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 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) . 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

1The code will be released after blind review.

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

---

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 $Q_1$ is to generate “local” drafts considering only $Q_1$. 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 $[\tilde{Q}, \tilde{P}]$, where each token is $\tilde{q}$ and $\tilde{p}$.

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:

\[
[\tilde{Q}, \tilde{P}] = W^M[Q, P] \tag{1}
\]

\[
\alpha_i = \sigma(W_e v_i + b_e), \tilde{q}, \tilde{p} \subset v \tag{2}
\]

\[
\bar{v}_i = \frac{1}{|N_i|} \left( \sum_{j \in N_i} \alpha_j W_{rji} v[j] \right) \tag{3}
\]
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. 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 $r_{ji} \in \{pp, pq, gp, qq\}$ (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 $p_i$ 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 $P(a_i|Q_i, P)$ to the deliberation form:

$$v_i' = ReLU(W^g_i + v_i') + b^g$$

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 $\mathcal{Q}_i, P_i \mid 1 \leq i \leq 2$ is sent to the drafting layer to produce $[P^d_i]_{i=1}^2$, then stacked with the original one $[P^d_i]_{i=1}^2$. 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 $\text{Attention}(Q, K, V)$ where $i, j \leq l$:

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

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

$$L = (L_{draft} + L_{refine})/2$$  \hspace{1cm} (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

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.
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)†.

<table>
<thead>
<tr>
<th>Models</th>
<th>Dev F1</th>
<th>EM</th>
<th>C</th>
<th>Test F1</th>
<th>EM</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT-large</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Naive</td>
<td>72.8</td>
<td>46.0</td>
<td>30.7</td>
<td>71.9</td>
<td>45.9</td>
<td>29.1</td>
</tr>
<tr>
<td>Current SOTA</td>
<td>73.5‡</td>
<td>46.5‡</td>
<td>31.8¶</td>
<td>72.6‡</td>
<td>45.1‡</td>
<td>30.1‡</td>
</tr>
<tr>
<td>TDN</td>
<td>73.1</td>
<td>47.2</td>
<td>32.6</td>
<td>72.3</td>
<td>46.5</td>
<td>29.8</td>
</tr>
<tr>
<td>DeBERTa-large</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Naive</td>
<td>75.8</td>
<td>50.1</td>
<td>34.9</td>
<td>75.0</td>
<td>49.8</td>
<td>34.3</td>
</tr>
<tr>
<td>TDN</td>
<td>77.4</td>
<td>52.7</td>
<td>40.1</td>
<td><strong>77.0</strong></td>
<td><strong>51.6</strong></td>
<td><strong>36.9</strong></td>
</tr>
<tr>
<td>RoBERTa-base</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Naive</td>
<td>72.2</td>
<td>44.5</td>
<td>28.7</td>
<td>72.5</td>
<td>45.7</td>
<td>29.9</td>
</tr>
<tr>
<td>Current SOTA</td>
<td>75.2‡</td>
<td>49.2‡</td>
<td>36.1¶</td>
<td>75.1†</td>
<td>47.1¶</td>
<td>32.7¶</td>
</tr>
<tr>
<td>TDN</td>
<td>73.8</td>
<td>48.9</td>
<td>34.7</td>
<td><strong>73.7</strong></td>
<td><strong>47.1</strong></td>
<td><strong>32.3</strong></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Models</th>
<th>F1</th>
<th>EM</th>
<th>C</th>
<th>C_p</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Draft + Refine (TDN)</td>
<td>77.6</td>
<td>53.6</td>
<td>40.3</td>
<td>11.7</td>
</tr>
<tr>
<td>(b) Refine</td>
<td>76.1</td>
<td>50.9</td>
<td>37.3</td>
<td>10.3</td>
</tr>
<tr>
<td>(c) CL</td>
<td>75.8</td>
<td>51.7</td>
<td>36.8</td>
<td>8.3</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Models</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>RoBERTa-large</td>
<td>60.0±1.1</td>
<td>62.8±3.2</td>
</tr>
<tr>
<td>ECONET</td>
<td>60.8±0.6</td>
<td>64.8±1.4</td>
</tr>
<tr>
<td>TDN</td>
<td>60.2±0.4</td>
<td>65.3±0.5</td>
</tr>
</tbody>
</table>

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.

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 \( F1 \geq 80\% \). 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.

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.

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

6Though 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.

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.

[1] 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. We implement our method based on the publicly available source code of ECONET (Han et al., 2020). 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.

8https://github.com/PlusLabNLP/ECONET
is unavailable in TB-Dense, we prepend two events $e_1, e_2$ to the passage $P$, and the model input is 
$"[CLS] + e_1 + e_2 + \text{SEP} + P + \text{SEP}"$. $e_1$ and $e_2$ 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 demonstrate 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


