

# Fusing Temporal Graphs into Transformers for Time-Sensitive Question Answering

Xin Su<sup>1,2</sup> Phillip Howard<sup>2</sup> Nagib Hakim<sup>2</sup> Steven Bethard<sup>1</sup>

<sup>1</sup>University of Arizona <sup>2</sup>Intel Labs

{xinsu, bethard}@arizona.edu, {phillip.r.howard, nagib.hakim}@intel.com

## Abstract

Answering time-sensitive questions from long documents requires temporal reasoning over the times in questions and documents. An important open question is whether large language models can perform such reasoning solely using a provided text document, or whether they can benefit from additional temporal information extracted using other systems. We address this research question by applying existing temporal information extraction systems to construct temporal graphs of events, times, and temporal relations in questions and documents. We then investigate different approaches for fusing these graphs into Transformer models. Experimental results show that our proposed approach for fusing temporal graphs into input text substantially enhances the temporal reasoning capabilities of Transformer models with or without fine-tuning. Additionally, our proposed method outperforms various graph convolution-based approaches and establishes a new state-of-the-art performance on SituatedQA and three splits of TimeQA.

## 1 Introduction

Long-document time-sensitive question answering (Chen et al., 2021) requires temporal reasoning over the events and times in a question and an accompanying long context document. Answering such questions is a challenging task in natural language processing (NLP) as models must comprehend and interpret the temporal scope of the question as well as the associated temporal information dispersed throughout the long document. For example, consider the time-sensitive questions about George Washington’s position provided in Figure 1. The relevant events and temporal information regarding George Washington’s position are scattered across many different sentences in the context document. Since there is no one single text segment containing the answer, the model must integrate and reason over events and times

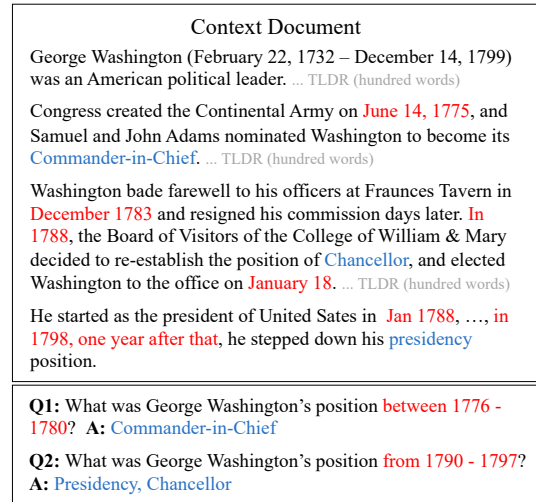


Figure 1: Time-sensitive question-answer pairs with a context document. The times and answers are in red and blue, respectively.

throughout the context document. Additionally, this example illustrates how changing the time expression in the question may also result in a change in the answer: in this case, replacing *between 1776 - 1780* with *from 1790 to 1797* changes the answer from *Commander-in-Chief* to *Presidency and Chancellor*.

Though not designed directly for question answering, there is a substantial amount of research on temporal information extraction (Chambers et al., 2014; Ning et al., 2018a; Zhang and Xue, 2018, 2019; Han et al., 2019; Ning et al., 2019; Vashishtha et al., 2019; Ballesteros et al., 2020; Yao et al., 2020; Zhang et al., 2022a). Such models can help reveal the structure of the timeline underlying a document. However, there is little existing research on combining such information extraction systems with question answering Transformer models (Izacard and Grave, 2021) to effectively reason over temporal information in long documents.

In this work, we utilize existing temporal information extraction systems to construct temporal

graphs and investigate different fusion methods to inject them into Transformer models.

We evaluate the effectiveness of each temporal graph fusion approach on long-document time-sensitive question answering datasets. Our contributions are as follows:

1. We introduce a simple but novel approach to fuse temporal graphs into the input text of question answering Transformer models.
2. We compare our method with prior approaches such as fusion via graph convolutions, and show that our input fusion method outperforms these alternative approaches.
3. We demonstrate that our input fusion approach can be used seamlessly with large language models in an in-context learning setting.
4. We perform a detailed error analysis, revealing the efficacy of our method in fixing temporal reasoning errors in Transformer models.

## 2 Related Work

### 2.1 Extracting Temporal Graphs

Research on extracting temporal graphs from text can be grouped into the extraction of event and time graphs (Chambers et al., 2014; Ning et al., 2018b), contextualized event graphs (Madaan and Yang, 2021), and temporal dependency trees and graphs (Zhang and Xue, 2018, 2019; Yao et al., 2020; Ross et al., 2020; Choubey and Huang, 2022; Mathur et al., 2022). Additionally, some prior work has focused on the problem of extracting temporal relations between times and events (Ning et al., 2018a, 2019; Vashishtha et al., 2019; Han et al., 2019; Ballesteros et al., 2020; Zhang et al., 2022a). The outputs of these temporal relation extraction systems are often used to construct temporal graphs. In this work, we use CAEVO (Chambers et al., 2014) and SUTime (Chang and Manning, 2012) to construct our temporal graphs because they are publicly available (unlike more recent models such as the temporal dependency graph parser proposed by Mathur et al. (2022)) and can easily scale to large amounts of long documents without requiring additional training.

### 2.2 Temporal Question Answering

Jia et al. (2018) decomposes questions and applies temporal constraints to allow general question-answering systems to answer knowledge-base-temporal questions. Saxena et al. (2021), Jia et al. (2021), Mavromatis et al. (2021), and Sharma et al.

(2023) use time-aware embeddings to reason over temporal knowledge graphs. Similar to our work, Huang et al. (2022) and Shang et al. (2021) answer temporal questions on text, but they focus on temporal event ordering questions over short texts rather than time-sensitive questions over long documents. Li et al. (2023) focus on exploring large language models for information extraction in structured temporal contexts. They represent the extracted time-sensitive information in code and then execute Python scripts to derive the answers. In contrast, we concentrate on temporal reasoning in the reading comprehension setting, using unstructured long documents to deduce answers. This poses more challenges in information extraction and involves more complex reasoning, which motivates our integration of existing temporal information extraction systems with transformer-based language models. The most similar work to ours is Mathur et al. (2022), which extracts temporal dependency graphs and merges them with Transformer models using learnable attention mask weights. We compare directly to this approach, and also explore both graph convolutions and input modifications as alternatives to fusing temporal graphs into Transformer models.

### 2.3 Fusing Graphs into Transformer Models

The most common approaches for fusing graphs into Transformer models are graph neural networks (GNN) and self-attention. In the GNN-based approach, a GNN is used to encode and learn graph representations which are then fused into the Transformer model (Yang et al., 2019; Feng et al., 2020; Yasunaga et al., 2021; Zhang et al., 2022b). In the self-attention approach, the relations in the graphs are converted into token-to-token relations and are then fused into the self-attention mechanism. For example, Wang et al. (2020) uses relative position encoding (Shaw et al., 2018) to encode a database schema graph into the BERT representation. Similarly, Bai et al. (2021) utilize attention masks to fuse syntax trees into Transformer models. We explore GNN-based fusion of temporal graphs into question answering models, comparing this approach to the attention-based approach of Mathur et al. (2022), as well as our simpler approach which fuses the temporal graph directly into the Transformer model’s input.

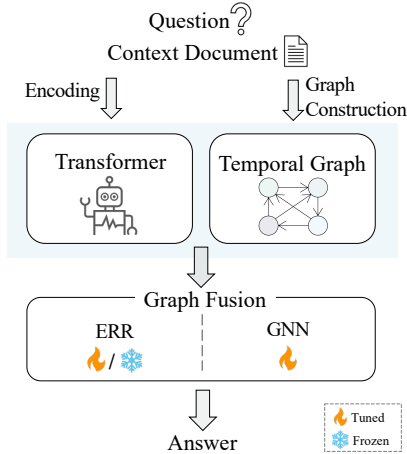


Figure 2: Overview of our approach. The ERR method allows for optional fine-tuning of the Transformer model, whereas for the GNN method requires fine-tuning.

### 3 Method

Our approach applies temporal information extraction systems to construct temporal graphs and then fuses the graphs into pre-trained Transformer models. We consider two fusion methods:

1. Explicit edge representation fusion (ERR): a simple but novel approach to fuse the graphs into the input text.
2. Graph neural network fusion: a GNN is used to fuse the graphs into the token embeddings or the last hidden layer representations (i.e., contextualized embeddings) of the Transformer model.

The overall approach is illustrated in Figure 2.

#### 3.1 Graph Construction

Given a time-sensitive question and a corresponding context document, we construct a directed temporal graph where events and time expressions are nodes and temporal relations are edges of the type BEFORE, AFTER, INCLUDES, INCLUDED BY, SIMULTANEOUS, or OVERLAP.

We extract the single timestamp included in each question, which is either explicitly provided by the dataset (as in SituatedQA (Zhang and Choi, 2021)) or alternatively is extracted via simple regular expressions (the regular expressions we use achieve 100% extraction accuracy on TimeQA (Chen et al., 2021)). We add a single question-time node to the graph for this time.

For the document, we apply CAEVO<sup>1</sup> to iden-

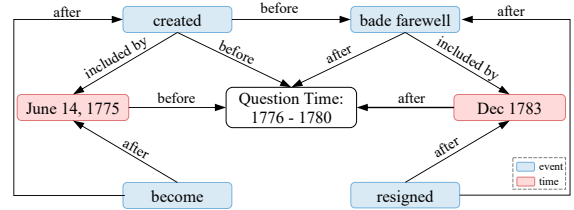


Figure 3: Temporal graph example.

tify the events, time expressions, and the temporal relations between them. CAEVO follows the standard convention in temporal information extraction that events are only simple actions (typically verbs) with linking of these actions to subjects, objects, and other arguments left to dependency parsers (Pustejovsky et al., 2003; Verhagen et al., 2009, 2010; UzZaman et al., 2013). We add document-event and document-time nodes for each identified event and time, respectively, and add edges between nodes for each identified temporal relation.

To link the question-time node to the document-time nodes, we use SUTime<sup>2</sup> to normalize time expressions to time intervals, and deterministically compute temporal relations between question time and document times as edges. For example, given a document-time node “the year 2022” and a question-time node “from 1789 to 1797” from the question “What was George Washington’s position from 1789 to 1797?”, the times will be normalized to [2022-01-01, 2022-12-31] and [1789-01-01, 1797-12-31] respectively, and the temporal relation between them can then be computed as AFTER.

To link the question-time node to document events, for each document-event node, we calculate the shortest path in the temporal graph between it and the question-time node and recursively apply standard transitivity rules (see Appendix A.1) along the path to infer the temporal relation. For example, given a path A is BEFORE B and B INCLUDES C, we can infer the relation between A and C is BEFORE. An example of a constructed temporal graph for Q1 in Figure 1 is illustrated in Figure 3.

#### 3.2 Graph Fusion

For both fusion methods, we concatenate the question and corresponding context document as an input sequence to the Transformer model. For example, given the question and document from Figure 1 Q1, the input is:

<sup>1</sup><https://github.com/nchambers/caevo>

<sup>2</sup><https://nlp.stanford.edu/software/sutime.html>

*question: What was George Washington's position between 1776 - 1780? context: ... Congress created the Continental Army on June 14, 1775 ...*

### 3.2.1 Explicit Edge Representation

In the ERR method, we mark a temporal graph's nodes and edges in the input sequence, using `<question time>` and `</question time>` to mark the question-time node and relation markers such as `<before>` and `</before>` to mark the nodes in the context document and their relations to the question time. Thus, the ERR input for the above example is:

*question: What was George Washington's position <question time>between 1776-1780</question time>? context: ... Congress <before>created</before> the Continental Army on <before>June 14, 1775</before>...*

This approach aims to make the model learn to attend to parts of the input sequence that may contain answer information. For instance, the model may learn that information related to the answer may be found near markers such as `<overlap>`, `<includes>`, `<included by>`, and `<simultaneous>`. Additionally, the model may learn that answer-related information may exist between `<before>` and `<after>`, even if the answer does not have any nearby temporal information.

### 3.2.2 GNN-based Fusion

In GNN-based fusion, we add `<e>` and `</e>` markers around each node, and apply a relational graph convolution (RelGraphConv; Schlichtkrull et al., 2018) over the marked nodes. RelGraphConv is a variant of graph convolution (GCN; Kipf and Welling, 2017) that can learn different transformations for different relation types. We employ the RelGraphConv to encode a temporal graph and update the Transformer encoder's token embedding layer or last hidden layer representations (i.e., contextualized embeddings). We utilize the RelGraphConv in its original form without any modifications.

Formally, given a temporal graph  $G = (V, E)$ , we use representations of the `<e>` markers from the Transformer model's token embedding layer or the last hidden layer as initial node embeddings. The output of layer  $l + 1$  for node  $i \in V$  is:

$$h_i^{l+1} = \sigma\left(\sum_{r \in R} \sum_{j \in \mathcal{N}^r(i)} \frac{1}{c_{i,r}} W_r^{(l)} h_j^{(l)} + W_0^{(l)} h_i^{(l)}\right)$$

where  $\mathcal{N}^r(i)$  denotes all neighbor nodes that have relation  $r$  with node  $i$ ,  $\frac{1}{c_{i,r}}$  is a normalization constant that can be learned or manually specified,  $\sigma$  is the activation function,  $W_0$  is the self-loop weight, and  $W_r$  is the learnable weights. We refer readers to Schlichtkrull et al. (2018) for more details.

## 4 Experiments

### 4.1 Datasets

We evaluate on two time-sensitive question answering datasets: Time-Sensitive Question Answering (TimeQA; Chen et al., 2021) and SituatedQA (Zhang and Choi, 2021). We briefly describe these two datasets below and provide statistics on each dataset in Table 9 of Appendix A.6.

**TimeQA** The TimeQA dataset is comprised of time-sensitive questions about time-evolving facts paired with long Wikipedia pages as context. The dataset has two non-overlapping splits generated by diverse templates which are referred to as TimeQA Easy and TimeQA Hard, with each split containing 20k question-answer pairs regarding 5.5K time-evolving facts and 70 relations. TimeQA Easy contains questions which typically include time expressions that are explicitly mentioned in the context document. In contrast, TimeQA Hard has time expressions that are not explicitly specified and therefore require more advanced temporal reasoning. For example, both questions in Figure 1 are hard questions, but if we replace the time expressions in the questions with *In 1788*, they will become easy questions. In addition, smaller Human-paraphrased Easy and Hard splits are also provided.

**SituatedQA** SituatedQA is an open-domain question answering dataset comprising two subsets: Temporal SituatedQA and Geographical SituatedQA. We focus on Temporal SituatedQA, which we will hereafter refer to as SituatedQA. Each question in SituatedQA is accompanied by a temporal annotation that could change the answer to the question if it is modified. For instance, the question "Which COVID-19 vaccines have been authorized for adults in the US as of Jan 2021?" has a corresponding answer of "Moderna, Pfizer." However, if we change the time to "Apr 10, 2021," the answer becomes "Moderna, Pfizer, J&J." As SituatedQA is a re-annotation of a subset of NQ-Open (Kwiatkowski et al., 2019a; Lee et al., 2019), we use the top 100 passages retrieved from Wikipedia



Method	Model Size	External Tool (Public Avail.)	Add. Data	Knowl. Fusion
BigBird (Chen et al., 2021)	128M	-	NaturalQ	-
FiD & Replicated FiD (Chen et al., 2021)	220M	-	NaturalQ	-
DocTime (Mathur et al., 2022)	220M	CAEVO (✓), TDGP (✗)	NaturalQ	Attention
BigBird + MTL (Chen et al., 2023a)	128M	TT (✓), TR (✗)	TriviaQA	Multi-Task
Longformer + MTL (Chen et al., 2023a)	435M	TT (✓), TR (✗)	TriviaQA	Multi-Task
DPR + Query Modified (Zhang and Choi, 2021)	110 M	-	-	Question Text
TSM (Cole et al., 2023)	11B	SUTime (✓)	Entire Wiki	Pre-Training
LongT5 <sub>ERR</sub> (Ours)	250M	SUTime (✓)	NaturalQ	Input Text

Table 1: Comparison of methods, base models’ size, external tools, additional data, and temporal knowledge fusion methods. NaturalQ: Natural Questions, TDGP: Temporal Dependency Graph Parser, TT: BERT-based Temporal Tagger, TR: timestamps retriever.

by FiD (Izacard and Grave, 2021)<sup>3</sup> for each question as context documents.

**Evaluation Metrics** We use the official evaluation methods and metrics provided in the code release of the datasets. For TimeQA, we report the exact match and F1 scores. For SituatedQA, we report the exact match score.

## 4.2 Baselines

We compare our models with the previous state-of-the-art on the TimeQA and SituatedQA, which we describe in this section and summarize in Table 1.

For TimeQA, we compare against:

**FiD & BigBird** Chen et al. (2021) adapt two long-document question answering models, BigBird (Zaheer et al., 2020) and FiD (Izacard and Grave, 2021), to the TimeQA dataset. Before fine-tuning the models on TimeQA, they also fine-tune on Natural Questions (Kwiatkowski et al., 2019b) and TriviaQA (Joshi et al., 2017), finding that training on Natural Questions results in the best performance. The best model from Chen et al. (2021) on TimeQA Easy and Hard is FiD, while BigBird performs best on the Human-paraphrased TimeQA Easy and Hard splits.

**Replicated FiD** We also report our replication of Chen et al. (2021)’s FiD model using the code provided by their GitHub repository<sup>4</sup>, but even with extensive hyperparameter tuning, we were unable to reproduce their reported performance of the FiD model on TimeQA Easy<sup>5</sup>.

**DocTime** DocTime (Mathur et al., 2022) first uses CAEVO to identify events and time expressions in context documents. Then, a custom-trained temporal dependency parser is applied to parse temporal dependency graphs. Finally, the parsed

temporal dependency graphs are fused into the attention mechanism of the FiD model, which is fine-tuned on Natural Questions.

**BigBird + MTL & Longformer + MTL** Chen et al. (2023a) inject temporal information into long text question answering systems using a multi-task learning (MTL) approach. They train BigBird and Longformer models on time-sensitive question answering tasks, along with three auxiliary temporal-awareness tasks. They explore first fine-tuning on Natural Questions and TriviaQA datasets, finding that TriviaQA results in the best performance. BigBird + MTL is their best model on TimeQA Easy, while Longformer + MTL performs best on TimeQA Hard.

For SituatedQA, we compare against:

**DPR + Query Modified** Zhang and Choi (2021) adapt a retrieval-based model, DPR (Karpukhin et al., 2020), for SituatedQA. They first fine-tune the retriever and reader of DPR on Natural Questions and then further fine-tune on SituatedQA. **TSM** (Cole et al., 2023) uses SuTime to identify and mask time expressions throughout all Wikipedia documents. They then fine-tune T5-1.1-XXL (Raffel et al., 2020) to predict the masked time expressions. Finally, they fine-tune the resulting T5-1.1-XXL model on SituatedQA.

## 4.3 Implementation Details

We use LongT5-base (Guo et al., 2022), a transformer sequence-to-sequence model, as our base model. Our experiments demonstrate that LongT5 outperforms the FiD model (Izacard and Grave, 2020) commonly used on long-document question answering tasks. To fairly compare with previous work (Chen et al., 2021; Mathur et al., 2022), we pre-train the LongT5 model on Natural Questions (Kwiatkowski et al., 2019a) and then fine-tune it

<sup>3</sup><https://github.com/facebookresearch/FiD>

<sup>4</sup>[wenhuchen/Time-Sensitive-QA](https://github.com/wenhuchen/Time-Sensitive-QA)

<sup>5</sup>Other researchers have reported the same issue [on GitHub](#)

	TimeQA Easy		TimeQA Hard		TimeQA <sub>HP</sub> Easy		TimeQA <sub>HP</sub> Hard		SituatedQA
	EM	F1	EM	F1	EM	F1	EM	F1	EM
BigBird (Chen et al., 2021)	51.2	60.3	42.4	50.9	47.6	53.7	38.8	45.9	-
FiD (Chen et al., 2021)	60.5	67.9	46.8	54.6	-	-	-	-	-
DocTime (Mathur et al., 2022)	<b>62.4</b>	<b>69.6</b>	48.2	56.4	-	-	-	-	-
BigBird + TML (Chen et al., 2023b)	55.4	63.8	45.2	55.3	-	-	-	-	-
Longformer + TML (Chen et al., 2023b)	52.4	62.6	48.7	58.5	-	-	-	-	-
DPR + Query Modified (Zhang and Choi, 2021)	-	-	-	-	-	-	-	-	23.0
TSM (Cole et al., 2023)	-	-	-	-	-	-	-	-	24.6
Replicated FiD	55.3	65.1	45.2	54.5	52.7	59.7	39.7	47.0	23.7
LongT5	55.4	64.3	49.9	58.4	53.7	61.6	45.3	52.3	24.7
LongT5 <sub>GNN</sub>	54.2	63.4	49.0	57.6	46.5	55.3	34.3	40.2	27.3
LongT5 <sub>ERR</sub>	56.9	66.2	<b>54.0</b>	<b>62.0</b>	<b>54.9</b>	<b>62.9</b>	<b>49.3</b>	<b>56.1</b>	<b>29.0</b>
Replicated FiD (re-annotated)	64.0	68.3							
LongT5 <sub>ERR</sub> (re-annotated)	<b>68.8</b>	<b>70.4</b>							

Table 2: Performance on the test sets. TimeQA<sub>HP</sub> denotes the human-paraphrased splits of TimeQA. The highest-performing model is bolded. Confidence intervals for our results are provided in table 7 in appendix A.3.

on TimeQA and SituatedQA, respectively. Appendix A.5 provides other implementation details such as hyperparameters, graph statistics, software versions, and external tool performance.

We perform model selection before evaluating on the test sets, exploring different graph subsets with both the ERR and GNN based fusion approaches introduced in Section 3.2. Table 6 in appendix A.2 shows that the best ERR method uses a document-time-to-question-time (DT2QT) subgraph and the best GNN method uses the full temporal graph by fusing it into the token embedding layer representations of the Transformer model. We hereafter refer to the LongT5 model fused with a DT2QT graph using the ERR method as LongT5<sub>ERR</sub>, and the LongT5 model fused with a full temporal graph using the GNN method as LongT5<sub>GNN</sub>.

#### 4.4 Main Results

We summarize the performance of baseline models and those trained with our graph fusion methods in Table 2.

**Which baseline models perform best?** On TimeQA, our LongT5 model without temporal graph fusion performs better than or equivalent to all other baseline models across every split and metric except for the Easy split. The best-performing model reported on TimeQA Easy is DocTime. On SituatedQA, LongT5 with no fusion performs as well as the best-reported results on this dataset.

**Which graph fusion methods perform best?** Using LongT5, we consider both of our ERR and GNN fusion methods described in Section 3.2. On

TimeQA, the LongT5<sub>GNN</sub> model fails to outperform LongT5 without fusion, while the LongT5<sub>ERR</sub> model improves over LongT5 on every split and dataset, exhibiting particularly large gains on the Hard splits. On Situated QA, both LongT5<sub>ERR</sub> and LongT5<sub>GNN</sub> models improve over the no-fusion LongT5 baseline, with ERR again providing the best performance. The somewhat inconsistent performance of the GNN fusion method across datasets (beneficial on SituatedQA while detrimental on TimeQA) suggests the need for a different GNN design for TimeQA, which we leave to future work.

To explore the differences between LongT5<sub>ERR</sub> and LongT5<sub>GNN</sub> models, we analyze 20 randomly sampled examples from TimeQA Hard where LongT5<sub>ERR</sub> is correct but LongT5<sub>GNN</sub> is incorrect. From our manual analysis of these 20 examples, all 20 examples share the same pattern: LongT5<sub>GNN</sub> fails to capture explicit temporal expressions in the context and relate them to the question’s timeline, which is crucial for deducing the right answer. This suggests that directly embedding pre-computed temporal relations between time nodes into the input is more efficient than implicitly doing so through the GNN, allowing the model to utilize them more easily. Table 14 of appendix A.11 shows three of the analyzed examples.

#### Does our approach outperform prior work?

On TimeQA, the LongT5<sub>ERR</sub> model achieves a new state-of-the-art on three of the four splits, with TimeQA Easy being the exception. On SituatedQA, the LongT5<sub>ERR</sub> model achieves a new state-of-the-art, outperforming the best reported results on this dataset. Our model excels on datasets that require

Type	Freq	#	Question	Relevant Context	Prediction	Answer
False Negative	40	1	Which team did the player Rivaldo belong to from 1996 to 1997?	After the Olympics ... Rivaldo moved to Spain as he joined <b>Deportivo La Coruña</b> in La Liga.	Deportivo La Coruña.	Deportivo.
		2	What was the position of Albert Reynolds from Dec 1992 to Nov 1994?	Albert Reynolds was an Irish Fianna Fáil politician ... <b>Leader of Fianna Fáil</b> from 1992 to 1994. ... He served as a <u>Teachta Dála</u> (TD) from 1977 to 2002.	Leader of Fianna Fáil.	Teachta Dála.
Semantic Understanding	3	3	Lucas Papademos was an employee for whom from 2010 to 2011?	In 2010 he served as an economic advisor to <u>Greek Prime Minister George Papandreou</u> .	N/A.	Greek Prime Minister George Papandreou.
Semantic Understanding & Insufficient Temporal Information	6	4	Which team did the player Miguel Veloso belong to from 2001 to 2003?	... Veloso started his football career at S.L. <b>Benfica</b> , but was rejected for being slightly overweight at the time...	Benfica.	N/A.
Semantic Understanding & Temporal Reasoning	1	5	What operated LMR 57 Lion from 1838 to 1845?	Lion was ordered by the <u>Liverpool &amp; Manchester Railway</u> in October 1837. ... In 1845 the LMR was absorbed by the Grand Junction Railway (GJR), which in turn was one of the constituents of the <b>London and North Western Railway</b> (LNWR) a year later.	London and North Western Railway.	Liverpool & Manchester Railway.

Table 3: Error categories and examples on the TimeQA Easy development set. Model predictions are in bold. Underlines denote the correct answers.

more temporal reasoning skills, like TimeQA Hard (where our model achieves a 5.8-point higher exact match score than DocTime) and SituatedQA (where our model achieves a 4.3-point higher exact match score than TSM).

Our approach offers further advantages over alternative models due to its simplicity, as summarized in Table 1. The best prior work on TimeQA Easy, DocTime, requires training a temporal dependency parser on additional data, using CAEVO, and modifying the Transformer model’s attention mechanism. The best prior work on SituatedQA, TSM, requires an 11-billion parameter T5 model which is pre-trained on the entirety of Wikipedia. In contrast, our approach only uses SUTime to construct a graph, requires only minor adjustments to the Transformer model’s input, and outperforms prior work using a model with only 250 million parameters.

**Why does our model not achieve state-of-art performance on TimeQA Easy as it does on other splits and datasets?** On TimeQA Easy, there is a performance gap between our LongT5<sub>ERR</sub> model and DocTime. Because the DocTime model has not been released we cannot directly compare with its predicted results. Instead, we randomly select 50 errors from our LongT5<sub>ERR</sub>’s output on the TimeQA Easy development set for error analysis. Table 3 shows that most of the errors are false negatives, where the model’s predicted answers are typically co-references to the correct answers (as in Table 3 example 1) or additional correct answers that are

applicable in the given context but are not included in the gold annotations (as in Table 3 example 2). The remaining errors are primarily related to semantic understanding, including the inability to accurately identify answer entities (e.g. identifying Greek Prime Minister George Papandreou as an employer in Table 3 example 3), the inability to interpret negation (e.g. in Table 3 example 4, where “rejected” implies that Veloso did not join Benfica), and the inability to reason about time with numerical calculations (e.g. “a year later” in Table 3 example 5 implies 1846). Addressing the semantic understanding errors may require incorporating additional entities and their types into the graphs, as well as better processing of negation information and relative times.

To better understand the extent of false negatives in TimeQA Easy, we re-annotated the 392 test examples where the predictions of the replicated FiD model and our LongT5<sub>ERR</sub> model are partially correct (i.e., EM = 0 and F1 > 0). We then incorporated additional coreferent mentions into the gold label set for these examples. For instance, if the original gold answer was “University of California,” we added its coreferent mention “University of California, Los Angeles” to the gold answers. We then evaluate both the replicated FiD (the best-performing model we can reproduce) and our LongT5<sub>ERR</sub> model on the re-annotated TimeQA Easy split. The last two rows of Table 2 show that while the exact match score for FiD increases by 8.7, the exact match score for our LongT5<sub>ERR</sub>

Model	TimeQA Easy		TimeQA Hard	
	EM	F1	EM	F1
ChatGPT	41.6	56.6	25.0	32.0
w/ ERR + DT2QT Subgraph	43.2	57.2	30.4	38.5

Table 4: Performance of ChatGPT on 500 sampled test examples, with and without our ERR method.

model increases by 11.9. This suggests that our model may be incurring greater penalties for finding valid coreferent answers than baseline methods.

**Does our ERR method benefit large language models using in-context learning?** We have focused so far on temporal graph fusion when fine-tuning models, but large language models such as ChatGPT (OpenAI, 2022) and LLaMA (Touvron et al., 2023) can achieve impressive performance without additional fine-tuning via in-context learning. Therefore, we tested the performance of ChatGPT (gpt-3.5-turbo) both with and without ERR for fusing the question-time-to-document-time graph. Following previous work (Khattab et al., 2022; Trivedi et al., 2022; Yoran et al., 2023) and considering the cost of ChatGPT’s commercial API, we randomly sample 500 examples from TimeQA Easy and TimeQA Hard for evaluation. The prompt format for ChatGPT remains the same as the input format described in Section 3.2.1, except that we concatenate in-context learning few-shot exemplars and task instructions before the input. We evaluate ChatGPT with and without graph fusion using an 8-shot setting. Examples of prompts are provided in Table 11 of Appendix A.8. Table 4 shows that our ERR graph fusion method improves the performance of ChatGPT on TimeQA Easy and particularly on TimeQA Hard. We note that this improvement is possible because our method can easily integrate with state-of-the-art large language models, as our approach to temporal graph fusion modifies only the input sequence. Prior work which relies on modifying attention mechanisms or adding graph neural network layers is incompatible with this in-context learning setting.

## 5 Analysis

In this section, we analyze our LongT5<sub>ERR</sub> model on the TimeQA development set.

**How do predictions differ compared to FiD?** We compare the predictions of LongT5<sub>ERR</sub> to the replicated FiD model in Table 5. While LongT5<sub>ERR</sub> and FiD both correct about the same number of

LongT5 <sub>ERR</sub>	FiD	Time QA Easy	TimeQA Hard
Correct	Correct	1427	1113
Correct	Wrong	269	494
Wrong	Correct	243	260
Wrong	Wrong	1082	1220

Table 5: Comparison of the predictions of LongT5<sub>ERR</sub> vs. the predictions of FiD.

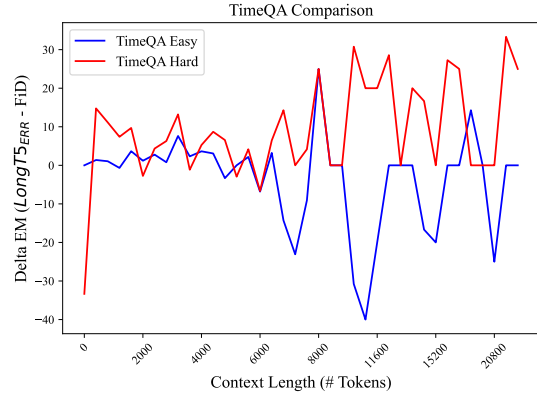


Figure 4: Comparison of the performance on different lengths of context documents.

each other’s errors on TimeQA Easy (269 vs. 243), LongT5<sub>ERR</sub> corrects many more of FiD’s errors than the reverse on TimeQA Hard (494 vs. 260). To further analyze these cases, we sampled 10 errors from the set where LongT5<sub>ERR</sub> was correct while FiD was incorrect as well as the set where the FiD was correct while LongT5<sub>ERR</sub> was incorrect. We did this across both TimeQA Easy and TimeQA Hard, totaling 40 examples. Among the 20 examples in which LongT5<sub>ERR</sub> was correct and FiD was incorrect, 17 have node markers near the answers in the ERR input sequence, and the most frequent ones are <included by> and <overlap>. The remaining 3 examples have unanswerable questions. In the examples in which FiD was correct while LongT5<sub>ERR</sub> was incorrect, we observe that 13 examples are additional correct answers (i.e., false negatives), while the other 7 examples are semantic understanding errors similar to those discussed previously. These results suggest that our ERR graph fusion approach is providing the model with useful targets for attention which allow it to produce more correct answers.

**How does the length of the document affect performance?** We compare the performance of LongT5<sub>ERR</sub> to the replicated FiD model on various document lengths, as depicted in Figure 4.



LongT5<sub>ERR</sub> performs less competitively than FiD on the Easy split for longer documents. This could be attributed to a high frequency of false negatives in LongT5<sub>ERR</sub>, as discussed previously. Additionally, it could be that LongT5<sub>ERR</sub> is less efficient at string matching on longer documents than FiD. Most of the question times in the Easy split are explicitly mentioned in the context document, which can be solved via string matching rather than temporal reasoning. However, our LongT5<sub>ERR</sub> model shows a substantial improvement on TimeQA Hard, outperforming the FiD model across most lengths.

## 6 Conclusion

In this paper, we compared different methods for fusing temporal graphs into Transformer models for time-sensitive question answering. We found that our ERR method, which fuses the temporal graph into the Transformer model’s input, outperforms GNN-based fusion as well as attention-based fusion models. We also showed that, unlike prior work on temporal graph fusion, our approach is compatible with in-context learning and yields improvements when applied to large language models such as ChatGPT. Our work establishes a promising research direction on fusing structured graphs with the inputs of Transformer models. In future work, we intend to use better-performing information extraction systems to construct our temporal graphs, enhance our approach by adding entities and entity type information to the graphs, and extend our method to spatial reasoning.

## Limitations

We use CAEVO and SUTime to construct temporal graphs because of their scalability and availability. Using more accurate neural network-based temporal information extraction tools may provide better temporal graphs but may require domain adaptation and retraining.

While we did not find graph convolutions to yield successful fusion on TimeQA, we mainly explored variations of such methods proposed by prior work. We also did not explore self-attention based fusion methods, as preliminary experiments with those methods yielded no gains, and DocTime provided an instance of such methods for comparison. But there may exist other variations of graph convolution and self-attention based fusion methods beyond those used in prior work that would make such methods more competitive with our

input-based approach to fusion.

We also did not deeply explore the places where graph neural networks failed. For example, the graph convolution over the final layer contextualized embeddings using the full temporal graph yielded performance much lower than all other graph convolution variants we explored. We limited our exploration of failures to the more successful explicit edge representation models.

## Ethics Statement

Wikipedia contains certain biases (Falenska and Çetinoğlu, 2021), and we use data from Wikipedia to train our model, thus we are potentially introducing similar biases into our models.

## References

- Jiangang Bai, Yujing Wang, Yiren Chen, Yaming Yang, Jing Bai, Jing Yu, and Yunhai Tong. 2021. [Syntax-BERT: Improving pre-trained transformers with syntax trees](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3011–3020, Online. Association for Computational Linguistics.
- Miguel Ballesteros, Rishita Anubhai, Shuai Wang, Nima Pourdamghani, Yogarshi Vyas, Jie Ma, Parminder Bhatia, Kathleen McKeown, and Yaser Al-Onaizan. 2020. [Severing the edge between before and after: Neural architectures for temporal ordering of events](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5412–5417, Online. Association for Computational Linguistics.
- Nathanael Chambers, Taylor Cassidy, Bill McDowell, and Steven Bethard. 2014. [Dense event ordering with a multi-pass architecture](#). *Transactions of the Association for Computational Linguistics*, 2:273–284.
- Angel X. Chang and Christopher Manning. 2012. [SU-Time: A library for recognizing and normalizing time expressions](#). In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC’12)*, pages 3735–3740, Istanbul, Turkey. European Language Resources Association (ELRA).
- Wenhu Chen, Xinyi Wang, and William Yang Wang. 2021. [A dataset for answering time-sensitive questions](#). In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.
- Ziqiang Chen, Shaojuan Wu, Xiaowang Zhang, and Zhiyong Feng. 2023a. [Tml: A temporal-aware multi-task learning framework for time-sensitive question answering](#). In *Companion Proceedings of the ACM*

- Web Conference 2023, WWW '23 Companion*, page 200–203, New York, NY, USA. Association for Computing Machinery.
- Ziqiang Chen, Shaojuan Wu, Xiaowang Zhang, and Zhiyong Feng. 2023b. [Tml: A temporal-aware multitask learning framework for time-sensitive question answering](#). In *Companion Proceedings of the ACM Web Conference 2023, WWW '23 Companion*, page 200–203, New York, NY, USA. Association for Computing Machinery.
- Prafulla Kumar Choubey and Ruihong Huang. 2022. [Modeling document-level temporal structures for building temporal dependency graphs](#). In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 357–365, Online only. Association for Computational Linguistics.
- Jeremy R. Cole, Aditi Chaudhary, Bhuwan Dhingra, and Partha Talukdar. 2023. [Salient span masking for temporal understanding](#). In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3052–3060, Dubrovnik, Croatia. Association for Computational Linguistics.
- Agnieszka Falenska and Özlem Çetinoğlu. 2021. [Assessing gender bias in Wikipedia: Inequalities in article titles](#). In *Proceedings of the 3rd Workshop on Gender Bias in Natural Language Processing*, pages 75–85, Online. Association for Computational Linguistics.
- Yanlin Feng, Xinyue Chen, Bill Yuchen Lin, Peifeng Wang, Jun Yan, and Xiang Ren. 2020. [Scalable multi-hop relational reasoning for knowledge-aware question answering](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1295–1309, Online. Association for Computational Linguistics.
- Mandy Guo, Joshua Ainslie, David Uthus, Santiago Ontanon, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. 2022. [LongT5: Efficient text-to-text transformer for long sequences](#). In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 724–736, Seattle, United States. Association for Computational Linguistics.
- Rujun Han, Qiang Ning, and Nanyun Peng. 2019. [Joint event and temporal relation extraction with shared representations and structured prediction](#). 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 434–444, Hong Kong, China. Association for Computational Linguistics.
- 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.
- Gautier Izacard and Edouard Grave. 2020. [Leveraging passage retrieval with generative models for open domain question answering](#).
- 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*, pages 874–880, Online. Association for Computational Linguistics.
- Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jan-nik Strötgen, and Gerhard Weikum. 2018. [Tequila: Temporal question answering over knowledge bases](#). In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM '18*, page 1807–1810, New York, NY, USA. Association for Computing Machinery.
- 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 & Knowledge Management, CIKM '21*, page 792–802, New York, NY, USA. Association for Computing Machinery.
- Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. [TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. [Dense passage retrieval for open-domain question answering](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.
- Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. 2022. [Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive nlp](#). *arXiv preprint arXiv:2212.14024*.
- Thomas N. Kipf and Max Welling. 2017. [Semi-supervised classification with graph convolutional networks](#). In *International Conference on Learning Representations*.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti,

- Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019a. [Natural questions: A benchmark for question answering research](#). *Transactions of the Association for Computational Linguistics*, 7:452–466.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019b. [Natural Questions: A Benchmark for Question Answering Research](#). *Transactions of the Association for Computational Linguistics*, 7:453–466.
- Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. [Latent retrieval for weakly supervised open domain question answering](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.
- 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](#).
- Aman Madaan and Yiming Yang. 2021. [Neural language modeling for contextualized temporal graph generation](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 864–881, Online. Association for Computational Linguistics.
- Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. [The Stanford CoreNLP natural language processing toolkit](#). In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.
- Puneet Mathur, Vlad Morariu, Verena Kaynig-Fittkau, Jiuxiang Gu, Franck Dernoncourt, Quan Tran, Ani Nenkova, Dinesh Manocha, and Rajiv Jain. 2022. [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*, pages 993–1009, Seattle, United States. Association for Computational Linguistics.
- Costas Mavromatis, Prasanna Lakkur Subramanyam, Vassilis N. Ioannidis, Soji Adeshina, Phillip Howard, Tetiana Grinberg, Nagib Hakim, and George Karypis. 2021. [Tempoqr: Temporal question reasoning over knowledge graphs](#). In *AAAI Conference on Artificial Intelligence*.
- Qiang Ning, Sanjay Subramanian, and Dan Roth. 2019. [An improved neural baseline for temporal relation extraction](#). 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 6203–6209, Hong Kong, China. Association for Computational Linguistics.
- Qiang Ning, Hao Wu, and Dan Roth. 2018a. [A multi-axis annotation scheme for event temporal relations](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1318–1328, Melbourne, Australia. Association for Computational Linguistics.
- Qiang Ning, Ben Zhou, Zhili Feng, Haoruo Peng, and Dan Roth. 2018b. [CogCompTime: A tool for understanding time in natural language](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 72–77, Brussels, Belgium. Association for Computational Linguistics.
- OpenAI. 2022. [ChatGPT: Optimizing language models for dialogue](#). OpenAI.
- James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, et al. 2003. [The timebank corpus](#). In *Corpus linguistics*, volume 2003, page 40. Lancaster, UK.
- 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](#). *The Journal of Machine Learning Research*, 21(1):5485–5551.
- Hayley Ross, Jonathon Cai, and Bonan Min. 2020. [Exploring Contextualized Neural Language Models for Temporal Dependency Parsing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8548–8553, Online. Association for Computational Linguistics.
- Apoorv Saxena, Soumen Chakrabarti, and Partha 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 (Volume 1: Long Papers)*, pages 6663–6676, Online. Association for Computational Linguistics.
- Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. [Modeling relational data with graph convolutional networks](#). In *The Semantic Web*, pages 593–607, Cham. Springer International Publishing.
- Chao Shang, Peng Qi, Guangtao Wang, Jing Huang, Youzheng Wu, and Bowen Zhou. 2021. [Open temporal relation extraction for question answering](#). In *3rd Conference on Automated Knowledge Base Construction*.



- Aditya Sharma, Apoorv Saxena, Chitranshu Gupta, Mehran Kazemi, Partha Talukdar, and Soumen Chakrabarti. 2023. [TwIRGCN: Temporally weighted graph convolution for question answering over temporal knowledge graphs](#). In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2049–2060, Dubrovnik, Croatia. Association for Computational Linguistics.
- Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. [Self-attention with relative position representations](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509*.
- Naushad UzZaman, Hector Llorens, Leon Derczynski, James Allen, Marc Verhagen, and James Pustejovsky. 2013. [SemEval-2013 task 1: TempEval-3: Evaluating time expressions, events, and temporal relations](#). In *Second Joint Conference on Lexical and Computational Semantics (\*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 1–9, Atlanta, Georgia, USA. Association for Computational Linguistics.
- Siddharth Vashishtha, Benjamin Van Durme, and Aaron Steven White. 2019. [Fine-grained temporal relation extraction](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2906–2919, Florence, Italy. Association for Computational Linguistics.
- Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Jessica Moszkowicz, and James Pustejovsky. 2009. The tempeval challenge: identifying temporal relations in text. *Language Resources and Evaluation*, 43:161–179.
- Marc Verhagen, Roser Sauri, Tommaso Caselli, and James Pustejovsky. 2010. Semeval-2010 task 13: Tempeval-2. In *Proceedings of the 5th international workshop on semantic evaluation*, pages 57–62.
- Bailin Wang, Richard Shin, Xiaodong Liu, Oleksandr Polozov, and Matthew Richardson. 2020. [RAT-SQL: Relation-aware schema encoding and linking for text-to-SQL parsers](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7567–7578, Online. Association for Computational Linguistics.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- An Yang, Quan Wang, Jing Liu, Kai Liu, Yajuan Lyu, Hua Wu, Qiaoqiao She, and Sujian Li. 2019. [Enhancing pre-trained language representations with rich knowledge for machine reading comprehension](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2346–2357, Florence, Italy. Association for Computational Linguistics.
- Jiarui Yao, Haoling Qiu, Bonan Min, and Nianwen Xue. 2020. [Annotating Temporal Dependency Graphs via Crowdsourcing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5368–5380, Online. Association for Computational Linguistics.
- Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. [QA-GNN: Reasoning with language models and knowledge graphs for question answering](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 535–546, Online. Association for Computational Linguistics.
- Ori Yoran, Tomer Wolfson, Ben Bogin, Uri Katz, Daniel Deutch, and Jonathan Berant. 2023. Answering questions by meta-reasoning over multiple chains of thought. *arXiv preprint arXiv:2304.13007*.
- Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33.
- Michael Zhang and Eunsol Choi. 2021. [SituatingQA: Incorporating extra-linguistic contexts into QA](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7371–7387, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Shuaicheng Zhang, Qiang Ning, and Lifu Huang. 2022a. [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.

Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D Manning, and Jure Leskovec. 2022b. [GreaseLM: Graph REASONing enhanced language models](#). In *International Conference on Learning Representations*.

Yuchen Zhang and Nianwen Xue. 2018. [Neural ranking models for temporal dependency structure parsing](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3339–3349, Brussels, Belgium. Association for Computational Linguistics.

Yuchen Zhang and Nianwen Xue. 2019. [Acquiring structured temporal representation via crowdsourcing: A feasibility study](#). In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (\*SEM 2019)*, pages 178–185, Minneapolis, Minnesota. Association for Computational Linguistics.

## A Appendix

### A.1 Transitivity rules

Figure 5 shows the standard temporal transitivity rules we apply to infer relations from paths through the temporal graph.

### A.2 Model Selection

For the ERR method introduced in Section 3.2.1, we consider a document-time-to-question-time (DT2QT) subgraph that contains all the edges connecting document-time nodes to the question-time node, and a document-time-event-to-question-time (DTE2QT) subgraph that builds on the document-time-to-question-time subgraph by adding all the edges from the document-event nodes to the question-time node. We do not fuse the entire temporal graph with ERR because doing so would significantly increase the length of the input. On average, this fusion will add a number of tokens equal to twice the number of edges because each edge needs to be represented by two special markers, e.g., <before> 2009 </before>.

For the GNN method, we consider both the entire temporal graph, and an “all time graph” derived from the DT2QT subgraph, with additional edges for the temporal relations between each time expression node and all other nodes.

We also consider whether to use all six relations or only three relations, merging (BEFORE, AFTER, SIMULTANEOUS, INCLUDES, INCLUDED BY, OVERLAP) into (BEFORE, AFTER, and OVERLAP).

We used the TimeQA Easy and Hard development sets for model selections. Table 6 shows the results of model selection. For ERR, merging the relations down to 3 slightly decreases performance on TimeQA Hard, the DT2QT graph is slightly better for TimeQA Hard, and the DTE2QT graph is slightly better for TimeQA Easy. Since the DT2QT is simpler, we select this model as our ERR model.

For GNN, no models outperform LongT5 alone, regardless of whether the GNN was applied to the token embeddings or the final contextualized embeddings, and whether applied to simpler or more complex temporal graphs.

### A.3 Confidence Intervals

We use the bootstrap resampling method to construct 95% confidence intervals on the test sets. The results are shown in Table 7.

### A.4 Software Licenses

We train question answering systems to answer English Wikipedia time-sensitive questions. Upon publication, we will release our code under the

	before	after	includes	included by	simultaneous	overlap
before	before	×	before	×	before	×
after	×	after	after	×	after	×
includes	×	×	includes	×	includes	×
included by	before	after	×	included by	included by	×
simultaneous	before	after	includes	included by	simultaneous	overlap
overlap	×	×	×	overlap	overlap	×

Figure 5: Transitivity rules.

Model	TimeQA Dev			
	Easy		Hard	
	EM	F1	EM	F1
LongT5	55.2	63.5	48.1	56.4
LongT5 + ERR + DT2QT Subgraph	56.1	65.1	<b>52.1</b>	<b>60.5</b>
LongT5 + ERR + DT2QT Subgraph (merge to 3 relations)	56.7	65.3	51.9	60.1
LongT5 + ERR + DTE2QT Subgraph	<b>57.0</b>	<b>65.6</b>	50.6	59.0
LongT5 + GNN Token Embed + All Time Graph	55.3	64.1	47.8	56.3
LongT5 + GNN Token Embed + Temporal Graph	55.5	64.3	48.1	56.8
LongT5 + GNN Context Embed + All Time Graph	11.5	11.5	13.4	13.4
LongT5 + GNN Context Embed + Temporal Graph	53.9	62.5	48.8	57.6

Table 6: Performance on TimeQA development datasets. The highest-performing model is bolded. DT2QT Subgraph is the document-time-to-question-time subgraph. DTE2QT Subgraph is the document-time-event-to-question-time subgraph.

Models	TimeQA Test				SituatdQA Test
	Easy	Hard	HP Easy	HP Hard	
	EM / F1	EM / F1	EM / F1	EM / F1	
Replicated FiD	55.3 / 65.1	45.2 / 54.5	52.7 / 59.7	39.7 / 47.0	-
LongT5	[53.9, 56.6] / [63.9, 66.3] 55.4 / 64.3	[43.8, 46.6] / [53.2, 55.7] 49.9 / 58.4	[50.2, 55.2] / [57.4, 61.9] 53.7 / 61.6	[37.3, 42.2] / [44.7, 49.4] 45.3 / 52.3	24.7
LongT5 <sub>GNN</sub>	[54.1, 56.7] / [63.1, 65.5] 54.2 / 63.4	[48.5, 51.2] / [57.2, 59.7] 49.0 / 57.6	[51.3, 55.9] / [59.3, 63.7] 46.5 / 55.3	[42.8, 47.6] / [50.0, 54.6] 34.3 / 40.2	[23.2, 26.4] 27.3
LongT5 <sub>ERR</sub>	[52.8, 55.6] / [62.1, 64.5] <b>56.9 / 66.2</b>	[47.6, 50.3] / [56.4, 58.8] <b>54.0 / 62.0</b>	[44.1, 48.8] / [53.1, 57.4] <b>54.9 / 62.9</b>	[32.0, 36.3] / [38.0, 42.3] <b>49.3 / 56.1</b>	[25.8, 29.3] <b>29.0</b>
	[55.5, 58.3] / [65.0, 67.5]	[52.6, 55.3] / [60.8, 63.2]	[52.5, 57.1] / [60.8, 65.0]	[46.9, 51.5] / [53.9, 58.1]	[27.3, 30.6]

Table 7: Confidence intervals on the test sets, generated via bootstrap resampling. HP = Human-Paraphrased.

MIT license. We list below the licenses of the data, software, and models we used. Our use is consistent with their intended uses.

- Stanford CoreNLP: GNU General Public License Version 3 <sup>6</sup>.
- CAEVO, Long-t5-tglobal-base model, HuggingFace Transformers: Apache License, Version 2.0 <sup>7</sup>.
- TimeQA dataset: BSD 3-Clause License <sup>8</sup>.
- Pytorch: BSD-style license <sup>9</sup>.

### A.5 Implementation Details

We use SUTime from Stanford CoreNLP V4.5.0 (Manning et al., 2014) to identify and normalize time expressions. We use the LongT5 implementation from Huggingface Transformers V4.20.1 (Wolf et al., 2020) and the LongT5 base model (250 million parameters) with transient-global attention checkpoint (long-t5-tglobal-base) from Google as the starting point for training. We follow the code provided in the FiD Github repository<sup>10</sup> to preprocess the Natural Questions dataset. We create new tokens in the tokenizer and model vocabulary for special marker tokens such as <e>. We tune the learning rate  $r \in \{1 \times 10^{-5}, 2 \times 10^{-5}, 3 \times 10^{-5}, 4 \times 10^{-5}, 5 \times 10^{-5}\}$ , batch size  $b \in \{4, 8, 16, 32\}$ , and number of RelGraphConv layers  $l \in \{1, 3, 6\}$  on the development set of TimeQA and use early stopping to monitor the exact match metrics. We use *ReLU* as activation functions in RelGraphConv layers. We set the hidden state size of RelGraphConv layers to the same as LongT5-Base. All experiments are conducted on 4 Nvidia A6000 GPUs.

<sup>6</sup>[www.gnu.org/licenses/gpl-3.0.html](http://www.gnu.org/licenses/gpl-3.0.html)

<sup>7</sup>[www.apache.org/licenses/LICENSE-2.0](http://www.apache.org/licenses/LICENSE-2.0)

<sup>8</sup>[opensource.org/licenses/BSD-3-Clause](http://opensource.org/licenses/BSD-3-Clause)

<sup>9</sup>[github.com/pytorch/pytorch/blob/master/LICENSE](https://github.com/pytorch/pytorch/blob/master/LICENSE)

<sup>10</sup><https://github.com/facebookresearch/FiD>

The total GPU hours of experiments are around 160. We will make our code publicly available upon publication of the paper<sup>11</sup>.

**Graph Statistics** The constructed graphs statistics are presented in Table 8.

**Performance of the External Tools** We use two tools: SUTime from Stanford CoreNLP and CAEVO. CAEVO’s reported precision is 0.92 for event-time relations and 0.88 for time-time relations. The accuracy of SUTime in recognizing time expression types and values is 0.96 and 0.82, respectively.

### A.6 Datasets

Table 9 presents the statistics of the datasets used for the experiments. Table 10 shows the input length of the LongT5 model before and after fusing the DT2QT sub-graph using the ERR method.

### A.7 Temporal Relations

We visualize the temporal relations between the time intervals in Figure 6.

### A.8 ChatGPT Prompts

We present the ChatGPT prompts we used in Table 11. Considering the overall input length constraint, for context documents in in-context learning examples, we use sentences containing answers as context for questions that have answers. For questions without answers, we randomly sample a sentence from the original context document as the context.

### A.9 GPT-4 Results

Following the in-context learning setup described in Section 4.4, we replicate the experiment using

<sup>11</sup>[https://github.com/IntelLabs/multimodal\\_cognitive\\_ai/tree/main/Fusing\\_Temporal\\_Graphs](https://github.com/IntelLabs/multimodal_cognitive_ai/tree/main/Fusing_Temporal_Graphs)

Dataset	Split	Avg. # Nodes	Avg. # Edges	In/Out Degree
TimeQA Easy	Train	161	323	2/2
TimeQA Easy	Dev	163	326	2/2
TimeQA Easy	Test	163	324	2/2
TimeQA Hard	Train	160	321	2/2
TimeQA Hard	Dev	162	325	2/2
TimeQA Hard	Test	162	323	2/2
TimeQA HP Easy	Train	163	317	2/2
TimeQA HP Easy	Test	194	376	2/2
TimeQA HP Hard	Train	163	317	2/2
TimeQA HP Hard	Test	194	376	2/2
SituatedQA TEMP	Train	214	213	1/1
SituatedQA TEMP	Dev	210	209	1/1
SituatedQA TEMP	Test	213	212	1/1

Table 8: The statistics of constructed graphs. HP = Human-Paraphrased.

Dataset	Split	# Questions	# Answerable	# Unanswerable	Avg. # Tokens
TimeQA Easy	Train	14308	12532	1776	2566
TimeQA Easy	Development	3021	2674	347	2635
TimeQA Easy	Test	2997	2613	384	2634
TimeQA Hard	Train	14681	12532	2149	2560
TimeQA Hard	Development	3087	2674	413	2636
TimeQA Hard	Test	3078	2613	465	2636
Human-Paraphrased TimeQA Easy	Train	1171	982	189	2907
Human-Paraphrased TimeQA Easy	Test	1171	982	189	2907
Human-Paraphrased TimeQA Hard	Train	1171	982	189	2907
Human-Paraphrased TimeQA Hard	Test	1171	982	189	2907
SituatedQA TEMP	Train	6009	6009	0	9923*
SituatedQA TEMP	Dev	3423	3423	0	9924*
SituatedQA TEMP	Test	2795	2795	0	9922*

Table 9: Statistics for the four datasets. Avg. # Tokens is the average number of tokens in the context document. We use the LongT5 model’s tokenizer to tokenize context documents. \*The length is based on the retrieved top 100 paragraphs from Wikipedia, which serve as a context document for each question. If the total length of a context document exceeds 10,000 tokens, we truncate the context accordingly.

the GPT-4 model. However, due to the costs of calling the commercial GPT-4 API, we only randomly sampled 100 examples each from TimeQA Easy and TimeQA Hard, resulting in a total of 200 samples for the experiment. The results are shown in Table 12. ERR method achieves similar performance improvements with GPT-4 as it does with ChatGPT (gpt-3.5-turbo).

### A.10 Additional Error Analysis

Table 13 presents error examples, in addition to the examples shown in Table 3.

### A.11 Analysis of the GNN Fusion Method

Table 14 shows examples from TimeQA Hard where LongT5<sub>ERR</sub> is correct but LongT5<sub>GNN</sub> is incorrect.

### A.12 Other Experimented Methods

The following are some of the methods we have tried but yielded lower performance than our main methods reported in Table 6.

1. We tried to use the coreference resolution tools to process the context documents and then construct and fuse temporal graphs based on the processed text, but we found that the coreference resolution preprocessing hurt the performance of the models.
2. We tried to fuse constructed temporal graphs into FiD models using relation-aware attention (Wang et al., 2020), but we found that the fused FiD models performed almost the same as the non-fused FiD models.
3. Our preliminary study found that the performance of the models can be significantly improved if only the most relevant paragraph in the context document is used as the context.



Dataset	# Tokens Pre-ERR	# Tokens Post-ERR
TimeQA	2617	2862
SituatedQA	9923	11515

Table 10: Average LongT5 input length before and after using the ERR method to fuse DT2QT subgraphs.

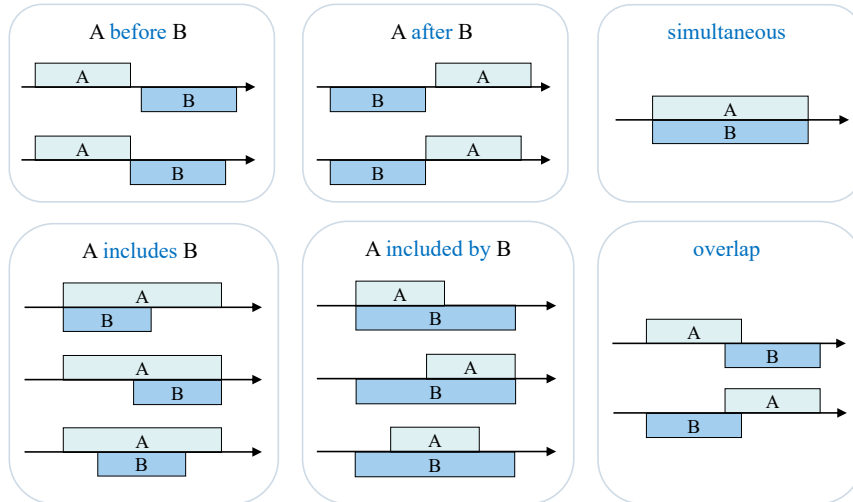


Figure 6: Visualization of temporal relations.

W/O Graph Fusion
<p><b>Instruction:</b> Answer the question based on the given context. If there is no answer, answer "no answer."</p> <p><b>Context:</b> In the 1955 general election, Cunningham was chosen as the new Ulster Unionist MP for South Antrim. He was a delegate to the Council of Europe and Western European Union Parliamentary Assembly from 1956 to 1959.</p> <p><b>Question:</b> Which position did Knox Cunningham hold before Apr 1956?</p> <p><b>Answer:</b> Ulster Unionist MP for South Antrim</p> <p><b>Context:</b> Broughton would go on to teach at Amherst College, Bryn Mawr College (1928-1965) and, later, serve as George L.</p> <p><b>Question:</b> Thomas Robert Shannon Broughton was an employee for whom before Jun 1926?</p> <p><b>Answer:</b> no answer</p>
W/ Document-Time-to-Question-Time Subgraphs Fused Using ERR Method
<p><b>Instruction:</b> Answer the question based on the given context. If there is no answer, answer "no answer." The temporal relations between the times mentioned in the context and the question are represented using XML-style tags.</p> <p><b>Context:</b> In the &lt;included by&gt;1955&lt;/included by&gt; general election, Cunningham was chosen as the new Ulster Unionist MP for South Antrim. He was a delegate to the Council of Europe and Western European Union Parliamentary Assembly &lt;overlap&gt;from 1956 to 1959&lt;/overlap&gt;.</p> <p><b>Question:</b> Which position did Knox Cunningham hold &lt;question time&gt;before Apr 1956&lt;/question time&gt;?</p> <p><b>Answer:</b> Ulster Unionist MP for South Antrim</p> <p><b>Context:</b> Broughton would go on to teach at Amherst College, Bryn Mawr College (&lt;after&gt;1928-1965&lt;/after&gt;) and, later, serve as George L.</p> <p><b>Question:</b> Thomas Robert Shannon Broughton was an employee for whom &lt;question time&gt;before Jun 1926&lt;/question time&gt;?</p> <p><b>Answer:</b> no answer</p>

Table 11: Examples of ChatGPT prompts.

Model	TimeQA Easy		TimeQA Hard	
	EM	F1	EM	F1
GPT-4	52.0	62.1	48.0	56.7
w/ ERR + DT2QT Subgraph	53.0	64.0	55.0	63.5

Table 12: Performance of GPT-4 on 100 randomly sampled test examples, with and without our ERR method.

Type	Freq	#	Question	Relevant Context	Prediction	Answer
False Negative	40	1	Where was Richard G. Hovannisian educated from 1954 to 1958?	Hovannisian received his B.A. in history (1954) from the <b>University of California, Berkeley</b> , and his M.A. in history (1958) and his Ph.D. (1966) from <u>University of California, Los Angeles (UCLA)</u> .	University of California, Berkeley	University of California, Los Angeles (UCLA)
		2	Which team did the player Simone Verdi belong to from 2014 to 2015?	On 18 June 2014, the co-ownership between Torino and <u>Milan</u> was renewed for a third year, with Verdi loaned to <b>Empoli</b> again.	Empoli	Milan
		3	Segenet Kelemu was an employee for whom from Nov 2013 to Nov 2014?	In 2013, Kelemu joined the <b>Alliance for a Green Revolution in Africa</b> (AGRA) as Vice President for Programs for about a year. In November 2013, Kelemu became the Director General of the <u>International Centre of Insect Physiology and Ecology</u> (icipe) ...	Alliance for a Green Revolution in Africa	International Centre of Insect Physiology and Ecology
Semantic Understanding	3	4	Donald Thobega played for which team in 2006?	... Between 1999 and 2006 he won a total of five caps for the <u>Botswana national football team</u> .	N/A	Botswana national football team
		5	Which school did Alexander Dvorkin go to from 1975 to 1978?	... in 1972, he became a student in the Faculty of Russian Language and Literature of <b>Moscow Pedagogical Institute</b> ... consequently, in the autumn of 1975, Dvorkin was expelled from the third year of the institute ...	Moscow Pedagogical Institute	N/A
Semantic Understanding & Insufficient Temporal Information	6	6	What was the capital of Kolomna from 1929 to Apr 2017?	Within the framework of administrative divisions, Kolomna serves as the administrative center of <u>Kolomensky District</u> ...	N/A	Kolomensky District
		7	Where did Ryan Gander work from 2007 to 2015?	... In 2004, he was made Cocheme Fellow at Byam Shaw School of Art, <u>London</u> ...	N/A	London

Table 13: Error categories and examples on the TimeQA Easy development set. Model predictions are in bold. Underlines denote the correct answers.

Thus, we tried to train a cross-encoder-based ranker to rank the paragraphs in the context documents, but we found that the pipeline approach using a cross-encoder as a paragraphs ranker and a LongT5 as a reader was not as good as using LongT5 directly to generate answers end-to-end.

---

<p><b>Question:</b> What position did Roland Burriss take in May 1989?  <b>LongT5<sub>ERR</sub> Prediction:</b> Illinois Comptroller  <b>LongT5<sub>GNN</sub> Prediction:</b> Illinois Attorney General  <b>Context:</b> ... In 1978, Burriss was the first African-American elected to statewide office in Illinois, when he was elected Illinois Comptroller. He served in that office until his election as Illinois Attorney General in 1990 ...  <b>Annotated explanation:</b> LongT5<sub>GNN</sub> prediction of "Illinois Attorney General" is based on the fact that Burriss was elected to this position in 1990. However, it failed to correctly link the timeline in the question "May 1989" to the context.</p>
---

---

<p><b>Question:</b> Where was Joe Burrow educated in Feb 2015?  <b>LongT5<sub>ERR</sub> Prediction:</b> Ohio State  <b>LongT5<sub>GNN</sub> Prediction:</b> Athens High School  <b>Context:</b> ... Burrow attended Athens High School (2011–14) in The Plains, Ohio ... After redshirting his first year at Ohio State in 2015, Burrow spent the next two years as a backup to J. T. Barrett ...  <b>Annotated explanation:</b> LongT5<sub>GNN</sub>'s guess of "Athens High School" is based on Burrow's education at Athens High School in The Plains, Ohio. However, it failed to discern how "Feb 2015" in the question aligns with the timeline in the context. As evident, Burrow was at Ohio State in 2015.</p>
---

---

<p><b>Question:</b> Which school did Kyriakos Mitsotakis attend in Feb 1985?  <b>LongT5<sub>ERR</sub> Prediction:</b> Athens College  <b>LongT5<sub>GNN</sub> Prediction:</b> Harvard University  <b>Context:</b> ... In 1986, he graduated from Athens College. From 1986 to 1990, he attended Harvard University and earned a bachelor's degree in social studies, receiving the Hoopes Prize ...  <b>Annotated explanation:</b> LongT5<sub>GNN</sub>'s prediction "Harvard University" is based on Mitsotakis attending Harvard from 1986 to 1990. Yet, the timeline "Feb 1985" in the question doesn't match the context. From the context, it's clear Mitsotakis graduated from Athens College in 1986, implying he was studying there in 1985.</p>
--

---

Table 14: Examples where LongT5<sub>ERR</sub> is correct, but LongT5<sub>GNN</sub> is incorrect.