ICLR International Conference On Learning Representations

KAIST

TDBench: Harnessing Temporal Databases for TSQA in LLMs

For automated, scalable, and reliable evaluation of Time-Sensitive QA (TSQA)

Soyeon Kim[1], Jindong Wang[2], Xing Xie[3], and Steven Euijong Whang[1]

[1]KAIST [2]William & Mary [3]Microsoft Research Asia

What is Time-Sensitive QA (TSQA)?

Asking questions where answers evolve over time

Answers Depend on Time

The correct factual answer changes depending on *when* the question is asked (e.g., "Current US President").

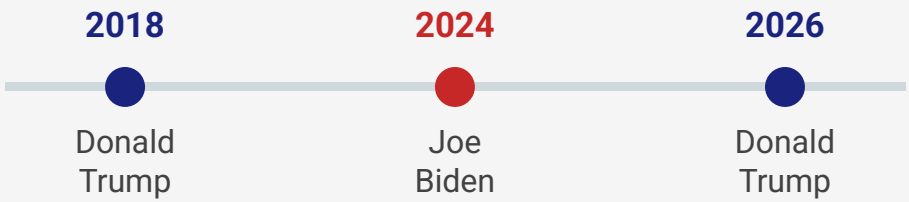
Requires Temporal Alignment

Models must distinguish between outdated and current knowledge, accurately reflecting the present state of the world.

Requires Temporal Reasoning

Going beyond fact retrieval to understand temporal relationships: *before*, *after*, *during*, or *overlapping* events.

Example: "Who is the current US President?"



"Static" QA fails here.

In TSQA, LLMs must answer the correct president based on the specific time constraint (e.g., "current", "in 2026", ...) mentioned in the query.

Background & Limitations

Critical Bottlenecks in Existing Time-Sensitive Question-Answering (TSQA) Benchmarks

High Manual Cost

- Heavy reliance on manual template design
- Limited to a small set of fixed templates (9–16 types), hindering scalability

Narrow Knowledge Source

- Centered almost exclusively on Wikipedia/Wikidata
- Poor coverage of specialized domains (e.g., Medical, Legal, Corporate)

Simplistic Temporal Logic

- Covers only basic relations like "Before" or "After"
- Misses complex intervals (“Overlap”, “Contain”, “Meet”)

Lack of Reasoning Verification

- Benchmarks only focus on the final answer string
- Fails to detect "Right Answer, Wrong Reason" (Reasoning Hallucination)

Motivation

Why We Need a Database-Driven Approach for TSQA

Prior Limitations

High Manual Cost & Simplistic Temporal Logics

Heavy reliance on manual template design.
Limited to a small set of fixed templates (9–16 types).

Lack of Reasoning Verification

Benchmarks focus on the final answer string.
Fails to detect "Right Answer, Wrong Reason".

Narrow Knowledge Source

Centered almost exclusively on Wikipedia/Wikidata.
Poor coverage of specialized domains.

TDBench Objectives

Automated Scale & Diverse Temporal Logics

Eliminate manual bottlenecks.
No need to write domain-specific templates.

Reasoning Verification Along with Answer

Benchmark focus on both the final answer and rationales.
Detect "Right Answer, Wrong Reason".

Generalizability to Arbitrary Domains

Extend beyond Wikipedia to domains like Legal/Medical.

TDBench Framework

Automated 3-Step Construction & Verifiable Evaluation Pipeline

TSQA CONSTRUCTION

1

Knowledge Selection

Filters time-varying attributes via **TFDs** (Temporal Functional Dependencies) to ensure dynamic context.

$$country,\ role \xrightarrow{T} name$$

2

Temporal SQL Generation

Constructs queries using **Allen's Algebra** (13 relations) and **Temporal Joins** for complex reasoning.

```
SELECT name, end FROM Leader
WHERE Country='Brazil' AND
Role='President' AND
date(end) = date('2019-05-01'
'-4 month')
```

3

Conversion to Natrual Language

Translates SQL to natural language via LLM, ensuring answers are **grounded in database facts**.

[Questions] "Who was the president of Brazil whose term ended exactly four months before May 1, 2019?", "Can you provide the president of Brazil who finished their term four months prior to May 1, 2019?"...
[Answer] Michel Temer **[Time reference]** 2019-01-01 (end)

TSQA EVALUATION

Answer Accuracy (A)

Checks if the final answer string is factually correct.

Time Accuracy (T)

Checks if the temporal rationale/evidence is valid.

Key Differences

What Makes TDBench Unique and Advanced to Prior TSQA Benchmarks

Fully Automated

Replaces human-written templates with database-driven generation. No manual effort required to create question contexts.

$$country,\ role \xrightarrow{T} name$$

13 Temporal Relations

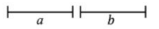
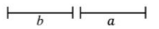
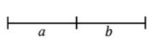
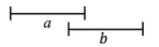
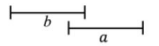
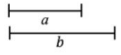
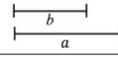
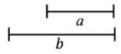
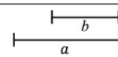
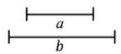
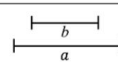
Covers the full spectrum of Allen's Interval Algebra (e.g., *Overlap*, *During*, *Meet*) using Temporal SQL.

Domain Scalability

Extensible to any domain with temporal data (Medical, Legal, Media), allowing for specialized benchmark creation.

Fine-grained Evaluation

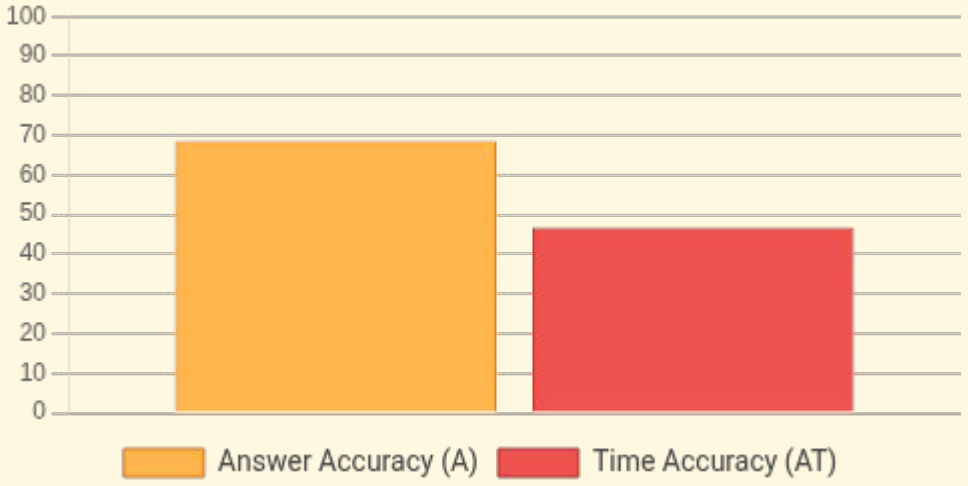
Introduces **Time Accuracy (T)** to verify if the model's rationale references the correct time period, exposing hallucinations.

Relation	Interval Diagram	SQL Condition	Example Temporal Context	Criteria
a before b		a.end < b.start	A president who ends before *September 20, 2000*	end
a after b		a.start > b.end	A president who starts after *March 20, 2001*	start
a meet b		a.end = b.end - b.length	A president who ends exactly *half a year* before *March 20, 2001*	end
a met-by b		a.start = b.start + b.length	A president who starts exactly *half a year* after *September 20, 2000*	start
a overlap b		a.start < b.start ∧ b.start < a.end < b.end	A president who starts before *September 20, 2000* and ends between *September 20, 2000* and *March 20, 2001*	start, end
		a.start < b.start ∧ a.end IS NULL	A president who is currently serving	start
a overlapped-by b		a.end > b.end ∧ b.start < a.start < b.end	A president who starts between *September 20, 2000* and *March 20, 2001* and ends after *March 20, 2001*	start, end
a equal b		a.start = b.start ∧ a.end = b.end	A president who starts in *September, 2000* and ends in *March, 2001*	start, end
a start b		a.start = b.start ∧ a.end < b.end	A president who starts in *September, 2000* and ends before *March, 2001*	start, end
a started-by b		a.start = b.start ∧ a.end > b.end	A president who starts in *September, 2000*	start
a finish b		a.start > b.start ∧ a.end = b.end	A president who starts after *September 20, 2000* and ends in *March, 2001*	start, end
a finished-by b		a.start < b.start ∧ a.end < b.end	A president who ends in *March, 2001*	end
a during b		a.start > b.start ∧ a.end < b.end	A president who starts after *September 20, 2000* and ends before *March 20, 2001*	start, end
a contain b		a.start < b.start ∧ a.end > b.end	A president who starts before *September 20, 2000* and ends after *March 20, 2001*	start, end

Experimental Results

Quantifying Hallucinations and Demonstrating Efficiency Across 8 LLMs

The "Reasoning Gap": 21.7% Drop



Finding: Models like GPT-4o and Llama 3.1-70B often get the correct answer string, but cite incorrect time periods (Reasoning Hallucination).

Efficiency: ~90% Cost Reduction

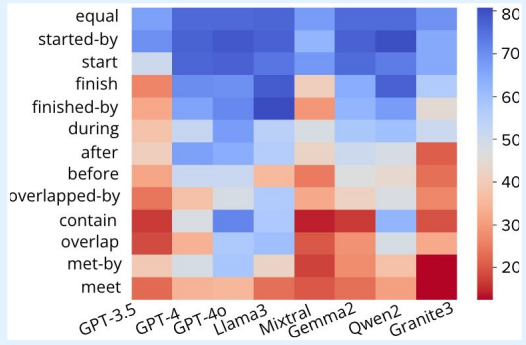
Netflix Dataset Token Usage (Per Question Generation)

Full Docs		1,288
TDBench		138

Qualitative Insights

Temporal Blind Spots

Diverse and complex temporal constraints identify model-specific weaknesses in temporal reasoning.



Conclusion & Takeaways

Summary of Contributions and Future Directions

Scalable Automation

Leveraging **Temporal Databases** and SQL eliminates manual bottlenecks, enabling the systematic generation of complex TSQA pairs at scale.

Trustworthy Evaluation

The **Time Accuracy (T)** metric uncovers "Right Answer, Wrong Reason" cases, revealing that LLMs hallucinate temporal rationales ~21.7% of the time.

Cost-Effective Scale

TDBench is ready for **domain-specific** benchmark (Medical, Legal) with minimal cost compared to LLM-only generation

Get TDBench Resources

Access the benchmark, code, and full paper.

https://github.com/ssoy0701/tdbench

https://iclr.cc/virtual/2026/poster/10009076

Thank You!

Questions? Reach out to us:

purplehibird@kaist.ac.kr