An Event-based Abductive Learning for Hard Time-sensitive Question Answering

Shaojuan Wu [*] , Jitong Li [*], Xiaowang Zhang [†] , Zhiyong Feng
College of Intelligence and Computing, Tianjin University
Tianjin Key Laboratory of Cognitive Computing and Application
Tianjin, China
{shaojuanwu, lijitong, xiaowangzhang, zyfeng}@tju.edu.cn

Abstract

Time-Sensitive Question Answering (TSQA) is to answer questions qualified for a certain timestamp based on the given document. It is split into easy and hard modes depending on whether the document contains time qualifiers mentioned in the question. While existing models have performed well on easy mode, their performance is significantly reduced for answering hard time-sensitive questions, whose time qualifiers are implicit in the document. The intuitive idea is to match temporal events in the given document by treating time-sensitive questions as a temporal event of missing objects. However, not all temporal events extracted from the document have explicit time qualifiers. In this paper, we propose an Event-AL framework in which a graph pruning model is designed to locate the timespan of implicit temporal events by capturing temporal relations between events. Moreover, we present an abductive reasoning module to determine proper objects while providing explanations. Besides, as the same relation may be scattered throughout the document in diverse expressions, a relation-based prompt is introduced to instruct LLMs in extracting candidate temporal events. Extensive experiments and results show that Event-AL outperforms strong baselines for hard time-sensitive questions, with a 12.7% improvement in EM scores. In addition, it also exhibits great superiority for multi-answer and beyond hard time-sensitive questions.

Keywords: Time-Sensitive Question Answering, Temporal Event, Abductive Reasoning, Graph Pruning

1. Introduction

Time-Sensitive Question Answering (TSQA) is to answer questions containing time qualifiers based on the given document (Chen et al., 2021; Wang et al., 2022). According to statistics, about 48% of the qualifiers are related to time in the Wikidata (Vrandecic and Krötzsch, 2014), widely used in daily life. So, it is crucial for applying language models to the real world (Tan et al., 2023). To evaluate the levels of models over temporal reasoning, TSQA is further split into easy and hard modes depending on whether the document contains time qualifiers mentioned in the question. As shown in Figure 1(a), time qualifiers are explicit in the document for easy questions, such as "*Which team did Attaphol Buspakom play for* **from 1985 to 1989**?". However, hard time-sensitive questions are implicit, such as "*Which team did Attaphol Buspakom play for* **from 1989 to 1990**?". Therefore, answering hard questions without explicit time mentioned in the given document is not trivial.

Existing models have struggled to address hard time-sensitive questions, though they have made great progress for easy mode (Zaheer et al., 2020; Izacard and Grave, 2021; Raffel et al., 2020), as shown in Figure 1(b). Pre-trained language models may excel at representing the text itself by fine-tuning large-scale data. Amazingly, Large Language Models (LLMs) perform poorly in TSQA, although they have recently demonstrated remarkable capabilities in solving various complex reasoning tasks, including mathematical reasoning (Imani et al., 2023; Stolfo et al., 2023), logical reasoning (Choudhary and Reddy, 2023). Several recent efforts indicate that LLMs may have difficulty understanding temporal concepts, including sequential, overlapping, and inclusive relationships between dates (Zhu et al., 2023; Nye et al., 2021). From another perspective, ChatGPT has comparable performance for answering easy and hard questions. Therefore, exploring hard questions implicit in the document and LLMs is interesting.

The intuitive idea is to match temporal events in the given document by treating the time-sensitive question as a temporal event without an object (i.e., $(s, r, ?o, t)$), where a temporal event consists of the subject s, relation r, object o and time qualifiers t. For example, "*Which team did Attaphol Buspakom play for from 1985 to 1989*?" could be parsed as "*(Attaphol Buspakom, play for, $?o$, from 1985 to 1989)*". However, not all candidate temporal events have explicit time qualifiers (i.e., $(s, r, o, ?t)$) since they have not existed in the document, such as temporal event 'E3' in Figure1. It severely hinders the feasibility of this idea.

In this paper, we propose a novel Event-based

[*] These authors contributed equally to this work and should be considered co-first authors.
[†] Corresponding author

LREC-COLING 2024, pages 1105–1115

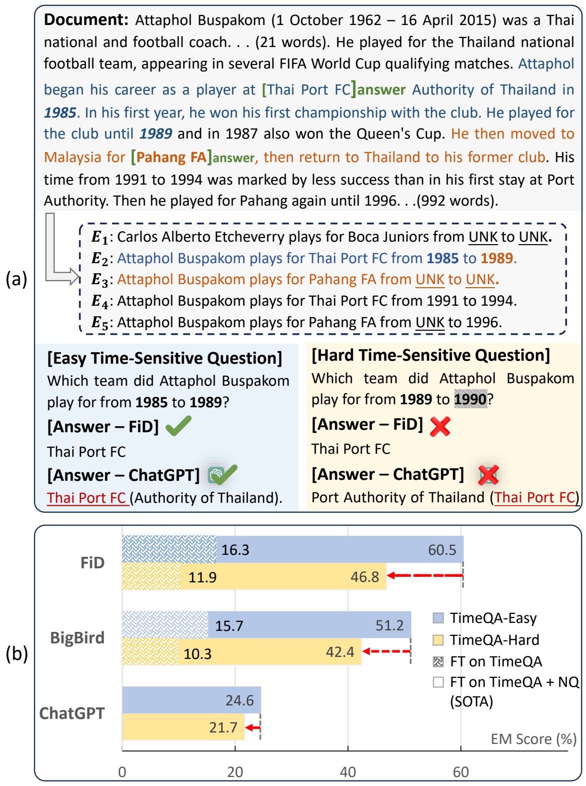


Figure 1: Comparison of easy and hard mode from TimeQA. (a) Examples of time-sensitive questions. (b) The Exact Match (EM) scores under Easy and Hard modes.

Abductive Learning (Event-AL) framework for mining answers to hard time-sensitive questions, as shown in Figure 2. Specifically, we first design a graph pruning model to locate the timespan of implicit temporal events through adjacent explicit temporal events. It is attributed to the fact that temporal events always occur linearly under the same subject and relation. However, multiple implicit temporal events are located in the same time span. For this reason, we propose an abductive reasoning module to determine proper objects while providing an explanation. From our experiments, models struggle to extract enough temporal events in many cases, as the same relations are often scattered throughout documents in diverse representations. To avoid these issues, we introduce a relation-based prompt to instruct LLMs to extract all possible temporal events that have the same relation with the time-sensitive question. Our experiments show that Event-AL significantly outperforms strong baselines, especially for hard and beyond-hard questions.

In short, our contributions are listed as follows:

- We present Event-AL, which locates the timespan of temporal events without explicit time qualifiers by adjacent temporal events. It is achieved by the graph pruning mechanism based on event temporal relation extraction.
- We propose an abductive reasoning module that determines proper objects as the answer while providing explanations.
- We conduct extensive experiments on two benchmarks, TimeQA and TempReason. Results show that Event-AL effectively answers time-sensitive questions and outperforms strong baselines for hard and beyond-hard questions by a large margin. Moreover, Event-AL achieves new state-of-the-art results in multi-answer time-sensitive questions.

2. Related Work

TSQA is proposed to diagnose the ability of models over temporal reasoning. It could be divided into two categories according to the style of given context. An important type is selecting an entity in the knowledge graph as the answer to time-sensitive questions (Jia et al., 2018a; Neelam et al., 2021; Jia et al., 2018b, 2021; Saxena et al., 2021; Chen et al., 2022). As it is less useful for real-world applications, another is proposed to answer time-sensitive questions based on the document (Wang et al., 2022; Zhang and Choi, 2021; Dhingra et al., 2022). In this line, existing methods(Chen et al., 2021; Tan et al., 2023) have made great progress by fine-tuning pre-trained language models (Zaheer et al., 2020; Izacard and Grave, 2021) on large-scale data. DocTime (Mathur et al.) incorporates the temporal dependency graph into the self-attention layer of Transformer models. In addition, LLMs have been used for temporal question answering. (Li et al., 2023) combines LLMs' extraction capability and a python solver's logical reasoning capability. QAaP (Zhu et al., 2023) first represents diversely expressed text as well-structured code by LLMs and thereby chooses answers through programming.

Our method is similar to the last line of works in that we also use LLMs for extracting temporal events. However, there are the following key differences. First, we focus on locating the timespan of temporal events without explicit time qualifiers rather than simply matching time-sensitive questions and temporal events extracted from the given document. Second, it is challenging for these prior methods to answer time-sensitive questions, especially for hard and multi-answer questions. In our experiments, we have shown results and examples by utilizing these methods and Event-AL together (see details in Table 1 and Figure 4). Finally, Besides answering questions, we found it crucial to utilize explanations for answer validation while answering questions, which is a unique and essential aspect of Event-AL.

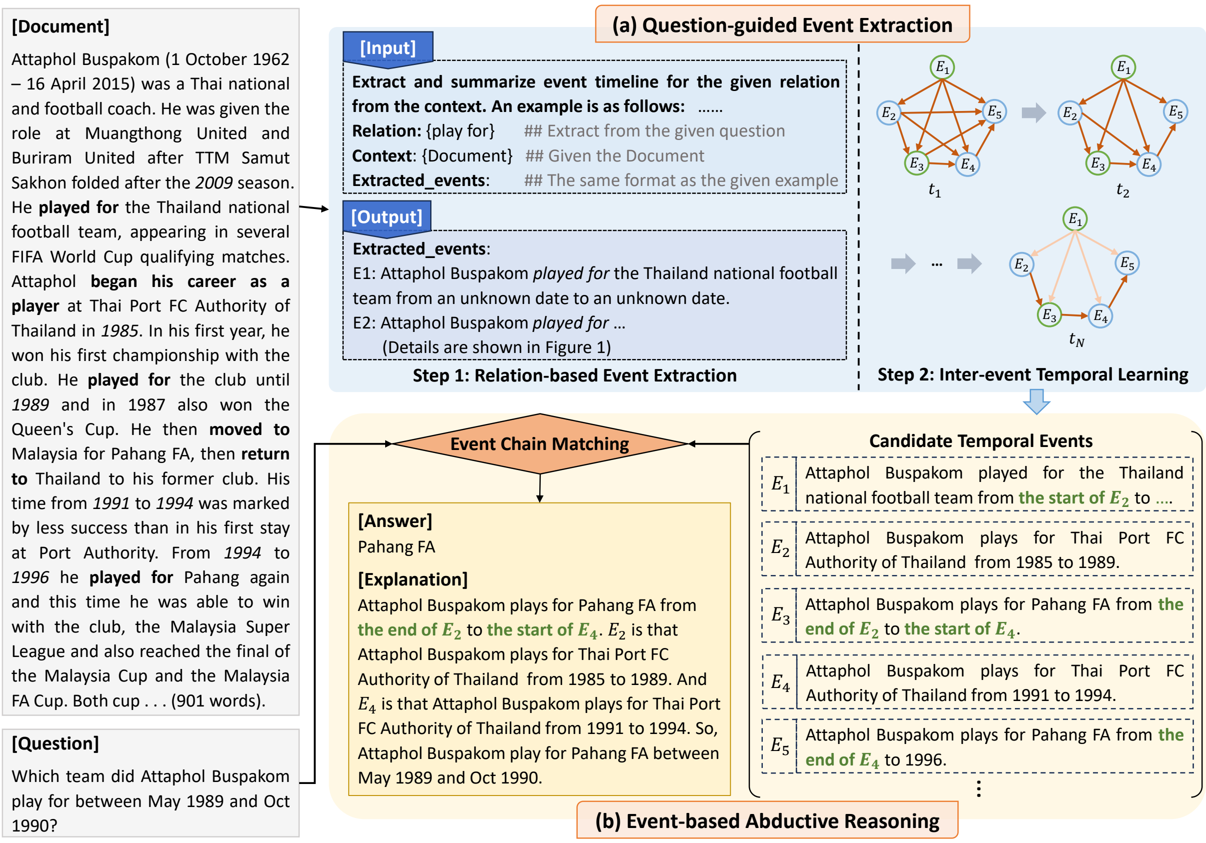


Figure 2: The overall framework of Event-AL. It consists of a question-guided event extraction module for capturing and completing candidate temporal events related to the question, and an event-based abductive reasoning module for obtaining and checking answers.

3. Our Approach

This section introduces Event-AL in three parts: problem statement, question-guided event extraction, and event-based abductive reasoning. The overall framework is shown in Figure 2.

3.1. Problem Statement

Time-Sensitive Question Answering (TSQA) is defined as a task to answer questions q containing time qualifiers t^q based on the corresponding long document d, given the dataset $D = \{(q_i, d_i, a_i), i = 1, 2, \ldots, N\}$, where N denotes the number of instances and each instance consists of the question q_i, the long document d_i and the ground-truth answer a_i. Specially, each question q_i contains subject s_i, relation r_i and a certain time qualifier t_i^q, such as, "*Which team did Attaphol Buspakom* (s) *play for* (r) *between 1989 and 1990* (t^q)?". The corresponding long document d_i covers various relations and timestamps of the subject s_i except for question mentions. Moreover, time qualifier t may be expressed by a single element, such as "in 1989" or a binary (t_s, t_e) consisting of a start time t_s and an end time t_e, such as "from 1989 to 1990". Therefore, TSQA requires the model to correctly understand temporal concepts and return an answer a_i within the long document d_i, such that the quaternary (s_i, r_i, a_i, t_i^q) conforms to the document d_i. In addition, we further explain why the answer a_i is valid, as shown in Figure 2.

3.2. Question-guided Event Extraction

Given an instance $x = (q_i, d_i)$, we first prompt LLMs to parse the time-sensitive question q_i as a temporal event missing object o, represented as $q_i = (s, r, ?, t)$. To obtain the target object, we extract temporal events with relation r from the given long document through event extraction and temporal learning. The following takes binary (t_s, t_e) as an example to explain.

Step 1: Relation-based Event Extraction
Considering documents are long and redundant, we first instruct LLMs to extract only temporal events with relation r by feeding it into the prompt. As shown in **Step 1** of Figure 2(a), we manually design the prompt consisting of four keys: task description I, demonstration D, relation r, and the given document d_i. The set of temporal events

$E = \{E_1, E_2, \ldots, E_n\}$ is generated, formulated as:

$$\text{LLMs}\,(I, D, r, d_i) = E \tag{1}$$

where $n = |E|$ is the number of temporal events with relation r. D is the set of examples, each consisting of the relation, document, and extracted temporal events, represented as $D = \{(r_j, d_j; E_{j1}, E_{j2}, \ldots, E_{jk}), j = 1, 2, \ldots, l; k = 1, 2, \ldots, j_m\}$, where l is the number of examples and j_m is the number of temporal events in the document d_j. We adopt one-shot example, setting l to 1, following previous works (Li et al., 2023).

Step 2: Inter-event Temporal Learning

After extracting temporal events, there are many events without explicit time qualifiers since they do not exist in the given document. To address this issue, we first construct a directed temporal event graph $G = (V, R)$, where $V = \{E_1, E_2, \ldots, E_n\}$ is a set of extracted temporal events, $\mathbf{R} \in \mathbb{R}^{n \times n}$ is an adjacency matrix and its element r_{ij} denotes the temporal relation between two events E_i and E_j, calculated as follows:

$$r_{ij} = \begin{cases} 1, & t(E_j) > t(E_i); \\ 0, & \text{else} \end{cases} \tag{2}$$

where $t(E_i)$ denotes the timestamp of the event E_i. If an event is missing the specific timestamp, we assume it has the same chronological order described in the document.

Then, we design a new graph pruning method to minimize the timespan of temporal events without explicit time qualifiers by reducing the number of edges. It is formalized as:

$$r_{ij}^{prun} = \begin{cases} 0, & r'_{(i-1)j} r_{i(j+1)} > r'_{ij}; \\ r'_{ij}, & \text{else} \end{cases} \tag{3}$$

$$r'_{ij} = r_{ij} * P(E_j | E_i, R_{next}) \tag{4}$$

$$P(E_j | E_i, R_{next}) = ETRE(seq[E_i, E_j]) \tag{5}$$

where $P(E_j|E_i, R_{next})$ is the predicted probability distribution of the next occurrence of E_j at E_i. $seq[E_i, E_j]$ represents the shortest sequence containing temporal events E_i and E_j extracted from document d_i. $ETRE(\cdot)$ denotes an event temporal relation extraction model. It is worth noting that we employ the event temporal relation extraction model(Wen and Ji) to predict the chronological relations between temporal events E_i and E_j. It is fine-tuned on the TSQA dataset based on a strong baseline (Huang et al., 2023) pretrained in the temporal relation extraction benchmark MATRES(Ning et al., 2018).

Finally, we fill in the 'UNK' of temporal events without explicit time qualifiers by their adjacent events and corresponding temporal relations. For example, the implicit temporal event E_5 "Attaphol Buspakom plays for Pahang FA from UNK to 1996." is updated to "Attaphol Buspakom plays for Pahang FA from the end of E_4 to 1996.", as shown in Figure 1(b). As a result, we end up with a temporal event graph completed by temporal relations between events.

Algorithm 1 Event Chain Matching

Input: query $q = (s, r, ?o, t)$ parsed by question, candidate temporal events $C_E = \{E_1, E_2, \ldots, E_n\}$ extracted from given document.
Output: predicted answer A and the explanation R_E.

for E_i in C_E **do**
  # parse candidate fact E_i
  E_i = (s, r, a_i, o_i)
  # Check the consistency of the question with the candidate fact
  $flag \leftarrow Check((s, r) \in q_i, (s, r) \in E_i)$
  if $flag$ is True **then**
    # Calculate matching score of the time
    $Scores \leftarrow Match(t \in q_i, t \in E_i)$
  else
    # mismatched event
  end if
end for
# Select the candidate temporal events E_a with the highest scores
$A \leftarrow o_a, o_a \in E_a$
$R_E \leftarrow EventChain(E_a)$
return A and R_E

3.3. Event-based Abductive Reasoning

The abductive reasoning module predicts answers and infers the most plausible explanations, critical for answering hard questions. The algorithm for abductive reasoning is presented in Algorithm 1.

For each time-sensitive question $q_i = (s, r, ?o, t)$, we take the set of extracted temporal events as candidate temporal events so that the interpretation provided is consistent with the given document, where each candidate fact E_j is also parsed into a python tuple similar to the question q_i:

$$C_E = \{E_1, E_2, \ldots, E_n\}$$

$$E_j = (s, r, o_j, t_j)$$

Since the candidate temporal events and questions are presented in code form, we can easily construct a task-specific execution function to find the best matching answer and the corresponding interpretation by a Python solver(Li et al., 2023), defined as follows:

$$A, R_E \leftarrow F(q, C_E)$$

For each candidate fact $E_i = (s, r, a_i, o_i)$, we first check that the two key values s, r are the same as the question q_i to ensure that the extracted fact is relationally consistent with the question fact. To obtain the final answer A, we use the intersection of times as the matching score to align the question time with the time of each candidate fact. Eventually, we select the candidate fact with the highest score as the answer.

Moreover, for each answer A, the corresponding chain of events can be easily oriented to the corresponding explanation R_E based on the previously constructed temporal event graph.

4. Experiments

4.1. Datasets

We evaluate our method on two widely used TSQA datasets containing numerous hard questions.

TimeQA (Chen et al., 2021) is proposed to investigate the TSQA task, in which question-answer pairs are generated based on the annotated time-sensitive fact obtained by hiring crowd workers. It is divided into two modes: 'hard' and 'easy', where easy questions tend to have explicit mentions in the document, while hard questions are implicit in the content and the mentioned time hardly ever appears directly, leading to them not being easily retrieved by key information matching.

TempReason (Tan et al., 2023) is a more comprehensive benchmark for time-sensitive question answering at present, created to probe the temporal reasoning ability of large language models more systematically. Our experiments are constructed on **Open Book Question Answering (OBQA)**, the most challenging among three different context settings, to evaluate the model's ability in temporal grounding and temporal reasoning. In addition, we choose two relatively complex questions, (L2) time-event and (L3) event-event, to highlight the ability of our model. It is worth noting that both L2 and L3 are hard questions.

4.2. Baselines

To provide a more comprehensive analysis of our method, we compared it with the following two types of baselines:

- To demonstrate the effectiveness of the abductive learning framework, compare it with the fine-tuned pre-trained language models: **BigBird** (Zaheer et al., 2020) proposes two attention mechanisms to capture local and global information, respectively. **FiD** (Izacard and Grave, 2021) generates the answer token by token in an auto-regressive fashion. We show the best performance of both BigBird and FiD, fine-tuned on large-scale Natural Questions and TimeQA datasets. **TempT5** (Tan et al., 2023) performs supervised fine-tuning of conventional T5 models.
- To prove the usefulness of temporal learning, compare with the existing well-known few-shot large language models: **Chain-of-Thought (CoT)** (Wei et al., 2022), focuses on generating the final answer step by step. **ReAct** (Yao et al., 2023) incorporates external knowledge through additional search and lookup actions. **QAaP** (Zhu et al., 2023) employs ChatGPT to extract structured facts and convert TSQA into program execution.

4.3. Implementation Details

To produce overall experimental results, we randomly selected 300 questions, ensuring that the existing model achieves similar performance with the test sets. It consists of four modes, involving easy and hard in the TimeQA as well as time-event and event-event in the TempReason. Due to cost concerns, we further sampled 200 questions from each of the three sub-datasets (except Event-Event, including only single-answer questions) for both single- and multi-answer questions. In addition, we employ GPT-3.5-Turbo as the model for all experiments unless otherwise specified.

For the event temporal relation extraction model in the inter-event temporal learning module, we first follow the experimental setup of the original paper(Huang et al., 2023). To adapt it to the long document scenario, we re-annotate the chronological relation between temporal events from the same document in the TimeQA dataset. Then, the ETRE model is further fine-tuned.

Followed by prior works (Chen et al., 2021; Tan et al., 2023), we evaluate the performance at the answer and token level using Exact Matching (EM) and F1 scores, respectively. In addition, we also evaluate multi-answer questions by Strict Exact Match (SEM), which is counted as correct if and only if all ground-truth answers are matched.

4.4. Results on TimeQA

Results are presented in Table 1. It could be observed that Event-AL achieves the best performance among all methods at the answer level (EM) and token level (F1) on both easy and hard modes. Specifically, compared with the fine-tuned pre-trained language models, Event-AL improves easy and hard modes by 6.7% and 12.7% in terms of EM scores, as well as results in 5.9% and 12.4% improvement in F1 scores. It demonstrates that our proposed method is effective by abductive reasoning, especially for hard questions. Compared

Backbone	Method	TimeQA				TempReason (OBQA)			
		Easy		Hard		Time-Event		Event-Event	
		EM	F1	EM	F1	EM	F1	EM	F1
Fine-tune (PLMs)	BigBird (Zaheer et al., 2020) ▷	51.2	60.3	42.4	50.9	-	-	-	-
	FiD (Izacard and Grave, 2021) ▷	60.5	67.9	46.8	54.6	-	-	-	-
	TempT5 (Tan et al., 2023) ▷	-	-	-	-	15.4	36.3	21.1	32.4
	BigBird (Zaheer et al., 2020) ◇	55.0	65.1	47.3	56.4	-	-	-	-
	FiD (Izacard and Grave, 2021) ◇	56.3	67.9	49.3	58.0	47.7	58.2	48.3	55.8
	TempT5 (Tan et al., 2023) ◇	31.7	39.7	30.3	38.5	43.3	54.5	41.0	48.9
Few-shot (ChatGPT)	CoT (Wei et al., 2022) ◇	40.3	53.3	25.3	29.8	35.7	41.4	36.7	48.3
	ReAct (Yao et al., 2023) ◇	45.0	55.1	28.3	34.4	39.3	45.6	42.7	50.8
	QAaP (Zhu et al., 2023) ◇	46.3	54.4	41.7	55.3	43.7	50.1	45.3	48.3
Ours	**Event-AL**	**63.0**	**73.8**	**61.7**	**70.4**	**55.3**	**62.8**	**58.0**	**59.5**
	w/o Temporal Learning	55.3	66.5	56.7	68.6	49.3	53.3	50.3	50.5
	w/o Abductive Reasoning	59.3	69.1	54.0	66.2	51.7	58.6	55.3	56.3

Table 1: Results on TimeQA and TempReason (OBQA), where '▷' denotes that results are from the original paper, tested on the entire test dataset and '-' means that no result was reported. '◇' denotes results are reproduced on our sampled subset.

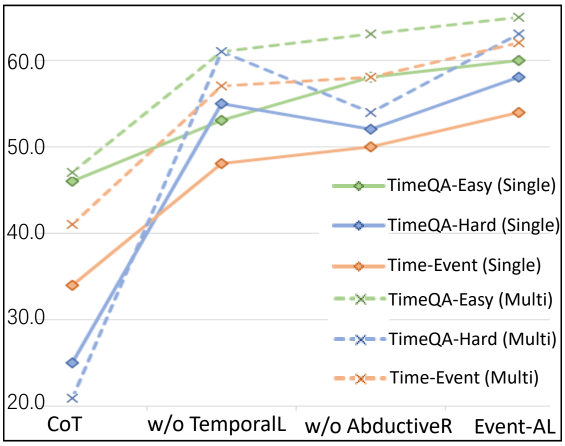
to other baselines based on ChatGPT, Event-AL increases EM scores by 16.7% and 20.0%, as well as F1 scores by 18.7% and 15.1% on easy and hard modes, respectively. This implies that Event-AL has a significant advantage in understanding temporal concepts by learning relations between events.

4.5. Results on TempReason

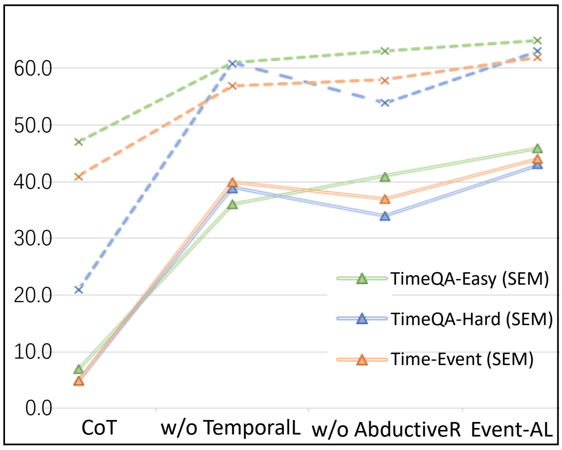
To further illustrate the model's effectiveness in dealing with inter-event relations, we evaluate models on TempReason with both time-event and event-event settings, as shown in Table 1. Event-AL also achieves the best performance among all methods on both metrics for time-event and event-event questions, which are hard questions. Event-AL achieves the EM score of 55.3% and 58.0% for time-event and event-event questions, which outperforms baselines by 7.6% and 9.7%. It suggests that Event-AL is superior in dealing with both hard questions. Event-AL performs remarkably well for event-event questions, maybe because it has performed explicit temporal learning to capture temporal relations between events. In addition, it is noted that the results of models are highly dependent on test samples selected from the TempReason. As a result, we reproduce all baselines and attain experimental results with the same test set, which are slightly higher than those reported in the original article.

5. Analysis and Ablation Study

In this section, we further analyze the performance of Event-AL for single- and multi-answer questions,



(a) EM scores for Single- and Multi-answer questions



(b) EM and SEM scores for Multi-answer questions

Figure 3: Ablation Study for Single- and Multi-answer questions on two modes of TimeQA and TempReason (Time-Event).

Models			Fine-tune (PLMs)			Few-shot (ChatGPT)			Event-AL
			BigBird	FiD	TempT5	CoT	ReAct	QAaP	(Ours)
TimeQA-Easy	Single	EM	52.0	59.0	32.5	46.0	51.0	46.0	**60.0**
		F1	64.7	66.8	47.8	57.2	58.9	53.6	**70.7**
	Multi	EM	53.0	61.0	37.0	47.0	43.0	48.0	**65.0**
		SEM	-	-	-	7.0	11.0	25.0	**46.0**
		F1	61.7	67.2	50.1	60.9	53.8	59.2	**74.5**
TimeQA-Hard	Single	EM	42.0	44.0	28.5	25.0	31.0	40.0	**58.0**
		F1	48.7	52.2	34.9	27.9	36.1	55.1	**70.2**
	Multi	EM	45.0	50.0	39.0	21.0	30.0	45.0	**63.0**
		SEM	-	-	-	5.0	8.0	23.0	**43.5**
		F1	52.9	62.2	52.8	27.1	42.8	56.6	**72.2**
Time-Event	Single	EM	-	45.0	42.0	34.0	39.0	43.0	**54.0**
		F1	-	54.6	52.2	39.9	45.2	50.8	**61.7**
	Multi	EM	-	53.3	44.0	41.0	38.0	50.0	**62.0**
		SEM	-	-	-	5.0	4.0	35.0	**44.0**
		F1	-	63.7	58.9	49.8	47.6	53.2	**69.9**

Table 2: Results of Single- and Multi-answer questions on TimeQA and TempReason (OBQA).

as well as the effectiveness of proposed modules: temporal learning and abductive reasoning.

5.1. Analysis of Single- and Multi-answer Questions

To further evaluate the performance of models on single- and multi-answer questions, we have conducted fine-grained experiments for two cases on both TimeQA and TempReason datasets separately, as shown in Table 2. Event-AL significantly outperforms all baselines for single- and multi-answer questions, especially for SEM scores of the latter. It performs 46.0%, 43.5%, and 44.0% on TimeQA-Easy, TimeQA-Hard, and Time-Event, which outperforms baselines by an average of around 20 points.

From the perspective of the same model, EM and F1 scores are mostly approximate for both single- and multi-answer questions. We conjecture that it is because models tend to treat single- and multi-answer questions in the same way. Moreover, multi-answer questions outperform single-answer for some models, including Event-AL, because it is often easier to get one of the answers than a single answer. While the SEM of multi-answer questions drops for almost all baselines, the reason could be these models generate the terminator as soon as one answer is generated, making them merely succeed in obtaining one of the answers. From the experiment, PLMs are almost unable to obtain multiple answers, maybe because they only select one of the answers for fine-tuning. In addition, since event-event questions only have one answer, the results of single-answer and overall questions are equivalent.

5.2. Effectiveness of Inter-event Temporal Learning

To have a clear view of the role that inter-event Temporal Learning (TemporalL) plays in Event-AL, we perform ablation studies by removing it from our model (i.e., w/o TemporalL), as shown in Table 1 and Figure 3.

Event-AL outperforms best for all three questions (overall, single- and multi-answer) on both datasets. As a whole, Event-AL obtains improved performances, especially for Event-Event, by comparing with *Event-AL* and *w/o Temporal Learning* in Table 1. It indicates that this module is helpful in learning temporal relations between events. As shown in Figure 3(a), there is the smallest drop on TimeQA-Hard for either single- or multiple-answer questions after removing TemporalL. The reason could be that temporal relations between events with explicit timestamps are easier to infer, making it skilled at easy questions. Moreover, SEM of multi-answer questions shows the largest decrease on TimeQA-Hard, compared to others, even though EM scores are all substantially lower than SEM, as shown in Figure 3(b). This suggests that temporal reasoning could facilitate the model to capture all possible answers, even for hard questions.

5.3. Effectiveness of Event-based Abductive Reasoning

To take a deep look into improvements contributed by event-based Abductive Reasoning (AbductiveR) in Event-AL, we perform ablation studies by removing it from our model (i.e., w/o AbductiveR), as shown in Table 1 and Figure 3.

Event-AL also performs ideally on all three types of questions. Firstly, by comparing *Event-AL* and

[Document]
Oliver Bulleid (19 September 1882-25 April 1970) was a British railway and mechanical engineer ... (81 words). **In 1901**, after a technical education at Accrington Grammar School, **he joined the Great Northern Railway (GNR) at Doncaster up to the age of 25**, during which time as an apprentice under H. A. Ivatt ... (35 words). Then, **he left to work in Paris with the French division of Westinghouse Electric Corporation as a Test Engineer from October 1908 to 1910**, and was soon promoted to Assistant Works Manager and Chief Draughtsman. Later that year, he married Marjorie Ivatt, Henry Ivatts youngest daughter. **A brief period working for the Board of Trade followed after Westinghouse**, arranging exhibitions in Brussels, Paris and Turin. **In December 1912, he rejoined the GNR as Personal Assistant to Nigel Gresley**, the new CME. Gresley was only six years Bulleids senior ... (1277 words).

Model	**[Input - Easy Question]** Which employer did Oliver Bulleid work for from 1908 to 1910?	**[Hard Question]** Oliver Bulleid was an employee for whom from 1907 to 1908?	**[Beyond-Hard Question]** Oliver Bulleid was an employee for whom in Dec 1911?
FiD	**[Output]** Westinghouse Electric Corporation	**[Output]** Westinghouse Electric Corporation	**[Output]** Great Northern Railway (GNR)
ChatGPT-CoT	**[Output]** Westinghouse Electric Corporation **(Explanation)** Oliver Bulleid worked for the French division of Westinghouse Electric Corporation from 1908 to 1910.	**[Output]** Westinghouse Electric Corporation **(Explanation)** From 1907 to 1908, Oliver Bulleid was an employee of the French division of Westinghouse Electric Corporation in Paris.	**[Output]** Unknown **(Explanation)** The provided context does not mention Oliver Bulleid's employment or activities in Dec 1911.
Event-AL	**[Output]** Westinghouse Electric Corporation **(Explanation)** Oliver Bulleid worked for the French division of Westinghouse Electric Corporation from Oct 1908 to 1910.	**[Output]** Great Northern Railway (GNR) **(Explanation)** Oliver Bulleid worked for Great Northern Railway (GNR) from 1901 to Oct 1908.	**[Output]** Board of Trade **(Explanation)** Oliver Bulleid worked for Board of Trade between end of E2 between start of E4. E2 is taht Oliver Bulleid worked for Westinghouse Electric Corporation from 1908 to 1910. E4 is that Oliver Bulleid worked for Great Northern Railway from 1912 to UNK.
	All three models answer correctly because they all locate the time mentioned in the question.	FiD and ChatGPT struggle to get relations of stated time and events, thus naturally notice events that are close to the mentioned time.	Only EVent-AL captures the timestamp of target event.

Figure 4: An example from Time-QA, including easy, hard and beyond hard questions, as well as their outputs utilizing three typical models: FiD, ChatGPT and Event-AL.

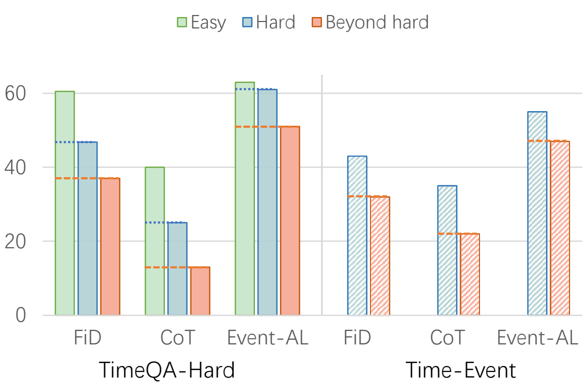


Figure 5: Comparison of performance (EM scores) on Easy, Hard and Beyond hard questions.

w/o Abductive Reasoning in Table 1, it can be noted that the overall performance decreases after removing AbductiveR, especially on TimeQA-Hard. The reason could be that the abductive reasoning framework forces the model to capture implicit information relevant to the question in the given document. Then, the performance of multi-answer questions decreases more than that of single-answer, as shown in Figure 3(a). In addition, the decline of EM is more moderate compared to SEM scores, as shown in Figure 3(b). The reason could be that the abductive reasoning framework enables the model to check the correctness of predicted answers while getting them. This eliminates situations where one answer is negated due to generating another, or multiple answers are generated to fill in the blanks.

6. How Effective is Beyond Hard Questions?

A follow-up question is how effective Event-AL becomes beyond hard questions. Beyond hard is evidenced by the fact that temporal qualifiers in the question do not explicitly appear in the given document, while the complete timestamp of its associated event is not mentioned. For example, neither temporal qualifiers "from 1907 to 1908" nor "Dec 1911" exist in the above document, as shown in Figure 4. Their difference is that the temporal event "Oliver Bulleid works for Great Northern Railway (GNR) from 1901 to 1908." has a clear timestamp, whereas it is ambiguous for the event "Oliver Bulleid works for Board of Trade from UNK to UNK.". Therefore, it requires capturing timestamps hidden in the given document and inferring the relations between the mentioned time and associated events to answer beyond hard questions.

To further evaluate the performance of our model in answering beyond-hard questions, we select beyond-hard questions from hard ones and run them in the same way as others. As shown in Figure 5, the performance of all three models gradually decreases on three classes of questions, aligned with our previous assumption. In spite of this, Event-AL remains the optimal performance in answering beyond questions, compared with the other two methods. This suggests that our model narrows down the time range of events without timestamps during temporal learning. In addition, it can be observed that Event-AL has a more robust performance for those three questions. This demonstrates the importance of abductive reasoning for understanding temporal concepts.

7. Conclusions

In this paper, we have proposed Event-AL, a framework that can provide an explanation while answering time-sensitive questions. It is achieved by capturing temporal relations between extracted temporal from the given document. Experiments demonstrate that Event-AL significantly outperforms strong baselines for hard and even beyond-hard questions and achieves new state-of-the-art results in multi-answer questions. As in previous work, we have only tested on a few samples due to budget constraints, but experimental results demonstrate that existing models achieve similar performance in both test sets. We believe that Event-AL inspires how to mine implicit representations from explicit text by exploiting sequence or structural relations, enhancing the robustness of LLMs in practical work. In the future, we will try to design an abductive reasoning module integrated with large language models to overcome the instability of results caused by LLMs.

Ethics Statement

In this paper, our proposed approach Event-AL can effectively address hard and even beyond hard time-sensitive questions. During experiments, we only utilized publicly available datasets and baseline models. And the acquisition, processing, and analysis of all data adhere to the principles of academic integrity during the research process. However, the generated answers may suffer from the phenomenon of hallucination, which is known to be prevalent in large language models. It is well-known that large language models possess unprecedented semantic understanding capabilities. It requires models with an advanced understanding of semantics while extracting temporal events from long documents, as the same relations are often scattered throughout documents in diverse representations. Therefore, we apply large language models for temporal event extraction, which may result in erroneous and misleading answers. To mitigate this issue, we designed an event-based abductive reasoning module, which has been verified for effectiveness in getting correct answers.

Acknowledgments

This work was supported by the Project of Science and Technology Research and Development Plan of China Railway Corporation (N2023J044).

8. Bibliographical References

Wenhu Chen, Xinyi Wang, and William Yang Wang. 2021. A dataset for answering time-sensitive questions. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks)*.

Ziyang Chen, Xiang Zhao, Jinzhi Liao, Xinyi Li, and Evangelos Kanoulas. 2022. Temporal knowledge graph question answering via subgraph reasoning. *Knowledge-Based Systems*, 251:109134.

Nurendra Choudhary and Chandan K. Reddy. 2023. Complex logical reasoning over knowledge graphs using large language models. *CoRR*, abs/2305.01157.

Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. 2022. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics (TACL)*, 10:257–273.

Quzhe Huang, Yutong Hu, Shengqi Zhu, Yansong Feng, Chang Liu, and Dongyan Zhao. 2023. More than classification: A unified framework for event temporal relation extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 9631–9646.

Shima Imani, Liang Du, and Harsh Shrivastava. 2023. Mathprompter: Mathematical reasoning using large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics: Industry Track (ACL)*, pages 37–42.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume (EACL)*, pages 874–880.

Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jannik Strötgen, and Gerhard Weikum. 2018a. Tempquestions: A benchmark for temporal question answering. In *Proceedings of the 37th International World Wide Web Conferences (WWW)*, pages 1057–1062.

Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jannik Strötgen, and Gerhard Weikum. 2018b. TEQUILA: temporal question answering over knowledge bases. In *Proceedings of the 27th*

ACM International Conference on Information and Knowledge Management (CIKM), pages 1807–1810.

Zhen Jia, Soumajit Pramanik, Rishiraj Saha Roy, and Gerhard Weikum. 2021. Complex temporal question answering on knowledge graphs. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 792–802.

Xingxuan Li, Liying Cheng, Qingyu Tan, Hwee Tou Ng, Shafiq Joty, and Lidong Bing. 2023. Unlocking temporal question answering for large language models using code execution. *CoRR*, abs/2305.15014.

Puneet Mathur, Vlad I. Morariu, Verena Kaynig-Fittkau, Jiuxiang Gu, Franck Dernoncourt, Quan Hung Tran, Ani Nenkova, Dinesh Manocha, and Rajiv Jain. Doctime: A document-level temporal dependency graph parser. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*.

Sumit Neelam, Udit Sharma, Hima Karanam, Shajith Ikbal, Pavan Kapanipathi, and Ibrahim Abdelaziz. 2021. SYGMA: system for generalizable modular question answering overknowledge bases. *CoRR*, abs/2109.13430.

Qiang Ning, Hao Wu, and Dan Roth. 2018. A multi-axis annotation scheme for event temporal relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Maxwell I. Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. 2021. Show your work: Scratchpads for intermediate computation with language models. *CoRR*, abs/2112.00114.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:140:1–140:67.

Apoorv Saxena, Soumen Chakrabarti, and Partha P. Talukdar. 2021. Question answering over temporal knowledge graphs. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL/IJCNLP)*, pages 6663–6676.

Alessandro Stolfo, Zhijing Jin, Kumar Shridhar, Bernhard Schölkopf, and Mrinmaya Sachan. 2023. A causal framework to quantify the robustness of mathematical reasoning with language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 545–561.

Qingyu Tan, Hwee Tou Ng, and Lidong Bing. 2023. Towards benchmarking and improving the temporal reasoning capability of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 14820–14835.

Denny Vrandecic and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Jiexin Wang, Adam Jatowt, and Masatoshi Yoshikawa. 2022. Archivalqa: A large-scale benchmark dataset for open-domain question answering over historical news collections. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 3025–3035.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems (NeurIPS)*.

Haoyang Wen and Heng Ji. Utilizing relative event time to enhance event-event temporal relation extraction. In *Proceedings of the 23rd Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS)*.

Michael J. Q. Zhang and Eunsol Choi. 2021. Situatedqa: Incorporating extra-linguistic contexts into QA. In *Proceedings of the 25th Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7371–7387.

Xinyu Zhu, Cheng Yang, Bei Chen, Siheng Li, Jian-Guang Lou, and Yujiu Yang. 2023. Question answering as programming for solving time-sensitive questions. *CoRR*, abs/2305.14221.