Temporal Reasoning on Implicit Events from Distant Supervision

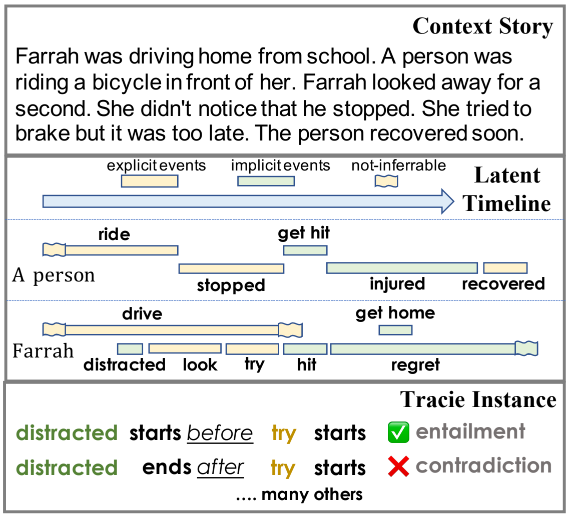
Ben Zhou[1,2] **Kyle Richardson**[1] **Qiang Ning**[3] **Tushar Khot**[1] **Ashish Sabharwal**[1] **Dan Roth**[2]
*[1]Allen Institute for AI [2]University of Pennsylvania [3]Amazon
{kyler,tushark,ashishs}@allenai.org {xyzhou,danroth}@cis.upenn.edu qning@amazon.com

Abstract

We propose TRACIE, a novel temporal reasoning dataset that evaluates the degree to which systems understand *implicit* events—events that are not mentioned explicitly in natural language text but can be inferred from it. This introduces a new challenge in temporal reasoning research, where prior work has focused on explicitly mentioned events. Human readers can infer implicit events via commonsense reasoning, resulting in a more comprehensive understanding of the situation and, consequently, better reasoning about time. We find, however, that state-of-the-art models struggle when predicting temporal relationships between implicit and explicit events. To address this, we propose a neuro-symbolic temporal reasoning model, SYMTIME, which exploits distant supervision signals from large-scale text and uses temporal rules to combine start times and durations to infer end times. SYMTIME outperforms strong baseline systems on TRACIE by 5%, and by 11% in a zero prior knowledge training setting. Our approach also generalizes to other temporal reasoning tasks, as evidenced by a gain of 1%-9% on MATRES, an explicit event benchmark.

1 Introduction

Understanding temporal relations between events in narrative text is a crucial part of text understanding. When reading a story, a human can construct a latent timeline about events' start and end times, similar to the one shown in Fig. 1 about an automobile accident. This timeline not only contains the placements of explicitly mentioned events (e.g., *ride a bicycle*), but also accounts for implicit events (e.g., Farrah was *distracted* so she looked away). Such a latent timeline explains the dynamics between events; for example, the possible chain of events between *ride* and *recovered* in this context contains *get hit* and *injured*. The ability to construct such a timeline is essential for understanding the causal dynamics of a situation. Without it, NLP systems cannot truly understand situations and reliably solve tasks such as temporal question-answering, causal inference, and scheduling assistance.


Context Story

Farrah was driving home from school. A person was riding a bicycle in front of her. Farrah looked away for a second. She didn't notice that he stopped. She tried to brake but it was too late. The person recovered soon.

distracted starts *before* try starts ✅ entailment

distracted ends *after* try starts ❌ contradiction

.... many others

Figure 1: A story, its latent timeline, and example TRACIE instances from it. For simplicity, events are shortened to single verbs and the timeline is exaggerated.

To better evaluate this ability, we introduce a new dataset called TRACIE (*TempoRAl Closure InfErence*) that focuses on temporal relations on implicit events in short stories. Our dataset contains high-quality annotations of both start and end time queries that test a system's understanding of the full temporal closure (i.e., both start and end time) of events. As a task that requires considerable commonsense knowledge, we follow Zhou et al. (2020) in minimizing the size of the training set, therefore making TRACIE mainly an evaluation set. The final TRACIE dataset contains a total of 5.4k human-curated instances, provided in a (multi-premise) textual entailment (TE) format, as illustrated at the bottom of Fig 1. A Pre-trained language model such as T5-Large (Raffel et al., 2020)

[*] Most of the work was done when the third author was employed at the Allen Institute for AI and the first author was an intern there.

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1361–1371
June 6–11, 2021. ©2021 Association for Computational Linguistics

fine-tuned on our new dataset achieves a modest binary prediction accuracy of 67.9%.[1] Consistent with other studies on temporal reasoning (Zhou et al., 2020), these results reveal serious limitations in existing pre-trained language models.

To build models better capable of understanding time with minimal direct training data, we propose a novel distant supervision technique that improves generalization by extracting temporal patterns in large-scale free text as part of an additional pre-training step. In contrast to other attempts at extracting temporal data through patterns at a sentence level (Gusev et al., 2011; Zhou et al., 2020), we extract over large windows of text such as paragraphs. This allows for capturing global information related to multiple events and extracting signals that do not appear in small-window local contexts. The resulting model, PTNTIME (Pattern-Time), achieves a 76.6% accuracy on TRACIE, a 9% gain over using standard T5-Large. We also show the applicability of PTNTIME on a standard temporal reasoning benchmark involving only explicit events, MATRES (Ning et al., 2018b), with a 9 point gain in a low-resource setting.

We achieve further improvements by coupling PTNTIME with a duration model from Zhou et al. (2020) to create a neural-symbolic reasoning model called SYMTIME. The key idea in SYMTIME is to *decompose* the computation of temporal relations to the predictions of relative distances between start times and those of durations. For example, in Fig 1, we can decide that *distracted* likely ends before *try* starts because the duration of *distracted* is likely to be shorter than the distance between the two start times. This allows for better prediction on the end time, which rarely appears in the natural text and has been previously shown to be difficult to annotate (Ning et al., 2018b). Such a symbolic computation involves a logical combination of the individual models in a way that formalizes part of the Allen interval algebra (Allen, 1983). This model, which supports a wider range of temporal computation and can be used with and without task-specific supervision, achieves a final accuracy of 78.9% on TRACIE's binary classification metric. We also show that SYMTIME is more robust to different distributions of the training data, demonstrating the benefits of using a temporal model with a transparent reasoning process.

In summary, we make the following 3 contributions: (1) a temporal relation dataset TRACIE focusing on implicit events (§3); (2) a distant supervision process for temporal understanding of implicit events (§4); and (3) a reasoning model that makes end-time comparisons using predictions of start-time distances and durations (§5). Finally, we demonstrate the effectiveness of our models on TRACIE, as well as the applicability of our approach to an existing temporal benchmark (§6).

2 Related Work

Temporal reasoning has received much attention in the NLP community, and to date, there are many datasets that focus on temporal ordering (Pustejovsky et al., 2003; Bethard et al., 2007; Cassidy et al., 2014; Reimers et al., 2016; O'Gorman et al., 2016; Ning et al., 2018b, 2020b), and other temporal knowledge (Pan et al., 2006; Zhou et al., 2019). We focus here on modeling implicit events, which has received relatively little attention. Multiple systems have been proposed as part of research into temporal ordering (Do et al., 2012; Moens and Leeuwenberg, 2017; Leeuwenberg and Moens, 2018; Meng and Rumshisky, 2018; Ning et al., 2018c; Han et al., 2019), duration prediction (Vashishtha et al., 2019) and other tasks. Our decision to use a textual entailment style follows recent work on natural language inference (Williams et al., 2017; Nie et al., 2020; Bhagavatula et al., 2020), which tends to not focus on time (for recent work on temporal NLI, see Vashishtha et al. (2020)). Many have used distant supervision for temporal reasoning (Gusev et al., 2011; Ning et al., 2018a; Zhou et al., 2020). Comparatively, our work captures longer-range dependencies in narrative text (for related ideas, see Ammanabrolu et al. (2021)).

We are inspired by structural predictions and constraints that combat the sparsity of temporal knowledge (Ning et al., 2017; Do et al., 2012), as well as neural module networks (Andreas et al., 2016; Gupta et al., 2019) and other decomposition-based approaches (Talmor and Berant, 2018; Khashabi et al., 2018; Li et al., 2019; Wolfson et al., 2020; Khot et al., 2021). In particular, we build neural-symbolic transformer models that operationalize some of the classical interval-based computations used in earlier work on temporal reasoning (Allen, 1983; Gerevini and Schubert, 1995) (for related ideas, compare with Leeuwenberg and Moens (2018); Vashishtha et al. (2019)).

[1] The same model achieves 77.4% on MATRES (Ning et al., 2018b) with a similar amount of training instances. All TRACIE numbers reported in this section are from Table 2.

Context Story (Premise)	**Hypothesis**	**Inference Label**
Tom needed to get braces. He was afraid of them. The dentist assured him everything would be fine. Tom had them on for a while. Once removed he felt it was worth it.	Tom avoids foods he can't eat with braces `starts` `before` the braces are removed.	`entailment`
We were all watching Spongebob as a family. It is a kid's show but all really enjoyed it. This one episode was especially funny for the adults. It has humor in it that is funny for kids and adults. It is something we can all watch...	The adults laughed at the jokes `ends` `before` we watch Spongebob as a family	`contradiction`
I was throwing the baseball with my son. He threw one past me that landed in the lake. I reached in to get the ball. I lost my balance and fell in. I got the ball and a bath all in one shot!	The ball was in the boys hand `starts` `after` he reached for the ball	`contradiction`

Figure 2: Example TRACIE instances. The **comparator** l ∈{`starts`,`ends`} and **relation** r ∈{`before`,`after`} in each hypothesis are highlighted, in addition to the corresponding explicit event from the story.

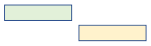
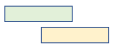
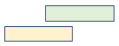
Illustration	**Allen's Relation**	**Tracie's Relation**
	Precedes, Meets	Starts Before Ends Before
	Overlaps, Finished-by, Contains, Starts, Equals, Started-by	Starts Before Ends After
	During, Finishes, Overlapped-by, Met-by, Preceded-by	Starts After Ends After

Figure 3: TRACIE's label definition and its relation to Allen's interval algebra, with a graph illustration between an implicit event and an explicit event.

This work is broadly related to works on causal dynamics (Pearl, 2009). The nature of combined temporal and causal focuses is also related to procedural text modeling (Tandon et al., 2018, 2020).

3 The TRACIE Dataset

In this section, we introduce the TRACIE dataset.[2]

3.1 Task Overview and Dataset Construction

The goal of TRACIE is to test a system's ability to compare start and end times of non-extractive implicit event phrases instead of extractive triggers from the context. Such tests in TRACIE take the form of multi-premise textual entailment (TE) (Lai et al., 2017). Each TRACIE instance contains 1) a **context story** (or premise) consisting of a sequence of *explicit* narrative events; 2) an **implicit event** in the form of a natural language phrase that is unmentioned but has some role in the story; 3) a **comparator** of either {`starts`, `ends`}; 4) an **explicit event** also in the form of a phrase, and 5) a **temporal relation** of either {`before`, `after`} that marks the relationship in the dimension defined by the *comparator* between the *implicit-event* and the *explicit-event*. With these 4 components, we are able to generate TE-style instances, using the context story as the premise and temporal queries about pair-wise relations between implicit and explicit events as hypotheses. For example, in the first positive instance shown in Fig. 1, "distracted" is the *implicit-event*, "starts" is the *comparator*, "try" is *explicit-event* and "before" is the *temporal-relation*. They form a positive hypothesis "distracted starts before try."[3] We flip the *temporal-relation* (i.e., "before" to "after" and vice versa) to create negative (contradiction) instances, as shown in the second example instance in Fig. 1.

Since the start times of *explicit-events* are more obvious to human annotators, we use them as reference points and compare the *implicit-event*'s start or end time with them (depending on the *comparator*), according to the label definitions shown in Fig. 3. In rare cases where two time points are the same (e.g., *hit* and *get hit* start at the same time in Fig.1), we use the causal relation to decide the order, so that *hit* starts before *get hit*. Such instances are created through a multi-stage annotation process as detailed (in respective order) below. All steps are implemented with the CrowdAQ platform (Ning et al., 2020a) with qualification exams.

Implicit Event Generation We randomly sample short stories from the ROCStories dataset (Mostafazadeh et al., 2016). For each story, one annotator writes 5 implicit event phrases that are not explicitly mentioned by the given story, but are inferable and relevant. The annotator additionally rewrites two explicit events closest to the implicit event's start and end time, respectively. With these two events, we can build two TRACIE instances (minus the *temporal-relation*) per implicit event, which accounts for 10 instances in total per story.

[2] We release TRACIE and its leaderboard at https://leaderboard.allenai.org/tracie

[3] All event phrases are shortened to triggers here for simplicity. See Fig. 2 for actual phrases.

Automatic Instance Generation We use AllenNLP (Gardner et al., 2018) to extract all verbs and relevant arguments with its semantic role labeling (SRL) model. With all the verbs and their arguments, we construct a pool of explicit events in the form of short phrases. For each implicit event, we randomly select two {*explicit-event*, *comparator*} pairs from the pool and build 10 additional instances (without *temporal-relation*).

Label Collection For each of the 20 instances per story, we annotate the *temporal-relation* with four different annotators. Annotators follow the label definition in §3.1 to produce four *temporal-relations* for each instance. We use the majority agreement as the final label and filter out unagreeable instances. Two authors additionally verify the instances with ambiguous verbs (e.g., "have") and corrected 5% of the end-time instances.

3.2 Splits and Analysis

We split the data under the independent and identically distributed (i.i.d.) assumption based on stories, with a 20/80 train/test ratio. We use a small training set, following Zhou et al. (2019), as we believe temporal relations involve much commonsense knowledge. As we later show in §6.3, it is infeasible to collect a large enough human-annotated training set to capture all the knowledge needed to tackle this problem completely, and a system must acquire knowledge from external resources. As a result, we use a small training set just to define the task, and at the same time, use an extensive testing set for more robust evaluation.

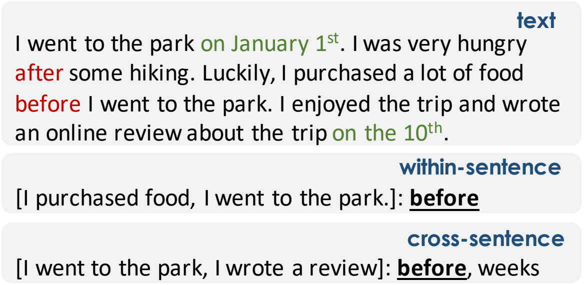
The authors conduct a human upper-bound analysis on 100 randomly sampled instances, following the procedure in Zhou et al. (2020). There is a 94% agreement and a 98% resolved accuracy,[4] suggesting that TRACIE has a high annotation quality.

4 Pattern-Based Pre-Training

As argued in §3.2, we believe that it is more efficient to build a model that learns the prior knowledge needed for the task with distant signals and only subsequently learns the task definition through a small training set. This section describes how we collect the distant signals related to events' start-time comparisons and pre-train a novel *temporally-aware* transformer model called PTNTIME. While PTNTIME will be used for fine-tuning directly on TRACIE, it will also form the basis of a more general temporal reasoning model called SYMTIME that we describe in §5.

4.1 Distant Supervision Collection

We describe the sources of distant supervision signals with the goal of understanding the relative order between two events' start times as well as the relative distance between them.



text

I went to the park on January 1st. I was very hungry after some hiking. Luckily, I purchased a lot of food before I went to the park. I enjoyed the trip and wrote an online review about the trip on the 10th.

within-sentence

[I purchased food, I went to the park.]: **before**

cross-sentence

[I went to the park, I wrote a review]: **before**, weeks

Figure 4: Extraction for start-time comparisons applied to an example paragraph.

Within-Sentence Extraction We collect start time comparisons between pairs of events heuristically from free-text using "before/after" keywords (following much prior work in temporal modeling and extraction (Do et al., 2012)). We use AllenNLP's SRL model to process each input sentence and find verbs with a temporal argument that starts with either "before" or "after", and contains at least another verb. If there are multiple verbs in the temporal argument, we take the one with the largest number of tokens as arguments. We match the two extracted verbs with the relation indicated by the first word of either "before" or "after". As the example in Fig. 4 shows, the extractor identifies that *purchase food* is before *go to park* as indicated by the "before" keyword mentioned in the text. We acquire 2.8 million instances from the May 2020 Wikipedia dump using this process.

Cross-Sentence Extraction The data collected from the within-sentence patterns does not reveal the relative distance between two start times. In addition, because writers often save trivial inferences for efficiency, certain event pairs rarely co-occur within a small textual window, making one event often implicit to the other one in these pairs. To better collect such signals, we employ a cross-sentence extraction that finds direct temporal expressions of hours and dates. Because these temporal expressions (e.g., `2021-01-01`) are globally comparable,

[4] This is obtained after the authors discuss and resolve any disagreements before comparing with the annotated labels.

the compared events can be anywhere in a document. Therefore, this process collects more supervision signals about *time-point comparisons* and their *relative distance* on event pairs with trivial causal relations. We apply the SRL model and find all temporal arguments and their associated verbs. We find the exact temporal values by filling unmentioned elements of a temporal expression with the nearest previous mention (e.g., we add "January" to the expression of "the 10th" in Fig. 4.) These extractions have high precision, as the SRL model does well on identifying temporal arguments.

We then construct supervision instances under the assumption that the extracted temporal expressions describe the start times of the associated verbs (e.g., *went* started on *January* 1^{st} in Fig. 4) . Each instance comprises an event pair, a temporal relation, and an estimation on the temporal difference between the two start times. Each event is a phrase constructed by taking all relevant arguments of the predicate verb in the SRL parses. We represent the differences between the two start times as one of seven coarse temporal units: {≤minutes, hours, days, weeks, months, years, ≥decades}. For example, we get *go to park* is **weeks** before *write review* as shown in Fig. 4. In addition to the event pairs, we randomly sample sentences within the paragraph to use as the context that better defines the events. We collect 700k instances from this cross-sentence extraction process from Wikipedia.

Language Model (LM) Pre-Training Data We couple the specialized temporal pre-training data described above with additional paragraphs that are used to perform conventional language model pre-training using the original denoising task proposed in Raffel et al. (2020). This is done to maintain part of the original language model's semantics and to avoid overfitting. We use the Gutenberg Dataset (Lahiri, 2014) as the source and collect 1 million paragraphs for this purpose.

Data Format We then format the within / cross-sentence extraction data to consistent instances that have input sequences of `event:[EventA] starts [Relation][EventB].story:[Paragraph]` and output sequences of `answer:[Label][Distance]`. Here `[EventA]` represents the tokens that describe the first event; `[EventB]` represents the ones that describe the second event; and `[Paragraph]` represents the tokens of the context, which is non-empty only for cross-sentence extractions. `[Relation]` is either `before` or `after`, and `[Label]` is either `positive` or `negative`. When the label is positive, the relation will be the gold relation extracted from the text; when it is negative, the relation will be the inverse of the extracted relation. We randomly make 50% of the instances negative. `[Distance]` is one of the 7 coarse temporal units represented with a set of blank tokens `[extra_id_N]`. We leave it to be blank for the within-sentence extractions so that the objective function will not include it in loss computations. The LM pre-training data follows the original format in Raffel et al. (2020).

4.2 Pattern-Based Temporal Model (PTNTIME)

We use a pre-trained sequence-to-sequence model as our base model and additionally pre-train this model using the data collected in §4.1 (for modeling details, see §6.1). We call the resulting model PTNTIME. As a result of this additional pre-training step, PTNTIME serves as new set of *temporally-aware* model weights that can be used in place of existing pre-trained models and fine-tuned on TRACIE. As we describe next, we also use PTNTIME to build a modular temporal reasoning model called SYMTIME that attempts to go beyond a standard language modeling approach and improve start and end point prediction.

5 Symbolic Temporal Reasoning Model (SYMTIME)

To address the challenge of predicting event end times for which it is difficult to obtain high-quality direct or distant supervision, we introduce a new reasoning model called SYMTIME in this section. This model makes end-time comparisons by symbolically combining start time distance and duration from separate predictions based on some of the components introduced in the previous section. Different from Leeuwenberg and Moens (2018) and Vashishtha et al. (2019), our model does not rely on explicit annotations on timepoints, but only relative comparisons between them.

5.1 Formulation

As described in §3.1, hypotheses in TRACIE make pair-wise comparisons between two events e_1 and e_2 using a *comparator* l from $\{\texttt{starts}, \texttt{ends}\}$ and a *query-relation* r from $\{\texttt{before}, \texttt{after}\}$ based on a provided story context. We associate each e_j with a latent start time $\mathbf{start}_j$ and an end

comparator l	relation $r_l(\mathbf{e}_1, \mathbf{e}_2)$=	
ends	**before**	if $\mathbf{end}_1 < \mathbf{start}_2$
	after	*otherwise*
starts	**before**	if $\mathbf{start}_1 < \mathbf{start}_2$
	after	*otherwise*

Figure 5: Decomposition of the relation functions that solve TRACIE instances (equal timepoints ignored).

time $\mathbf{end}_j$, as well as, for convenience, a duration $\mathbf{duration}_j = \mathbf{end}_j - \mathbf{start}_j$. Under this formulation, a symbolic approach to solving TRACIE involves computing the *relation functions* r_l shown in Figure 5. For example, given exact numeric values $\mathbf{end}_1$ and $\mathbf{start}_2$, as one would assume in a classical interval-based approach to temporal reasoning (Allen, 1983)[5], determining if the first event *ends before* the second involves simply computing whether $\mathbf{end}_1$ is less than $\mathbf{start}_2$.

Given that the exact values of start and end times are latent, we use the intervals to do the same comparisons, as they are more context-invariant. For example, we do not need the exact date to know that *lunch* starts before *dinner* in the same day, because there is a typical distribution of the relative distance between the two start times. Based on this idea, we build a neural-symbolic model that learns approximations of these simple functions in Fig. 5 in a differentiable way. Specifically, we use individual neural modules that make predictions about event intervals via distance and duration functions $\mathrm{dist}(e_i, e_j)$ and $\mathrm{dur}(e_j)$, respectively.

To understand this decomposition, we define the distance and duration functions computed by these two modules as $\mathrm{dist}(e_i, e_j) = \mathbf{start}_i - \mathbf{start}_j$ and $\mathrm{dur}(e_j) = \mathbf{duration}_j$. By exploiting the rule that an end point $\mathbf{end}_j$ can be computed as $\mathbf{end}_j = \mathbf{start}_j + \mathbf{duration}_j$, we can, for example, decompose the relation $r_{ends}(e_1, e_2) = \mathbf{before}$ (i.e., e_1 *ends before* e_2) in terms of our two modules as follows via simple algebraic manipulation:

$$\begin{aligned} & r_{ends}(e_1, e_2) = \mathbf{before} \\ \Leftrightarrow\ & \mathbf{end_1} < \mathbf{start_2} \\ \Leftrightarrow\ & \mathbf{start_1} + \mathbf{duration_1} < \mathbf{start_2} \\ \Leftrightarrow\ & \left(\mathbf{start_1} - \mathbf{start_2}\right) + \mathbf{duration_1} < 0 \\ \Leftrightarrow\ & \mathrm{dist}(e_1, e_2) + \mathrm{dur}(e_1) < 0 \end{aligned}$$

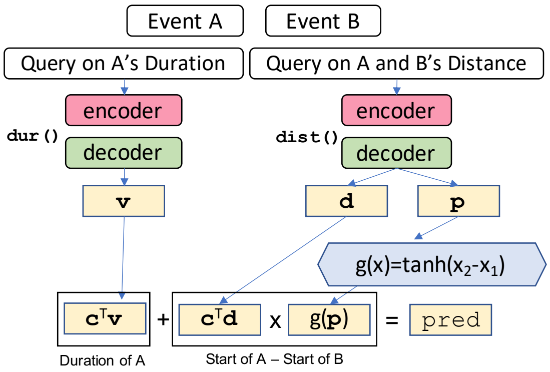


Figure 6: A schematic overview of SYMTIME to compare event A's end time with event B's start time via modular predictions about A's duration and distance from B and their symbolic combination (bottom).

Hence, we have reduced the computation of the relation *ends before* to a *symbolic computation* over two numeric intervals. Conversely, we have $r_{ends}(e_1, e_2) = \mathbf{after} \Leftrightarrow \mathrm{dist}(e_1, e_2) + \mathrm{dur}(e_1) > 0$,[6] For the **starts** comparator, we have $r_{starts}(e_1, e_2) = \mathbf{before} \Leftrightarrow \mathrm{dist}(e_1, e_2) < 0$ and vice versa for the **after** relation.

In what follows, we describe how we approximate the values of the two functions via individual neural modules (see illustration in Fig. 6).

5.2 Duration Estimation

To obtain a model to estimate $\mathrm{dur}(\cdot)$, we pre-train a sequence-to-sequence model with the duration data from Zhou et al. (2020), which is similarly collected from pattern-based extraction. The data contains over 1 million events with their corresponding duration values. We map each instance to an input sequence `event:[Event]story:[Story]` and a corresponding output sequence `answer:[Value]`, where `[Event]` represents the tokens of an event with the trigger verb marked by a special token to its left, `[Story]` represents down-sampled tokens from the context, and `[Value]` is one of the 7 unit labels as described in §4.1 (i.e., { ≤minutes, hours, days, weeks, months, years, ≥decades }).

5.3 Computation and Learning

We use the output from PTNTIME to approximate the function $\mathrm{dist}(\cdot)$. Following the sequence formulation of PTNTIME in §4, we replace `[EventA]` with the textual description of e_1, `[EventB]` with

[5] In the Allen algebra, the values $\mathbf{end}_x$ and $\mathbf{start}_y$ correspond to the right and left end points x^+, y^- in the intervals $(x^-, x^+), (y^-, y^+)$. Likewise, our $\mathbf{duration}_x$ corresponds to the value $(x^+ - x^-)$.

[6] We note that one drawback of this inference rule is that it does not predict causal relations and, therefore, cannot handle instances where $\mathbf{end_1} = \mathbf{start_2}$ as our label definitions describe in §3.1. We leave this problem for future research.

the textual description of e_2, and `[Paragraph]` with the context (premise), and fix `[Relation]` to be *before*. By taking the values of the vocabulary indices corresponding to "positive" and "negative" from the logits of `[Label]` and applying a $\mathrm{softmax}$ operation, we get P_{before} and P_{after}. These are the probability of e_1 starting before and after e_2, respectively, and are used to define the vector $\mathbf{p} = [P_{\text{before}}, P_{\text{after}}]$. Similarly, we apply $\mathrm{softmax}$ to the logits of `[Distance]` over the 7 words representing the temporal units to obtain 7 values that approximate the probabilities of the distance between two events' start times being closest to each temporal unit. We place the 7 values in temporal units' increasing order in vector $\mathbf{d}$. To represent $|\mathbf{start_1} - \mathbf{start_2}|$ with a single value, we dot product the probabilities with an incremental constant vector $\mathbf{c} = [0, 1, 2, 3, 4, 5, 6]$. To get the direction, we apply the $\tanh$ function to the difference between the probabilities in $\mathbf{p}$.[7] As a result, we have:

$$\begin{aligned} \mathrm{dist}(\cdot) &= \mathbf{start_1} - \mathbf{start_2} \\ &= \mathbf{c}^T\mathbf{d} * \tanh(\mathrm{INT}_{max} * (\mathbf{p}_2 - \mathbf{p}_1)) \end{aligned} \quad (1)$$

We use the pre-trained model in §5.2 to approximate the function $\mathrm{dur}(\cdot)$. Because the model is pre-trained with markers to the left of trigger verbs, we run a part-of-speech tagger on input phrases and add a marker to the left of the first verb. We apply $\mathrm{softmax}$ to the logit values of `[Value]` over the 7 temporal unit words and get, as above, 7 values representing the probabilities of the input event's duration being closest to each unit. We form $\mathbf{v}$ by placing these values at the temporal unit's increasing order. With the same constant vector, we have:

$$\mathrm{dur}(\cdot) = \mathbf{duration_1} = \mathbf{c}^T\mathbf{v} \quad (2)$$

For hypotheses with comparator `starts`, we use PtnTime and its sequence-to-sequence objective to learn (i.e., we take the input hypothesis and context as is and use `[Label]` directly as the prediction). For hypotheses where the comparator is `ends`, we use the inference process in §5.1 and the computation process described above to construct $logits = [\mathbf{pred}, -\mathbf{pred}], \mathbf{pred} = \mathrm{dist}(e_1, e_2) + \mathrm{dur}(e_1)$ as detailed in Fig. 6. We find the *gold-temporal-relation* in each training instance and compute a two-class cross-entropy loss with *logits*. The PtnTime that predicts `starts` hypotheses shares weights with the one used in computing *logits*. The final model SymTime can also be used to predict Tracie instances without any task-specific supervision as the two functions are initialized with distant supervision.

6 Experiments

In this section, we detail our experimental setup (§6.1-6.2) and report our main results (§6.3-6.5).[8]

6.1 Baselines and Systems

We use T5-Large implemented by Wolf et al. (2019) as our base sequence-to-sequence model for both PtnTime and the duration model in §5.2 as it provides for faster iterations. We use early stopping, batch size of 32 and other default parameters. PtnTime converges after 45k steps (∼1.4M instances) and the duration model converges after 80k steps (∼2.6M instances). We use these pre-trained weights in SymTime as well as SymTime-ZeroShot which uses no Tracie supervision.

We compare with our proposed models with a host of baselines based on the same pre-trained language model, including **BaseLM**: T5-Large, and **BaseLM-MATRES**: T5-Large fine-tuned on 20k MATRES training data. We also compare with other architectures/models, including **BiLSTM** as used in Williams et al. (2017), **Roberta-Large** (Liu et al., 2019) and **T5-3B**. All models and baselines follow a standard TE setup and default parameters. We report a 3-run average and each model is run until convergence.

6.2 Metrics and Settings

We measure system performance on Tracie separately for start-time hypotheses and end-time hypotheses. We also employ a story-wide exact match metric, which is the percentage of stories with all its related hypotheses answered correctly.

In addition to Tracie's standard i.i.d. split, we propose a pruned version of the training set with balanced prior distributions. For example, in the i.i.d. training set, 70% of the examples with the comparator `ends` and relation *after* are positive. We randomly remove instances from the majority classes to produce a uniform-prior training set such that a model can no longer rely on such prior distributions. We believe this setting better evaluates a system's true understanding of the task.

[7] To ensure that tanh returns a value close to 1 or -1, we multiply the distance by a big number denoted as INT_{max}.

[8] We release the systems for reproduction at `http://cogcomp.org/page/publication_view/937`

System	Start	End	All	Story
Majority	57.3	69.8	64.1	18.1
BiLSTM	53.7	63.5	59.1	10.9
Roberta-Large	78.5	78.3	78.4	26.1
T5-3B	79.4	77.4	78.3	26.9
BaseLM (T5-large)	75.5	75.4	75.4	22.6
BaseLM-MATRES	76.7	76.3	76.5	25.3
PTNTIME (ours)	81.4	77.5	79.3	31.0
SYMTIME (ours)	**82.1**	**79.4**	**80.6**	**32.0**
SYMTIME-ZEROSHOT	77.0	73.1	74.9	21.6

Table 1: Performance on TRACIE, best numbers in **bold**. BaseLM is T5-large; Story is the percentage of story-wide exact match; Majority is based on the comparator and temporal-relation distribution; Zeroshot uses no TRACIE instance as supervision.

System	Start	End	All	ΔAll
Random	50.0	50.0	50.0	-14.1
BiLSTM	50.5	51.2	50.9	-8.2
Roberta-Large	75.1	68.1	71.3	-7.1
T5-3B	72.8	68.6	70.5	-7.8
BaseLM (T5-large)	68.1	67.8	67.9	-7.5
BaseLM-MATRES	76.3	69.9	72.8	-3.7
PTNTIME (ours)	80.6	73.2	76.6	-2.7
SYMTIME (ours)	**81.2**	**77.0**	**78.9**	-1.7
SYMTIME-ZEROSHOT	77.0	73.1	74.9	**0.0**

Table 2: Performance on TRACIE uniform-prior training setting. ΔAll compares the difference with Table 1; Majority is equivalent to random guessing.

6.3 Main Results

Table 1 shows system performance on TRACIE's i.i.d. setting. We observe that PTNTIME improves on all metrics over the base language model, with 6% on start-time comparisons and 8% on story-wide exact match. It also outperforms BaseLM-MATRES, suggesting that distant supervision is more efficient than extensive human annotation.

With a symbolic end-time inference, SYMTIME further improves on all metrics, with 7%, 4%, and 9% gains over the base language model on start time, end time and story-wide exact match, respectively. SYMTIME can further improve the performance on start-time hypotheses over PTNTIME even though they use the same model to predict start-time queries. This is because PTNTIME is not designed to understand end time from pre-training, and fine-tuning on such data hurts its representation in general. This illustrates the benefits of models using explicit and sensible reasoning processes.

Table 2 compares systems in the uniform-prior training setting. Compared to the setting in Table 1, a system cannot exploit prior knowledge about the label distribution when making predictions. Given this, we see that all baselines produce a much lower performance, e.g., the BiLSTM, which is a model that lacks much of the pre-requisite knowledge for reasoning, suddenly performs near random chance. Compared to the baseline models, PTNTIME only drops 2.7%, suggesting that it is more invariant to evaluation settings and better understands temporal common sense. SYMTIME has the smallest drop among all models (1.7%) because of its explicit reasoning process on end-time hypotheses. SYMTIME-ZEROSHOT does not use any TRACIE training examples, so it has the same performance in the uniform-prior setting which outperforms all supervised baselines including T5-3B.

System	OT-NS	OT	OT-MS	PT
Wang et al. (2020)	85.9	-	-	-
BaseLM	86.0	87.5	77.4	69.0
SYMTIME	87.3	89.6	86.1	75.1

Table 3: Performance on MATRES. Wang et al. (2020) is not strictly comparable with the rest.

6.4 Extrinsic Evaluation

To show that our model is not limited to the TRACIE dataset and is general in temporal relation reasoning, we also evaluate on MATRES (Ning et al., 2018b), a temporal relation dataset focused on comparing explicit events' start times. We train and evaluate only the instances with a label of either "before" or "after", which accounts for about 80% of all instances. We compare the performance of SYMTIME[9] with BaseLM. We report four results - **OT-NS (original test, no story)**: train and test with only the sentences containing the trigger verbs; **OT**: train and test with the entire document (down-sampled to be below the maximum sequence length) as an auxiliary input; **OT-MS (original test, minimal supervision)**: train with 1.2k (6%) training instances; **PT (perturbed test)**: train with the complete training set and test on a perturbed test set from Gardner et al. (2020). In OT-NS, we also report a SOTA system from Wang et al. (2020) under the same two-label[10] setting.

Table 3 shows the performance of our model and the baselines. We see that our model is consistently

[9] This is virtually the same as using PTNTIME as MATRES does not evaluate duration nor end times.

[10] Wang et al. (2020) is trained with two additional labels. We constraint the output space to only "before" and "after" using argmax, but this process makes it not directly comparable.

Sys.	BaseLM	PTNTIME	SYMTIME	Human
Acc.	52.6	72.2	75.3	82.5

Table 4: Performance on *no-story* TRACIE under the uniform-prior training setting.

Sys.	PTNTIME	cross-sentence	within-sentence
Acc.	80.6	79.9	63.7

Table 5: Comparison of pre-training data sources on TRACIE's start time prediction accuracy, under the uniform-prior training setting.

better than BaseLM, and at the same time, comparable to Wang et al. (2020). Our model benefits more from input contexts, and only drops 4% in the OT-MS setting with minimal supervision (from 89.6 to 86.1), comparing to the 10% drop from T5-Large. This shows the effectiveness of our distant signals in §4.1, which are also designed to encourage contextual understandings.

6.5 Ablation Studies and Analysis

To better understand the improvements from our models, we conduct several ablation studies.

Table 4 shows the results on TRACIE where the story is not provided as part of the inputs to systems (a *no-story* setting). While such a setting bares some resemblance to the *partial-input* baselines often employed in TE (Poliak et al., 2018), in our setting, it is often possible to predict temporal relations in the absence of stories because of strong commonsense priors. Indeed, we estimate that 65% of the instances can be correctly predicted from the hypotheses alone, based on expert analysis in § 3.2. This suggests a 82.5% human upper-bound[11] in this *no-story* setting. Hence, such a setting partly evaluates a model's ability to incorporate commonsense priors when making decisions.

We see that BaseLM is close to random chance, whereas PTNTIME and SYMTIME improve 20% and 22% respectively. This suggests that our models better understand temporal common sense through the distant supervision on both start times and duration. On the other hand, we observe much smaller drops in our model's performances in this *no-story* setting. This suggests that our models do not improve as much on the 35% instances that require multi-hop timeline constructions over more than two events, motivating future work.

Table 5 compares the two pre-training sources described in §4.1 by individually pre-training two models with only within-sentence or cross-sentence extracted data. We see that the cross-sentence extraction brings the most performance gain on TRACIE's start-time binary metric under the uniform-prior training setting. This suggests that the global extraction rule is able to introduce new knowledge that is not seen in localized language model pre-training. Combining the within-sentence data further improves the performance.

Through analysis on the interval predictions made by SYMTIME, we notice a tendency for the model to predict "after" for end-time instances, possibly due to overly-estimated durations: a byproduct of natural biases in text. Given the weak signal used to learn such intervals and these potential biases, this is not altogether surprising. We leave the task of learning more robust and faithful interval representations for future work.

7 Conclusion

We introduce a challenging dataset TRACIE, to evaluate systems' temporal understanding of implicit events. We propose a distant supervision process that improves language models' understanding of start times of both explicit and implicit events. We further combine this process with a distantly supervised model that estimates events' duration to compare event end times, under the explicit rule that end times are start times plus durations. We show that our model improves over TRACIE and MATRES, suggesting the effectiveness of high-precision pre-training and symbolic temporal reasoning. Despite these advances, TRACIE continues to be a challenging task for future work on general temporal reasoning.

Acknowledgments

This research is based upon work supported in part by the office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA Contract No. 2019-19051600006 under the BETTER Program, and by Contract FA8750-19-2-1004 with the US Defense Advanced Research Projects Agency (DARPA). The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

[11] We assume that the remaining 35% non-predictable instances are decided by random guessing.

References

James F Allen. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843.

Prithviraj Ammanabrolu, Wesley Cheung, William Broniec, and Mark O Riedl. 2021. Automated Storytelling via Causal, Commonsense Plot Ordering. In *AAAI*.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and D. Klein. 2016. Neural module networks. *CVPR*.

Steven Bethard, James H. Martin, and Sara Klingenstein. 2007. Timelines from Text: Identification of Syntactic Temporal Relations. *ICSC*.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Scott Wen-tau Yih, and Yejin Choi. 2020. Abductive Commonsense Reasoning. *In ICLR*.

Taylor Cassidy, Bill McDowell, Nathanel Chambers, and Steven Bethard. 2014. An Annotation Framework for Dense Event Ordering. In *ACL*.

Quang Do, Wei Lu, and D. Roth. 2012. Joint Inference for Event Timeline Construction. In *EMNLP-CoNLL*.

Matt Gardner, Yoav Artzi, Victoria Basmova, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hanna Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, R. Tsarfaty, Eric Wallace, A. Zhang, and Ben Zhou. 2020. Evaluating Models' Local Decision Boundaries via Contrast Sets. In *Findings of EMNLP*.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew E. Peters, M. Schmitz, and L. Zettlemoyer. 2018. AllenNLP: A Deep Semantic Natural Language Processing Platform. *NLP-OSS*, abs/1803.07640.

Alfonso Gerevini and Lenhart Schubert. 1995. Efficient algorithms for qualitative reasoning about time. *Artificial intelligence*, 74(2):207–248.

Nitish Gupta, Kevin Lin, Dan Roth, Sameer Singh, and Matt Gardner. 2019. Neural Module Networks for Reasoning over Text. *In ICLR*.

Andrey Gusev, Nathanael Chambers, Divye Raj Khilnani, Pranav Khaitan, Steven Bethard, and Dan Jurafsky. 2011. Using Query Patterns to Learn the Duration of Events. In *IWCS*.

Rujun Han, Qiang Ning, and Nanyun Peng. 2019. Joint Event and Temporal Relation Extraction with Shared Representations and Structured Prediction. In *EMNLP*.

Daniel Khashabi, Tushar Khot, Ashish Sabharwal, and Dan Roth. 2018. Question Answering as Global Reasoning over Semantic Abstractions. In *AAAI*.

Tushar Khot, Daniel Khashabi, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2021. Text Modular Networks: Learning to Decompose Tasks in the Language of Existing Models. In *NAACL*.

Shibamouli Lahiri. 2014. Complexity of Word Collocation Networks: A Preliminary Structural Analysis. In *Proceedings of ACL-SRW*.

Alice Lai, Yonatan Bisk, and Julia Hockenmaier. 2017. Natural Language Inference from Multiple Premises. *IJCNLP*.

A. Leeuwenberg and Marie-Francine Moens. 2018. Temporal information extraction by predicting relative time-lines. In *EMNLP*.

Tao Li, Vivek Gupta, Maitrey Mehta, and V. Srikumar. 2019. A logic-driven framework for consistency of neural models. In *EMNLP*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv*, abs/1907.11692.

Yuanliang Meng and Anna Rumshisky. 2018. Context-aware neural model for temporal information extraction. In *ACL*.

Marie-Francine Moens and A. Leeuwenberg. 2017. Structured learning for temporal relation extraction from clinical records. In *EACL*.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A Corpus and Evaluation Framework for Deeper Understanding of Commonsense Stories. In *NAACL*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A New Benchmark for Natural Language Understanding. *In ACL*.

Qiang Ning, Z. Feng, and D. Roth. 2017. A Structured Learning Approach to Temporal Relation Extraction. In *EMNLP*.

Qiang Ning, H. Wu, H. Peng, and D. Roth. 2018a. Improving Temporal Relation Extraction with a Globally Acquired Statistical Resource. In *NAACL*.

Qiang Ning, Hao Wu, Pradeep Dasigi, Dheeru Dua, Matt Gardner, Robert L. Logan IV, and Zhen Nie. 2020a. Easy, Reproducible and Quality-Controlled Data Collection with CROWDAQ. In *EMNLP*.

Qiang Ning, Hao Wu, Rujun Han, Nanyun Peng, Matt Gardner, and Dan Roth. 2020b. Torque: A reading comprehension dataset of temporal ordering questions. In *EMNLP*.

Qiang Ning, Hao Wu, and Dan Roth. 2018b. A multi-axis annotation scheme for event temporal relations. In *ACL*.

Qiang Ning, Ben Zhou, Z. Feng, H. Peng, and D. Roth. 2018c. CogCompTime: A Tool for Understanding Time in Natural Language. In *EMNLP*.

Tim O'Gorman, Kristin Wright-Bettner, and Martha Palmer. 2016. Richer event description: Integrating event coreference with temporal, causal and bridging annotation. In *CNS*.

Feng Pan, Rutu Mulkar-Mehta, and Jerry R Hobbs. 2006. Extending TimeML with Typical Durations of Events. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*.

J. Pearl. 2009. Causal inference in statistics: An overview. *Statistics Surveys*, 3:96–146.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only Baselines in Natural Language Inference. *Proceedings of *SEM*.

James Pustejovsky, José M Castano, Robert Ingria, Roser Sauri, Robert J Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R Radev. 2003. TimeM: Robust Specification of Event and Temporal Expressions in Text. In *New Directions in Question Answering*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-text Transformer. *JMLR*, 21(140):1–67.

Nils Reimers, Nazanin Dehghani, and Iryna Gurevych. 2016. Temporal Anchoring of Events for the Timebank corpus. In *ACL*.

Alon Talmor and Jonathan Berant. 2018. The Web as a Knowledge-base for Answering Complex Questions. *In NAACL*.

Niket Tandon, Bhavana Dalvi, Joel Grus, Wen tau Yih, Antoine Bosselut, and Peter Clark. 2018. Reasoning about actions and state changes by injecting commonsense knowledge. *ArXiv*, abs/1808.10012.

Niket Tandon, Keisuke Sakaguchi, Bhavana Dalvi Mishra, Dheeraj Rajagopal, Peter Clark, Michal Guerquin, Kyle Richardson, and Eduard Hovy. 2020. A dataset for tracking entities in open domain procedural text. In *EMNLP*.

Siddharth Vashishtha, Adam Poliak, Yash Kumar Lal, Benjamin Van Durme, and Aaron Steven White. 2020. Temporal Reasoning in Natural Language Inference. In *Finding of EMNLP*.

Siddharth Vashishtha, Benjamin Van Durme, and Aaron Steven White. 2019. Fine-grained Temporal Relation Extraction. In *ACL*.

Haoyu Wang, Muhao Chen, Hongming Zhang, and Dan Roth. 2020. Joint constrained learning for event-event relation extraction. In *EMNLP*.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *In NAACL*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Tomer Wolfson, Mor Geva, Ankit Gupta, Matt Gardner, Yoav Goldberg, Daniel Deutch, and Jonathan Berant. 2020. Break it Down: A Question Understanding Benchmark. *TACL*, 8:183–198.

Ben Zhou, Daniel Khashabi, Qiang Ning, and D. Roth. 2019. "going on a vacation" takes longer than "going for a walk": A study of temporal commonsense understanding. In *EMNLP*.

Ben Zhou, Qiang Ning, Daniel Khashabi, and Dan Roth. 2020. Temporal Common Sense Acquisition with Minimal Supervision. In *ACL*.