# CogCompTime: A Tool for Understanding Time in Natural Language

**Qiang Ning,**[1] **Ben Zhou,**[1] **Zhili Feng,**[2] **Haoruo Peng,**[1] **Dan Roth**[1,3]
Department of Computer Science
[1]University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA
[2]University of Wisconsin-Madison, Madison, WI 53706, USA
[3]University of Pennsylvania, Philadelphia, PA 19104, USA
{qning2,xzhou45,hpeng7}@illinois.edu, zfeng49@cs.wisc.edu, danroth@seas.upenn.edu

## Abstract

Automatic extraction of temporal information in text is an important component of natural language understanding. It involves two basic tasks: (1) Understanding time expressions that are mentioned explicitly in text (e.g., *February 27, 1998* or *tomorrow*), and (2) Understanding temporal information that is conveyed implicitly via relations. In this paper, we introduce CogCompTime, a system that has these two important functionalities. It incorporates the most recent progress, achieves state-of-the-art performance, and is publicly available.[1] We believe that this demo will be useful for multiple time-aware applications and provide valuable insight for future research in temporal understanding.

## 1 Introduction

Time is an important dimension when we describe the world because many facts are time-sensitive, e.g., one's place of residence, one's employment, or the progress of a conflict between countries. Consequently, many applications can benefit from temporal understanding in natural language, e.g., timeline construction (Do et al., 2012; Minard et al., 2015), clinical events analysis (Jindal and Roth, 2013; Bethard et al., 2015), question answering (Llorens et al., 2015), and causality inference (Ning et al., 2018a).

Temporal understanding from natural language requires two basic components (Verhagen et al., 2007, 2010; UzZaman et al., 2013). The first, also known as the Timex component, requires extracting explicit time expressions in text (i.e., "Timex") and normalize them to a standard format. In Example 1, the Timex is ***February 27, 1998*** and its *normalized* form is '1998-02-27". Note that normalization may also require a reference time for Timexes like "tomorrow", for which we need to know the document creation time (DCT). In addition to DATE, there are also other Timex types including TIME (e.g., ***8 am***), DURATION (e.g., ***3 years***), and SET (e.g., ***every Monday***).

Timexes carry temporal information *explicitly*, but temporal information can also be conveyed *implicitly* via temporal relations (i.e., "TempRel"). In Example 2, there are two events: ***e1:exploded*** and ***e2:died***. The text does not tell readers exactly when they happened, but we do know that there is a TempRel between them, i.e., ***e1:exploded*** happened *before* ***e2:died***. The second basic component of temporal understanding is thus the TempRel component, which requires that TempRels are extracted automatically from text. While the Timex component provides absolute time anchors, the TempRel component provides the relative order of events. These two together provide a complete picture of the temporal dimension of a story, so they are naturally the most important building blocks towards temporal understanding.

| |
|---|
| **Example 1:** Presidents Leonid Kuchma of Ukraine and Boris Yeltsin of Russia signed an economic cooperation plan on (***t1:February 27, 1998***). |
| **Example 2:** A car (***e1:exploded***) in the middle of a group of men playing volleyball. More than 10 people have (***e2:died***), police said. |

In this paper, we present CogCompTime (see Fig. 1), a tool with both the Timex and TempRel components, which are conceptually built on Zhao et al. (2012) and Ning et al. (2017), respectively. CogCompTime is a new implementation that integrates both components and also incorporates the most recent advances in this area (Ning et al., 2018a,b,c). Two highlights are: First, CogCompTime achieves comparable performance to state-of-the-art Timex systems, but is almost two times faster than the second fastest, HeidelTime (Strötgen and Gertz, 2010). Second,

[1] http://groupspaceuiuc.com/temporal/

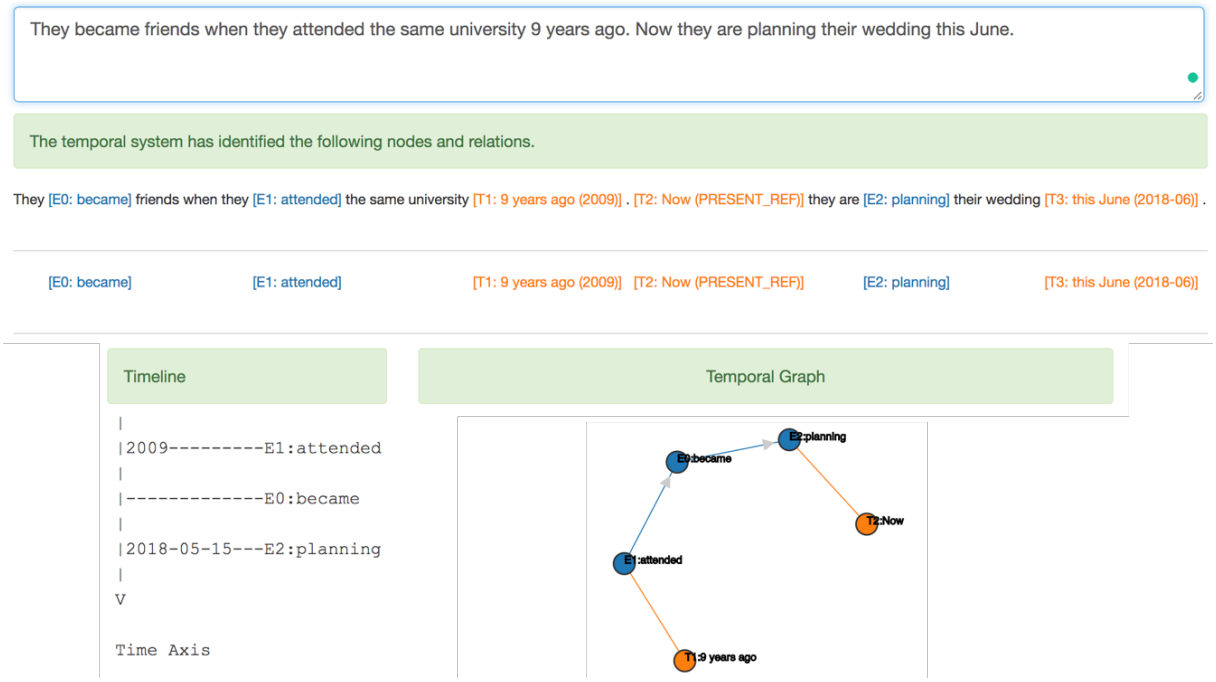


Figure 1: **A snapshot of the interface of CogCompTime.** From top to bottom: Input box, event and Timex highlight, and two visualizations (timeline and graph). The DCT was chosen to be 2018-05-15.

CogCompTime improves the performance of the TempRel component by a large margin, from the literature's $F_1 \approx 50$ (UzZaman et al., 2013) to $F_1 \approx 70$ (see details in Sec. 3). Given these two contributions, we believe that CogCompTime is a good demonstration of the state-of-the-art in temporal understanding. In addition, since CogCompTime is publicly available, it will provide easy access to users working on time-aware applications, as well as valuable insight to researchers seeking further improvements.

We briefly review the literature and explain in detail the processing pipeline of CogCompTime in Sec. 2: the Timex component, the Event Extraction component, and the TempRel component. Following that, we provide a benchmark evaluation in Sec. 3 on the TempEval3 and the MATRES datasets (UzZaman et al., 2013; Ning et al., 2018c). Finally, we point out directions of future work and conclude this paper.

## 2 System

The system pipeline of CogCompTime is shown in Fig. 2: It takes raw text as input and uses CogCompNLP (Khashabi et al., 2018) to extract features such as lemmas, part-of-speech (POS) tags, and semantic role labelings (SRL). Then CogCompTime sequentially applies the Timex component, the event extraction component and the TempRel component. Finally, both a graph visualization and a timeline visualization are provided for the users. In the following, we will explain these main modules in detail.

### 2.1 Timex Component

Existing work on Timex extraction and normalization falls into two categories: rule-based and learning-based. Rule-based systems use regular expressions to extract Timex in text and then deterministic rules to normalize them (e.g., HeidelTime (Strötgen and Gertz, 2010) and SUTime (Chang and Manning, 2012)). Learning-based systems use classification models to chunk out Timexes in text and normalize them based on grammar parsing (e.g., UWTime (Lee et al., 2014) and ParsingTime (Angeli et al., 2012)). CogCompTime adopts a mixed strategy: we use machine learning in the Timex extraction step and rule parsing in the normalization step. This mixed strategy, while maintaining a state-of-the-art performance, significantly improves the computational efficiency of the Timex component, as we show in Sec. 3.

Technically, the Timex extraction step can be formulated as a generic text chunking problem and the standard B(egin), I(nside), and O(utside) labeling scheme can be used. CogCompTime proposes TemporalChunker, by retraining Illinois-Chunker (Punyakanok and Roth, 2001) on top of the Timex chunk annotations provided by the TempEval3

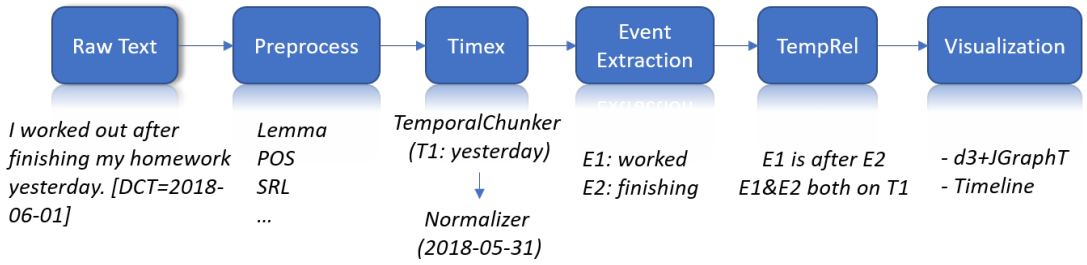


Figure 2: **System pipeline of CogCompTime**: It preprocesses raw text input using CogCompNLP and then applies Timex, Event Extraction, and TempRel components sequentially, with two user-friendly visualizations (i.e., graph-type visualization and timeline-type visualization) provided at the end.

workshop (UzZaman et al., 2013). Here a machine learning based extraction algorithm significantly improves the computational efficiency by quickly sifting out impossible text chunks, as compared to regular expression matching, which has to check every substring of text against regular expressions and is often slow. However, we do admit that learning-based extraction handles corner cases not as well as rule-based systems because of limited training examples.

After Timexes are extracted, we apply rules to normalize them. We think rule-based methods are generally more natural for normalization: On one hand, the desired formats of various types of Timexes are already defined as rules by corresponding annotation guidelines; on the other hand, the intermediate steps of how one Timex is normalized are not annotated in any existing datasets (it is inherently hard to do so), so learning-based methods usually have to introduce latent variables and need more training instances as a result. Therefore, we have adopted a rule-based normalization method. However, we note that an obvious drawback is that the rule set needs to be redesigned for every single language.

## 2.2 Event Extraction Component

Event extraction is closely related to how events are defined. Generally speaking, an event is considered to be an action associated with corresponding participants. In the context of temporal understanding, events are usually represented by their head verb token, so unlike the generic chunking problem in Timex extraction, event extraction can be formulated as a classification problem for each token. Specifically, CogCompTime only considers those *main-axis* events as defined by the MATRES annotation scheme (Ning et al., 2018c), so event extraction is simply a binary classification problem (i.e., whether or not a token is a main-axis event or not). We extract features based on the lemmas and POS tags within a fixed window, SRL, and prepositional phrase head, and train a sparse averaged perceptron classifier for event extraction.

## 2.3 TempRel Component

Temporal relations can be generally modeled by a graph (called temporal graph), where the nodes represent events and Timexes, and the edges represent TempRels. With all the nodes extracted (by previous steps), the TempRel component is to make predictions on the labels of those edges. In this paper, the label set for Event-Event TempRels is *before*, *after*, *equal*, and *vague* and for Event-Timex TempRels is *equal* and *not-equal*.[2] State-of-the-art methods include, e.g., ClearTK (Bethard, 2013), CAEVO (Chambers et al., 2014), and Ning et al. (2017). The TempRel task is known to be very difficult. Ning et al. (2018c) attributes the difficulty partly to the low inter-annotator agreement (IAA) of existing TempRel datasets and proposes a new Multi-Axis Temporal RElations dataset of Start-points (MATRES) with significantly improved IAA, so for the TempRel task, we have chosen MATRES as the benchmark in this paper.[3]

We also incorporate the recent progress of Ning et al. (2017, 2018a,b). The feature set used for TempRel is shown in Fig. 3, which contains features derived individually from each node and jointly from a node pair. Since a node can be either an event or a Timex, an edge can also be either an Event-Event edge or an Event-Timex edge and the features have to vary a bit, as detailed

[2]The simplification of Event-Timex label set is due to our observation that other labels have very low accuracies in our setting. As a demo paper, we have chosen not to use them. However, we think it is interesting and worth further investigation.

[3]Specifically, we only need to replace the TempRel annotations in TempEval3 by the new annotations provided in MATRES.

by Fig. 3. Note that for Event-Event edges, we incorporate features from TemProb,[4] which encodes prior knowledge of typical temporal orders of events (Ning et al., 2018b). With these features, we also adopt the constraint-driven learning algorithm for TempRel classification proposed in Ning et al. (2017) with sparse averaged perceptron. Then our TempRel component assigns local prediction scores (i.e., soft-max scores) to each edge and solves an integer linear programming (ILP) problem via Gurobi (Gurobi Optimization, 2015) to achieve globally consistent temporal graphs (please refer to Ning et al. (2018a) for details). CogCompTime is a unique package so far that incorporates all these recent progresses.

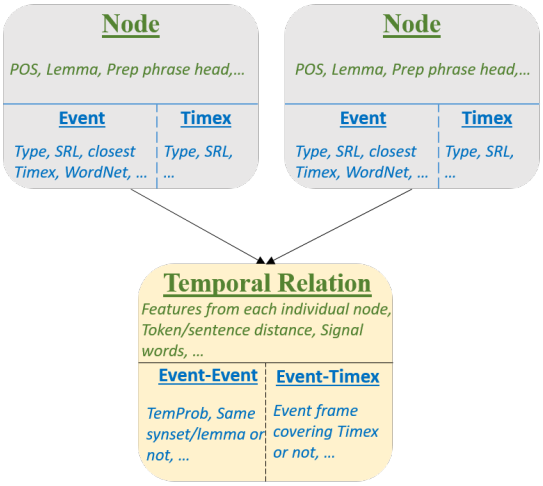


Figure 3: **The primary features used in the TempRel component.** Since there are two types of nodes (i.e., event and Timex), and two types of TempRels (i.e., Event-Event and Event-Timex), we put the common features above and split specific feature sets below. Conjunctive features are not listed exhaustively here.

### 2.4 Visualization

As shown in Fig. 1, we highlight the extracted Timexes and events in text. Specifically for Timexes, we also annotate their normalized values along with their chunks.

We provide two forms of visualization for the extracted TempRels. Since TempRels can be naturally modeled by a graph, a graph visualization is an obvious choice and we use d3 (`https://d3js.org/`) in CogCompTime. Additionally, we provide a more compact visualization to those graphs via timeline construction. Since a graph is only partially ordered (as opposed to a timeline which is fully ordered), we resort to the appearance order of events in timeline construction when the temporal order is *vague* based on the corresponding graph.

[4] `http://cogcomp.org/page/resource_view/114`

## 3 Benchmark Experiment

We used the articles provided by the TempEval3 workshop, with the original train/test split in our experiment: TimeBank and AQUAINT were for training (256 articles), and Platinum was for testing (20 articles). Note that we also replaced the TempRel annotations in the original TempEval3 datasets by MATRES (Ning et al., 2018c) due to its higher IAA. In the Timex component, TemporalChunker by default takes 10% of the train set as dev, and in other components, 5-fold cross-validation was used for parameter tuning.

Table 1 evaluates the Timex component of CogCompTime, comparing with state-of-the-art systems. The "normalization" and "end-to-end" columns were evaluated based on gold Timex extraction and system Timex extraction, respectively. The fact that CogCompTime had the best extraction $F_1$ and normalization accuracy but not the best end-to-end performance is due to our mixed strategy: Timexes extracted by our learning-based TemporalChunker sometimes cannot be normalized correctly by our rule-based normalizer. This phenomenon is relatively more severe in CogCompTime comparing to systems that are consistently rule-based or learning-based in both extraction and normalization. However, the computational efficiency is improved significantly by reducing the runtime of the second fastest, HeidelTime, by more than 50%.

Table 2 shows the performance of the Event Extraction and TempRel components. We also copied the Timex extraction performance from Table 1. Note that CogCompTime only extracts those main-axis events as defined by MATRES (Ning et al., 2018c). Since Ning et al. (2018c) did not propose an event extraction method, Table 2 is in fact the first reported performance of event extraction on MATRES and as we see, both the precision and recall are better than those numbers reported in TempEval3. Note that since CogCompTime works on different annotations, this does not indicate that our event extraction algorithm is better than those participants in TempEval3; instead, this indicates that the event extraction problem in MATRES is an easier machine learning task (and probably better-defined).

The performance of TempRel extraction is fur-

| Timex Systems | Extraction | | | Normalization | End-to-end | Runtime |
|---|---|---|---|---|---|---|
| | P | R | $F_1$ | Accuracy | $F_1$ | Seconds |
| HeidelTime (Strötgen and Gertz, 2010) | 84.0 | 79.7 | 81.8 | 78.1† | 78.1 | 18 |
| SUTime (Chang and Manning, 2012) | 80.0 | 81.1 | 80.6 | 69.8† | 69.8 | 16 |
| UWTime (Lee et al., 2014) | **86.7** | 80.4 | 83.5 | 84.4 | **82.7** | 400 |
| CogCompTime | 86.5 | **83.3** | **84.9** | **84.7** | 76.8 | **7** |

Table 1: **Performance of our Timex component** compared with state-of-the-art systems on a benchmark dataset, the Platinum dataset from the TempEval3 workshop (UzZaman et al., 2013). The "extraction" and "normalization" columns are the two intermediate steps. "Normalization" was evaluated given gold extraction, while "end-to-end" means system extraction was used. Runtimes were evaluated under the same setup.

†HeidelTime and SUTime have no clear-cut between extraction and normalization, so even if gold Timex chunks are fed in, their extraction step cannot be easily skipped.

ther evaluated in Table 2, both when gold event and Timex extraction is used and when system extraction is used. As for Event-Event TempRels, we also introduce a new relaxed metric[5], where predictions of *before/after* are not penalized when the gold label is *vague*. This is based on the definition of *vague* in MATRES, i.e., to assign *vague* labels when either *before* or *after* is reasonable. We think this relaxed metric is even more suitable when creating timelines from temporal graphs, where an order must be picked anyhow when two events have a *vague* relation. When system extraction was used, the TempRel performance saw a large drop. However, the performance here, although it is perhaps still not sufficiently good for some applications, is already a significant step forward in temporal understanding. As a reference point, the best system in TempEval3, ClearTK (Bethard, 2013), had P=34.08, R=28.40, $F_1$=30.98 (using system extraction) and P=37.32, R=35.25, $F_1$=36.26 (using gold extraction). Again, given the dataset difference, these numbers are not directly comparable, but it indicates that the MATRES dataset used here probably has the TempRel task better defined and we hope this demo paper will be a good showcase of the new state-of-the-art.

| | P | R | $F_1$ |
|---|---|---|---|
| Event Extraction | 83.5 | 87.0 | 85.2 |
| Timex Extraction | 86.5 | 83.3 | 84.9 |
| **Gold Extraction** | | | |
| Event-Event | 61.6 | 70.9 | 65.9 |
| Event-Event (Relaxed) | 75.2 | 74.8 | 75.0 |
| Event-Timex | 84.6 | 84.6 | 84.6 |
| **System Extraction** | | | |
| Event-Event | 48.4 | 58.0 | 52.8 |
| Event-Event (Relaxed) | 75.6 | 61.8 | 68.0 |
| Event-Timex | 79.5 | 61.1 | 69.0 |

Table 2: **Performance of the Event/Timex Extraction and TempRel components** when gold/system extraction is used. The relaxed metric does not penalize the system if a *before/after* prediction is made on a *vague* relation. Please also refer to the text about this metric.

[5]This relaxed metric does not apply to Event-Timex TempRels since the label set is only *equal* and *not-equal*, .

## 4 Future Work

We plan to further improve CogCompTime in the following directions. First, the MATRES dataset (Ning et al., 2018c) only considers verb events, but nominal events are also very common and important, so we plan to incorporate nominal event extraction and corresponding TempRel extraction. Second, CogCompTime currently does not incorporate an event coreference component. Since coreference is important for bridging long-distance event pairs, it is a desirable feature. We can adopt existing event coreferecing techniques such as Peng et al. (2016) in the next step. Third, CogCompTime currently only works on the main-axis events as defined in MATRES. How to incorporate other axes, e.g., intention axis, opinion axis, and hypothesis axis, requires further investigation.

## 5 Conclusion

In this paper, we present CogCompTime, a new package that, given raw text, (1) extracts time expressions (Timex) and normalizes them to a standard format, and (2) extracts events on the main time axis of a story (per the definition of Ning et al. (2018c)) and the temporal relations between events and Timexes. CogCompTime takes advantage of many recent advances and achieves state-of-the-art performance in both tasks. We think this demo will be interesting for a broad audience because it is useful not only for identifying the shortcomings of existing methods, but also for applications that depend on temporal understanding of natural language text.

## References

Gabor Angeli, Christopher D. Manning, and Daniel Jurafsky. 2012. Parsing time: Learning to interpret time expressions. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*.

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

Steven Bethard, Leon Derczynski, Guergana Savova, James Pustejovsky, and Marc Verhagen. 2015. SemEval-2015 Task 6: Clinical TempEval. In *Proc. of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 806–814.

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

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

Quang Do, Wei Lu, and Dan Roth. 2012. Joint inference for event timeline construction. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Inc. Gurobi Optimization. 2015. Gurobi optimizer reference manual.

Prateek Jindal and Dan Roth. 2013. Extraction of events and temporal expressions from clinical narratives. *Journal of Biomedical Informatics (JBI)*.

Daniel Khashabi, Mark Sammons, Ben Zhou, Tom Redman, Christos Christodoulopoulos, and Vivek Srikumar et al. 2018. CogCompNLP: Your swiss army knife for NLP. In *11th Language Resources and Evaluation Conference*.

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

Hector Llorens, Nathanael Chambers, Naushad UzZaman, Nasrin Mostafazadeh, James Allen, and James Pustejovsky. 2015. SemEval-2015 Task 5: QA TempEval - evaluating temporal information understanding with question answering. In *Proc. of the 9th International Workshop on SemEval*.

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

Qiang Ning, Zhili Feng, and Dan Roth. 2017. A structured learning approach to temporal relation extraction. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1038–1048, Copenhagen, Denmark. Association for Computational Linguistics.

Qiang Ning, Zhili Feng, Hao Wu, and Dan Roth. 2018a. Joint reasoning for temporal and causal relations. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 2278–2288, Melbourne, Australia. Association for Computational Linguistics.

Qiang Ning, Hao Wu, Haoruo Peng, and Dan Roth. 2018b. 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, New Orleans, Louisiana. Association for Computational Linguistics.

Qiang Ning, Hao Wu, and Dan Roth. 2018c. 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, Melbourne, Australia. Association for Computational Linguistics.

Haoruo Peng, Yangqiu Song, and Dan Roth. 2016. Event detection and co-reference with minimal supervision. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Vasin Punyakanok and Dan Roth. 2001. The use of classifiers in sequential inference. In *Proc. of the Conference on Neural Information Processing Systems (NIPS)*, pages 995–1001. MIT Press.

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

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

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

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

Ran Zhao, Quang Do, and Dan Roth. 2012. A robust shallow temporal reasoning system. In *Proc. of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.