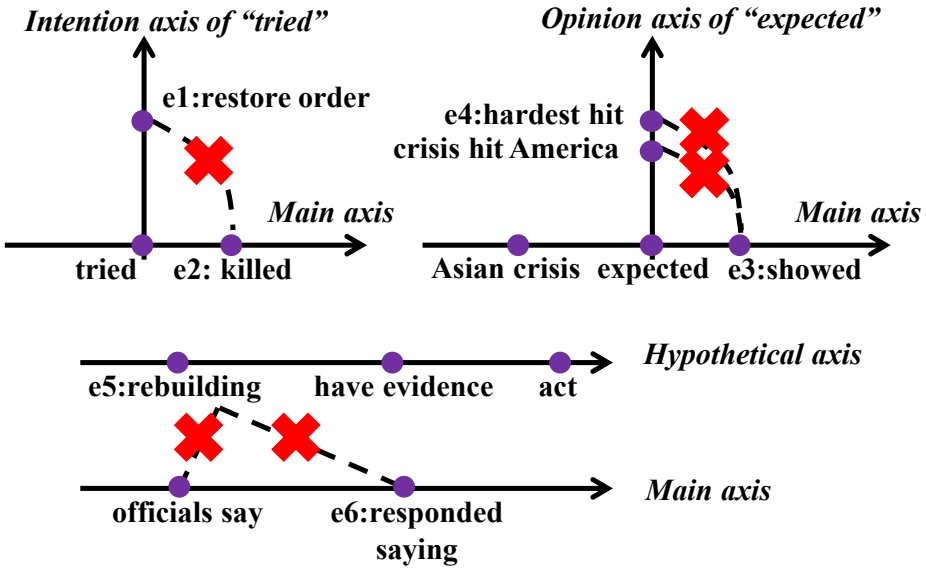


Figure 5: A multi-axis view of Examples 8-10. Only events on the same axis are compared.

rics of task difficulty when only t^1_{start} vs. t^2_{start} or t^1_{end} vs. t^2_{end} are annotated. Non-anchorable events were removed for both jobs. We can see that when annotating the relations between end-points, only one out of ten crowdsourcers (11%) could successfully pass our qualifying test; and even if they had passed it, half of them (56%) would have been kicked out in the middle of the task. The third line is the overall accuracy on gold set from all crowdsourcers (excluding those who did not pass the qualifying test), which drops from 67% to 37% when annotating end-end relations. The last line is the average response time per annotation and we can see that it takes much longer to label an end-end TempRel (52s) than a start-start TempRel (33s). This important discovery indicates that the TempRels between end-points is probably governed by a different linguistic phenomenon, so we ignore the comparison of end-points in this work, although event duration is indeed, another important task. When the end-points are ignored, our label set now becomes *before, after, equal,* and *vague*.

Metric	t^1_{start} vs. t^2_{start}	t^1_{end} vs. t^2_{end}
Qualification pass rate	50%	11%
Survival rate	74%	56%
Accuracy on gold	67%	37%
Avg. response time	33s	52s

Table 7: Annotations involving the end-points of events are found to be much harder than only comparing the start-points.

5.4 Experiments

We name our new dataset *MATRES* for Multi-Axis Temporal RElations for Start-points. We apply our existing TempRel extractor on MATRES, assuming that all the events and axes are given. We adopted the same train/dev/test split of TB-Dense, where there are 22 documents in train, 5 in dev, and 9 in test. Parameters were tuned on the train-set to maximize its F_1 on the dev-set, after which the classifier was retrained on the union of train and dev. A detailed analysis of the baseline system is provided in Table 8. The performance on *equal* and *vague* is lower than on *before* and *after*, probably due to shortage in these labels in the training data and the inherent difficulty in event coreference and temporal vagueness. We can see, though, that the overall performance on MATRES is much better than those in the literature for TempRel extraction, which used to be in the low 50's Chambers et al. (2014), Ning et al. (2017). The same system was also retrained and tested on the original annotations of TimeBank-Dense (Line "Original"), which confirms the significant improvement if the proposed annotation scheme is used.

6 Proposed Remaining Research

As described in Secs. 3-5, my work on TempRel extraction has been from three perspectives: structured machine learning, injection of human knowledge, and dataset collection. My

	Training			Testing		
	P	R	F_1	P	R	F_1
Before	.74	.91	.82	.71	.80	.75
After	.73	.77	.75	.55	.64	.59
Equal	1	.05	.09	-	-	-
Vague	.75	.28	.41	.29	.13	.18
Overall	**.73**	**.81**	**.77**	**.66**	**.72**	**.69**
Original	.44	.67	.53	.40	.60	.48

Table 8: Performance of the proposed baseline system on MATRES. Line "Original" is the same system retrained on the original TimeBank-Dense and tested on the same subset of event pairs. Due to the limited number of *equal* examples, the system did not make any *equal* predictions on the testset.

remaining research will extend my current work also from these three perspectives. In addition, I also plan to apply my TempRel extraction technique on timeline generation.

6.1 Structured Learning from Partial Annotations

As we introduced in Sec. 3, the TempRel extraction problem is inherently a structured learning task given the transitivity property of temporal graphs. When annotating TempRel in practice, it is often the case that the temporal graphs are not completed annotated (either by choice or by mistake). For example, it is well-known that the TimeBank dataset has many missing edges (see Fig. 6; after all, this is why TimeBank-Dense was proposed). Then a natural question is whether we can still use these partially annotated temporal graphs (denoted by $\mathcal{P}$ hereafter, in contrast to fully annotated graphs $\mathcal{F}$) as supervision to improve existing structured learning methods. A preliminary study has been published in Ning et al. (2018d), and the algorithm to jointly learn from both $\mathcal{F}$ and $\mathcal{P}$ is shown in Algorithm 2, which is an extension of the Constraint-Driven Learning (CoDL) Chang et al. (2007) algorithm. CoDL is a semi-supervised learning algorithm, which first learns from $\mathcal{F}$, and then labels an unannotated dataset using the learned system while enforcing structural constraints; CoDL can thus also be considered as a structured learning version of self-learning or bootstrapping. Algorithm 2 extends CoDL such that the annotations in $\mathcal{P}$, albeit partial, are enforced along with those structural constraints. We think enforcing both partial annotations and structural constraints will maximize the benefit we can gain from $\mathcal{P}$ because those partial annotations in $\mathcal{P}$ will also put restrictions on other parts of the graph.

Partial annotations in general structured learning tasks are interesting from two perspectives. First, *incidental supervision* Roth (2017). In practice, supervision signals may not always be perfect: they may be noisy, only partial, based on different annotation schemes, or even on different (but relevant) tasks; incidental supervision is a general paradigm that aims at making use of the abundant, naturally occurring data, as supervision signals. Second, *data collection for structured learning tasks*. The fact that $\mathcal{P}$ can provide useful supervision signals poses a further question: What is the optimal data collection scheme for structure learning tasks, fully annotated, partially annotated, or a mixture of both?

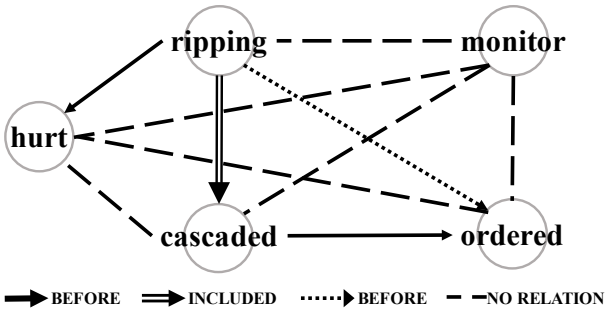


Figure 6: The human-annotation for Example 3 provided in TempEval3 UzZaman et al. (2013), where many TempRels are missing due to the annotation difficulty. Solid lines: original human annotations. Dotted lines: TempRels inferred from solid lines. Dashed lines: missing relations.

Algorithm 2: Joint learning from $\mathcal{F}$ and $\mathcal{P}$ by bootstrapping

Input: $\mathcal{F}$, $\mathcal{P}$, Learn, Inference

$S_{\mathcal{F}} = \text{Learn}(\mathcal{F})$

Initialize $S_{\mathcal{F}+\mathcal{P}} = S_{\mathcal{F}}$

while *convergence criteria not satisfied* **do**

  $\tilde{\mathcal{P}} = \emptyset$

  foreach $p \in \mathcal{P}$ **do**

    $\hat{\mathbf{y}} = \text{Inference}(p; S_{\mathcal{F}+\mathcal{P}})$

    $\tilde{\mathcal{P}} = \tilde{\mathcal{P}} \cup \{(\mathbf{x}, \hat{\mathbf{y}})\}$

  $S_{\mathcal{F}+\mathcal{P}} = \text{Learn}(\mathcal{F} + \tilde{\mathcal{P}})$

return $S_{\mathcal{F}+\mathcal{P}}$

As a preliminary work, Ning et al. (2018e) studies partial annotations using TempRel extraction as a showcase, but the idea is quite general and the result may have a broad impact to our understanding of structured learning. Recently, we have made interesting progress and we discover that collecting data partially (or in other words, early stopping) can make better use of a fixed annotation budget, both from the standpoint of information theory and also supported by experimental results. I plan to continue my study in structured learning, develop more effective algorithms to make use of partial data and gain more understanding of the general incidental supervision scheme.

6.2 Common Sense of Temporal Ordering

In Sec. 4, we argued that humans have a prior knowledge of the typical ordering of events (e.g., "explosion" should be before "casualties"). When generating this prior knowledge, certain level of abstractions are actually needed in order to generalize. For example, given two events, "*Jack* is arrested because of robbery" and "*John* is arrested because of robbery", one question to ask is "do we take them samely or differently?". One may think that they

are different due to their difference between arguments (i.e., "Jack" vs. "John"), but an obvious downside is that there are too many entities of different surface forms to account for in a limited dataset; more importantly, "rob" leading to "being arrested" is likely to be a common pattern in which the subjects play a minor role. Based on this intuition, we build TemProb with the assumption that two events are considered to be in the same category as long as they share the same predicate. As a result, TemProb encodes this information as a prior distribution $h_r(v_i, v_j)$ in Eq. (8). As shown in Sec. 4, this assumption is working reasonably good. However, there are potential improvements that we can make.

Intuitively, if we group verbs that are semantically close (e.g., "bomb" and "explode", and "attack" and "ambush"), then the prior distribution can more accurately reflect humans' common sense. To investigate this, we have performed k-means clustering in a preliminary study, using off-the-shelf Word2Vec Mikolov et al. (2013) and GloVe Pennington et al. (2014) word embeddings. With clustering, the definition of prior distribution $h_r(v_i, v_j)$ needs to change: assuming that the set of verbs in the same cluster as v is $g(v) \subseteq V$, we modify the prior distribution with clustering as

$$\tilde{h}_r(v_i, v_j) = \frac{\sum_{v \in g(v_i)} \sum_{v' \in g(v_j)} C(v, v', r)}{\sum_{r' \in R} \sum_{v \in g(v_i)} \sum_{v' \in g(v_j)} C(v, v', r')}.$$

Obviously, $\tilde{h}_r(v_i, v_j) = \tilde{h}_r(v_k, v_m)$ as long as v_i and v_k are in the same cluster and v_j and v_m are in the same cluster. We have indeed observed performance improvement if these events are clustered based on those pretrained word embeddings.

Furthermore, clustering based on pretrained embeddings may not be optimal. For example, "cancer", "explode", and "suicide", although they are quite different semantically, may all temporally precede "death". The pretrained word embeddings represent the semantics in the "general" sense of how words are used in a context, but to better represent the common sense of temporal ordering, we can instead train a "temporal embedding" of those words. In addition, the arguments are currently all omitted when collecting TemProb. Again, this works reasonably good in our experiments, but obviously there are cases that the arguments are important. For example, after "Jack was robbed by John", then what would happen to Jack and John are totally different: Jack may be injured and John may be arrested. I plan to investigate ways to incorporating them in training the temporal embeddings.

6.3 Further Improvement on Data Collection

In Ning et al. (2018c), we have proposed to model the temporal structure by multiple time axes and the preliminary study therein is very promising: Crowdsourcers can reliably tell main-axis events from other events and also label the TempRels between them. I plan to further improve it from the following perspectives.

First, many of the axis assignment of events are very straightforward and can be handled by syntactic rules. For example, the majority of the Intentions are of the form of *to do*; most Negations are the verbs following negative words, e.g., *not* or *fail*. Using high

precision rules to assign events to different axes helps reducing the required annotation cost and also improving the quality of the dataset.

Second, Ning et al. (2018c) does not have a crowdsourcing scheme to do full axis assignment (only main-axis was handled). Our subsequent study showed that simply asking crowdsourcers to work on all types of axes is too demanding. I plan to decompose the task into smaller tasks so that crowdsourcers only work on smaller tasks and can understand the guidelines better.

Third, we ignored nominal events (e.g., *explosion* is the nominal form of the verb event *explode*) when collecting MATRES in Ning et al. (2018c) for convenience. However, since temporal graphs are structured, some relations are interrelated through nominal events and incorporating them will help obtain a more complete picture of a story.

Fourth, crowdsourcers will inevitably make mistakes and I have observed that some temporal graph even violate the transitivity constraints. Since we have the overall accuracy of each crowdsourcer and their answers to each question, I have tried to use these statistics to clean the data. Specifically, I use the confidence scores of each answer as the $f_r(ij)$ in Eq. (2) and perform ILP inference on top of the crowdsourced annotations. Later I plan to further use those statistics during training so that the dataset is treated as "soft" annotations.

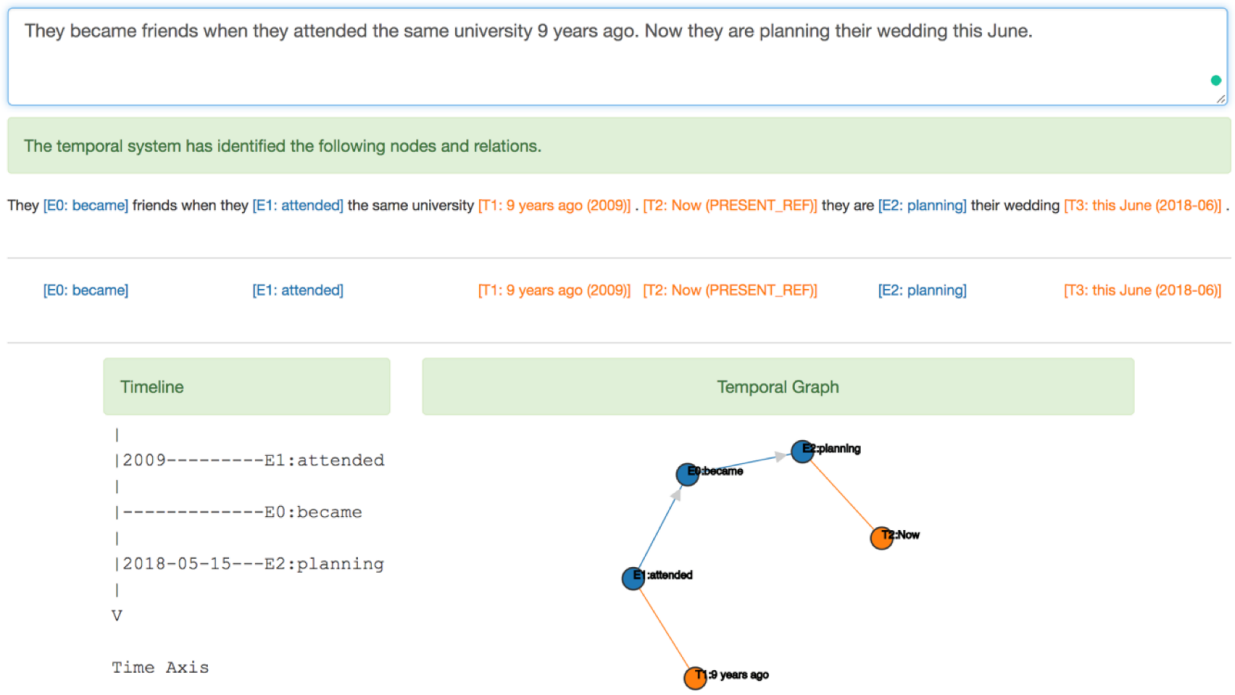


Figure 7: **A snapshot of the interface of CogCompTime.** From top to bottom: Input box, event and Timex highlight, and two visualizations (timeline and graph). The document creation time was chosen to be 2018-05-15.

6.4 Timeline Construction

Ultimately, we would like to use TempRel extraction techniques in real-world applications. Timeline generation is an important application since it allows human readers to grasp the progress of a complex social event more easily. Existing works are mostly from the approach of information retrieval, e.g., relying on knowledge bases and the document creation times (DCT) of news articles. However, to truly understand what is going on, we need also information from the level of natural language, indicating the ordering of events besides DCTs.

In a recent work on mine, CogCompTime (Ning et al., 2018e), we have attempted to do timeline construction for the temporal relations (our online demo is available at `http://groupspaceuiuc.com/temporal/`; see Fig. 7 for a snapshot of it). CogCompTime performs timeline generation with two improvements. First, we use our own Timex extractor and normalizer to parse out all the absolute time points in text. Note that temporal graphs can actually be extended to have both event and Timex nodes. The extended temporal graphs still possess the symmetry and transitivity properties. Depending on the types of the two nodes of an edge, we have Event-Event (EE) TempRels, Event-Timex (ET) TempRels, and Timex-Timex (TT) TempRels. It has been shown in Ning et al. (2018a) that the extended temporal graphs are helpful since ET TempRels and TT TempRels often provide unique information for determining (long-distance) EE TempRels, so these extracted and normalized Timexes, along with the relations between them, are also incorporated into the ILP inference step. Second, in Eq. (2), one detail we neglected earlier is that only event pairs that are within a same sentence or from adjacent sentences are kept, because the dataset only has annotations in this way; we find in CogCompTime that even if we do not have supervision signals for distant event pairs, incorporating them into the ILP formulation will still be helpful.

Lastly, long-distance TempRels are mainly constructed via Timexes and event coreference. We have incorporated information from Timexes above, but event coreference remains as a missing component in the current technique. I plan to make use of existing event coreference techniques (e.g., Peng et al. (2016)) in my work. In addition, I plan to extend both the event coreference and TempRel extraction work to generate timelines from multiple document.

References

James F Allen. Towards a general theory of action and time. *Artificial intelligence*, 23(2): 123–154, 1984.

Steven Bethard. ClearTK-TimeML: A minimalist approach to TempEval 2013. In *Second Joint Conference on Lexical and Computational Semantics (* SEM)*, volume 2, pages 10–14, 2013.

Steven Bethard, James H Martin, and Sara Klingenstein. Timelines from text: Identification

of syntactic temporal relations. In *Semantic Computing, 2007. ICSC 2007. International Conference on*, pages 11–18. IEEE, 2007.

Steven Bethard, Leon Derczynski, Guergana Savova, James Pustejovsky, and Marc Verhagen. SemEval-2015 Task 6: Clinical TempEval. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 806–814, Denver, Colorado, June 2015. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/S15-2136`.

P. Bramsen, P. Deshpande, Y. K. Lee, and R. Barzilay. Inducing temporal graphs. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, pages 189–198, Sydney, Australia, July 2006. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W/W06/W06-1623`.

Taylor Cassidy, Bill McDowell, Nathanel Chambers, and Steven Bethard. An annotation framework for dense event ordering. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 501–506, 2014.

N. Chambers. Event schema induction with a probabilistic entity-driven model. In *EMNLP*, volume 13, pages 1797–1807, 2013a.

Nathanael Chambers. NavyTime: Event and time ordering from raw text. Technical report, DTIC Document, 2013b.

Nathanael Chambers and Daniel Jurafsky. Unsupervised Learning of Narrative Event Chains. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, ACL 2008*, pages 789–797, 2008.

Nathanael Chambers, Shan Wang, and Dan Jurafsky. Classifying temporal relations between events. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 173–176. Association for Computational Linguistics, 2007.

Nathanael Chambers, Taylor Cassidy, Bill McDowell, and Steven Bethard. Dense event ordering with a multi-pass architecture. *Transactions of the Association for Computational Linguistics*, 2:273–284, 2014.

Angel X Chang and Christopher D Manning. SUTIME: A library for recognizing and normalizing time expressions. In *LREC*, volume 2012, pages 3735–3740, 2012.

M. Chang, L. Ratinov, and D. Roth. Guiding semi-supervision with constraint-driven learning. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 280–287, Prague, Czech Republic, 6 2007. Association for Computational Linguistics. URL `http://cogcomp.org/papers/ChangRaRo07.pdf`.

Ming-Wei Chang, Vivek Srikumar, Dan Goldwasser, and Dan Roth. Structured output learning with indirect supervision. In *Proc. of the International Conference on Machine Learning (ICML)*, 2010. URL `http://cogcomp.org/papers/CSGR10.pdf`.

Michael Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proc. of ACL*, 2002.

T. G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1924, 1998.

Quang Do, Wei Lu, and Dan Roth. Joint inference for event timeline construction. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2012. URL `http://cogcomp.org/papers/DoLuRo12.pdf`.

Brian S Everitt. *The analysis of contingency tables*. CRC Press, 1992.

Y. Freund and R. Schapire. Large margin classification using the Perceptron algorithm. In *Proc. of the Annual ACM Workshop on Computational Learning Theory (COLT)*, pages 209–217, 1998.

David Graff. The AQUAINT corpus of english news text. *Linguistic Data Consortium, Philadelphia*, 2002.

Gurobi Optimization, Inc. Gurobi optimizer reference manual, 2012. URL `http://www.gurobi.com`.

Prateek Jindal and Dan Roth. Using domain knowledge and domain-inspired discourse model for coreference resolution for clinical narratives. 20(2):356–362, 3 2013. URL `http://cogcomp.org/papers/JindalRo12.pdf`.

Natsuda Laokulrat, Makoto Miwa, Yoshimasa Tsuruoka, and Takashi Chikayama. UTTime: Temporal relation classification using deep syntactic features. In *Second Joint Conference on Lexical and Computational Semantics (* SEM)*, volume 2, pages 88–92, 2013.

Kenton Lee, Yoav Artzi, Jesse Dodge, and Luke Zettlemoyer. Context-dependent semantic parsing for time expressions. In *ACL (1)*, pages 1437–1447, 2014.

Hector Llorens, Nathanael Chambers, Naushad UzZaman, Nasrin Mostafazadeh, James Allen, and James 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)*, pages 792–800, 2015.

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

Inderjeet Mani, Ben Wellner, Marc Verhagen, and James Pustejovsky. Three approaches to learning TLINKs in TimeML. *Technical Report CS-07–268, Computer Science Department*, 2007.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *The Conference on Advances in Neural Information Processing Systems (NIPS)*. 2013.

Anne-Lyse Minard, Manuela Speranza, Eneko Agirre, Itziar Aldabe, Marieke van Erp, Bernardo Magnini, German Rigau, Ruben Urizar, and Fondazione Bruno Kessler. SemEval-2015 Task 4: TimeLine: Cross-document event ordering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 778–786, 2015.

T. Mitamura, Y. Yamakawa, S. Holm, Z. Song, A. Bies, S. Kulick, and S. Strassel. Event nugget annotation: Processes and issues. In *Proceedings of the Workshop on Events at NAACL-HLT*, 2015.

Nasrin Mostafazadeh, Alyson Grealish, Nathanael Chambers, James Allen, and Lucy Vanderwende. CaTeRS: Causal and temporal relation scheme for semantic annotation of event structures. In *Proceedings of the 4th Workshop on Events: Definition, Detection, Coreference, and Representation*, pages 51–61, 2016.

Qiang Ning, Zhili Feng, and Dan Roth. A structured learning approach to temporal relation extraction. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, pages 1038–1048, Copenhagen, Denmark, 9 2017. Association for Computational Linguistics. URL `http://cogcomp.org/papers/NingFeRo17.pdf`.

Qiang Ning, Zhili Feng, Hao Wu, and Dan Roth. Joint reasoning for temporal and causal relations. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 2278–2288. Association for Computational Linguistics, 7 2018a. URL `http://cogcomp.org/papers/NingFeWuRo18.pdf`.

Qiang Ning, Hao Wu, Haoruo Peng, and Dan Roth. Improving temporal relation extraction with a globally acquired statistical resource. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, pages 841–851. Association for Computational Linguistics, 6 2018b. URL `http://cogcomp.org/papers/NingWuPeRo18.pdf`.

Qiang Ning, Hao Wu, and Dan Roth. A multi-axis annotation scheme for event temporal relations. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 1318–1328. Association for Computational Linguistics, 7 2018c. URL `http://cogcomp.org/papers/NingWuRo18.pdf`.

Qiang Ning, Zhongzhi Yu, Chuchu Fan, and Dan Roth. Exploiting partially annotated data for temporal relation extraction. In *The Joint Conference on Lexical and Computational Semantics (*Proc. of the Joint Conference on Lexical and Computational Sematics)*, pages 148–153. Association for Computational Linguistics, 6 2018d. URL `http://cogcomp.org/papers/NingYuFaRo18.pdf`.

Qiang Ning, Ben Zhou, Zhili Feng, Haoruo Peng, and Dan Roth. Cogcomptime: A tool for understanding time in natural language. In *EMNLP (Demo Track)*, Brussels, Belgium, 11 2018e. Association for Computational Linguistics. URL http://cogcomp.org/papers/NZFPR18.pdf.

Tim O'Gorman, Kristin Wright-Bettner, and Martha Palmer. Richer event description: Integrating event coreference with temporal, causal and bridging annotation. In *Proceedings of the 2nd Workshop on Computing News Storylines (CNS 2016)*, pages 47–56, Austin, Texas, November 2016. Association for Computational Linguistics.

Haoruo Peng, Yangqiu Song, and Dan Roth. Event detection and co-reference with minimal supervision. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016. URL http://cogcomp.org/papers/PengSoRo16.pdf.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.

V. Punyakanok, D. Roth, W. Yih, and D. Zimak. Learning and inference over constrained output. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1124–1129, 2005. URL http://cogcomp.org/papers/PRYZ05.pdf.

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

Nick Rizzolo and Dan Roth. Learning based java for rapid development of nlp systems. In *Proc. of the International Conference on Language Resources and Evaluation (LREC)*, Valletta, Malta, 5 2010. URL http://cogcomp.org/papers/RizzoloRo10.pdf.

D. Roth and W. Yih. A linear programming formulation for global inference in natural language tasks. In Hwee Tou Ng and Ellen Riloff, editors, *Proc. of the Conference on Computational Natural Language Learning (CoNLL)*, pages 1–8. Association for Computational Linguistics, 2004. URL http://cogcomp.org/papers/RothYi04.pdf.

Dan Roth. Incidental supervision: Moving beyond supervised learning. In *Proc. of the Conference on Artificial Intelligence (AAAI)*, 2 2017. URL http://cogcomp.org/papers/Roth-AAAI17-incidental-supervision.pdf.

Zhiyi Song, Ann Bies, Stephanie Strassel, Tom Riese, Justin Mott, Joe Ellis, Jonathan Wright, Seth Kulick, Neville Ryant, and Xiaoyi Ma. From light to rich ere: Annotation of entities, relations, and events. In *Proceedings of the The 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 89–98, Denver, Colorado, June 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W15-0812.

Jannik Strötgen and Michael Gertz. HeidelTime: High quality rule-based extraction and normalization of temporal expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 321–324. Association for Computational Linguistics, 2010.

William F Styler IV, Steven Bethard, Sean Finan, Martha Palmer, Sameer Pradhan, Piet C de Groen, Brad Erickson, Timothy Miller, Chen Lin, Guergana Savova, et al. Temporal annotation in the clinical domain. *Transactions of the Association for Computational Linguistics*, 2:143, 2014.

Naushad UzZaman, Hector Llorens, James Allen, Leon Derczynski, Marc Verhagen, and James Pustejovsky. SemEval-2013 Task 1: TEMPEVAL-3: Evaluating time expressions, events, and temporal relations. *Second Joint Conference on Lexical and Computational Semantics*, 2:1–9, 2013.

Marc Verhagen and James Pustejovsky. Temporal processing with the TARSQI toolkit. In *22nd International Conference on on Computational Linguistics: Demonstration Papers*, pages 189–192. Association for Computational Linguistics, 2008.

Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz, and James Pustejovsky. SemEval-2007 Task 15: TempEval temporal relation identification. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 75–80. Association for Computational Linguistics, 2007.

Marc Verhagen, Roser Sauri, Tommaso Caselli, and James Pustejovsky. SemEval-2010 Task 13: TempEval-2. In *Proceedings of the 5th international workshop on semantic evaluation*, pages 57–62. Association for Computational Linguistics, 2010.

Ran Zhao, Quang Do, and Dan Roth. A robust shallow temporal reasoning system. In *Proc. of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 6 2012. URL `http://cogcomp.org/papers/ZhaoDoRo12.pdf`.