Timeline Deliberation for Fine-grained Temporal Ordering

Anonymous ACL submission

Abstract

Despite recent advances, language models still struggle to capture temporal orders between events. For example, it is not trivial to teach the fine-grained difference between two questions "happened right before" or "happened often before". Previous solutions have relied on weak supervision, namely answer overlaps, as a proxy label to contrast similar and dissimilar pairs. In contrast, we claim that answer overlap on the question pair is too weak a signal for contrastive learning (also known as *shortcut* problem). So we propose to leverage question "bundles", a related question subset we group with respect to the events in the passage, as a stronger supervision to approximate a timeline of a passage. We introduce the Timeline Deliberation Network (TDN), which reasons over the timeline in a two-level process: The drafting layer drafts answers based on semantic and syntactic evidence. The refinement layer aggregates over contrast question groups as a set of inputs and collectively refines answers to maintain temporal consistency. Results on TORQUE and TB-dense datasets demonstrate that TDN outperforms previous methods, by effectively resolving the shortcut problem [1].

1 Introduction

Temporal ordering is a challenging area in natural language processing that involves understanding and reasoning about temporal relations between events (Ning et al., 2020; Zhou et al., 2019; Chen et al., 2021). Conventional approaches to incorporate the knowledge of temporal orders into the model only considered a limited number of coarse relations however, such as before/after/simultaneous.

Meanwhile, our focus is temporal machine reading comprehension (TMRC) task, such as TORQUE (Ning et al., 2020) aims at fine-grained understanding of temporal expressions that capture real-world diversity of temporal relations. For example, it requires the model to distinguish finer granularity like *"what event **happen right before** X"* and *"what **happen often before** X"*.

Specifically, we study "weakly-supervised" contrastive learning method that leverages answer overlaps between related questions (Shang et al., 2021), which performs comparably or outperforms baselines requiring stronger but expensive human-annotated categorization (Han et al., 2020; Huang et al., 2022), as we show in Section 4. For example, in Figure 1, $Q1$ *"what event **started before** X"* and $Q3$ *"what **happened before** X"* share the overlapping answer "debate" and "protection". On the other hand, $Q1$ does not have any common answer with $Q5$ *"what **happened when** X **was made**"*. Contrastive objective trains to pull $Q1$ and $Q3$ closer than $Q1$ and $Q5$.

The use of weak supervision in TMRC tasks, however, poses a potential threat of "shortcuts" or "spurious overlap": To illustrate, question $Q2$ and $Q3$, *"What happened before X"* and *"What happened after X"*, in Figure 1 are temporally distinct, but shares answers "protection" and "debate". In such scenarios, contrastive learning may overlook the temporal meanings of "before" and "after" by solely depending on answer overlap to determine semantic relations between temporal expressions.

Our distinction is discerning meaningful overlaps ($Q1$ and $Q3$) from spurious overlaps ($Q2$ and $Q3$), by adding another dimension of timeline. Figure 1-(b) shows a full-structured timeline, where event and question are annotated as time spans (e.g. start time and end time) [2]. This additional information can teach the model that $Q2$ and $Q3$ spans are disjoint and the overlap of "protection" and "debate" is a coincidence. Despite its importance, previous work does not consider a timeline, as supervision

[1] The code will be released after blind review.

[2] Relations are simple for illustration purposes, but can also represent events that might happen (uncertain), or, often happen (repetitive) (Ning et al., 2020).

(a) But the government **[e1: decided]** to **[e2: suspend]** the **[e3: cull]** in the face of strong public opposition. The public **[e4: debate]** on the conflict between wildlife **[e5: protection]** and farmers' livelihoods is still **[e6: going]** on here.

Q1. What event started before the decision?	A. debate, protection
Q2. What happened after the decision?	A. suspend, debate, protection, going
Q3. What happened before the decision?	A. protection, debate, going
Q4. What happened during the decision?	A. cull, debate, protection, going
Q5. What happened when the decision was made?	A. suspend

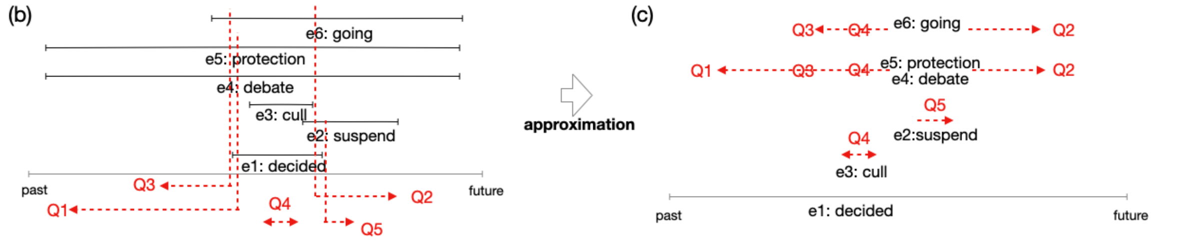

Figure 1: (a) Example of passage and related question set in temporal machine reading comprehension. Events in the passage are in bold. The passage's timelines are shown in (b) a full-structured manner and (c) approximated timeline using questions, i.e. temporal bundles. The timeline is centered around the event e1.

to build one is lacking in most scenarios.

Our distinction is approximating the timeline, without requiring supervision, by "bundling" questions that are related to each event. For example, in Figure 1-(c), protection (e5) is related to $Q1$, $Q2$ and $Q3$. When aggregated into a set of questions, this **temporal bundle** for $e5$ can be used to infer a consistent timeline among these questions. We illustrate the process using the example in Figure 1. For instance, the starting and ending points of the two events, debate (e4) and protection (e5) can be inferred from the answers to questions Q1 ("*What event started before*") and Q2 ("*What happened after*"), respectively. In this way, we know that $Q2$ and $Q3$ are disjoint and answer overlaps for these pairs are coincident, such that attention to the spurious overlaps can be safely reduced.

We propose a novel approach for effective reasoning over approximated timelines, which views temporal ordering as deliberation with constraints inspired by the human cognitive process of iterative refinement. Our Timeline Deliberation Network (TDN) consists of two levels: a Drafting Layer that generates semantic and syntactic evidence for each temporal ordering question, and a Refinement Layer that uses an attention mechanism to aggregate temporal relationships from multiple question-answer pairs. The resulting temporal information acts as a constraint on the original question and compels the model to refine the answer for consistency with the given temporal context.

We evaluate TDN on TORQUE, a reading comprehension dataset for temporal ordering questions. We achieve state-of-the-art performance on the public leaderboard.[3] We quantitatively and qualitatively analyze TDN's effectiveness in dealing with shortcuts by the timeline understanding, especially by a new "passage consistency" metric. Lastly, we confirm its generalizability to related tasks through the performance gain on TB-Dense.

Our main contributions are three-fold:

- We point out the shortcut issue in fine-grained temporal understanding and propose a novel approach to resolve it.
- We develop a framework for TMRC based on the human cognitive process: draft and refine.
- TDN effectively captures fine-grained temporal orders and outperforms other approaches.

2 Related Work

Our work is related to the following areas of research: temporal reading comprehension (TMRC), deliberation networks, and graph networks.

Temporal ordering reasoning Conventional temporal ordering tasks are temporal relation extraction (TRE) (Cassidy et al., 2014; Ning et al., 2018), whose goal is to categorize the temporal order into pre-defined categories. MATRES (Ning et al., 2018) groups the temporal relations into 4 categories: $Before/After/Simultaneous/Vague$. TB-Dense (Cassidy et al., 2014) considers 2 more

[3] https://leaderboard.allenai.org/torque/submissions/public. To be published after blind review.

classes, $Includes$ and $IsIncluded$. Our proposed approach can also benefit these tasks as we discuss in Section 5.

However, our main task is the TMRC task TORQUE (Ning et al., 2020) requiring finer-granular understanding of temporal ordering in question form to reflect the real-world diversity of temporal relations. Previous approaches to the TMRC task include continuously pre-training a PLM (Han et al., 2020) and question decomposition methods (Huang et al., 2022; Shang et al., 2021). ECONET (Han et al., 2020) continually pre-trains the PLM to inject the knowledge of temporal orders. Question decomposition approaches (Huang et al., 2022; Shang et al., 2021) divide the question into the event part and temporal relation expression part to better capture the complex semantics. All of the above use contrastive learning to understand different temporal relations, either by contrasting relations with human annotations (Han et al., 2020; Huang et al., 2022) or annotated answers (Shang et al., 2021). However, the former can be costly or imprecise, while the latter may rely on shortcuts. Our distinction is avoiding costly annotations but reduce shortcuts using timeline structure.

Deliberation networks Deliberation networks (Xia et al., 2017) incorporate the concept of human deliberation into the decision-making process. The idea behind the network is to simulate the human decision-making process by having multiple levels in the network, each representing a different stage in the deliberation process. The lower levels use local cues to identify relevant options, while the higher levels aggregate global information and make the final decision. However, they have been only applied to a sequence-to-sequence model (Xiong et al., 2018; Hu et al., 2021), to deal with its limited left-to-right attention. We are the first to apply them in temporal ordering using encoder-only models (Devlin et al., 2018; Liu et al., 2019), where the local information corresponds to each question and the global information is the timeline representing relations between questions and events.

Graph networks Graph Networks (Kipf and Welling, 2016; Velickovic et al., 2017) learn features through message passing on graph structures. These networks have demonstrated their effectiveness in tasks requiring complex reasoning skills, such as numerical reasoning (Ran et al., 2019; Chen et al., 2020) and logical reasoning (Huang et al., 2021). Graph networks also have been applied to TRE (Mathur et al., 2021; Zhang et al., 2022), though their effectiveness in TMRC has not been investigated.

3 Proposed Method

As overviewed in Figure 2, our approach is composed of two steps: Drafting (subsection 3.1) and Refinement (subsection 3.2). For example, in Figure 1, the first step in answering $Q1$ is to generate "local" drafts considering only $Q1$. The second step, then follows to collect answers from multiple questions, and checks if there are temporal inconsistency (by building semi-structured timeline). These global constraints help that semantics of temporal relations such as "started before" and "happened after" are not misinterpreted.

3.1 Drafting Layer

We formulate local drafting for query Q as a binary classification for every token in the given passage P, determining whether it is an answer to Q. For this goal, first, PLM encodes the question-passage pairs to get the contextual representation for each token. It takes the concatenated sequence of pair as input $[Q, P]$ and outputs the representation $[\acute{Q}, \acute{P}]$, where each token is $\acute{q}$ and $\acute{p}$.

After that, we build a syntax-aware graph that captures word-to-word dependency, following the convention of (Cheng and Miyao, 2017; Mathur et al., 2021; Zhang et al., 2022). However, unlike prior work mainly focusing on temporal relations on passage and not on question, comprehending both is critical for TMRC. To consider both, we build dependency trees for the question and passage then connect root nodes and co-mentioned event words bidirectionally. Here event words refer to nouns and verbs. Next, we followed graph reasoning step in (Ran et al., 2019) for question-passage interaction. The connections of nodes are categorized into 4 types: (1) question-question (qq) (2) passage-passage (pp) (3) passage-question (pq) (4) question-passage (qp). Each node in the graph is the corresponding word in question and passage.

The full pipeline is as follows:

$$[\bar{Q}, \bar{P}] = W^M[\acute{Q}, \acute{P}] \tag{1}$$

$$\alpha_i = \sigma(W_v v_i + b_v), \bar{q}, \bar{p} \subset v \tag{2}$$

$$\tilde{v}_i = \frac{1}{|N_i|}\left(\sum^{j\in N_i} \alpha j W^{r_{ji}} v[j]\right) \tag{3}$$

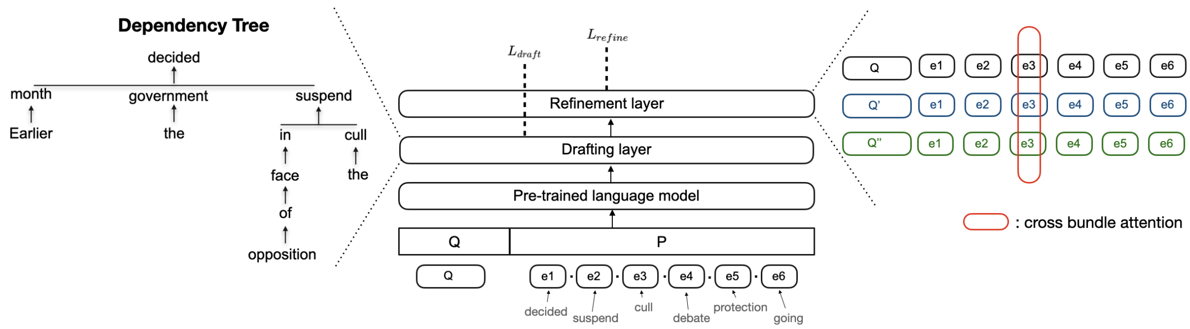

Figure 2: Overview of TDN. Drafting layer extracts evidence from syntactic and semantic features. Deliberation layer aggregates information from temporal bundles and answer the question. We only represent event tokens and part of the dependency tree for simplicity.

$$v_i^{\prime} = ReLU(W_i^g + \tilde{v}_i^{\prime}) + b^g \tag{4}$$

The PLM's hidden outputs pass the projection layer $W^M \in \mathbb{R}^{h \times h}$ for node initialization (Equation 1). If a word is tokenized into multiple tokens, we use the first token embedding of the word.[4] The weight for each node is computed to find the relevant nodes for answering temporal ordering questions (Equation 2). In the message propagation step, the adjacency matrix $W^{r_{ji}}$ guides the distinguished message passing for each type $r_{ji} \in \{pp, pq, qp, qq\}$ (Equation 3). The message representation is added with the corresponding nodes (Equation 4), where W^g is weight and b^g is the bias term. We iterate the reasoning step (Equation 2, Equation 3, Equation 4) for T times. Finally, each passage word representation is summed with $\acute{p_i}$ and normalized. The resulting passage representation is P^d and each word is p^d.

3.2 Refinement Layer

Our second and principal objective is to aggregate local question-answer drafts to approximate an internally consistent timeline. We design a refinement layer with a specialized attention structure to allow optimizing its timeline, constrained by the temporal bundle. The temporal bundle is defined as a set of questions from the same contrast groups in the dataset, or a set of questions asking about the same event. This temporal bundle serves as an approximated timeline for refining the draft. If a temporal bundle with l questions is given, we transform the conventional equation in MRC to answer the i-th question $P(a|Q_i, P)$ to the deliberation form:

$$P(a_i|Q_i, P, P(a_1|Q_1, P), P(a_2|Q_2, P), ...,, P(a_l|Q_l, P)) \tag{5}$$

Due to the unavailability of gold answers during inference, we regard the predictions of the drafting layer as answers for both training and inference. A naive method is to concatenate all the related question-answer pairs and expand them to the original passage. However, since the predictions are used directly as answer events without proper filtering, they may lose the signal of the prediction's uncertainty or importance in answering questions. Therefore, we gather the bundle on the embedding space. The related questions $[Q_i, P_i]_{i=2}^{l}$ is sent to the drafting layer to produce $[P_i^d]_{i=2}^{l}$, then stacked with the original one $[P_i^d]_{i=1}^{l}$. Here, the previous drafting layer encodes the question information into the passage, so passage tokens can independently capture temporal relations related to the question event, and create an approximated timeline.

Then the refinement layer utilizes the timeline structure that is weaved by the temporal bundle. Our key component is the extended multi-head attention mechanism "cross-bundle attention" that attends to the information from the temporal bundle, which is otherwise neglected in the original transformer (Vaswani et al., 2017) and deliberation network (Xia et al., 2017). In detail, each passage token p_k attends to the same positioned token from other instances. The equation is as follows where multi-head attention is $(Attention(Q, K, V))$ where $i, j \leq l$:

$$CrossBundleAttention = Attention(p_{ik}, p_{jk}, p_{jk}) \tag{6}$$

The refinement layer inserts the cross-bundle attention following the self-attention module in the

[4] Note that we also indicate the number of words in a passage as n, as long as there is no confusion between the two.

Models	Dev			Test		
	F1	EM	C	F1	EM	C
RoBERTa-large	75.7	50.4	36	75.2	51.1	34.5
ECONET	76.9	52.2	37.7	76.3	52	37
UBA	77.5	52.2	37.5	76.1	51	**38.1**
OTR-QA	77.1	51.6	**40.6**	76.3	52.6	37.1
TDN (RoBERTa-large)	<u>**77.6**</u>	<u>**53.6**</u>	<u>40.3</u>	<u>**76.9**</u>	<u>**52.8**</u>	**38.1**

Table 1: Comparison between TDN and baselines on TORQUE dataset. All reported results are statistically significant ($p \leq .05$). Underline denotes statistically significant ($p \leq .01$) improvement over the RoBERTa-large baseline, using a paired t-test. The best performance is denoted in bold.

transformer encoder layer. Via the cross-bundle attention, temporal bundles are successfully passed to the original instance. Each event in the passage grabs the semantics of the temporal relations and the timeline that are made by other questions. This refinement step corrects the answer by revisiting the passage and leveraging the approximated timelines, mitigating the shortcut problem we mentioned in section 1. We iterate the refinement T' times to enhance the process of deliberation.

3.3 Learning Objectives and Answer Prediction

For each deliberation level, the last output is fed to the FFN to get the probability of whether the token is an answer to the question or not. During the training stage, We adopt the loss minimization approach by (Xiong et al., 2019; Li et al., 2019). At each level, the last output is fed to the FFN and the resulting loss for answer prediction is computed. The final loss is the average value of the losses at each level:

$$L = (L_{draft} + L_{refine})/2 \tag{7}$$

where L_{draft} is the answer prediction loss from the draft layer's output, and L_{refine} is the loss from the refinement layer's. During the inference stage, the outputs of the refinement layer pass our FFN to be our final logits.

4 Experiment

4.1 Dataset and Evaluation Metrics

We evaluate our proposed model on TORQUE dataset (Ning et al., 2020), which is a temporal reading comprehension dataset. It has 3.2k passages and 21.2k user-provided questions. Each instance has a question asking the temporal relationships between events described in a passage of text. TORQUE's annotation provides groups of questions, where one group consists of questions that were created by modifying the temporal nuance of an original seed question that dramatically changes the answers. Following (Ning et al., 2020), we use the official split and evaluation metrics. All instances are split into 80%/5%/15% for train/dev/test without common passages. We use Macro F1, exact-match (EM), and consistency (C) as evaluation metrics. C (consistency) is the percentage of question groups for which a model's predictions have $F1 \geq 80\%$ for all questions in a group.

4.2 Baselines

We compare our model against several baselines, including a naive PLM and models that use contrastive methods to teach the model temporal relations. Specifically, **OTR-QA** (Shang et al., 2021) reformulates the TORQUE task as open temporal relation extraction and uses contrastive loss to model temporal relations. As they target TORQUE without any external supervision like our method, they are our main baseline. **ECONET** (Han et al., 2020) is a continual pre-training approach with adversarial training that aims to equip models with knowledge about event temporal relations. They use external corpus for continual learning, and compile a lexicon of 40 common temporal expressions to use the discriminator for contrastive learning. **UBA** (Huang et al., 2022) employ the attention-based question decomposition to understand fine-grained questions. **RoBERTa-large** (Liu et al., 2019) is a baseline model provided together with the TORQUE dataset. As RoBERTa-large is the model that the previous works are based on, we choose it for the naive PLM baseline. They also utilize a dictionary of temporal expressions as additional supervision, to capture the distinctions in temporal relationships.

4.3 Experimental Settings

We search hyperparameters, T and T' is {2, 3} for the graph iteration step and for refining step. For the attention mechanism in the refinement layer, each layer has 8 attention heads with a hidden size of 1024. Feedforward layers have dimensions $\{1024, 2048\}$. A temporal bundle consists of questions from the same question group in TORQUE. During the fine-tuning, the gradient accumulation step is set to 1, dropout ratio is set to 0.2 and other settings are identical with (Ning et al., 2020). (Shang et al., 2021) only report the best single

Models	Dev			Test		
	F1	EM	C	F1	EM	C
BERT-large						
Naive	72.8	46.0	30.7	71.9	45.9	29.1
Current SOTA	73.5‡	46.5‡	31.8‡	**72.6**‡	45.1‡	**30.1**‡
TDN	73.1	47.2	32.6	72.3	**46.5**	29.8
DeBERTa-large						
Naive	75.8	50.1	34.9	75.0	49.8	34.3
TDN	77.4	52.7	40.1	**77.0**	**51.6**	**36.9**
RoBERTa-base						
Naive	72.2	44.5	28.7	72.6	45.7	29.9
Current SOTA	75.2†	49.2†	36.1†	73.5‡	**47.1**†	**32.7**†
TDN	73.8	48.9	34.7	**73.7**	**47.1**	32.3

Table 2: Comparison with with PLM variants. Reported results are marked. Naive results are from TORQUE (Ning et al., 2020). Current SOTA results are from OTR-QA (Shang et al., 2021)†, UBA (Huang et al., 2022)‡.

Models	F1	EM	C
TDN	**77.6**	**53.6**	**40.3**
(d) TDN - Self-Attention	77.4	52.2	38.9
(c) TDN - Cross-Bundle Attention	76.3	51.4	38.6
(b) TDN - Refinement Layer	76.0	51.9	38.1
(a) TDN - G_{syn}	76.1	50.9	37.3

Table 3: Ablation study on the dev set of TORQUE. Results are based on RoBERTa-large. The best performance is denoted in bold.

model results for all sets, and (Huang et al., 2022) report single model results on test set, to make the fair comparison with the baselines, we report the averaged score on the dev set and the best score on the test set. But to establish the concrete result, We conducted paired t-tests for both set against the naive baseline to establish the statistical significance of our method. We use the PyTorch 1.11 library, and NVIDIA GeForce RTX 3090 GPUs.

4.4 Experimental Results

Table 1 compares our approach to the baseline methods. The baseline performances are provided by previous works (Ning et al., 2020; Han et al., 2020; Shang et al., 2021; Huang et al., 2022). For the RoBERTa-large model, the results show that TDN outperforms all compared baselines on both splits of TORQUE, even surpassing ECONET and UBA, which use a human-annotated dictionary of temporal expressions. One exception is the consistency score (C) of OTR-QA on dev set. But we note that TDN outperforms it in F1 and EM and generalizes better to the test set, indicated by a much smaller dev-test gap in C (3.5 for OTR-QA vs 2.2 for TDN). On the test set, the result shows that TDN significantly outperforms all the baselines, achieving state-of-the-art results on the TORQUE leaderboard.

4.5 PLM variants

Table 2 displays the results for PLM encoder variants. First, Our method shown to be generalizable to the BERT model, and its performance is comparable to other previous methods. We also implement our method on DeBERTa (He et al., 2021) together with the naive baseline, which is known to perform better than RoBERTa on natural language understanding (NLU) tasks. When using a naive PLM encoder, we found that DeBERTa encoder is slightly worse than RoBERTa in most of the metrics. However, with the addition of TDN, our method achieves the best F1 score, demonstrating the effectiveness and generalizability of our method even with other PLM variants. Lastly, when using the RoBERTa-base model, our results are again comparable to other baselines and surpass them in terms of F1 score, highlighting the scalability of TDN.

4.6 Ablation Study

To validate the effectiveness of each model component, we conduct an ablation study on dev set and report the results in Table 3. In (a) we remove the syntactic graph network component G_{syn} in the draft layer and find the performance decreases significantly. This suggests that syntactic graph reasoning helps the downstream process of deliberation by collecting temporal cues and creating more fine-grained question-aware passage token representations. For the refinement layer, we first remove (b) the whole layer, (c) the cross-bundle attention layer, and (d) the self-attention layer. The performance drops significantly with (b), indicating the importance of the refinement layer. Comparison between (c) and (d) indicates that the refinement layer helps performance gain by virtue of cross-bundle attention. It is the leading part of deliberation by attending over the global temporal bundle for the timeline. Meanwhile, (d) removing the simple stack of the transformer's self-attention part has the least impact on the performance.

5 Discussion

While we empirically validated the effectiveness of TDN, its implication and generalizability can be

Models	F1	EM	C	C_p
(a) Draft + Refine (TDN)	**77.6**	**53.6**	**40.3**	**11.7**
(b) Refine	76.1	50.9	37.3	10.3
(c) *CL*	75.8	51.7	36.8	8.3

Table 4: Comparison of contrastive learning (*CL*), and TDN on the dev set of TORQUE. The best performance is denoted in bold.

further clarified by the following discussion questions:

- Q1: Does TDN mitigate shortcuts?
- Q2: Does TDN generalize to another task?

5.1 Q1: Mitigating shortcuts

We first address the question of whether the performance gain of TDN can be attributed to a better comprehension of the passage timeline (approximated as temporal bundles).

To quantitatively measure whether TDN understands passage timelines, we adopt a passage-level consistency score C_p (Gardner et al., 2020; Ning et al., 2020): If a model understands the passage timeline, its answers will be internally consistent with respect to all questions, which C_p quantifies as a ratio of questions with $F1 \geq 80\%$ [5]. We compare TDN with the model equipped with contrastive learning, which is implemented following OTR-QA's contrastive loss (Shang et al., 2021).

Table 4 shows that C_p of TDN is significantly higher than that of *CL*. To isolate the effect of the Refine phase, where the temporal bundle is used, we also present ablated results removing the draft layer– We observe that even without a draft layer, ours outperforms *CL*, which indicates that the improved understanding of timeline plays a critical role for performance gains [6].

Figure 3 groups F1 gains, by bundle sizes, from which the gap from *CL* widens as the size grows. It is coherent with our hypothesis that TDN gains effectiveness from refining local answers, by comparing with other question-answer in the bundle, which would be more effective for a larger bundle size. Moreover, our method persistently outperforms contrastive loss, even with a small bundle size with a margin of $2.3pp$.

[5]The threshold of 80% follows the convention of (Gardner et al., 2020; Ning et al., 2020).

[6]Though one may argue adding Drafting layer with CL may further improve CL, we found this was not the case (F1 and EM of 75.4 and 50.7 respectively), which is why we report CL itself.

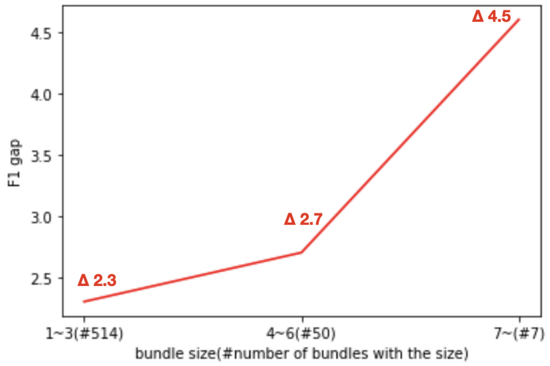

Figure 3: Plot of the relationship between bundle size and F1 score. X-axis is the bundle size, binned into groups of 3. The number of bundles in each bin is denoted in brackets. Y-axis is the gap between the average F1 score of TDN and *CL*, in percentage.

Models	Dev	Test
RoBERTa-large	60.0($\pm$1.1)	62.8($\pm$3.2)
ECONET	60.8($\pm$0.6)	64.8($\pm$1.4)
TDN	60.2($\pm$0.4)	**65.3**($\pm$0.5)

Table 5: Micro-F1 score on the TB-Dense dataset. The best performance on the test set is denoted in bold.

Lastly, as qualitative observations, Figure 4 compares answers from TDN with *CL*: *CL* fails to clearly distinguish the semantic difference between Q1 and Q2, while our understanding of the timeline avoids such mistakes. Ours is aware that "exploded" occurred before the tour ($Q3$), and not after the tour ($Q2$), so it cannot be during the same time as the tour ($Q4$). while *CL* fails. In addition, during the refining process, ours finds the unmentioned events (e.g. "arrested" in $Q1$) and puts them in the right place on the timeline.

5.2 Q2: Generalization

To investigate whether our proposed approach generalizes to a related temporal ordering task, we evaluate on TB-Dense (Cassidy et al., 2014), which is a public benchmark for temporal relation extraction.

For TB-Dense, when the passage and two event points in the passage are given, the model must classify the relations between events into one of 6 types [7]. We implement our method based on the publicly available source code of ECONET (Han et al., 2020) [8]. For the drafting layer, as the question

[7]Though the granularity of temporal understanding required in this task is coarser than in TORQUE, there are no other fine-grained datasets available to evaluate generalizability.

[8]https://github.com/PlusLabNLP/ECONET

P1. After **touring** Tanzanian capital Dar es Salaam Thursday and **meeting** with Kenyan police leaders Friday morning, the FBI chief also **said** that he is very satisfied with the close and effective **cooperation** among the FBI agents and the police in Kenya and Tanzania. The man who **hurled** a grenade at security guards at the U.S. embassy here seconds before the bomb **exploded** was positively **identified** Thursday as two more suspects -- one Arab , one Sudanese -- who had been **arrested**, Kenya 's national newspapers **reported** Friday .	
Q1. What events had started before the FBI chief toured the Tanzanian capital?	*CL*: cooperation, hurled, exploded, __ TDN: cooperation, hurled, exploded, identified, arrested
Q2. What events occured after the FBI chief toured the Tanzanian capital?	*CL*: meeting, said, reported TDN: said, reported
Q3. What events occured before the FBI chief toured the Tanzanian capital?	*CL*: hurled, exploded, __ TDN: hurled, exploded, arrested
Q4. What events occured during the same time that the FBI chief toured the Tanzanian capital?	*CL*: meeting, cooperation, exploded, identified TDN: meeting

Figure 4: Qualitative analysis of contrastive learning and TDN. Events in the passage are highlighted in bold. In answers, correct events are denoted in blue, and incorrect events are denoted in red. Missing events are underlined.

is unavailable in TB-Dense, we prepend two events $e1$, $e2$ to the passage P, and the model input is "[CLS] + e1 + e2 + [SEP] + P + [SEP]". $e1$ and $e2$ have self-linked edges and are bidirectionally connected to their original positions in the passage. For the refinement layer, since no contrast group is available in TB-Dense, we manually group data instances that are asked on the same passage, and they work as a temporal bundle. Hyperparameters for fine-tuning are the same as ECONET. Micro-F1 score is reported by averaging the runs from 3 different seeds. Since ECONET is the only model that targets both fine-grained and coarse-grained temporal ordering, we compare our results with it.

Our method achieves an F1 score of 65.3% on this task, compared to a RoBERTa-large baseline that achieves an F1 score of 62.8%. Moreover, our method outperforms ECONET, which unlike ours, uses an external corpus. These results demonstrates that TDN's ability to build and utilize an approximate timeline is effective at various granularities, and as such, our method has broader applicability beyond the fine-grained temporal ordering task.

6 Conclusion

We introduce a novel approach for temporal machine reading comprehension, Timeline Deliberation Network (TDN), which captures fine-grained temporal orders between events in a passage. To mitigate the shortcut problem in existing works introduced by reliance on answer overlap, we introduce a new dimension of temporal reasoning to the model in the form of a timeline. TDN approximates an internally consistent timeline using question bundles, grouped with respect to events in the passage, as a form of stronger supervision.

TDN consists of a drafting layer which extracts evidence by encoding syntax and semantics of the passage, and a refinement layer which utilize the timeline through a novel attention mechanism. Results on TORQUE and TB-dense datasets demonstrate that TDN outperforms previous methods by effectively mitigating the shortcut problem.

7 Limitations

Despite the promising results, there are some limitations to our approach. One limitation is that our target, fine-grained temporal ordering, while a more realistic setting, is not commonly encountered in current NLP tasks. However, we argue that this is an important area that needs more active research, especially considering applications of NLP models in real-world and real-time scenarios.

Relatedly, there is a lack of standardized datasets for evaluating models in the fine-grained temporal ordering task, and more datasets are required to effectively evaluate models in this setting. We have tried to remedy this issue by testing generalizability on TB-Dense, a related task with lower granularity.

References

Taylor Cassidy, Bill McDowell, Nathanael Chambers, and Steven Bethard. 2014. An annotation framework for dense event ordering. In *Annual Meeting of the Association for Computational Linguistics*.

Kunlong Chen, Weidi Xu, Xingyi Cheng, Zou Xiaochuan, Yuyu Zhang, Le Song, Taifeng Wang, Yuan Qi, and Wei Chu. 2020. Question directed graph attention network for numerical reasoning over text. In *Conference on Empirical Methods in Natural Language Processing*.

Wenhu Chen, Xinyi Wang, and William Yang Wang. 2021. A dataset for answering time-sensitive questions. *arXiv preprint arXiv:2108.06314*.

Fei Cheng and Yusuke Miyao. 2017. Classifying temporal relations by bidirectional lstm over dependency paths. In *Annual Meeting of the Association for Computational Linguistics*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, et al. 2020. Evaluating models' local decision boundaries via contrast sets. *Findings of Empirical Methods in Natural Language Processing*.

Rujun Han, Xiang Ren, and Nanyun Peng. 2020. Econet: Effective continual pretraining of language models for event temporal reasoning. *arXiv preprint arXiv:2012.15283*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention.

Ke Hu, Ruoming Pang, Tara N. Sainath, and Trevor Strohman. 2021. Transformer based deliberation for two-pass speech recognition. *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 68–74.

Hao Huang, Xiubo Geng, Guodong Long, and Daxin Jiang. 2022. Understand before answer: Improve temporal reading comprehension via precise question understanding. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 375–384, Seattle, United States. Association for Computational Linguistics.

Yinya Huang, Meng Fang, Yu Cao, Liwei Wang, and Xiaodan Liang. 2021. Dagn: Discourse-aware graph network for logical reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5848–5855.

Thomas Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *ArXiv*, abs/1609.02907.

Zekang Li, Cheng Niu, Fandong Meng, Yang Feng, Qian Li, and Jie Zhou. 2019. Incremental transformer with deliberation decoder for document grounded conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 12–21, Florence, Italy. Association for Computational Linguistics.

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*.

Puneet Mathur, Rajiv Jain, Franck Dernoncourt, Vlad Morariu, Quan Hung Tran, and Dinesh Manocha. 2021. TIMERS: Document-level temporal relation extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 524–533, Online. Association for Computational Linguistics.

Qiang Ning, Hao Wu, Rujun Han, Nanyun Peng, Matt Gardner, and Dan Roth. 2020. TORQUE: A reading comprehension dataset of temporal ordering questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1158–1172, Online. Association for Computational Linguistics.

Qiang Ning, Hao Wu, and Dan Roth. 2018. A multi-axis annotation scheme for event temporal relations. In *Annual Meeting of the Association for Computational Linguistics*.

Qiu Ran, Yankai Lin, Peng Li, Jie Zhou, and Zhiyuan Liu. 2019. NumNet: Machine reading comprehension with numerical reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2474–2484.

Chao Shang, Peng Qi, Guangtao Wang, Jing Huang, Youzheng Wu, and Bowen Zhou. 2021. Open temporal relation extraction for question answering. In *Conference on Automated Knowledge Base Construction*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio', and Yoshua Bengio. 2017. Graph attention networks. *ArXiv*, abs/1710.10903.

Yingce Xia, Fei Tian, Lijun Wu, Jianxin Lin, Tao Qin, Nenghai Yu, and Tie-Yan Liu. 2017. Deliberation networks: Sequence generation beyond one-pass decoding. In *NIPS*.

Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang. 2018. Modeling coherence for discourse neural machine translation. *ArXiv*, abs/1811.05683.

Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang. 2019. Modeling coherence for discourse neural machine translation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7338–7345.

Shuaicheng Zhang, Qiang Ning, and Lifu Huang. 2022. Extracting temporal event relation with syntax-guided graph transformer. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 379–390, Seattle, United States. Association for Computational Linguistics.

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 *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3363–3369.