January 13th, 2026

Mario Kuzmanov

X-TEMP NLP

***Time**-aware **R**etrieval-Augmented Large Language Models for Temporal Knowledge Graph Question Answering*

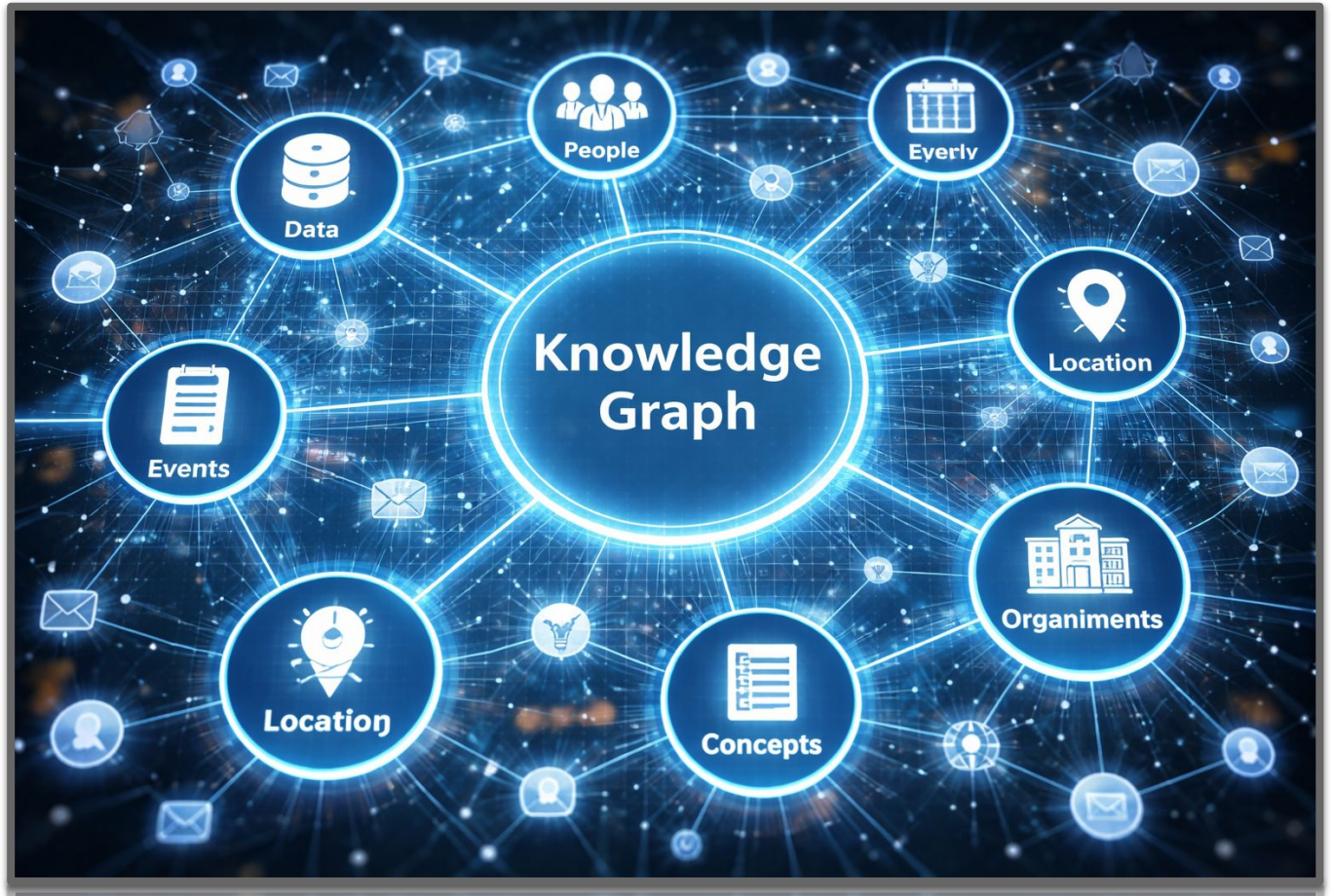
TimeR^4

Xinying Qian, Ying Zhang, Yu Zhao, Baohang Zhou, Xuhui Sui, Li Zhang, Kehui Song

OUTLINE (1)

Preliminaries

- *Introduction*
- *Related Works*
- *Motivation*

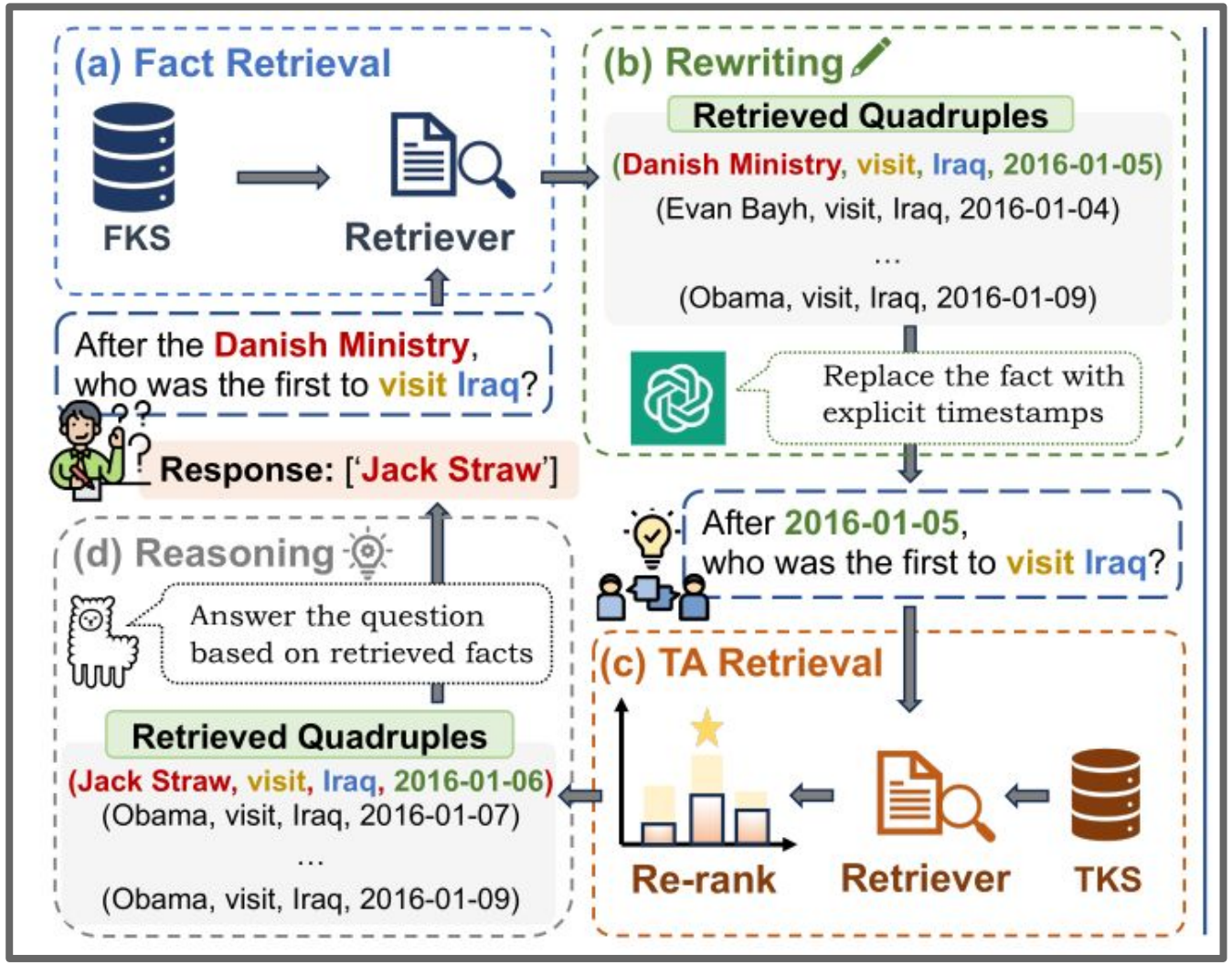


src

OUTLINE (2)

TimeR^4

- Fact Retrieval
- *Rewrite*
- *Time-aware Retrieval*
- *Rerank*
- *Reasoning*

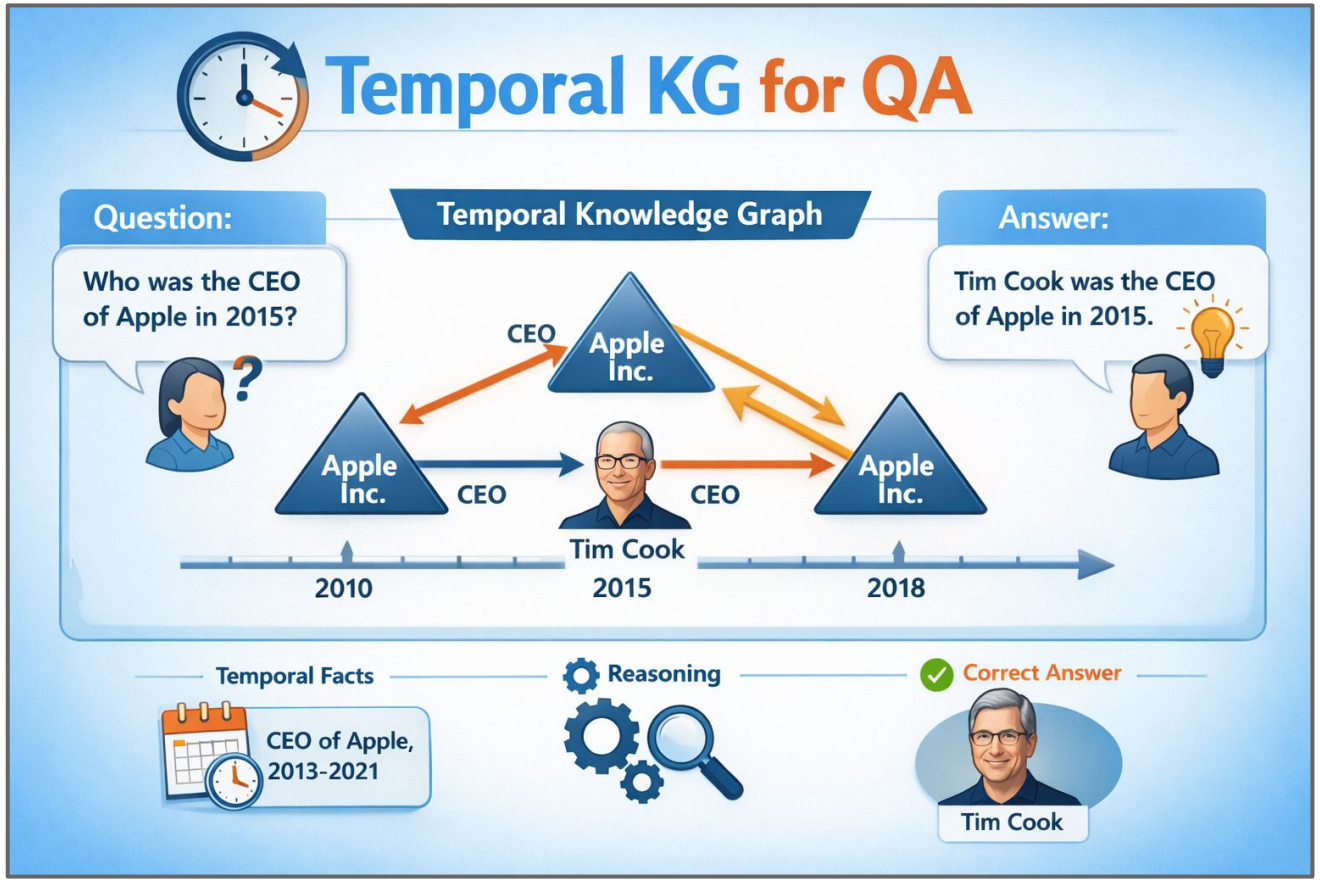


TimeR^4 architecture

OUTLINE (3)

Experimental Work

- *Datasets*
- *Results*
- *Ablation Study*
- *Further Discussion*
- Case Study



src

OUTLINE (4)

Key Takeaways

- *Conclusions &*
  Limitations/Criticism

src

Preliminaries: Formal Introduction

- Let **G** be a directed temporal knowledge graph (TKG)

$$G = \{\mathcal{E}, \mathcal{P}, \mathcal{T}, \mathcal{F}\}$$

, then:

1. **E -** the nodes are represented by the set of **entities**
2. **P & T -** the edges are expressed by the set of **predicates** with **timestamps**
3. **F** - the temporal facts are the set of quadruples **F**, such that each f ∈ F = (**s** ∈ E, **p** ∈ P, **o** ∈ E, **t** ∈ T)

- Thus, **TKGQA** is:

  - to infer the correct answer of a question **q** ∈ Q, where Q is the set of natural language questions based on the relevant quadruples **f**. The answer can be either an entity name or a timestamp.

Preliminaries: Informal Introduction

`Question: "Who is the president of the United States after Obama?"`

- **KG** - { (Obama, president, US), (Trump, president, US) }
- **TKG** - {(Obama, president, US, 01-20-2017), (Trump, president, US, 01-21-2017)}
- **TKG Question Answering (TKGQA)** - answer the question based on the knowledge from the graph.

Preliminaries: Related Works

KGQA Methods

- retrieval-based reasoning
- path-based reasoning
- agent-based reasoning

TKGQA Methods

- TempoQR
- MultiQA
- EXAQT
- TwiRGCN
- LGQA
- ARI

KGQA Methods

Retrieval-based

- retrieve relevant subgraphs/triples from the KG.
- then a LLM processes and reasons over the retrieved information.
- based on semantic similarity, neglects temporal constraints.

Path-based

- paths in the KG that possibly answer the question are generated with a LLM.
- misses a temporal dimension, therefore not directly applicable to TKGQA.

Agent-based

- an LLM-based agent searches and prunes the KGs and find answers.
- however, greedy decision-making accumulates errors and complex reasoning is an overhead, requiring many LLM-calls.

TKGQA Methods (1)

TempoQR - augments question embeddings with context, entities and time-aware information

EXAQT- utilizes RGCN (Relational Graph Convolution Networks) layer + dictionary matching

MultiQA- encodes temporal information via Transformer layers

TKGQA Methods (2)

ARI - integrates large-scale LLMs in a knowledge adaptability framework with methodological guidance

LGQA - multi-hop message passing GNN to combine global and local features

TwiRGCN - adopts temporally weighted graph convolution + answer gating

Preliminaries - Motivation

- Turns out, existing **TKGQA** methods have not overcame two significant challenges:

  1. **Hallucinations**
  2. **Lack of temporal knowledge**

Hallucinations

Take the following implicit temporal question:

“After the Danish Ministry, who was the first to visit Iraq?"

Problem - no explicit timestamps, extra steps for reasoning, increased chance of hallucinations

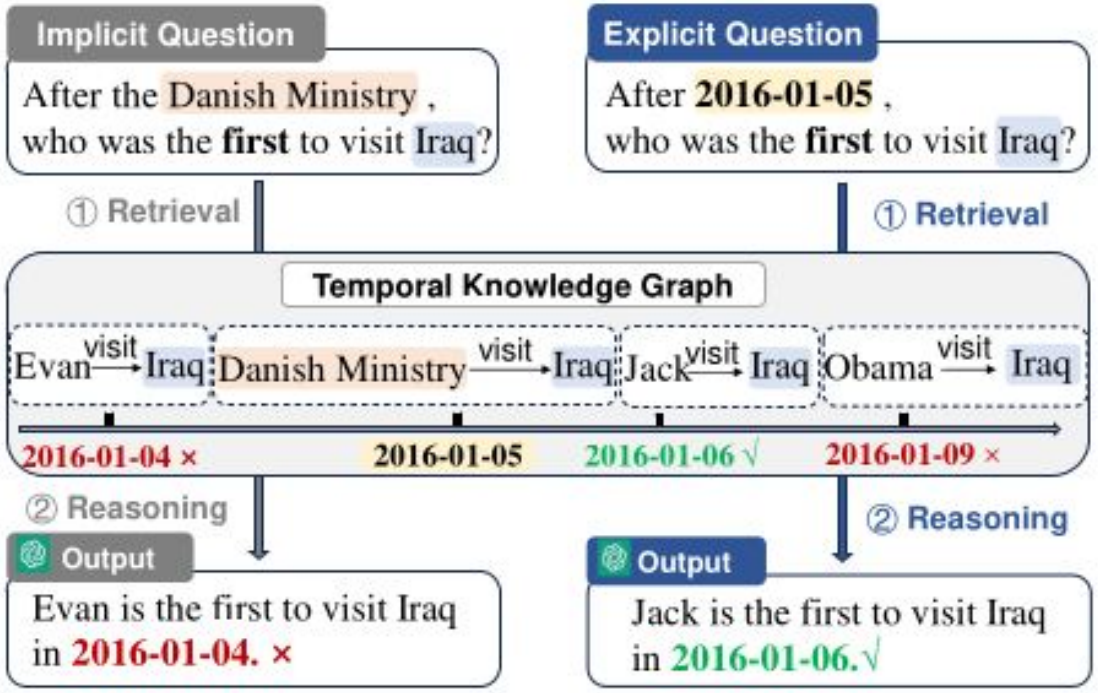


Figure 1: Examples of challenges in integrating temporal knowledge graphs with large language models.

Preliminaries - Motivation

- Turns out, existing **TKGQA** methods have not overcame two significant challenges:

  1. **Hallucinations**
  2. **Lack of temporal knowledge**

Hallucinations

Take the following implicit temporal question:

"After the Danish Ministry, who was the first to visit Iraq?"

Solution - take the explicit question, simplify the task for LLMs

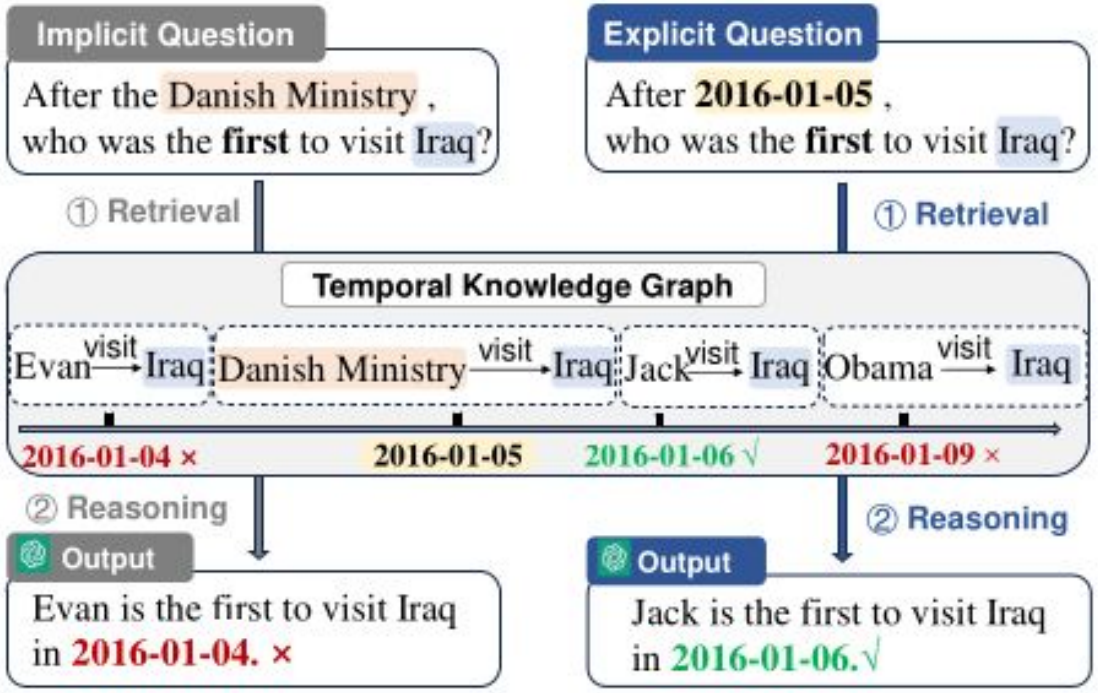


Figure 1: Examples of challenges in integrating temporal knowledge graphs with large language models.

Preliminaries - Motivation

- Turns out, existing **TKGQA** methods have not overcame two significant challenges:

  1. **Hallucinations**
  2. **Lack of temporal knowledge**

Lack of temporal knowledge

Take the following quadruple

"(Evan, visit, Iraq, 2016-01-04)"

Problem - if the timestamp is not taken into account (standard retrieves like BM25), reasoning becomes ineffective.

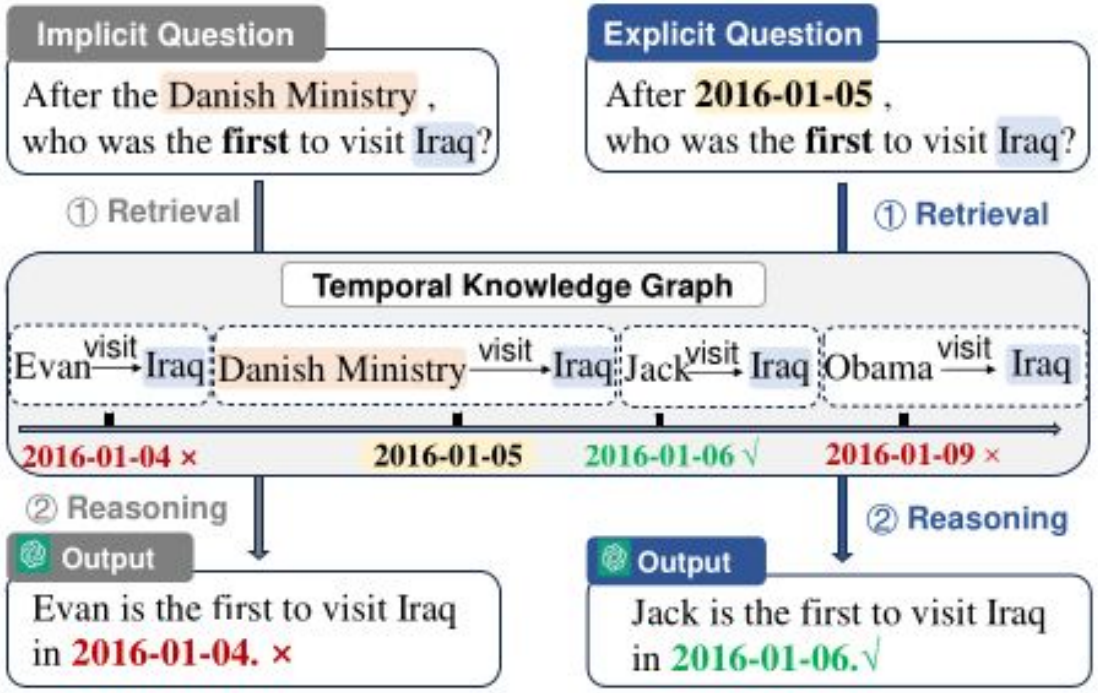


Figure 1: Examples of challenges in integrating temporal knowledge graphs with large language models.

Preliminaries - Motivation

- Turns out, existing **TKGQA** methods have not overcame two significant challenges:

  1. **Hallucinations**
  2. **Lack of temporal knowledge**

Lack of temporal knowledge

Take the following quadruple

"(Evan, visit, Iraq, 2016-01-04)"

Solution - a retriever that pays attention to semantic similarity and temporal constraints.

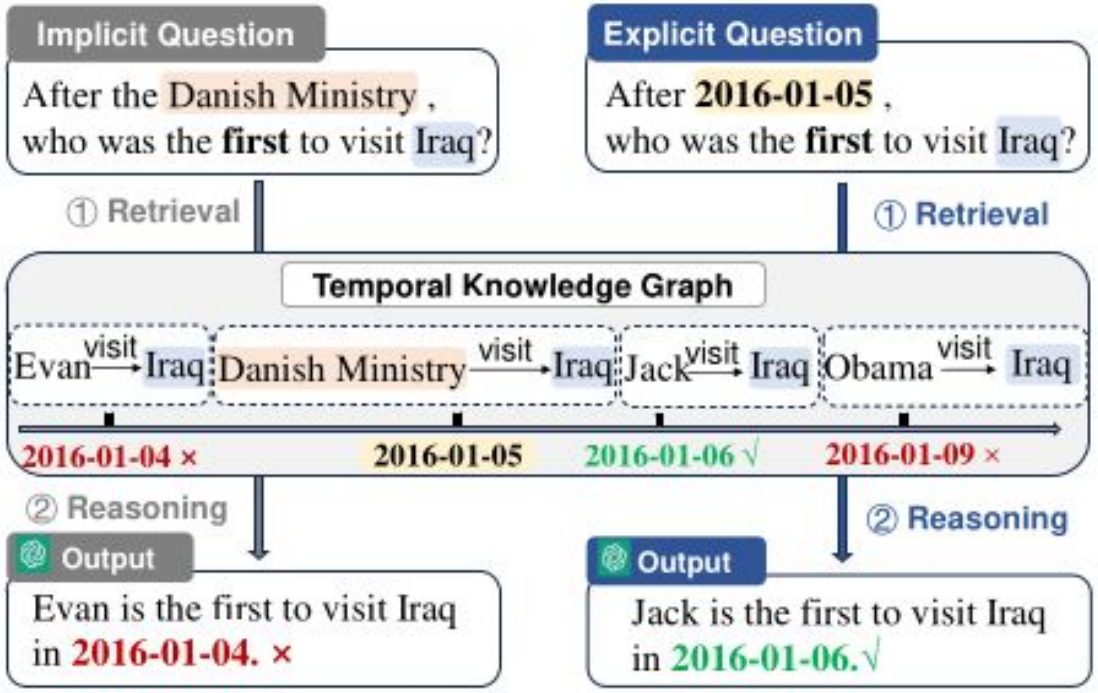


Figure 1: Examples of challenges in integrating temporal knowledge graphs with large language models.

This work...

1. Uses a **retrieve-rewrite** strategy to mitigate hallucinations by converting implicit questions to explicit ones.
2. Employs a contrastive time-aware **retrieve-rerank** strategy to simultaneously capture semantics and temporal constraints.
3. Fine-tunes **LLMs** on two **datasets**. Both open-source.
4. Demonstrates **huge improvements** over existing approaches and enhanced reasoning capabilities.

TimeR ^{4}: Fact Retrieval

1. Each quadruple is converted to a natural language sequence, no entity linking.

2. PLM embeds all quadruples.

3. Fact Knowledge Store (**FKS**) stores the representations.

$$\text{FKS} = \{\mathbf{E}_f | \mathbf{E}_f = LM(S(s,p,o,t)), (s,p,o,t) \in \mathcal{G}\} \quad (1)$$

4. The Questions are also embedded.

$$\mathbf{E}_q = LM(q) \in \mathbb{R}^d \qquad (2)$$

TimeR^4: Fact Retrieval

5. FAISS is used for indexing and similarity calculation.

6. The **k-nearest** semantic quadruples are based on their similarity score with the question:

$$\phi_{FKS}(\mathbf{E}_q, \mathbf{E}_t) = \cos(\mathbf{E}_q, \mathbf{E}_f) = \mathbf{E}_q \cdot \mathbf{E}_f \qquad (3)$$

$$\mathbf{f} = \arg\max \phi_{FKS}(\mathbf{E}_q, \mathbf{E}_f) \qquad (4)$$

TimeR ^{4}: Rewrite

7. The background facts are retrieved through the **FKS** and then given to the LLM for rewriting.

8. Questions are translated to contain explicit timestamps*

$$q^* = LLM(Prompt(q, f)), q \in Q \qquad (5)$$

Rewriting Prompt Template

Replace the temporal fact in questions with explicit timestamps from the provided facts or your knowledge without any explanation. If you are not sure about the answer, return the original questions.

For instance, from the fact:
"[Juan Carlos I, Praise or endorse, Vietnam, 2006-02-22]",
We can modify the question:
"After Vietnam, who was the first to praise Juan Carlos I?"
to "After 2006-02-22, who was the first to praise Juan Carlos I?"

Here is your turn:
Facts: 〈fact〉
Question: 〈question〉

Figure 5: Rewriting prompt template.

TimeR ^{4}: Time-Aware Retrieval

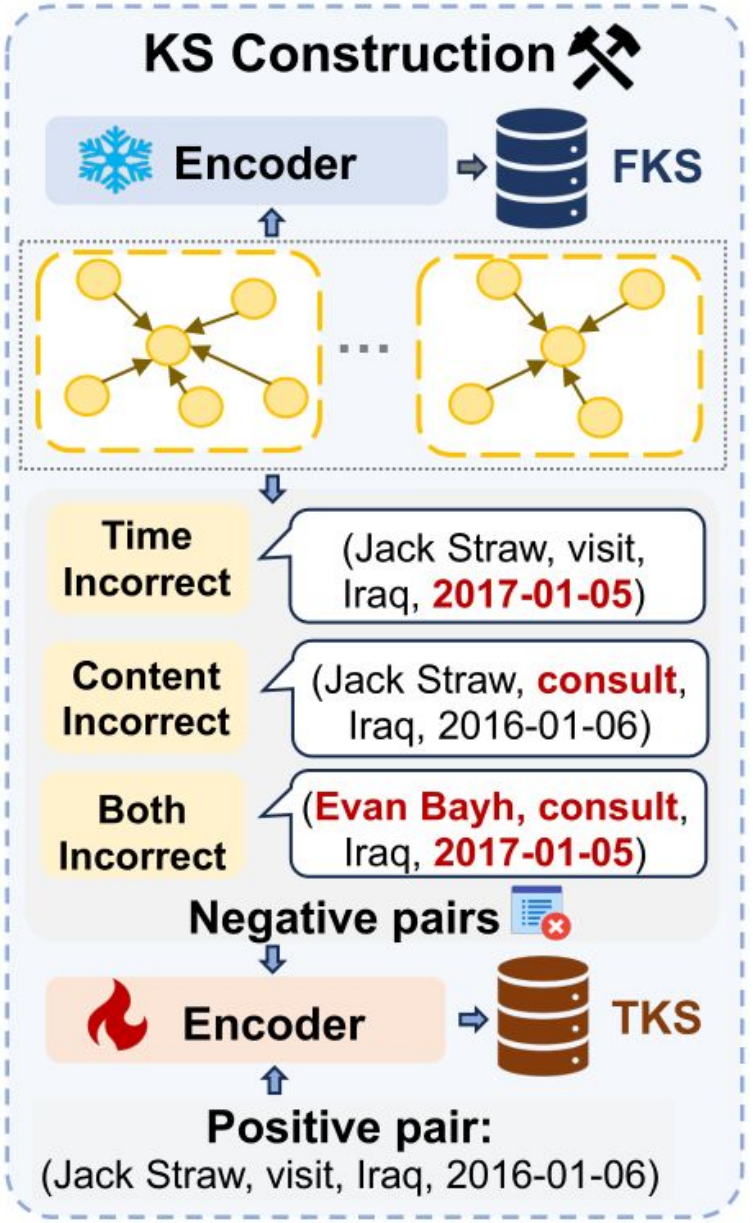
9. Construct the temporal knowledge store (**TKS**)

$$\text{TKS} = \{\mathbf{E}_t | \mathbf{E}_t = LM_t(S(s,p,o,t)), (s,p,o,t) \in \mathcal{G}\} \quad (6)$$

10. With Contrastive Time-aware Retrieval strategy

How?

- Creating hard negative examples for each positive.
- **Time incorrect**, **content incorrect**, and **both incorrect**



TimeR^4: Time-Aware Retrieval

11. Meaning, Minimize the Contrastive Loss Function

$$\mathcal{L} = \sum_i [w_p Y \cdot \exp(\phi_{TKS}) + w_n (1 - Y) \cdot \exp(1 - \phi_{TKS})] \tag{8}$$

wp - weight for positive example

wn - weight for negative example

$$\phi_{TKS} = \cos(\mathbf{E}_{q^*}, \mathbf{E}_t) \tag{7}$$

Goal

- minimize the distance between positive pairs and maximize the distance between negative pairs. What happens when Y=1, Y=0?

TimeR^4: Rerank

- **12.** Create a time-filtering function for irrelevant facts and more focus on time-related ones

  Purposes

  - Each (s, p, o, t) e F e G is filtered such that it falls between the range tq - t. Time difference is normalized:

$$\phi_t(t_q, t) = \begin{cases} 1 - \frac{|t_q - t|}{max(t_q - t)}, & \text{if } (t_q - t) > 0 \\ -100, & \text{otherwise} \end{cases} \quad (9)$$

- **13.** Rendering:

$$\phi(q, t) = \mu \cdot \phi_{TKS}(\mathbf{E}_{q^*}, \mathbf{E}_t) + (1 - \mu) \cdot \phi_t(t_q, t) \quad (10)$$

TimeR ^{4}: Reasoning

14. Reasoning over the retrieved quadruples is formulated as:

$$\mathcal{L} = \max_{\Phi} \sum_{(q^*, a) \in \hat{\mathcal{Q}}} \sum_{t=1}^{|a|} \log \left(P_\Phi \left(a_t \mid (q^*, f^+), a_{<t} \right) \right) \quad (11)$$

- **Maximizing** the probability of the answer **a** to question **q** from the knowedge graph **G** by using the retrieved quadruples **f +**
- LLMs are **generating answers** based on the instruction tuning prompt template.

Reasoning Prompt Template

Based on the historical facts, please answer the given question. Please keep the answer as simple as possible and return all the possible answers as a list.
Historical facts: 〈fact〉
Question: 〈question〉

Figure 6: Reasoning prompt template.

TimeR^4: Training

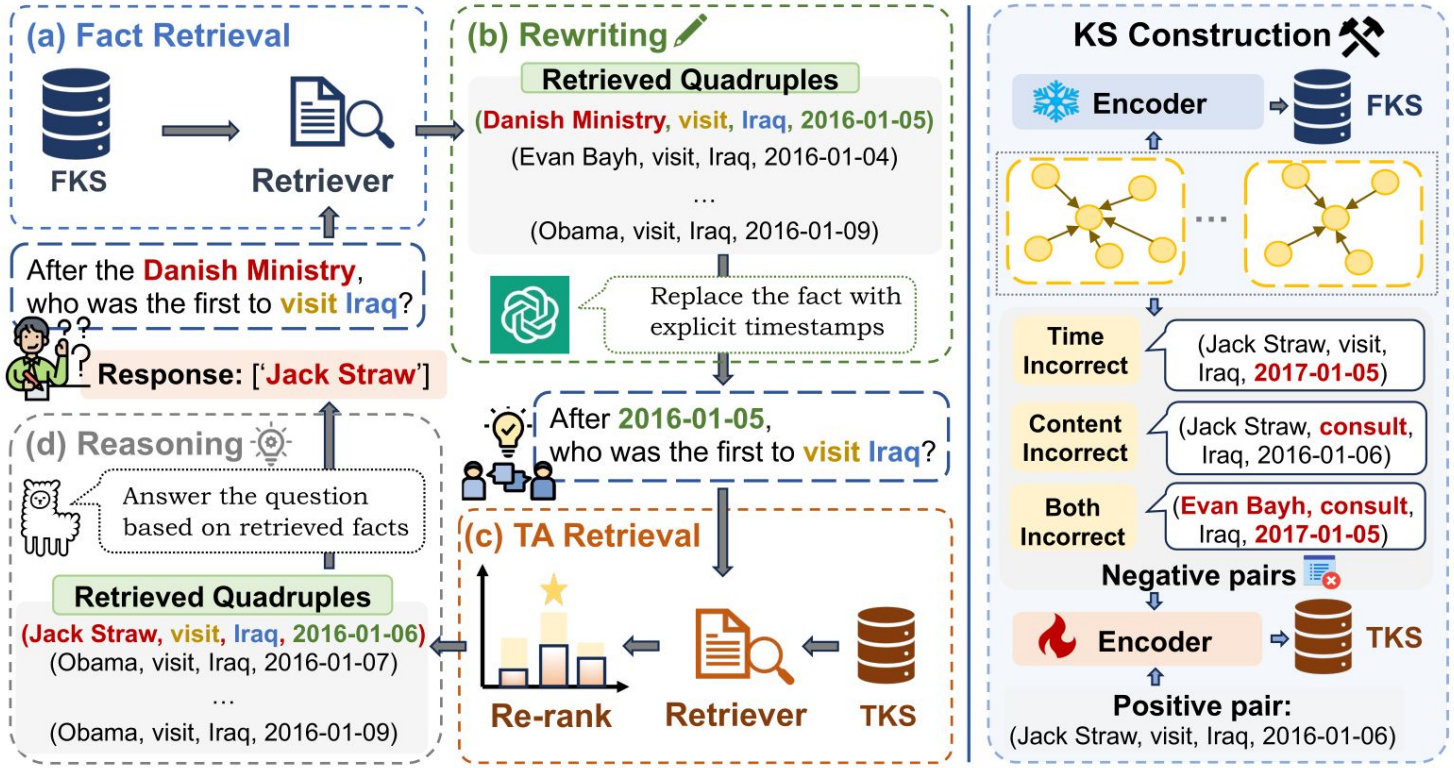


Figure 2: The architecture of TimeR^4 can be divided into four modules, fact retrieval, rewriting, time-aware retrieval, and reasoning. The right part shows how we construct the Knowledge Store (KS).

TimeR^4 full architecture

Algorithm 1: The training procedure of TimeR^4

Input : TKG $\mathcal{G}$, Questions $\mathcal{Q}$, Ground Truth gt, Language Model LM, raw LLM M

Output : Fine-tuned LLM M'

negatives $\leftarrow$ `GenerateNegatives`($\mathcal{G}$, $\mathcal{Q}$, gt)

$LM_t \leftarrow$ Optimize LM as Equation 8

$FKS \leftarrow$ `ConstructKS`($\mathcal{G}$, LM)

$TKS \leftarrow$ `ConstructKS`($\mathcal{G}$, LM_t)

for $\{q\} \in loader(\mathcal{Q})$ **do**

  $f \leftarrow$ `Retrieve`(FKS, q, LM);

  $q^* \leftarrow$ `ReWrite`(q, f);

  $f' \leftarrow$ `Retrieve`(TKS, q^*, LM_t);

  $f^+ \leftarrow$ `ReRank`(f', q);

  $M' \leftarrow$ Optimize M as Equation 11

end

Function `Retrieve`(KS, q, LM):

  $\mathbf{E_q} \leftarrow$ `LM`(q);

  $\phi_s \leftarrow$ `cos`($\mathbf{E_q}$, KS);

  $f^+ \leftarrow$ `TopN`(ϕ_s);

  return f^+;

Function `ConstructKS`($\mathcal{G}$, LM):

  $KS \leftarrow$ `LM`($\mathcal{G}$);

  return KS;

TimeR^4 training procedure

Experimental Work: Datasets

- **MULTITQ** - the largest **TKGQA** dataset, with 500K unique QA pairs, featuring multiple temporal granularities - years, months and days, and questions spanning over 3600 days.

- **TimeQuestions** - 16K temporal questions with time granularity on years.

Category		Train	Dev	Test
Single	Equal	135,890	18,983	17,311
	Before/After	75,340	11,655	11,073
	First/Last	72,252	11,097	10,480
Multiple	Equal Multi	16,893	3,213	3,207
	After First	43,305	6,499	6,266
	Before Last	43,107	6,532	6,247
Total		386,787	587,979	54,584

Table 1: Statistics of MULTITQ dataset.

Category	Train	Dev	Test
Explicit	2,724	1,302	1,311
Implicit	651	291	292
Temporal	2,657	1,073	1,067
Ordinal	938	570	567
Total	6,970	3,236	3,237

Table 2: Statistics of TIMEQUESTIONS dataset.

Datasets: Baselines

MULTITQ

1. Pretrained LMs - **BERT, ALBERT**

2. Embedding-based - **EmbedKGQA, CronKGQA,MultiQA**

3. LLM-based - **ARI, LLaMA2, ChatGPT**

TimeQuestions

1. KGQA - **PullNet**, **GRAFTNet**

2. TKGQA - **CronKGQA**, **TempoQR**, **EXAQT**, **LGQA**, **TwiRGCN**

3. LLM-based - **LLaMA2**, **ChatGPT**

Experimental Work: TimeR4 Setup

- **LLM** backbone - LLaMA2-Chat-7B
    - fine-tuned for 2 epochs on 2 NVIDIA A6000 GPUs
- **TimeQuestions** is left as it is, but **MULTITQ** is discarded on 80%
    - 100k examples left
- **Fact Retriever** - SentenceBERT
    - off-the-shelf
- **Time-Aware Retriever** - SentenceBERT
    - fine-tuned for 10 epochs
- **Rewriting** - gpt-3.5-turbo
    - OPENAI_API with 0.4 temperature

Experimental Work: Results

Model	Overall	Question Type		Answer Type	
		Single	Multiple	Entity	Time
BERT	8.3	9.2	6.1	10.1	4
ALBERT	10.8	11.6	8.6	13.9	3.2
EmbedKGQA	20.6	23.5	13.4	29	0.1
CronKGQA	27.9	13.4	13.4	32.8	15.6
MultiQA	29.3	34.7	15.9	34.9	15.7
ARI	38.0	68.0	21.0	39.0	34.0
LLaMA2	18.5	22.0	10.1	23.9	5.5
ChatGPT	10.2	14.7	7.7	13.7	2
TimeR[4]	**72.8**	**88.7**	**33.5**	**63.9**	**94.5**

Table 3: Performance comparison of different models (in percentage) on MUULTITQ.

metric: hits@1

Model	Overall	Question Type		Answer Type	
		Explicit	Implicit	Temporal	Ordinal
PullNet	10.5	2.2	8.1	23.4	2.9
Uniqorn	33.1	31.8	31.6	39.2	20.2
GRAFT-Net	45.2	44.5	42.8	51.5	32.2
CronKGQA	46.2	46.6	44.5	51.1	36.9
TempoQR	41.6	46.5	3.6	40	34.9
EXAQT	57.2	56.8	51.2	64.2	42
TwiRGCN	60.5	60.2	58.6	64.1	51.8
LGQA	52.9	53.2	50.6	60.5	40.2
LLaMA2	27.1	26.8	32.5	27.9	23.4
ChatGPT	45.9	43.3	51.1	46.5	48.1
GenTKGQA	58.4	59.6	61.1	56.3	57.8
TimeR[4]	**78.1**	**82.3**	**73.0**	**83.0**	**64.9**

Table 4: Performance comparison of different models (in percentage) on TimeQuestions.

metric: hits@1

Results: Discussion

- **PLMs** and **KGQA** have the worst results (expected)
- **TimeR4** has significant gains over other methods (ARI's ChatGPT on MULTITQ and ChatGPT, LLaMA2 on TimeQuestions)
- **ChatGPT** and **LLaMA2** perform much better on TimeQuestions but TimeQuestions is built on Wikidata, which is in their pretraining data.

Experimental Work: Ablation Study

- **Effect of Fact Retrieval**
  - replacing it with NER/entity linking
- **Effect of Rewriting**
  - instead relying on the original questions and the retrieved facts
- **Effect of Temporal Retrieval**
  - replacing it with the Fact Retrieval
- **Effect of Reranking**
  - removing the strategy completely

Model	Hit@1	
	MULTITQ	TimeQuestions
TimeR[4]	**72.78**	**78.1**
w/o fact retrieval	41.04	54.3
w/o rewrite	61.12	77.2
w/o temporal retrieval	<u>70.34</u>	77.3
w/o rerank	63.04	<u>77.9</u>

Table 5: Results of the ablation study. "w/o" means removing the module.

Experimental Work: Further Discussion

- **Comparison with LLMs**

Model	MULTITQ					Timequestions				
	Overall	Question Type		Answer Type		Overall	Question Type		Answer Type	
		Single	Multiple	Entity	Time		Explicit	Implicit	Temporal	Ordinal
TimeR^4	**72.8**	**88.7**	33.5	**63.9**	**94.5**	**78.1**	**82.3**	**73.0**	**83.0**	**64.9**
LLaMA2	18.5	22.0	10.1	23.9	5.5	28.9	26.8	41.9	33.7	33.8
LLaMA2 *w/ finetuned*	33.9	38.4	22.7	45.0	7.8	45.8	44.4	46.0	51.9	37.8
LLaMA2 *w/ TimeR*^{4}	39.1	44.2	26.6	37.0	44.2	59.3	57.4	51.5	73.4	41.0
ChatGPT	10.2	14.7	7.7	13.7	2	45.9	43.3	51.1	46.5	42.1
ChatGPT *w/ TimeR*^{4}	41.4	58.5	**41.2**	56.1	57.1	64.8	66.0	52.9	77.6	45.5

Table 6: Effects of integrating the TimeR^4 framework with different LLMs for reasoning.

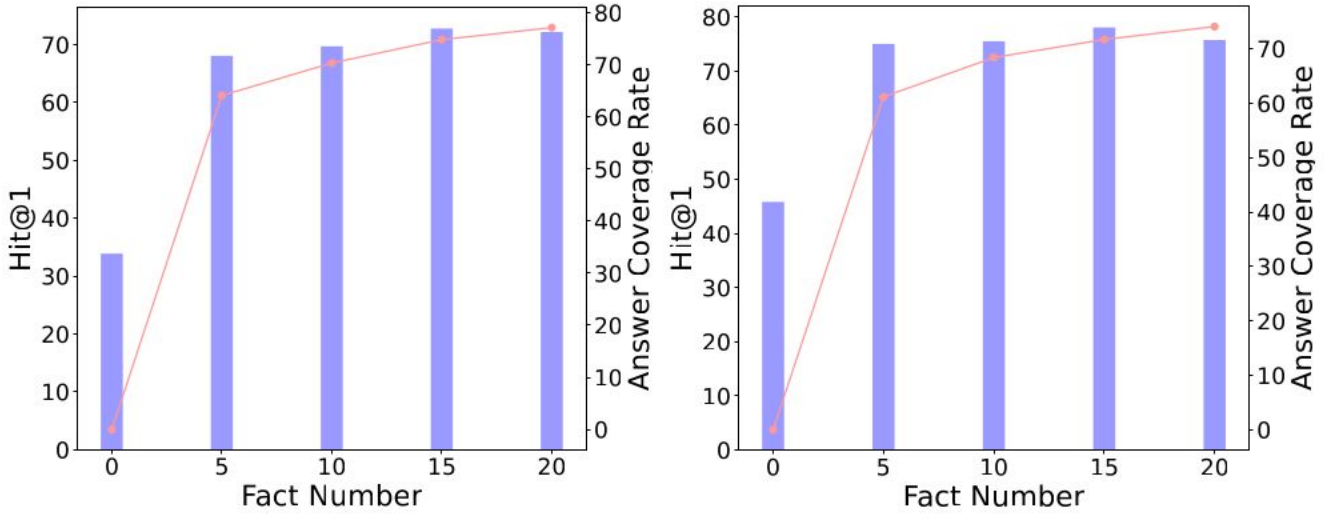
w/finetuned - LLMs are fine-tuned with only questions

w/TimeR4 - retrieved facts and rewritten questions are the input

Experimental Work: Further Discussion

Number of Retrieved Facts

- The model achieves peak performance with 15 relevant facts
- Too little facts do not provide sufficient knowledge
- Too many facts introduce noise



(a) Results on MULTITQ. (b) Results on TimeQuestions.

Figure 3: The Hits@1 results of different fact numbers.

Experimental Work: Further Discussion

Effectiveness of Retrieval

First-row:

- The Fact Retriever and the Time-Aware Retriever exhibit largely overlapping answer coverage.

Second-row:

- “Time-Aware Retriever answers most questions correctly that the Fact Retriever does, while also handling a significantly larger set of complex questions.”

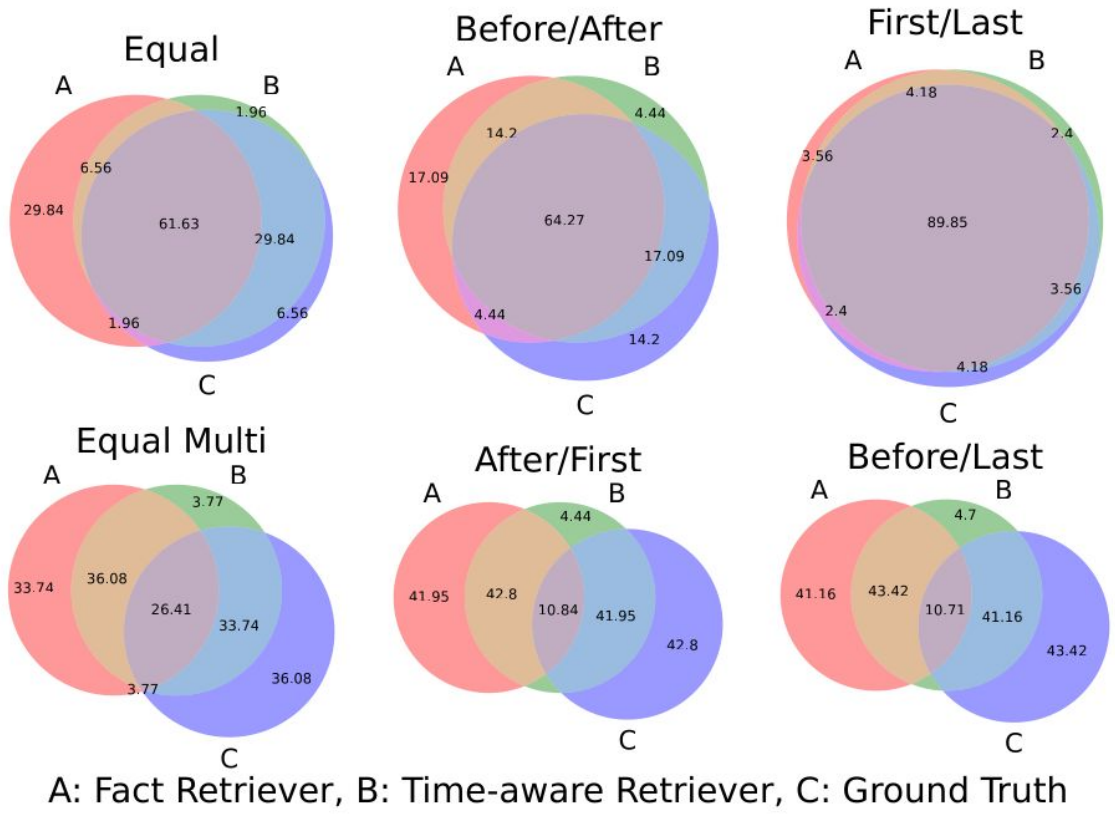


Figure 4: Venn diagrams for the answers coverage overlap of time-aware retrieval, fact retrieval, and ground truth for six question types in MULTITQ dataset.

Experimental Work: Case Study

Type	Question	Response		
		ChatGPT *w/ TimeR*^{4}	**LLaMA2 *w/ TimeR*^{4}**	**TimeR^4**
Equal	In 2005-01, who used light weapons to attack Thailand?	-Insurgent (Thailand) **-Citizen (Thailand)**	1. Insurgent (Thailand) 2. **Citizen (Thailand)** 3. Armed Opposition (Thailand)	**['Citizen (Thailand)', 'Armed Gang (Thailand)']**
Before/ After	Who investigated China after 22 July 2015?	1.Police (South Korea) 2. Mainland Affairs Council 3. Police (South Africa)	1. Japan 2. South Korea 3. France 4. China	**['Xi Jinping']**
First/ Last	In which month did Benny Gantz first visit China?	- **May** - August	**May**, August, or July	**['2012-05']**
Equal Multi	Who was the first to praise Iraq in 2015?	Iran	1. Iran 2. Iraq 3. el-Tayeb	**['Foreign Affairs (France)']**
After /First	After Okada Katsuya, who wish to visit Cambodia first?	- South Korea - Thailand - **Foreign Affairs (South Korea)**	1. Okada Katsuya 2. John Faulkner 3. Anupong Paochinda	**['Foreign Affairs(South Korea)']**
Before /Last	Who did Zimbabwe's Foreign Minister praise last before Kuwait?	- South Sudan - **Iran**	Guy Scott Mark Simmonds Tony Blair Faith Pansy Tlakula	**['Iran']**

Table 7: Comparison of responses to six different question types between our TimeR^4 and ChatGPT *w/ TimeR*^{4}, LLaMA2 *w/ TimeR*^{4}. The correct answers are highlighted in **bold font**.

Case Study: Discussion

- **Without fine-tuning**, LLMs are mostly providing random answers.

- **However,** LLMs respond to questions like “In which month …” with “May”, while the correct answer being “2012-05”. Is this really a false answer?

Key Takeaways: Conclusions

- **Fetches** relevant facts in the FKS and integrates specific timestamps into the questions (retrieve-rewrite) to mitigate LLM hallucinations.
- **Retrieves** facts from the TKS and reranks them based on temporal constraints (retrieve-rerank).
- **Enhances** significantly the temporal reasoning abilities of LLMs based on the achieved empirical results, however requires fine-tuning.

Key Takeaways: Limitations/Criticism

- The paper has inconsistent notation in places with sometimes highly ambiguous explanations.
- A line for future research is enhancing the retrieval of complex temporal facts.
- The models are limited without fine-tuning/standardizing them to the task format and/or evaluation methods cannot accurately capture their responses.

References

1. TimeR4 - please note all images are taken from this paper except explicitly indicated otherwise (with an embedded link).
2. GitHub Repository

THANK YOU FOR YOUR ATTENTION!

Q&A