Annotating Temporal Dependency Graphs via Crowdsourcing

Jiarui Yao[1], Haoling Qiu[2], Bonan Min[2], and Nianwen Xue[1]

[1]Brandeis University
{jryao, xuen}@brandeis.edu
[2]Raytheon BBN Technologies
{haoling.qiu, bonan.min}@raytheon.com

Abstract

We present the construction of a corpus of 500 Wikinews articles annotated with *temporal dependency graphs* (TDGs) that can be used to train systems to understand temporal relations in text. We argue that temporal dependency graphs, built on previous research on narrative times and temporal anaphora, provide a representation scheme that achieves a good balance between completeness and practicality in temporal annotation. We also provide a crowdsourcing strategy to annotate TDGs, and demonstrate the feasibility of this approach with an evaluation of the quality of the annotation, and the utility of the resulting data set by training a machine learning model on this data set. This data set is publicly available[1].

1 Introduction

Understanding temporal relations between events in a text is an important part of understanding the "meaning" of text. With the wide adoption of machine learning methods in natural language processing, the ability to achieve a large-scale high-quality temporally annotated data set has been the bottleneck in advancing the state of the art in this area. Even though the first temporal annotation scheme, TimeML (Pustejovsky et al., 2003a; Saurí et al., 2006), was proposed over a decade ago, temporally annotated data is still relatively scarce. The largest data set that we are aware of is the data set used in TempEval-3 (UzZaman et al., 2013), and it consists of 276 articles from the TimeBank Corpus (Pustejovsky et al., 2003b) and the AQUAINT Corpus. This data set was later re-annotated by (Ning et al., 2018) to improve its annotation consistency using a crowdsourcing approach.

There are many challenges that have contributed to this state of affairs. Temporal relations are often confounded with modalities (How do you order an event that actually happened with one that might happen?). Some events are ambiguous between an instantaneous and stative reading (Does "marriage" refer to the start of the marriage or does it refer to the state when marriage is in effect?) and this complicates its temporal relation with other events. While these have all contributed to the difficulty in temporal annotation, the main challenge is a practical one. The general assumption in temporal annotation has been that the temporal relation between every pair of events in a text has to be specified in order to fully understand the temporal relations in a text. This amounts to constructing a fully connected graph in which every event is connected to another event. With this *pair-wise* approach, a text of n events has $\binom{n}{2}$ event pairs that need to be considered. As the value of n increases, the number of event pairs quickly becomes very large. In practice, attempts to achieve complete annotation often fell far short and had to settle with covering all event pairs within a short text window (e.g., within adjacent sentences). Even with this restriction, it is still difficult to produce a large data set. For example, there are only 36 articles in TimeBank-Dense (Cassidy et al., 2014). On the other end of the spectrum, approaches that allow the annotators to select a subset of the event pairs to annotate often end up with sparse and inconsistent annotation, as different annotators often select different event pairs to annotate. For example, while the TimeBank corpus has relatively more articles, but only annotates a relatively small number of temporal relations (6,418 in total). There are also efforts that report improved annotation consistency by focusing on specific syntactic constructions (Bethard et al., 2007) or one aspect of temporal annotation (Reimers et al., 2016), but this comes at the cost of incomplete annotation.

One promising recent approach to get out of this

[1] https://github.com/Jryao/temporal_dependency_graphs_crowdsourcing

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 5368–5380,
November 16–20, 2020. ©2020 Association for Computational Linguistics

dilemma is to focus on *dependencies* between time expressions, between time expressions and events, and between events (Zhang and Xue, 2018b), based on the observation that time expressions and events are often expressed in relative terms and their temporal location needs to be understood with a *reference time* in mind. Consider the examples in (1):

(1) a. He **left** on Friday. He left home at 9:00am.
b. The Pentagon **said** today that it would **reexamine** the question.
c. The Pentagon said today that it will **reexamine** the question.
d. Ricky New **entered** the store carrying a large stick, **demanded** money, **assaulted** the clerk with the stick, and **left** with an undisclosed amount of money.

In (1a), the interpretation of the time expression *9:00am* depends on another time expression *Friday*, which in turn depends on when the time this sentence is written, generally known as the document creation time (DCT). In (1b), the temporal location of *re-examine* can only be understood in relation to *said* (it happens after the saying event). Note that it does not depend on *today*, as it may or may not happen on that day. In contrast, in (1c), the temporal location of *re-examine* can only be understood with respect to the DCT, not *said*. Another example of one event depends on another event for its temporal interpretation is (1d), where the temporal interpretation of *demanded* depends on *entered* (it happened after *entered*), the temporal interpretation of *assaulted* depends on *demanded*, and the temporal interpretation of *left* depends on *assaulted*. This is a linguistic phenomenon known as *temporal anaphora* (Reichenbach, 1949; Partee, 1973, 1984; Hinrichs, 1986; Webber, 1988; Bohnemeyer, 2009) that has been extensively studied in computational linguistics. The working hypothesis of this *dependency-based* approach is that by annotating the dependencies, additional temporal relations can be inferred, via transitivity, or via common sense reasoning. This hypothesis seems to have been born out in (1d): Based on the dependencies, one can additionally infer that *assaulted* happened after *entered*, for example.

Zhang and Xue (2018b) made the assumption that there is exactly one reference time for each event or time expression. With this assumption, the temporal relations between time expressions and events in a text will form a *temporal dependency tree* (TDT). This means that each event or time expression only relates to one other event or time expression, making TDT a much scalable annotation problem in practice. However, there are reasons to believe that this assumption is too stringent, and in some cases, multiple reference times may be needed to properly interpret the temporal location of an event. In this paper, we extend the temporal dependency tree to *temporal dependency graph* (TDG), allowing each event to have a reference time expression, a reference event, or both. Compared with TDT, TDG does not substantially increase the number of temporal relations in a text that need to be annotated while improving its expressiveness.

We also investigate the feasibility of annotating TDGs from scratch via crowdsourcing, meaning we start with identifying events and time expressions, and then annotate the temporal relations between them. Previous work on crowdsourcing temporal annotations relies on the events and time expressions already identified in the TimeBank (Ning et al., 2018; Zhang and Xue, 2019), and this limits the possibility of expanding temporally annotated datasets beyond what already exist. We show that with a carefully designed annotation strategy, annotating TDGs via crowdsourcing is feasible. We annotated a corpus of 500 Wikinews articles with this approach, and created the largest corpus in terms of the number of articles and the number of event or time expression pairs.

The remainder of the paper is organized as follows. We provide a brief overview of the TDT representation and propose our extension in Section 2. We present our crowdsourcing strategy in Section 3. We present a quantitative analysis of our corpus in Section 4, and then evaluate the quality of our annotation in Section 5. In Section 6, we retrain a neural ranking parser (Zhang and Xue, 2018a) on this data set to demonstrate its utility and establish a baseline for fellow researchers. We discuss related work in Section 7, and conclude in Section 8.

2 From Temporal Dependency Tree to Temporal Dependency Graph

2.1 Temporal Dependency Tree

Zhang and Xue (2018b) defines a temporal dependency tree as a rooted directed edge-labeled tree in which nodes are events and time expressions as well as a few pre-defined meta nodes (e.g. DCT). The parent of each node is its reference time. The

Relations	Definitions
Before	A before B
After	A after B
Overlap	The temporal interval of A overlaps that of B. Applies only to events
Includes	Time expression A includes B

Table 1: Temporal relations used in TDT

temporal dependency tree obeys the following constraint: the reference time of a time expression can only be a time expression or a meta node and it cannot be an event. The reference time of an event, on the other hand, can be a time expression, a meta node or another event. For example, in (2), the reference time for the time expression *yesterday* is *DCT*, and the reference time for *went* and *had* is *yesterday*. The edges in the temporal dependency tree are labeled with temporal relations, which are simplified version of what is used in TimeML. The full set of temporal relations are presented in Table 1. The relation between time expressions and meta nodes is represented as "Depend-on".

(2) Yesterday, I **went** to the museum, then **had** dinner with my friends.

2.2 Temporal Dependency Graph

In a temporal dependency tree, each child node can only have one parent or one reference time. However, there are reasons to believe that this assumption is too stringent. In (2), for example, the reference time of the *had* event can be *went*, or the time expression *yesterday*. To precisely determine the temporal location of *had*, we need to say it happened *yesterday* and after *went*. To account for linguistic phenomena like this, we extend the temporal dependency tree to temporal dependency graph, where each event always has a reference time that is a time expression or a meta node. We call this the *reference timex*. Optionally it can also have another event as its reference time, and we call this the *reference event*. The reference event is optional because not all events have an reference event. For example, in (2), *went* does not have another event as its reference time and only has a reference timex. In TDG, the reference timex of an event is the most specific (i.e., the smallest) *narrative time* (Pustejovsky and Stubbs, 2011) that the event can be placed into. If such a narrative time is not available, this event should be anchored to DCT. The reference event of an event is the event that gives this event the most specific temporal location.

Figure 1 provides a contrast between the TDT and TDG for the example in (2). The solid lines indicate edges for TDT, and the dotted line indicates the additional edge in its TDG.

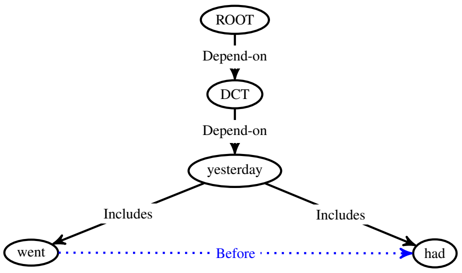


Figure 1: Temporal Dependency Structure for (2).

3 Crowdsourcing Strategy

Crowdsourcing is generally accepted as a cost-effective alternative to the traditional annotation approach where annotators are provided detailed guidelines and carefully trained to meet certain consistency threshold before productive annotation can start. In a crowdsourcing setting, we oftentimes hire a much larger set of annotators that are not professionally trained and may be only working on the task sporadically. Therefore, it is infeasible to ask them to follow detailed guidelines. As a result, successful crowdsourcing tasks tend to be simple and intuitive. To make crowdsourcing practical for TDGs, we adopt a divide-and-conquer strategy that decomposes the annotation of a TDG into five steps. A top-level flowchart of our annotation process is in Figure 2. After all annotation steps are completed, we assemble the TDG for each text, and an example that illustrates the step-by-step construction of the TDG of a text is provided in Figure 3.

In all steps, each annotation is completed by three crowd workers using the Amazon Mechanic Turk platform. Unless otherwise specified, the majority-voted answer is designated as the gold annotation. We explain each annotation step in greater detail in the rest of the section.

3.1 Step 1: Time Expression Identification

In the spirit of simplifying the task as much as possible, we present crowd workers a candidate time expression and ask them to decide if it is indeed a time expression instead of asking crowd workers to select time expressions from raw text directly. This makes this task a binary decision rather than an open-ended text selection. This means we

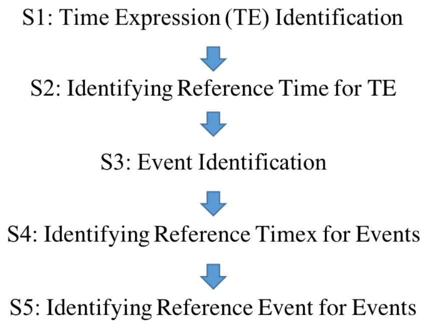

S1: Time Expression (TE) Identification
S2: Identifying Reference Time for TE
S3: Event Identification
S4: Identifying Reference Timex for Events
S5: Identifying Reference Event for Events

Figure 2: A top-level flow chart for the annotation tasks. S1, S2, ..., S5 refer to the steps 1-5.

need a reliable way to generate candidate time expressions without excluding true time expressions. To achieve this, we supplement candidate time expressions extracted with HeidelTime (Strötgen and Gertz, 2013) with numeric expressions extracted with regular expression patterns.

3.2 Step 2: Identifying Reference Time for Time Expressions

With the assumption that the reference time for a time expression can only be another time expression, the search space for the reference time of a time expression is greatly reduced. Following the practice of (Zhang and Xue, 2018b), we classify time expressions into different types: *locatable* time expressions that can be placed on a timeline and *unlocatable* time expressions. Locatable time expressions include concrete *absolute* time expressions (e.g., "2020"), concrete *relative* time expressions (e.g., "this year") and *vague* time expressions (e.g., "nowadays"). Unlocatable time expressions include durations (e.g., "two months"), set (e.g., "every month"). Concrete absolute time expressions do not need a reference time to be resolved, and they are directly attached to the root of the dependency graph without going through an annotation process. Vague time expressions belong to a closed set and they can be anchored to pre-defined meta nodes (Present/Past/Future_reference) in a deterministic manner. Unlocatable time expressions cannot be resolved with reference times and are ignored. The focus is on identifying the reference time for concrete relative time expressions, which can be another concrete time expression (absolute or relative) or the DCT. To classify the time expressions, we use regular expression patterns in the case of absolute time expressions and dictionaries when there is a closed set of expressions for that particular type.

As many concrete relative time expressions can be resolved to DCT, we further split this step into two subtasks. In the first subtask, we ask crowd workers if a time expression can be resolved to DCT. If the answer is "No", in the second subtask, crowd workers will be asked to find the reference time for this time expression. However, it turned out that in most cases the reference time is the DCT. After the first subtask was completed, we found that only for fewer than 200 time expressions (less than 10% of all time expressions), their reference time is not DCT. We decide that it is not worth setting up another crowdsourcing task and to have experts annotate the second subtask.

3.3 Step 3: Event Identification

Following the same approach with time expression identification, we give crowd workers an event candidate and ask them if that is an event. To collect event candidates, we first construct a list of common event trigger words[2]. Then, we parse the raw text with Stanford CoreNLP dependency parser (Chen and Manning, 2014; Manning et al., 2014) and add the verbs of each sentence as well as the root of its dependency parse as event candidates. We exclude modal events, negative events and stative events in this work. In total, we collected 27,487 event candidates.

3.3.1 Quality Control for Step 3

We set up a qualification test for this pass. Crowd workers have to achieve at least 70% accuracy in order to be eligible to work on this task. In addition, 5 questions with gold answers are inserted into each HIT. If a crowd worker's accuracy on the 5 test questions drops below 60%, he or she will be blocked from the task and his/her annotation will be discarded.

3.3.2 Post-processing

We performed post-processing procedures to filter out some trivial mistakes in the crowdsourced annotation. For event annotation, we excluded negative events in this project since negative events do not have temporal locations. However, even though we made this clear in our instructions to the workers, some crowd workers still annotated negative events as events. We take advantage of the Stanford CoreNLP dependency parser (Chen and Manning, 2014) to filter out some of these unwanted events.

[2] This list is publicly available along with the annotated data set.

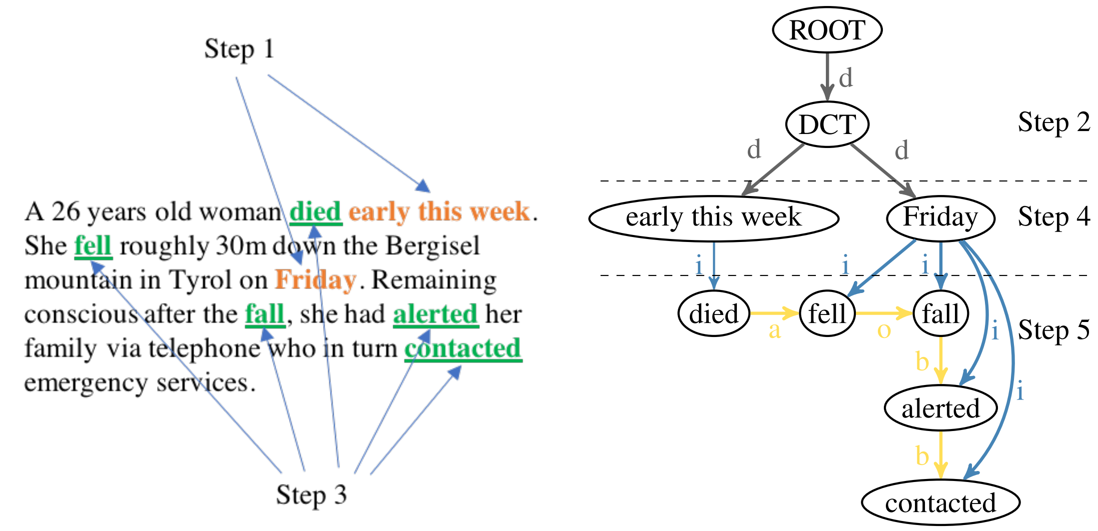

Step 1
A 26 years old woman died early this week.
She fell roughly 30m down the Bergisel
mountain in Tyrol on Friday. Remaining
conscious after the fall, she had alerted her
family via telephone who in turn contacted
emergency services.
Step 3
ROOT
d
DCT
Step 2
d
d
early this week
Friday
Step 4
i
i
i
died
a
fell
o
fall
i
Step 5
b
alerted
i
b
contacted

Figure 3: Constructing a TDG in 5 steps.

3.4 Step 4: Identifying Reference Timex for Events

In this step, the crowdworkers are asked to select a reference timex for a given event from a list of candidate time expressions. In theory, all the time expressions in a text can be considered as candidate reference times. However, in practice, we observe that in the vast majority of cases the reference timex of an event is in the same paragraph as the event itself. We also observe that in news reports, key events and time expressions are usually described in the first paragraph. To reduce the number of candidate time expressions and simplify this task, for each event, we present crowd workers with time expressions in the same paragraph as well as time expressions in the first paragraph as its candidate reference timexes.

To participate in this task, crowd workers need to achieve 70% accuracy on the qualifying test. Four questions with gold answers are added to each HIT.

3.4.1 Answer Aggregation

Additionally, after computing each worker's average accuracy on all the tasks they submitted, we found that some workers were able to maintain a higher average accuracy that was between 0.7 to 0.8. This discovery inspires us to come up with a tiered-approach with weighted answer aggregation. Specifically, we compute the average accuracy of each worker and create a "best workers" group which consists of the crowd workers whose average accuracy is above 0.7. Then, for each question, if the three crowd workers give the same answer, that answer becomes the gold answer. Otherwise, if one crowd worker is in the "best workers" group, his/her answer becomes the gold answer; else, the majority answer is the gold answer.

3.5 Step 5: Identifying Reference Event for Events

In addition to reference timexes, some events also have a reference event, which gives it the most specific temporal location. In a crowdsourcing setting, given an event, the crowd worker is provided with a list of candidate events that are potential reference times for the event. This is a challenging problem as the list of candidates can be very long in a typical text, and there are now no obvious heuristics that can be used to shrink down the list. We rely on linguistic insights from research on *temporal anaphora* to identify where potential reference events are (Hinrichs, 1986; Webber, 1988), and split this task into subtasks that reflect different scenarios of how an event is related to its reference time.

We split this task into two subtasks: a within-sentence reference event identification task and a cross-sentence reference event identification task. In the first subtask, given a non-sentence-initial event, crowdworkers are asked to identify its reference event in the same sentence. In the second subtask, given a sentence-initial event, crowdworkers are asked to identify its reference event from previous sentences.

Within-sentence annotation When the reference time of an event is from the same sentence, we can take advantage of syntactic patterns to identify candidate reference events. For example, for events in complement clauses their reference events are typically the matrix events. The event in the subject of a sentence depends on the main verb for temporal interpretation (3a). In (3b), the event in a purpose clause depends on the main verb. When there is a temporal conjunction (3c), it provides clue for the temporal dependency between the event in the temporal modifier and the event in the main clause. Based on this discovery, we extract event pairs in the following structures with Stanford dependency parser (Chen and Manning, 2014): complement clauses, relative clauses, temporal conjunctions, arguments and predicates, and purpose expressions.

(3) a. The **landslide hit** the village.
    b. I **got** up at 6am to **take** the train.
    c. Right **before** I **got** to the station, the train **left**.

Cross-sentence annotation In wikinews articles, each paragraph is usually a self-contained discourse segment. We assume that the first event of a discourse segment starts a new temporal chain and does not have a reference event. In addition, to make the annotation problem practical, we limit the maximum number of reference event candidates to be 4 when proposing reference event candidates from previous sentences for crowd-workers to choose from.

In Step 5, 11K events are given to crowd workers for reference event resolution. Crowd workers need to achieve 0.6 accuracy on the qualification test. We use the same answer aggregation approach as Step 4.

4 Corpus Statistics

The news articles that we use for our annotation are sampled from English Wikinews[3] and extracted with the publicly available WikiExtractor.py script to remove hypertext markups.[4]

Table 2 presents a comparison of this corpus and some other temporally annotated corpora. TDT-Crd (Zhang and Xue, 2019) is a crowdsourced TDT corpus annotated on top of TimeBank (Pustejovsky et al., 2003b), while TB-Dense (Cassidy et al., 2014) is annotated on a subset of TimeBank. MATRES (Ning et al., 2018) was first annotated on TB-Dense, then extended[5] to the TempEval-3 (UzZaman et al., 2012) data set. TDT-Crd includes events that are matrix verbs. MATRES annotates verb events on the main axis and orthogonal axes (see Ning et al., 2018 for their axis types), and does not annotate the relations between events and time expressions. Compared to the four TimeBank-based corpora, our corpus is much larger on every count, with 500 news articles, 14,974 events, 2,485 time expressions, and 28,350 temporal relations.

	Docs	Timex	Events	Rels
TimeBank	183	1,414	7,935	6,418
TB-Dense	36	289	1,729	12,715
MATRES	275	-	1,790	13,577
TDT-Crd	183	1,414	2,691	4,105
This work	500	2,485	14,974	28,350

Table 2: Events, time expressions and temporal relations in various corpora.

A more detailed analysis of temporal relations in our corpus shows that for reference timex identification, 19% of the events have a reference timex that is in the same sentence, while 17% of the events have a reference timex that is in different sentences. Around 64% of the events have DCT as the reference timex. This indicates that in the majority of cases, the reference timex of an event cannot be found in the same sentence, and our TDG annotation is able to capture these relations as a document-level annotation framework. Our analysis also shows that for reference event identification, 27% of the events do not have an reference event, and these are usually the first event of a paragraph. Table 3 shows the distribution of temporal relations in reference event identification.

No RE	Before	After	Overlap
27%	24%	25%	24%

Table 3: Distribution of temporal relations between events and events. RE refers to reference event.

5 Annotation Evaluation

We evaluate the annotation quality of our data set with two evaluation metrics. The first metric measures the agreement between crowd workers and experts, and the second metric, Worker Agreement With Aggregate (WAWA) (Ning et al., 2018), measures the agreement among the crowd-workers. Both metrics have their advantages and disadvantages but in conjunction, they provide a fuller picture of the annotation consistency of our data set.

[3] https://en.wikinews.org/

[4] https://github.com/attardi/wikiextractor

[5] https://cogcomp.seas.upenn.edu/page/publication_view/834

To measure the agreement between the expert and crowd-workers, ten percent of the articles are double annotated by experts and crowd workers. As shown in Table 4, high agreements are achieved in the first two steps. The post-processing effort in Step 3 brings the agreement from 0.79 to 0.83. Table 5 presents the agreement scores of the reference time identification for events. In Step 4 and 5, annotations are conducted both on crowdsourced events and time expressions and gold events and time expressions. Agreement scores are calculated for both labeled (L) and unlabeled (U) annotation. Unlabeled agreement evaluates reference time identification, while labeled agreement evaluates both reference time identification and relation annotation between a given event and its reference time. We achieve a labeled (unlabeled) F1-score of 0.77 (0.85) on gold events and time expressions in the reference timex identification, and a labeled (unlabeled) F1-score of 0.75 (0.83) on gold events in the reference event identification. There is an error propagation effect when crowdsourced events and time expressions are used, and the agreement scores are lower.

We also evaluate our annotation with the WAWA metric, which measures the average agreement between crowd workers' annotation and the aggregate answer. The WAWA score measures the agreement among crowd workers, and as such it is sensitive to the number of crowd workers and whether there are outliers. Nevertheless, it is a useful metric, assuming that when an annotation task is well-defined, there should be less variation among the annotators. When computing WAWA, we used the majority aggregation instead of the weighted majority aggregation, and we only computed the labeled agreement. The WAWA scores for the subtasks are also reported in Tables 4 and 5.

Task	Agreement	WAWA
S1: Timex ID	0.96	0.97
S2: Timex RT	0.89	0.95
S3: Event ID	0.79	0.84

Table 4: Agreement F1 and WAWA for time expression identification (ID), time expression reference time (RT) identification and event identification.

Relation only annotation evaluation is also performed for Step 4 and Step 5 on gold events and time expressions. Specifically, we compute the portion of correct relation when the reference time is correctly annotated. As we can see in Table 6, our relation-only annotation agreements between crowd workers and experts for S4 and S5 are 0.91 and 0.85. This shows that finding the appropriate reference timex and reference event is the more challenging aspect of the annotation. The relation-only agreement is in the ballpark of annotation frameworks such as Ning et al. (2018) that do not require the identification of reference events or timexes, although a strict comparison is impossible given different data sets are used.

Task	Node	L	U	WAWA
S4: RT ID	Gold	0.77	0.85	0.81
	Crowd	0.61	0.67	0.78
S5: RE ID	Gold	0.75	0.83	0.75
	Crowd	0.52	0.59	0.70

Table 5: Agreement F1 for reference timex (RT) and reference event (RE) identification for events. The third column evaluates the labeled (L) annotation, the fourth column evaluates the unlabeled (U) annotation.

	S4	S5
Agreement	0.91	0.85

Table 6: Relation only annotation agreement.

5.1 Error Analysis for Reference Event Identification

The most challenging aspect of this project is identifying the reference event for a given event and determining their temporal relations. To gain a better understanding of the quality of the crowdsourced data set, we did a manual error analysis of this pass. We randomly sampled 100 instances where the crowdsourced reference events are different from that identified by the expert. We then decide if the crowdsourced annotation is simply wrong or is different from the expert annotation but is still reasonable. For example, in (4), the reference event identified by crowdworkers for event *discovered* is *pursued*. However, the *pursued* event happened before the *incident* event, and the *incident* event happened before the *discovered* event, so we get the most specific temporal location for event *discovered* when *incident* is used as the reference event. The crowdsourced annotation in this case is simply wrong. Example (5) is an edge case where it is reasonable to say the *fight* event happened before event *lose*, but it's also reasonable to say that the *lose* event is a part of the *fight* process, so the *fight* event overlaps the *lose* event.

(4) The **incident** took place after three youngsters on bicycles **pursued** two youths who sought cover inside the store. Investigators have **discovered** that Kamaleswaran's mother was also inside the store during the shooting.

(5) Terror organisations and their pawns are targeting our innocent citizens in the most immoral and heartless way as they **lose** the **fight** against our security forces.

In the 100 instances, 36 of them are wrong, while the other 64 are different from that of the expert but reasonable. As we can see in Table 7, in the 36 wrong annotations, 21 (58%) of them are caused by the crowd worker identifying the incorrect reference event while the other 15 (42%) of them are annotated with incorrect temporal relations. This shows that identifying the correct reference time is more challenging than determining the temporal relation.

Structure	Relation	Total
21 (58%)	15 (42%)	36

Table 7: Distribution of the cause of the wrong annotations in the 100 sampled instances.

6 Experiments

We test our data with an attention-based neural ranking temporal dependency parser[6] that Zhang and Xue (2018a) developed for TDT, which parses the temporal dependency tree by ranking the candidate parents for each node. To apply the tree parser to the graph data, we first add a meta node as reference event for events that only have a reference timex. Then, we rank all the time expressions for events and pick the one with the highest score as its reference timex, and rank all the events and select its reference event. To help the model learn the relations between DCT and events, a POS tag feature is added which only distinguishes present tense verb events with other events. This feature is represented as a one hot vector. We use the same hyperparameter values as Zhang and Xue (2018a). In the 500 documents, 400 are used as training data, 50 as validation data, and 50 as test data. The test data is annotated by experts, and the validation data is generated from crowdsourced annotation as follows: if there is no agreement for one question, i.e. three crowd workers chose three different answers, then experts annotate that question.

We also develop a heuristic baseline system as follows. First, each time expression is attached to DCT. For each event, if there is a time expression in the same sentence, we attach the event to that time expression, and designate the relation as "Include". Otherwise that event is attached to DCT, and the relation is "Before". For the reference events, we attach each event to the immediately previous event in the text, and designate the relation as "Overlap", which is the most common relation between events in experts' annotation. As shown in Table 8, the neural ranking system achieves 0.66 labeled F1-score on the test data, compared with a baseline of 0.51. Table 8 also includes a breakdown of accuracy for different subtasks. The neural ranking model outperforms the baseline by a large margin for all subtasks. Overall, these results show that temporal dependency parsing is a very challenging task, and by making this data set available, it will aid in the development of more sophisticated machine learning models to advance the state of the art in this area.

		Unlabeled F		Labeled F	
		dev	test	dev	test
Baseline	te,te	0.80	0.82	0.80	0.82
	e,te	0.54	0.70	0.46	0.58
	e,e	0.64	0.61	0.26	0.34
	full	0.62	0.68	0.41	0.51
Neural	te,te	0.88	0.93	0.88	0.93
	e,te	0.62	0.77	0.53	0.66
	e,e	0.7	0.77	0.5	0.58
	full	0.69	0.79	0.55	0.66

Table 8: Experiment results of the baseline system and the neural ranking model.

7 Related Work

7.1 Temporal Dependency Structure

Kolomiyets et al. (2012) are the first work that use the term temporal dependencies, and they extract timelines from narrative stories as temporal dependency trees. However, in their work, only events are included as nodes in the dependency tree, and the parent of each node is not explicitly defined as the reference event of the child event. Zhang and Xue (2018b) first defined a temporal dependency tree structure that have both events and time expressions as nodes in the tree, and attempted

[6] https://github.com/yuchenz/tdp_ranking

to explicitly define the parent of each event or time expression as the reference event or time expression of the child node. This temporal dependency tree has been applied to both Chinese (Zhang and Xue, 2018b) and English (Zhang and Xue, 2019) data, and to both news reports and narrative stories, indicating this framework can be applied across languages and genres. The present work extends temporal dependency trees to the temporal dependency graphs, and crowd-sourced temporal dependency graphs on English news articles.

7.2 Crowdsourcing Temporal Relations

Early studies on crowdsourcing temporal relations usually focus on some subtask of this problem. Snow et al. (2008) crowdsources the relations of a subset of verb event pairs from TimeBank (Pustejovsky et al., 2003b) whose relations are either "strictly before" or "strictly after". Ng and Kan (2012) only focuses on the relation between events and time expressions from news data. Caselli et al. (2016) conducts crowdsourcing experiments on both temporal relation annotation and event / time expression extraction. In the time expression extraction experiments, they ask crowd workers to select time expressions directly from the raw text. In contrast, we give crowd workers time expression candidates and ask them binary questions. Our approach prevents crowd workers from selecting wrong textual spans. Ning et al. (2018) comes up with a multi-axis approach for event temporal relation annotation (see Ning et al., 2018 Section 2 and Appendix A for more details about their multi-axis model). The multi-axis approach is a way of factoring out modalities in event annotation, and combined with the decision to only consider the start point of events, they are able to achieve high accuracy in annotating temporal relations assuming gold events are provided. Our annotation is more challenging in that crowdworkers also need to identify time expressions and events, in addition to annotating temporal relations.

Zhang and Xue (2019) crowdsourced a temporal dependency tree (TDT) corpus, and is the work that is the most related to ours. The differences between their work and this work are as follows. First, our work extends the temporal dependency tree to temporal dependency graph, where events always have a reference timex and optionally also have a reference event. In TDT, events only have one reference time, either a reference timex or a reference event, but not both. The second difference is that the TDT corpus is constructed on top of TimeBank (Pustejovsky et al., 2003b), without having to annotate events and time expressions. In contrast, we construct the TDG corpus from scratch in that we first extract events and time expressions, then annotate the relations between them as part of the graph structure.

8 Conclusion

In this paper, we proposed a temporal annotation scheme called temporal dependency graphs which extend previous research on temporal dependency trees. The temporal dependency graphs, like temporal dependency trees, draw inspiration from previous research on narrative times and temporal anaphora, allow a good trade-off between completeness and practicality in temporal annotation. We proposed a crowdsourcing strategy and demonstrated its feasibility with a comparative analysis of the quality of the annotation. We also demonstrated the utility of the data set by training a neural ranking model on this data set, and the data set is publicly available.

Acknowledgments

We thank the anonymous reviewers for their helpful comments, Jayeol Chun and Yi Zhang for discussing the project with us. This work is supported in part by a grant from the IIS Division of National Science Foundation (Award No. 1763926) entitled "Building a Uniform Meaning Representation for Natural Language Processing" awarded to the fourth author. All views expressed in this paper are those of the authors and do not necessarily represent the view of the National Science Foundation.

This work was supported in part by DARPA/I2O and U.S. Army Research Office Contract No. W911NF-18-C-0003 under the World Modelers program, and the Office of the Director of National Intelligence (ODNI) and Intelligence Advanced Research Projects Activity (IARPA) via IARPA Contract No. 2019-19051600006 under the BETTER program. The views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies, either expressed or implied, of ODNI, IARPA, the Department of Defense or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

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

Jürgen Bohnemeyer. 2009. Temporal anaphora in a tenseless language. *The expression of time in language*, pages 83–128.

Tommaso Caselli, Rachele Sprugnoli, and Oana Inel. 2016. Temporal information annotation: Crowd vs. experts. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3502–3509, Portorož, Slovenia. European Language Resources Association (ELRA).

Taylor Cassidy, Bill McDowell, Nathanael Chambers, and Steven Bethard. 2014. An annotation framework for dense event ordering. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 501–506, Baltimore, Maryland. Association for Computational Linguistics.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar. Association for Computational Linguistics.

Erhard Hinrichs. 1986. Temporal anaphora in discourses of english. *Linguistics and philosophy*, pages 63–82.

Oleksandr Kolomiyets, Steven Bethard, and Marie-Francine Moens. 2012. Extracting narrative timelines as temporal dependency structures. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 88–97, Jeju Island, Korea. Association for Computational Linguistics.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Jun-Ping Ng and Min-Yen Kan. 2012. Improved temporal relation classification using dependency parses and selective crowdsourced annotations. In *Proceedings of COLING 2012*, pages 2109–2124, Mumbai, India. The COLING 2012 Organizing Committee.

Qiang Ning, Hao Wu, and Dan Roth. 2018. A multi-axis annotation scheme for event temporal relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1318–1328, Melbourne, Australia. Association for Computational Linguistics.

Barbara H Partee. 1984. Nominal and temporal anaphora. *Linguistics and philosophy*, pages 243–286.

Barbara Hall Partee. 1973. Some structural analogies between tenses and pronouns in english. *The Journal of Philosophy*, 70(18):601–609.

James Pustejovsky, José M Castano, Robert Ingria, Roser Sauri, Robert J Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R Radev. 2003a. Timeml: Robust specification of event and temporal expressions in text. *New directions in question answering*, 3:28–34.

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

James Pustejovsky and Amber Stubbs. 2011. Increasing informativeness in temporal annotation. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 152–160. Association for Computational Linguistics.

Hans Reichenbach. 1949. Elements of symbolic logic.

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

Roser Saurí, Jessica Littman, Bob Knippen, Robert Gaizauskas, Andrea Setzer, and James Pustejovsky. 2006. Timeml annotation guidelines. *Version*, 1(1):31.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. 2008. Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 254–263, Honolulu, Hawaii. Association for Computational Linguistics.

Jannik Strötgen and Michael Gertz. 2013. Multilingual and cross-domain temporal tagging. *Language Resources and Evaluation*, 47(2):269–298.

Naushad UzZaman, Hector Llorens, James F. Allen, Leon Derczynski, Marc Verhagen, and James Pustejovsky. 2012. Tempeval-3: Evaluating events, time expressions, and temporal relations. *CoRR*, abs/1206.5333.

Naushad UzZaman, Hector Llorens, Leon Derczynski, James Allen, Marc Verhagen, and James Pustejovsky. 2013. Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 1–9.

Bonnie Lynn Webber. 1988. Tense as discourse anaphor. *Computational Linguistics*, 14(2):61–73.

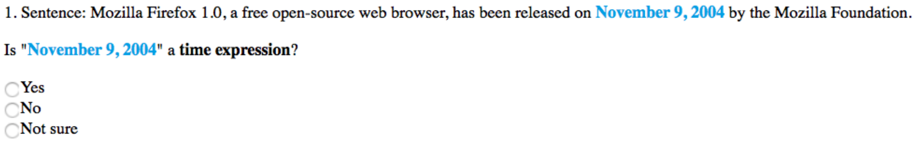
Yuchen Zhang and Nianwen Xue. 2018a. Neural ranking models for temporal dependency structure parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3339–3349, Brussels, Belgium. Association for Computational Linguistics.

Yuchen Zhang and Nianwen Xue. 2018b. Structured interpretation of temporal relations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Yuchen Zhang and Nianwen Xue. 2019. Acquiring structured temporal representation via crowdsourcing: A feasibility study. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 178–185, Minneapolis, Minnesota. Association for Computational Linguistics.

A Appendix

In this section, we give examples of the annotation interface of each step. In step 4 and 5, the number of options we gave to crowd workers is varied. The template we have here for step 5 has the maximum number of options. The options are also ranked by likelihood: the first event is usually the most likely choice.

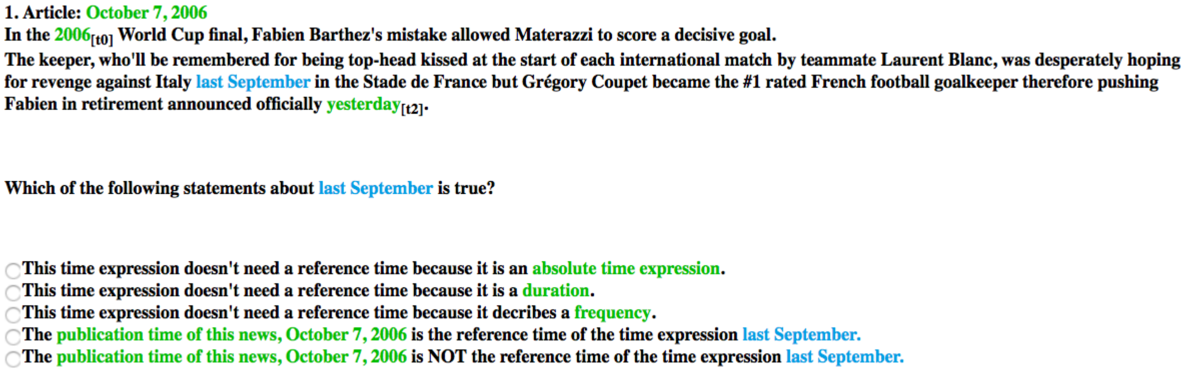


1. Sentence: Mozilla Firefox 1.0, a free open-source web browser, has been released on November 9, 2004 by the Mozilla Foundation.

Is "November 9, 2004" a **time expression**?

○ Yes
○ No
○ Not sure

Figure 4: Annotation interface for time expression identification.

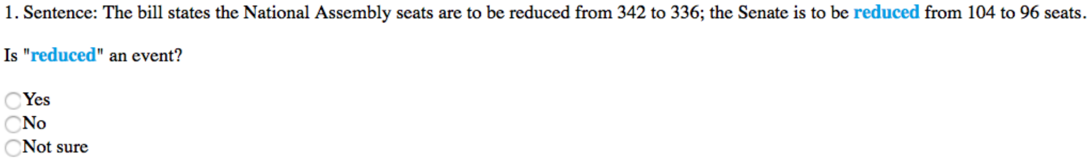


1. Article: October 7, 2006
In the 2006[t0] World Cup final, Fabien Barthez's mistake allowed Materazzi to score a decisive goal.
The keeper, who'll be remembered for being top-head kissed at the start of each international match by teammate Laurent Blanc, was desperately hoping for revenge against Italy last September in the Stade de France but Grégory Coupet became the #1 rated French football goalkeeper therefore pushing Fabien in retirement announced officially yesterday[t2].

Which of the following statements about last September is true?

○ **This time expression doesn't need a reference time because it is an absolute time expression.**
○ **This time expression doesn't need a reference time because it is a duration.**
○ **This time expression doesn't need a reference time because it decribes a frequency.**
○ **The publication time of this news, October 7, 2006 is the reference time of the time expression last September.**
○ **The publication time of this news, October 7, 2006 is NOT the reference time of the time expression last September.**

Figure 5: Annotation interface for reference time resolution for time expressions.

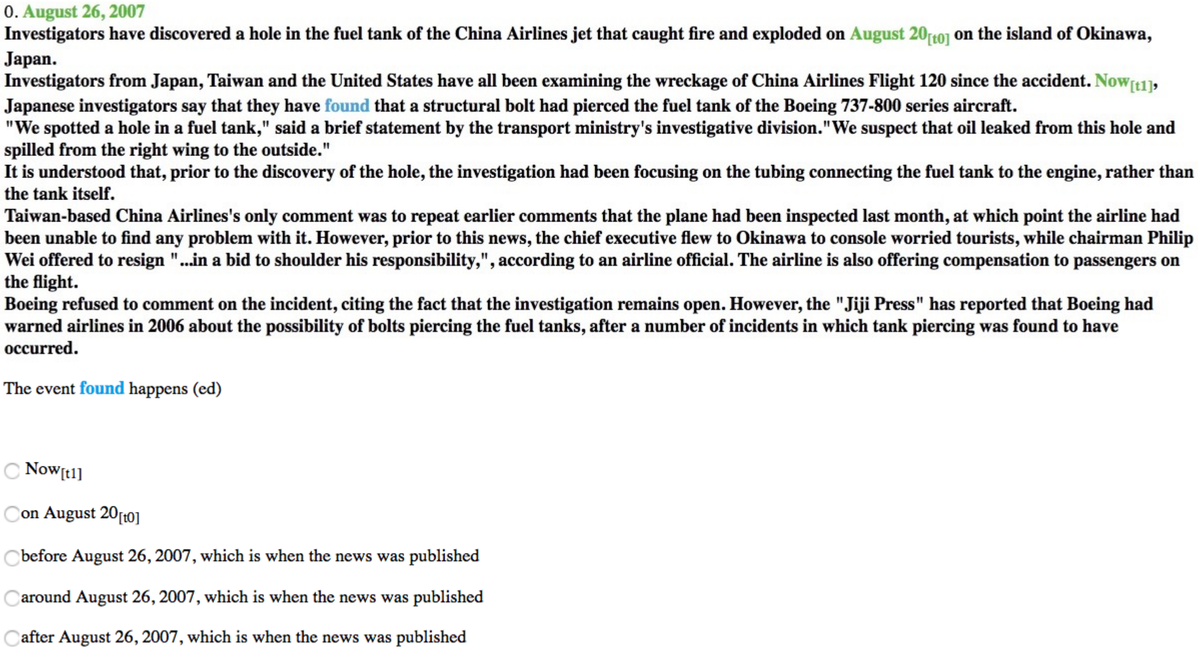


1. Sentence: The bill states the National Assembly seats are to be reduced from 342 to 336; the Senate is to be **reduced** from 104 to 96 seats.

Is "**reduced**" an event?

○ Yes
○ No
○ Not sure

Figure 6: Annotation interface for event identification.



0. **August 26, 2007**
Investigators have discovered a hole in the fuel tank of the China Airlines jet that caught fire and exploded on August 20[t0] on the island of Okinawa, Japan.
Investigators from Japan, Taiwan and the United States have all been examining the wreckage of China Airlines Flight 120 since the accident. Now[t1], Japanese investigators say that they have found that a structural bolt had pierced the fuel tank of the Boeing 737-800 series aircraft.
"We spotted a hole in a fuel tank," said a brief statement by the transport ministry's investigative division."We suspect that oil leaked from this hole and spilled from the right wing to the outside."
It is understood that, prior to the discovery of the hole, the investigation had been focusing on the tubing connecting the fuel tank to the engine, rather than the tank itself.
Taiwan-based China Airlines's only comment was to repeat earlier comments that the plane had been inspected last month, at which point the airline had been unable to find any problem with it. However, prior to this news, the chief executive flew to Okinawa to console worried tourists, while chairman Philip Wei offered to resign "...in a bid to shoulder his responsibility,", according to an airline official. The airline is also offering compensation to passengers on the flight.
Boeing refused to comment on the incident, citing the fact that the investigation remains open. However, the "Jiji Press" has reported that Boeing had warned airlines in 2006 about the possibility of bolts piercing the fuel tanks, after a number of incidents in which tank piercing was found to have occurred.

The event **found** happens (ed)

○ Now[t1]

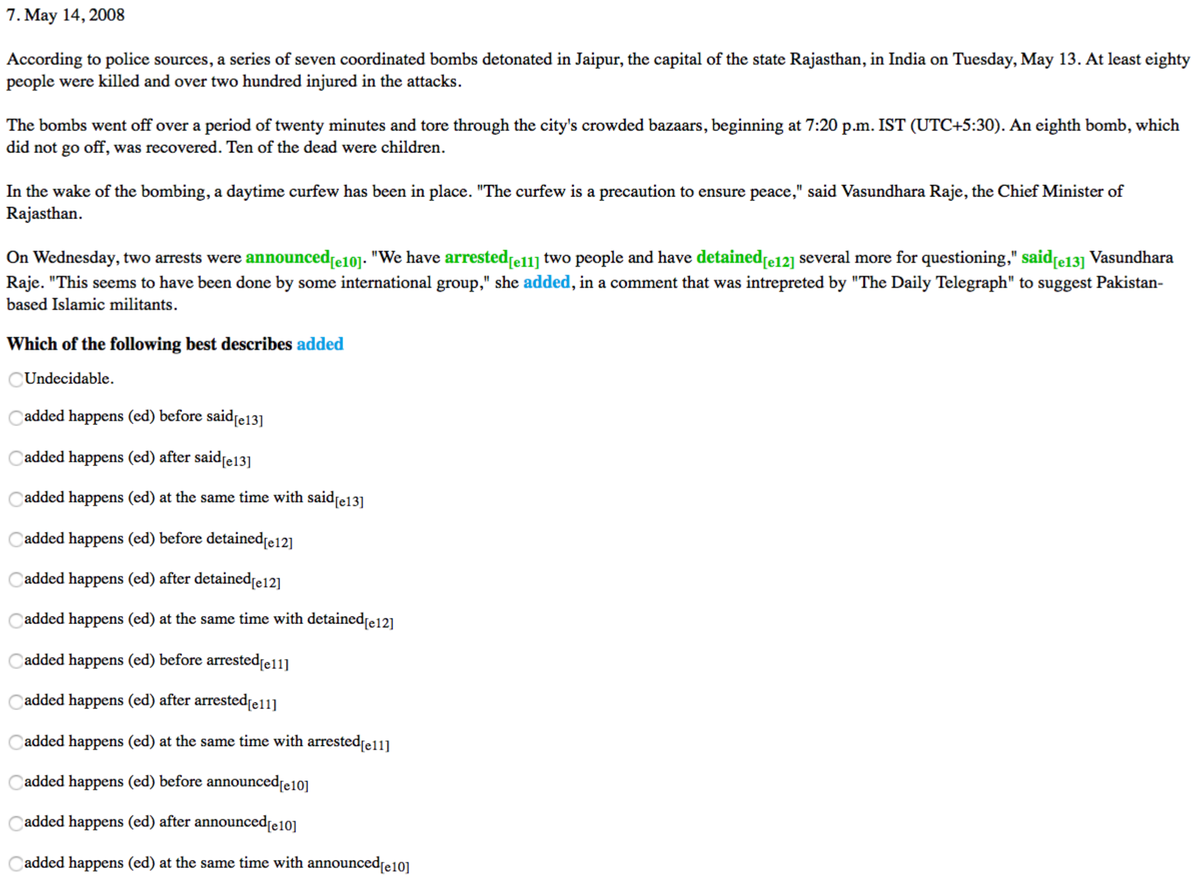
○ on August 20[t0]

○ before August 26, 2007, which is when the news was published

○ around August 26, 2007, which is when the news was published

○ after August 26, 2007, which is when the news was published

Figure 7: Annotation interface for resolving reference timex for events.



7. May 14, 2008

According to police sources, a series of seven coordinated bombs detonated in Jaipur, the capital of the state Rajasthan, in India on Tuesday, May 13. At least eighty people were killed and over two hundred injured in the attacks.

The bombs went off over a period of twenty minutes and tore through the city's crowded bazaars, beginning at 7:20 p.m. IST (UTC+5:30). An eighth bomb, which did not go off, was recovered. Ten of the dead were children.

In the wake of the bombing, a daytime curfew has been in place. "The curfew is a precaution to ensure peace," said Vasundhara Raje, the Chief Minister of Rajasthan.

On Wednesday, two arrests were **announced**$_{[e10]}$. "We have **arrested**$_{[e11]}$ two people and have **detained**$_{[e12]}$ several more for questioning," **said**$_{[e13]}$ Vasundhara Raje. "This seems to have been done by some international group," she **added**, in a comment that was intrepreted by "The Daily Telegraph" to suggest Pakistan-based Islamic militants.

Which of the following best describes added

○ Undecidable.

○ added happens (ed) before said$_{[e13]}$

○ added happens (ed) after said$_{[e13]}$

○ added happens (ed) at the same time with said$_{[e13]}$

○ added happens (ed) before detained$_{[e12]}$

○ added happens (ed) after detained$_{[e12]}$

○ added happens (ed) at the same time with detained$_{[e12]}$

○ added happens (ed) before arrested$_{[e11]}$

○ added happens (ed) after arrested$_{[e11]}$

○ added happens (ed) at the same time with arrested$_{[e11]}$

○ added happens (ed) before announced$_{[e10]}$

○ added happens (ed) after announced$_{[e10]}$

○ added happens (ed) at the same time with announced$_{[e10]}$

Figure 8: Annotation interface for resolving reference events for events.