Temporal Common Sense Acquisition with Minimal Supervision

Ben Zhou,[1] **Qiang Ning,**[2] **Daniel Khashabi,**[2] **Dan Roth**[1]
[1]University of Pennsylvania, [2]Allen Institute for AI
{xyzhou,danroth}@cis.upenn.edu {qiangn,danielk}@allenai.org

Abstract

Temporal common sense (e.g., duration and frequency of events) is crucial for understanding natural language. However, its acquisition is challenging, partly because such information is often not expressed explicitly in text, and human annotation on such concepts is costly. This work proposes a novel sequence modeling approach that exploits explicit and implicit mentions of temporal common sense, extracted from a large corpus, to build TACOLM,[1] a **t**empor**a**l **co**mmon sense **l**anguage **m**odel. Our method is shown to give quality predictions of various dimensions of temporal common sense (on UDST and a newly collected dataset from RealNews). It also produces representations of events for relevant tasks such as duration comparison, parent-child relations, event coreference and temporal QA (on TimeBank, HiEVE and MCTACO) that are better than using the standard BERT. Thus, it will be an important component of temporal NLP.

1 Introduction

Time is crucial when describing the evolving world. It is thus important to understand time as expressed in natural language text. Indeed, many natural language understanding (NLU) applications, including information retrieval, summarization, causal inference, and QA (UzZaman et al., 2013; Chambers et al., 2014; Llorens et al., 2015; Bethard et al., 2016; Leeuwenberg and Moens, 2017; Ning et al., 2018b), rely on understanding time.

However, understanding time in natural language text heavily relies on *common sense* inference. Such inference is challenging since commonsense information is rarely made explicit in text (e.g., how long does it take to open a door?) Even when such information is mentioned, it is often affected by another type of *reporting bias*: people rarely say the obvious, in order to communicate more efficiently, but sometimes highlight rarities (Schubert, 2002; Van Durme, 2009; Gordon and Van Durme, 2013; Zhang et al., 2017; Bauer et al., 2018; Tandon et al., 2018).

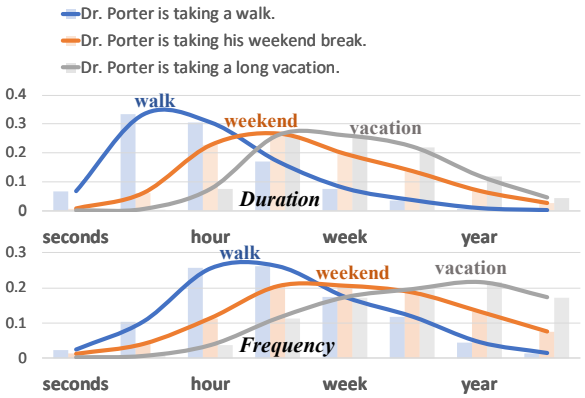

Figure 1: Our model's predicted distributions of event **duration** and **frequency**. The model is able to attend to contextual information and thus produce reasonable estimates.

This is an even more pronounced phenomenon when it comes to temporal common sense (TCS) (Zhou et al., 2019). In Example 1, human readers know that a typical vacation is likely to last at least a few days, and they would choose *"will not"* to fill in the blank for the first sentence; instead, with a slight change of context "vacation" → "walk outside," people typically prefer *"will"* for the second one. Similarly, any system which correctly answers this example for the right reason would need to incorporate TCS in its reasoning.

Example 1: choosing from *"will" or "will not"*
Dr. Porter is now (*e1:taking*) a vacation and _____ be able to see you soon.
Dr. Porter is now (*e2:taking*) a walk outside and _____ be able to see you soon.

Acquiring the multiple dimensions of TCS (e.g., *duration* and *frequency*) is challenging. As shown

[1]https://cogcomp.seas.upenn.edu/page/publication_view/904

in Example 1, the duration of "taking a vacation" and "taking a walk" are not expressed explicitly, so that systems are required to read between the lines to support the inference. A pre-trained language model may not handle this issue well, as it cannot identify the TCS dimensions in temporal mentions and effectively learn from them. As a result, it cannot generalize well to similar events without explicit temporal mentions. To handle this problem, we design syntactic rules that can collect a vast amount of explicit mentions of TCS from unannotated corpus such as Gigaword (Graff et al., 2003) (§3.3). We use this data to pre-train our model so that it distinguishes different dimensions.

A second challenge occurs when the text is highlighting rare and special cases. As a result, temporal mentions in natural text may follow a distorted distribution in which certain kinds of "common" events are under-represented. For instance, we may rarely see mentions of "I opened the door in 3 seconds," but we may see "it took me an hour to open this door" in text. To overcome this challenge, we exploit the joint relationship among temporal dimensions. Although we rarely observe the *true* duration of "opening the door" in free-form text, we may see phrases like "I opened my door during the fire alarm," providing an upper-bound to the duration of the event (i.e., "opening the door" does not take longer than the alarm.) We believe that we are the first to exploit such phenomena among temporal dimensions.

This paper studies several important dimensions of TCS inference: *duration* (how long an event takes), *frequency* (how often an event occurs) and *typical time* (when an event typically happens).[2] As a highlight, Fig. 1 shows the distributions (over time units) we predict for the *duration* and *frequency* of three events. We can see that "taking a vacation" lasts from days to months while "taking a walk" lasts from minutes to hours. As shown, our model is able to produce different and sensible distributions for the "take" event, depending on the context in which "take" occurs.

Our work builds upon pre-trained contextual language models (Peters et al., 2018; Devlin et al., 2019; Liu et al., 2019). However, a standard language modeling objective does not lead to a model that handles the two challenges mentioned above; in addition, other systematic issues limit its ability to handle TCS. In particular, language models do not directly utilize the ordinal relationships among temporal units. For example, "hours" is longer than "minutes," and "minutes" are longer than "seconds."[3] Fig. 2 shows that BERT does not produce a meaningful duration distribution for a set of events with a gold duration of "day" (extracted in §3.3). Our proposed system, on the other hand, is able to utilize the ordinal relationships and produce unimodal distributions around the correct labels in both Fig. 1 and Fig. 2 .

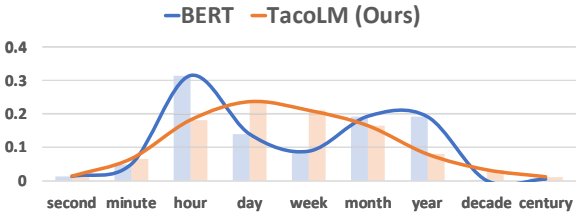

Figure 2: The predictive distribution of two models (ours and vanilla BERT) for a set of events labeled with "days" as their duration. Experiments show our model is about 40% better on duration predictions. (RealNews corpus; details in §4.2).

Contributions. This work proposes an augmented pre-training for language models to improve their understanding of several important temporal phenomena. We address two kinds of reporting biases by effectively acquiring weak supervision from free-form text and utilizing it to learn multiple temporal dimensions jointly. Our model incorporates other desirable properties of time in its objective (ordinal relations between temporal phrases, the circularity of certain dimensions, etc.) to improve temporal modeling. Our experiments show 19% relative improvement over BERT in intrinsic evaluations, and 5-10% improvements in most extrinsic evaluations done on three time-related datasets. Furthermore, the ablation study shows the value of each proposed component of our construction. Overall, this is the first work to incorporate a wide range of temporal phenomena within a contextual language model.

The rest of this paper is organized as follows. We distinguish our work with the prior work in §2. The core of our construction, including extraction of cheap supervision from raw data and augmenting a language model objective function with temporal signals, is in §3 . We conclude by showing intrinsic and extrinsic experiments in §4.

[2]E.g., typical time in a day (the morning), typical day of a week (on Sunday), and typical time of a year (summer).

[3]The relationship can be more complex. E.g., "hours" is closer to "minutes" than "centuries" is; days of a week forms a circle: "Mon." is followed by "Tue." and preceded by "Sun."

2 Related Work

Common sense has been a popular topic in recent years, and existing NLP works have mainly investigated the acquisition and evaluation of common sense reasoning in the physical world. These works include but are not limited to size, weight, and strength (Bagherinezhad et al., 2016; Forbes and Choi, 2017; Elazar et al., 2019), roundness and deliciousness (Yang et al., 2018), and intensity (Cocos et al., 2018). A handful of these works uses cheap supervision. For example, Elazar et al. (2019) recently proposed a general framework that discovers distributions of quantitative attributes (e.g., length, mass, speed, and duration) from explicit mentions (or co-occurrences) of these attributes in a large corpus. However, Elazar et al. (2019) restrict events to be verb tokens, while we handle verb phrases containing more detailed information (e.g., "taking a vacation" is very different from "taking a break," although they share the same verb "take"). Besides, there has been no report on the effectiveness of this method on temporal attributes.

On the other hand, *time* has long been an important research area in NLP. Prior works have focused on the extraction and normalization of temporal expressions (Strötgen and Gertz, 2010; Angeli et al., 2012; Lee et al., 2014; Vashishtha et al., 2019), temporal relation extraction (Ning et al., 2017, 2018c; Vashishtha et al., 2019), and timeline construction (Leeuwenberg and Moens, 2018). Recently, MCTACO (Zhou et al., 2019) summarizes five types of TCS and the three temporal dimensions studied here are all in their proposal.[4] MCTACO shows that modern NLU techniques are still a long way behind humans on TCS understanding, suggesting that further research on this topic is needed.

There have been works on temporal common sense, such as event duration (Pan et al., 2006; Gusev et al., 2011; Williams, 2012; Vempala et al., 2018; Vashishtha et al., 2019), typical temporal ordering (Chklovski and Pantel, 2004; Ning et al., 2018a,b), and script learning (i.e., what happens next after certain events) (Granroth-Wilding and Clark, 2016; Li et al., 2018; Peng et al., 2019). Those on duration are highly relevant to this work. (Pan et al., 2006) annotates a subset of documents from TimeBank (Pustejovsky et al., 2003) with "less-than-one-day" and "more-than-one-day" annotations and provides the first baseline system for this dataset. Vempala et al. (2018) significantly improve earlier work by using additional aspectual features for this task. Vashishtha et al. (2019) annotate the UDS-T dataset with event duration annotations and propose a joint method that extracts both temporal relations and event durations. Our approach has two notable differences from this line of work. First, we work on duration, frequency, and typical time—jointly on three dimensions of TCS, while the works above only focused on duration. Second, we focus more on obtaining cheap supervision signals from unlabeled data, while these other works all have access to human annotations. With respect to harnessing cheap supervision, Williams (2012); Gusev et al. (2011) propose to mine web data using a collection of hand-designed query patterns. In contrast to our approach, they are based on counting instead of machine learning and cannot handle the contextualization of events.

3 Temporally Focused Joint Learning with Minimal Supervision

In this section, we elaborate our approach to designing and pre-training TACOLM, a time-aware language model.

3.1 Scope

In this work, we focus on three major temporal dimensions of events, namely *Duration*, *Frequency* and *Typical Time*. Here, *Typical Time* means the typical occurring time of events during a day, day of a week, and month or season of a year. We follow the same definition to each of the dimensions (also called properties) in Zhou et al. (2019).

3.2 Joint Learning and Auxiliary Dimensions

As mentioned earlier, commonsense information extraction comes with the challenge of reporting biases. For example, people may not report the duration of "opening the door," or the frequency of "going to work." However, it is often possible to get supportive signals from other dimensions, as people mention "going to work" associated mostly with "a day" in a week, hence we may know the frequency of such an event.

We argue that many temporal dimensions are interrelated and a joint learning scheme would suit this task. Beyond duration, frequency and typical time, we also introduce auxiliary dimensions that are not meant to be used by themselves but will

[4] They additionally propose *typical order of events* and *stationarity* (whether a state holds for a very long time or indefinitely).

Figure 3: Examples of the extraction process for each temporal dimensions. The temporal arguments are marked orange and the result of the extraction are tuples of the form (`event`, `value`, `dimension`).

help the prediction of other dimensions. The auxiliary dimensions we define here are event *Duration Upper-bound* and event *Relative Hierarchy*. The former represents values that are upper-bounds to an event's duration but not necessarily the exact duration. The latter consists of two sub-relations, namely *temporal ordering* and *duration inclusion* of event-pairs.

3.3 Cheap Supervision from Patterns

We collect a few pattern-based extraction rules based on SRL parses for each temporal dimension (including the auxiliary dimensions). We design the rules to have high precision, while not compromising too much on recall. We overcome the potential sparsity issue (and the resulting low recall problem) by extracting from a massive amount of data. Fig. 3 provides some examples of the input/output for each dimension, as we describe the specific extraction process below.

We first process the entire Gigaword (Graff et al., 2003) corpus and use AllenNLP's SRL model (Gardner et al., 2018; Shi and Lin, 2019) to annotate all sentences. We extract the ones that contain at least one temporal argument (i.e., the *arg-tmp* constituent of SRL annotations) and use textual patterns to categorize each sentence into a corresponding dimension with respect to an associated verb. These patterns are inspired by earlier works and are extensively improved with iterative manual error analysis. The rest of this section is devoted to explaining the key design ideas used for these patterns.

Duration. We check if the temporal argument starts with "for," extract the numerical value and the temporal unit word, and normalize them into the nearest unit among the nine units in our scope: ("second," "minute," "hour," "day," "week," "month," "year," "decade," "century.") We ignore particular phrases such as "for a second chance" where the semantic of "second" is not temporal related. We found that "for" is the only high-precision preposition that indicates exact values of duration.

Frequency. Such temporal arguments are usually composed of a duration phrase and a numerical head (e.g., "four times per") indicating the frequency within the duration (e.g., "week"). Thus, we check for multiple keywords that indicate the start of a frequency expression, including "every," "per," "once," ... "times." If so, we extract the duration value as well as the numerical head's value. We ignore any temporal phrases that contain "when" since they often convey semantics that does not fit any of our temporal categories; e.g., "when everyday life..." is not describing the frequency of the corresponding verb. We represent the frequency with duration $\mathbf{d}$, with a definition of occurring *once* every $\mathbf{d}$ elapses. For example, the frequency of "four times per week" is represented as "1.75 days." Similarly, we normalize them into the nearest unit among the nine duration units described above, and "1.75 days" is extracted as "days."

Typical Time. We pre-define a list of typical time keywords, including the *time of day* (e.g., "morning" etc.), *time of week* (e.g., "Monday" etc.), *month* (e.g., "January" etc.) and *season* (e.g., "winter" etc.) We check if any of the typical time keywords appear in the temporal argument and verify if the temporal argument is, in fact, describing the time of occurrence. This is done by filtering out the temporal arguments that contain a set of invalid prepositions, including "until," "since," "following," since such keywords often do not indicate the actual time of occurrence.

Duration Upper-bound. Many temporal arguments describe the duration *upper-bound* instead of the exact duration value. For example, as described in (Gusev et al., 2011), "did `[activity]` yesterday" indicates something that happened within a day. We extend the set of patterns to include "in `[temporal expression]`" or keywords such as "next" (e.g., "the next day"), "last" (e.g., "last week"), "previous" (e.g., "previous month"), or "recent" (e.g., "recent years"). We normalize the values into the same label set of the nine unit words as the duration dimension.

Event Relative Hierarchy. A system can learn about an event with comparisons to other ones, as we show in §1. To acquire hierarchical relationships between events, we check whether the SRL temporal argument starts with a keyword that indicates a relation between the main event and another event phrase. We consider five such keywords, namely "before," "after," "during," "while" and "when." We use these keywords to label the relative relationship between the two events. Here, we assume that "during" and "while" are the same, which indicates that the main event is not longer than the one in the argument. Note that certain keywords might have meanings that do not suggest temporal relationships (e.g., "while" has a different sense similar to "whereas.") We rely on SRL annotations to identify the appropriate sense of the keywords. We use the temporal keyword as labels, but keep the entire event phrase in the SRL temporal argument for later use in §3.5.

Resulting data. We collect 25 million instances that are successfully parsed into one of our temporal dimensions from the entire Gigaword corpus (Graff et al., 2003). Each instance is in the form of (`event,value,dimension`) tuples (Fig. 3), with a dimension distribution shown in Fig. 4. For all events, we remove the related temporal argument so that it does not contain direct information about the dimension or the value. For example, as shown in Fig. 3, "for 2 hours" is removed, and only "Jack rested before the speech" is kept so that the target duration does not present in the event. Note that `value` is also called and used as "label" in later contexts related to classification tasks.

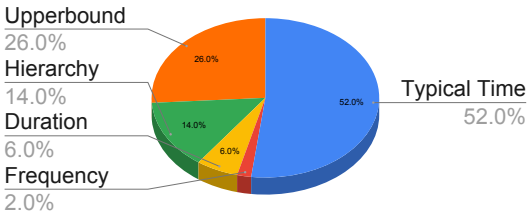

Figure 4: The distributions of different temporal dimensions in the collected data.

3.4 Soft Cross-Entropy Objective for Ordinal Classification

The temporal values in one dimension are naturally related to each other via a certain ordering and appropriate distance measures. To account for and utilize this external knowledge, we use a soft cross-entropy to encourage predictions that are aligned with the external knowledge.

Consider $\mathbf{x}$ as a system's output logits across labels, and we express our soft loss function as follows:

$$\ell = -\sum_{i \in D} \mathbf{y}_i^\top \log(\text{softmax}(\mathbf{x}_i)), \qquad (1)$$

where D is the instances in the training data and $\mathbf{y}$ represent the degree to which the target labels align with the external knowledge. Thus, $\mathbf{y}$ is a probability vector, i.e., has non-zero values and sum to 1.0.

Now we describe how we construct $\mathbf{y}$ to apply the aforementioned external knowledge. *Duration*, *Frequency*, and *Upper-bound* take the same set of labels of duration units. We first define a function logsec(.) which takes a unit and normalizes it to its logarithmic value in "seconds" (e.g., "minute" $\rightarrow$ 60 $\rightarrow$ 4.1). For each instance in these dimensions, with an observed gold label g, we assume a normal distribution with a mean value of $\mu = \text{logsec}(g)$ and a fixed standard deviation of $\sigma = 4$. Then, we construct $\mathbf{y}$ so that,

$$\mathbf{y}[i] = \frac{1}{\sigma\sqrt{2\pi}} e^{-(\text{logsec}(l)-\mu)^2/2\sigma^2} \qquad (2)$$

where l is the i^{th} label. We apply softmax at the end to ensure $\mathbf{y}$ sums to 1.

For *typical time*, the labels are placed with approximately equal distances in a circular fashion. For example, "Monday" is before "Tuesday" and after "Sunday." We assume adjacent units have a distance of 1, and we generate $\mathbf{y}$ based on a Gaussian distribution with a standard deviation of 0.5. In other words, we assume the two immediate neighbors of a gold label are reasonably possible.

For *hierarchy*, we construct $\mathbf{y}$ as a one-hot vector where only the gold label has a value of 1, and the rest are zeroes.

3.5 Sequential Language Modeling

Our goal is to build a model that is able to predict temporal labels (values) given events and dimensions. Instead of building a classification layer on top of a pre-trained model, we follow previous work (Huang et al., 2019) and place the label into the input sequence. We mask the label in the sequence and use the masked token prediction objective as the classification objective. To produce more general representations, we also keep the temporal label and mask the event tokens instead at a certain probability, so that we are

able to maximize both $\mathbb{P}(\texttt{Tmp-Label}|\texttt{Event})$ and $\mathbb{P}(\texttt{Event}|\texttt{Tmp-Label})$ in the same learning process, where `Tmp-Label` refers to the temporal label associated with the event.

Specifically, we use the reserved "unused" tokens in BERT-base model lexicon to construct a 1-to-1 mapping from every value in every dimension to the new vocabulary. We choose not to use the existing representations for temporal terms that are already included in BERT's "in-use" lexicon, such as "minutes" or "weeks," because these keywords have different temporal semantics in different dimensions. Instead, we assign unique and separate lexicon entries to different values in different dimensions, even though the values may share the same surfaces. Consider each (`event,value,dimension`) tuple, we map `value` and `dimension` to their new vocabularies `[Val]` and `[Dim]`, and we use [$\texttt{W}_1$, $\texttt{W}_2$, ..., $\texttt{W}_n$] to represent the tokens in the sentence, and $\texttt{W}_{verb}$ the event verb anchor from SRL. We now form a sequence [$\texttt{W}_1$, $\texttt{W}_2$,...`[Vrb]`, $\texttt{W}_{verb}$, ...$\texttt{W}_n$, `[SEP]`, `[Vrb]`, `[Dim]`, `[Val]`, `[Arg-Tmp-Event]`], where `[Vrb]` is a marker token that is the same across all instances. `[Arg-Tmp-Event]` is the event phrase in the SRL temporal argument, as described in *hierarchy*. `[Arg-Tmp-Event]` is empty for all dimensions other than *hierarchy*.

We mask `[Val]` with probability p_{mask} and `[Dim]` with probability p_{dim}. We individually mask each event tokens with probability p_{event} when we do not mask `[Val]` nor `[Dim]`. Soft cross-entropy is used when predicting `[Val]`, and a regular Cross-entropy is used for other tokens. We use the pre-trained token-recovery layer, and follow BERT's setting to randomly keep a token's surface or change it to noise during recovery.

In the experiments, we explore a set of configurations of the system. We explore the effect of having only one sentence or the two additional neighboring sentences as input contexts. We also experiment with all-event-masking, where we mask tokens in the event with a much higher probability. The goal of this masking scheme is to reduce the predictability of event tokens based on other event tokens to alleviate prior biases and focus more on the temporal argument. For example, BERT predicts "coffee" for the `[MASK]` in "I had a cup of `[MASK]` this evening" because of the strong prior of "cup of." By masking more tokens in the event, the remaining ones will be more conditioned to the temporal cue.

3.6 Label Weight Adjustment

The label imbalance in the training data largely hinders our goal, as we should not assume a prior distribution as expressed in natural language. For example, "seconds" appears around ten times less than "years" in the data we collected for *duration*, leading to a biased model. We use weight adjustment to fix this. Specifically, we apply weight adjustment to the total loss with a weight factor calculated as the observed label's count relative to the number of all instances.

4 Experiments

4.1 Variations and Settings

We experiment with several variants of the proposed system to study the effect of each change.

Input Size. A model with three input sentences (including the event sentence's left/right neighbors) are labeled with MS. Non MS models use only one sentence in which the event occurs.

All Event Masking. A model with $p_{event} = 0.6$ is labeled as AM, and $p_{event} = 0.15$ otherwise.

Final Model. Our final model includes all auxiliary dimensions (AUX) (mentioned in §3.2), uses soft cross-entropy loss (SL) and applies weight adjustment (ADJ) (mentioned in §3.6). We study each changes' effect by ablating them individually.

To deal with the skew present in the training data (§3), we down-sample to ensure roughly the same occurrences of each dimension (except for frequency because of its low quantity). As a result, 4.3 million sentences were used in pre-training (down-sampled from 25 million mined sentences). We employ a learning rate of 2e-5 with 3 epochs and set $p_{mask} = 0.6$ and $p_{dim} = 0.1$. Other parameters are the same as those of the BERT base model. We use epoch 2's model for extrinsic evaluations to favor generalization, and epoch 3's model for intrinsic evaluations as it achieves the best performance across tasks.

4.2 Intrinsic Evaluation

We evaluate our method on the temporal value recovery task, where the inputs are a sentence representing the event, an index to the event's verb, and a target dimension. The goal is to recover the temporal value of the given event in the given dimension.

Datasets. To ensure a fair comparison, we sample instances from a new corpus RealNews (Zellers et al., 2019) that have no document overlap with

our pre-training data and, at the same time making the data not strictly in-domain. We apply the same pattern extraction process mentioned in §3.3 on the new data and collect instances that are uniformly distributed across dimensions and values. In addition, we ask annotators on Mechanical Turk to filter out the events that cannot be recovered by common sense. For example, "I brush my teeth [Month]" will be discarded because all candidate answers are approximately uniformly distributed so that one cannot identify a subgroup of labels to be more likely.

Specifically, we ask one annotator to select from 4 choices regarding each (event, temporal value) tuple. The choices are 1) the event is unclear and abstract; 2) the event has a uniform distribution across all labels within the dimension; 3) the given label is one of the top 25% choices among all other labels within the dimension and 4) the given label is not very likely. We keep the instances for which the annotator selects option 3), verifying that the label is a very likely choice for the given dimension. For the RealNews corpus, we annotate 1,774 events that are roughly uniformly distributed across dimensions and labels, among which 300 events are preserved.

We also apply the same process to UDST dataset. We find the majority of the original annotation to be unsuitable, as there are many annotations to events that are seemingly undecidable by common sense. We first apply an initial filtering by using only events of which the anchor word is a verb and require all existing annotations from (Vashishtha et al., 2019) of the same instance to have an average distance less than two units. We then use our method to annotate 1,047 events, and eventually, 142 instances are left.

Systems. In both datasets, we compare our proposed system with BERT. To use BERT's predictions on temporal values without supervision, we artificially add prepositions querying the target dimension as well as a masked token right after the verb. For example, "I ran to the campus" will be transformed as "I ran for 1 `[MASK]` to the campus". The specific prepositions added are "for 1" (duration), "every" (frequency), "in the" (time of the day), "on" (week), "in" (month), and "in" (season). We then rank the temporal keywords (singular) in the given dimension according to the masked token's predictions. For our model, we follow the sequence formulation described above, recover and rank the masked `[Val]` token.

In addition, we also compare with a baseline system called *BERT + naive finetune*, which is BERT fine-tuned on the same pre-training data we used for our proposed models, with a higher probability of masking a temporally related keyword (i.e., all values we used in all dimensions). Unlike our model, we only use soft cross-entropy loss and do not distinguish the dimensions each keyword is expressing.

Metrics. Following Vashishtha et al. (2019), we employ a metric "distance" that measures the rank difference between a system's top prediction and the gold label with respect to an ordered label set. For duration and frequency where values are in a one-directional order, we use the absolute difference of the label ranks. For other dimensions where the labels are in circular relationships, we use the minimal distance between two labels in both directions, so that "January" will have a distance 1 with "December." This is similar to an *MAE* metric, and we report the averaged number across instances.

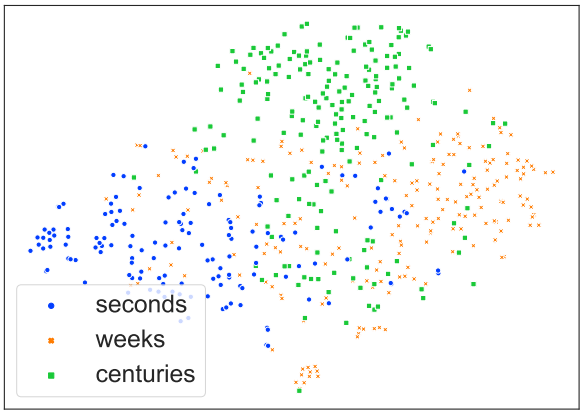

Figure 5: Representations of events (whose durations were labeled as seconds, weeks, or centuries) obtained from the original BERT base model.

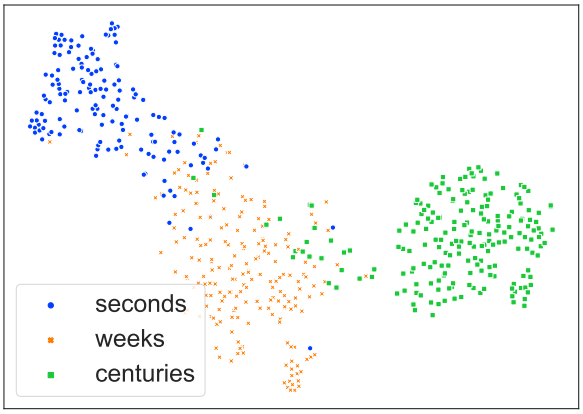

Figure 6: Representations of the same set of events as in Fig. 5 obtained from the proposed method.

Systems	RealNews						UDS-T
			Typical Time				
	Duration	Freq	Day	Week	Month	Season	Duration
BERT	1.33	1.68	1.75	1.53	3.78	0.87	1.77
BERT + naive finetune	1.21	1.45	1.47	1.28	3.28	1.08	2.06
TACOLM (ours)	**0.75**	**1.17**	**1.72**	**1.19**	**3.42**	**0.63**	**1.49**
TACOLM (ours), normalized	8%	13%	22%	17%	29%	16%	17%
TACOLM -ADJ	0.84	1.20	1.82	1.08	2.47	0.74	1.68
TACOLM -SL	0.77	1.30	1.88	1.06	2.50	0.74	1.50
TACOLM -AUX	0.77	1.28	1.61	1.31	3.25	0.78	1.51
TACOLM -MS	0.84	1.12	1.82	1.5	3.17	0.61	1.69
TACOLM -AM	0.68	1.20	1.86	1.31	3.11	0.70	1.58

Table 1: Performance on intrinsic evaluations. The "normalized" row is the ratio of the distance to the gold label over the total number of labels in each dimension. Smaller is better.

The results on the filtered RealNews dataset and filtered UDST dataset are shown in Table 1. We see that our proposed final model is mostly better than other variants, and achieves 19% improvement over BERT on average on the normalized scale.

We plot the embedding space of events with duration of "seconds" "weeks" or "centuries" in Fig 5 and Fig 6. We take the verb's contextual representation, apply PCA to reduce the dimension from 768 to 50, and then t-SNE to reduce it further to 2. Comparing the two plots, we see that the clusters formed by BERT embeddings have a wider distribution over the space, and the clusters have more points in overlap, even though the three sets of events have drastically different duration values. Our proposed model's embedding is able to better cluster the events based on this temporal feature, which is expected.

4.3 TimeBank Evaluation

Beyond unsupervised intrinsic experiments, we also evaluate the capability of the event temporal representation as a product of our model. That is, we finetune both BERT baseline and our model with the same process to compare the internal representations of the transformers. We use TimeML (Sauréi et al., 2005; Pan et al., 2006), a dataset with event duration annotated as lower and upper bounds. The task is to decide whether a given event has a duration longer or shorter than a day. This is a suitable task to evaluate the embeddings because deciding longer/shorter than a day requires reasoning with more than one label, and would also benefit from auxiliary dimensions like duration upper-bound.

The dataset contains 2,251 events, and we split the events based on sentences into 1,248/1,003 train/test. We formulate the training as a sequence classification task by taking the entire sentence and adding a special marker to the left of the verb indicating its position. The marker is unseen to both BERT and our model. We use the transformer output of the first token and feed it to an MLP for classification. We use a learning rate of 5e-5 and train for 3 epochs, and we repeat every reported number with 3 different random initialization and take the average.

System F1	Accuracy	<Day F1	≥Day F1
BERT	73.7	63.7	79.0
TACOLM	**81.7**	**74.8**	**85.6**

Table 2: Performance on TimeBank Classification

Table 2 shows the results of the TimeBank experiment. We see around 7-11% improvement over BERT on this task. Comparing with the state-of-the-art (Vempala et al., 2018) with a different training/testing split, our model is within 1.5% of the best results but uses 25% less training data.

4.4 Subevent Relation Extraction

We apply our event representations to the task of event sub-super relation extraction. This is a proper evaluation because the task naturally benefits from temporal commonsense knowledge. Intuitively, short duration or high frequency indicates the event being at a lower hierarchy and vice versa. We test if the temporal focused event representations will improve.

We use HiEVE (Glavaš et al., 2014), a dataset with annotations of four event relationships: no relation (NoRel), coreference (Coref), Child-Parent (C-P) and Parent-Child (P-C). There is no official split for this dataset, so we randomly 80/20 split

the data at the document level and down-sample negative NoRel instances with a probability of 0.4.

Similarly, we formulate the problem as a sequence classification task, where two events are put into one sequence separated by "`[SEP]`," and verbs are marked by adding a marker token to their left. We use the representations of the first token and feed it to an MLP for classification. We train each model with a 5e-5 learning rate and 3 epochs. Each reported number is an average from 3 runs under different random initialization. During inference time, the probability scores for non-negative relations are averaged from the same event pair's sequences in both orders.

Table 3 shows the results of the HiEVE experiment. We see that TACOLM improves by 4% and 8% on the coreference task and the parent-child tasks over BERT, respectively.

Systems F1	NoRel	Coref	C-P	P-C
BERT	90.5	47.9	40.7	40.6
TACOLM	**91.3**	**51.5**	**49.4**	**48.5**

Table 3: Performance on HiEVE. The numbers are in percentages. Higher is better.

4.5 Temporal Question Answering

We also evaluate on MCTACO (Zhou et al., 2019), a question answering dataset that requires comprehensive understandings of temporal common sense and reasoning. We compare the exact-match score across the 5 dimensions defined in MCTACO, although this work only focuses on 3 of them. We use the original baseline system and interchange transformer weights to compare between BERT and ours. However, because our model replaces temporal expressions with special tokens, it is at disadvantage to be directly evaluated on the original dataset with temporal expressions in natural language. To fix this, we run the same extraction system in §3.3 with modifications to identify the dimension a question is asking, and augment candidate answers with our special tokens representing the temporal values (if any) mentioned. This introduces rule-based dimension identification as well as coarse unit normalization to the systems, so we train/evaluated BERT baseline with the same modified data as well. Each number is an average of 5 runs with different random initializations.

Results on MCTACO are shown in Table 4. As expected, we find that our model achieves better performance on the three dimensions that are focused in this work (i.e., duration, frequency, and typical time) as well as stationarity. However, the improvements are not very substantial, indicating the difficulty of this task and motivates future works. The model also does slightly worse on ordering, which is worth investigating in future works.

System	Duration	Ordering	Stationarity	Frequency	Typical Time
BERT	33.4	**36.5**	57.6	43.3	39.5
TACOLM	**34.6**	35.1	**57.9**	**45.1**	**40.9**

Table 4: Performance on MCTACO. Numbers are percentages and indicate exact match (EM) metric. Higher is better.

5 Conclusion

Temporal common sense (TCS) is an important yet challenging research topic. Despite the existence of several prior work on event duration, this is the first attempt to jointly model three key dimensions of TCS—duration, frequency, and typical time—from cheap supervision signals mined from unannotated free text. The proposed sequence modeling framework improves over BERT in terms of handling reporting bias, taking into account the ordinal relations and exploiting interactions among multiple dimensions of time. The success of this model is confirmed by intrinsic evaluations on RealNews and UDS-T (where we see a 19% improvement), as well as extrinsic evaluations on TimeBank, HiEVE and MCTACO. The proposed method may be an important module for future applications related to *time*.

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. This research is also supported by a grant from the Allen Institute for Artificial Intelligence (allenai.org).

References

Gabor Angeli, Christopher D Manning, and Daniel Jurafsky. 2012. Parsing time: Learning to interpret time expressions. In *NAACL-HLT*, pages 446–455.

Hessam Bagherinezhad, Hannaneh Hajishirzi, Yejin Choi, and Ali Farhadi. 2016. Are elephants bigger than butterflies? reasoning about sizes of objects. In *AAAI*.

Lisa Bauer, Yicheng Wang, and Mohit Bansal. 2018. Commonsense for generative multi-hop question answering tasks. In *EMNLP*, pages 4220–4230.

Steven Bethard, Guergana Savova, Wei-Te Chen, Leon Derczynski, James Pustejovsky, and Marc Verhagen. 2016. SemEval-2016 Task 12: Clinical TempEval. In *SemEval*, pages 1052–1062.

Nathanael Chambers, Taylor Cassidy, Bill McDowell, and Steven Bethard. 2014. Dense event ordering with a multi-pass architecture. *TACL*, 2:273–284.

Timothy Chklovski and Patrick Pantel. 2004. VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations. In *EMNLP*, pages 33–40.

Anne Cocos, Veronica Wharton, Ellie Pavlick, Marianna Apidianaki, and Chris Callison-Burch. 2018. Learning scalar adjective intensity from paraphrases. In *EMNLP*, pages 1752–1762.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Yanai Elazar, Abhijit Mahabal, Deepak Ramachandran, Tania Bedrax-Weiss, and Dan Roth. 2019. How large are lions? inducing distributions over quantitative attributes. In *ACL*.

Maxwell Forbes and Yejin Choi. 2017. Verb physics: Relative physical knowledge of actions and objects. In *ACL*, volume 1, pages 266–276.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. Allennlp: A deep semantic natural language processing platform. In *NLP-OSS*, pages 1–6.

Goran Glavaš, Jan Šnajder, Marie-Francine Moens, and Parisa Kordjamshidi. 2014. HiEve: A corpus for extracting event hierarchies from news stories. In *LREC*, pages 3678–3683, Reykjavik, Iceland. European Language Resources Association (ELRA).

Jonathan Gordon and Benjamin Van Durme. 2013. Reporting bias and knowledge acquisition. In *AKBC*, pages 25–30. ACM.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English gigaword. *Linguistic Data Consortium, Philadelphia*, 4(1):34.

Mark Granroth-Wilding and Stephen Christopher Clark. 2016. What happens next? event prediction using a compositional neural network model. In *ACL*.

Andrey Gusev, Nathanael Chambers, Pranav Khaitan, Divye Khilnani, Steven Bethard, and Dan Jurafsky. 2011. Using query patterns to learn the duration of events. In *IWCS*, pages 145–154.

Luyao Huang, Chi Sun, Xipeng Qiu, and Xuanjing Huang. 2019. GlossBERT: BERT for word sense disambiguation with gloss knowledge. In *EMNLP*, pages 3507–3512.

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

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

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

Zhongyang Li, Xiao Ding, and Ting Liu. 2018. Constructing narrative event evolutionary graph for script event prediction. *IJCAI*.

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 preprint arXiv:1907.11692*.

Hector Llorens, Nathanael Chambers, Naushad UzZaman, Nasrin Mostafazadeh, James Allen, and James Pustejovsky. 2015. SemEval-2015 Task 5: QA TEMPEVAL - evaluating temporal information understanding with question answering. In *SemEval*, pages 792–800.

Qiang Ning, Zhili Feng, and Dan Roth. 2017. A structured learning approach to temporal relation extraction. In *EMNLP*.

Qiang Ning, Zhili Feng, Hao Wu, and Dan Roth. 2018a. Joint reasoning for temporal and causal relations. In *ACL*.

Qiang Ning, Hao Wu, Haoruo Peng, and Dan Roth. 2018b. Improving temporal relation extraction with a globally acquired statistical resource. In *NAACL*, pages 841–851.

Qiang Ning, Ben Zhou, Zhili Feng, Haoruo Peng, and Dan Roth. 2018c. CogCompTime: A tool for understanding time in natural language. In *EMNLP*.

Feng Pan, Rutu Mulkar, and Jerry R Hobbs. 2006. Extending TimeML with typical durations of events. In *ARTE*, pages 38–45. Association for Computational Linguistics.

Haoruo Peng, Qiang Ning, and Dan Roth. 2019. KnowSemLM: A Knowledge Infused Semantic Language Model. In *CoNLL*.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237.

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

Roser Sauréi, Jessica Littman, Bob Knippen, Robert Gaizauskas, Andrea Setzer, and James Pustejovsky. 2005. Timeml annotation guidelines.

Lenhart Schubert. 2002. Can we derive general world knowledge from texts? In *HLT*, pages 94–97. Morgan Kaufmann Publishers Inc.

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*.

Jannik Strötgen and Michael Gertz. 2010. HeidelTime: High quality rule-based extraction and normalization of temporal expressions. In *SemEval*, pages 321–324. Association for Computational Linguistics.

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. In *EMNLP*, pages 57–66.

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

Benjamin Van Durme. 2009. *Extracting implicit knowledge from text*. Ph.D. thesis, University of Rochester.

Siddharth Vashishtha, Benjamin Van Durme, and Aaron Steven White. 2019. Fine-grained temporal relation extraction. In *ACL*, pages 2906–2919.

Alakananda Vempala, Eduardo Blanco, and Alexis Palmer. 2018. Determining event durations: Models and error analysis. In *NAACL*, volume 2, pages 164–168.

Jennifer Williams. 2012. Extracting fine-grained durations for verbs from twitter. In *ACL-SRW*, pages 49–54. Association for Computational Linguistics.

Yiben Yang, Larry Birnbaum, Ji-Ping Wang, and Doug Downey. 2018. Extracting commonsense properties from embeddings with limited human guidance. In *ACL*, volume 2, pages 644–649.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *NeurIPS*, pages 9054–9065. Curran Associates, Inc.

Sheng Zhang, Rachel Rudinger, Kevin Duh, and Benjamin Van Durme. 2017. Ordinal common-sense inference. *TACL*, 5(1):379–395.

Ben Zhou, Daniel Khashabi, Qiang Ning, and Dan Roth. 2019. "Going on a vacation" takes longer than "Going for a walk": A Study of Temporal Commonsense Understanding. In *EMNLP*.