Recursive Question Understanding for Complex Question Answering over Heterogeneous Personal Data

Philipp Christmann and Gerhard Weikum

Max Planck Institute for Informatics, Saarland Informatics Campus, Germany

ReQAP-Website

ReQAP decomposes questions recursively for deriving an executable operator tree

MOTIVATION

★ Devices collect **vast amounts** of **personal data**

Calendar, Mail, Movies, Notes, Music, Shopping, Workouts, ...

★ **Personal data** naturally in **heterogeneous form**

★ **Real user questions** often quite **complex**
(verified in a user study collecting 2000 questions)

- *„How often did I eat Italian food after playing football?“*
- *„When did I last have pizza with both my parents?“*
- *„Which books did I buy for my friends last Christmas?“*
- *„Daily kilometers I ran during business trips?“*

★ Large language models **(LLMs)** inherently **limited**

LIMITATIONS OF THE STATE-OF-THE-ART

★ **Verbalization**[1] predominant for **heterogeneous QA**
⇒ **Evidence** from heterogeneous sources **linearized into textual form** and provided as input for retrieval-augmented generation (**RAG**)
⇒ RAG has limited context windows size
⇒ LLMs struggle with aggregation, grouping or joins, especially when numerical data involved

★ Alternative: **NL-to-SQL translation**[2]
⇒ Input and DB schema information **translated into code** (e.g., a SQL query), **for execution**
⇒ Works well with structured data and well-defined schema
⇒ Not the case here: **combination** of data from **structured** and **unstructured** content required (see user data on the right)
⇒ **Unstructured textual** content **not** easily **accessible**

1: Oğuz et al., UniK-QA: Unified Representations of Structured and Unstructured Knowledge for Open-Domain QA, NAACL 2022.
2: Katsogiannis-Meimarakis and Koutrika, A Survey on Deep Learning Approaches for Text-to-SQL, VLDB Journal 2023.

„How often did I eat Italian food after playing football?“

Calendar

18:00, 3 October 2024
Football practice

21:00, 3 October 2024
Dinner with friends
📍 Ristorante Napoli

20:30, 11 October 2024
Pizzaaa time!!

Workouts

19:02 – 20:43
11 October 2024
Workout: football
Duration: 1:40:46
Calories: 1,145 kcal
Ø-Heart rate: 146 bpm

Social media

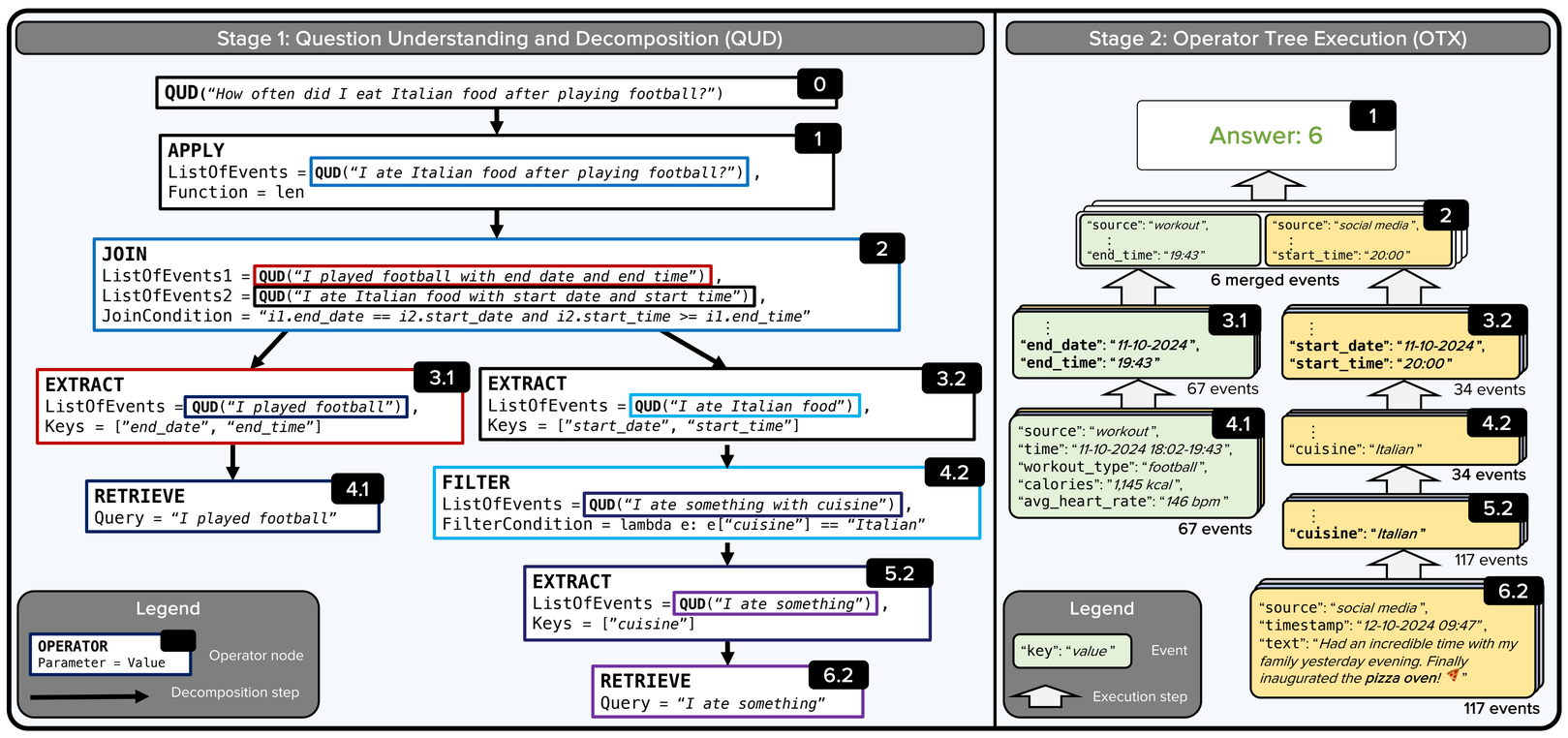
09:47, 12 October 2024
Had an incredible time with my family yesterday evening. Finally inaugurated the pizza oven! 🍕

17:02, 20 October 2024
Kept the three points at home today! 😍⚽

Mails

19:31, 15 October 2024
Dinner on Sunday? *To: Tom*
Hey Tom,
how is it going?
I was wondering if you and Anna would like to come over on Sunday evening? We finally got the new pizza oven up and running – would be great to fire it up together!
Felix and Lisa are coming too, so it should be fun.
Regards, Oliver

Excerpt of heterogeneous data for the example question.



Overview of the two stages in ReQAP, based on its inference for an example question.

ReQAP METHODOLOGY

(Recursive Question Understanding for Complex Question Answering over Personal Data)

★ **Novel** approach for **complex QA,** involving joins, grouping, aggregations or temporal constraints, over **heterogeneous sources**

★ **ReQAP combines** the strengths of **verbalization** and **translation** paradigms

★ **Stage 1: Decomposition** of question complexity
⇒ Each **recursive decomposition step** generates executable operator with simplified sub-question(s)
⇒ Construction of executable operator tree
⇒ Training via **in-context learning**, inference via **fine-tuned LM**
⇒ Strengths of translation paradigm

★ **Stage 2:** Novel **operators** for **processing text**
⇒ `RETRIEVE` enables **retrieval from heterogeneous sources** for sub-questions like *“my workouts”*
⇒ `EXTRACT` performs **on-the-fly information extraction** to augment retrieval results with structured key-value pairs
⇒ Strengths of verbalization paradigm

★ **Answers** by ReQAP are **traceable to sources**

★ Implementation with **small-scale models** for inference **on end user devices** (trained via LLMs)

OTHER OPERATORS

`RETRIEVE` (Query) `EXTRACT` (ListOfEvents, Keys)	Novel operators
`JOIN` (ListOfEvents1, ListOfEvents2, JoinCondition) `GROUP_BY` (ListOfEvents, Keys)	Standard SQL
`FILTER` (ListOfEvents, FilterCondition) `MAP` (ListOfEvents, Function) `APPLY` (ListOfEvents, Function) `UNNEST` (ListOfEvents, Key)	Basic functions
`ARGMIN, ARGMAX, MIN, MAX, SUM, AVG`	

EXTRACT OPERATOR

★ Augment events with structured key-value pairs
(e.g., adding value ”*Italian*” for a new key `cuisine` to a mail)

★ Efficient implementation via small-scale LM
(per-event processing with >10K inputs events)

★ LM input: verbalized event and key to extract
(notable user information, e.g. a list of friends, can be added)

★ LM output: extracted value
(when 70% of first 50 inputs lead to same input-key a mapping is extracted)

RETRIEVE OPERATOR

★ Candidate retrieval via SPLADE
(low threshold of 0.1 for high recall)

★ Identification of patterns to detect sets of events
(e.g., events from same source, or mails with same sender)

★ Classification of sets of events via cross-encoders
(3-way classifier: (i) keep all events, (ii) drop all events, (iii) decide per event)

★ For challenging sets, events classified per-event
(2-way classifier: (i) keep event, (ii) drop event; based on cross-encoder)

PerQA is challenging for QA systems, with complex questions and realistic user data

PerQA BENCHMARK

NEW!

★ **20** handcrafted **personas** with detailed information

★ **2,000** complex question **templates** (LLM-generated)

★ **3,500** complex **questions** (after template initialization)

★ **40,000 events** per **persona**
(calendar, mails, movies, music, purchases, TV series, social media, workouts)

★ LLM-verbalization of canonicalized events for realistic data

Canonicalized event:
```
“event”: “meeting”,
“participants”: [”Julian Groß”]
“location”: “restaurant”,
“cuisine”: “Italian”,
“restaurant”: “La Bella Napoli”
```

LLM-verbalized event:
```
“source”: “social media”,
“timestamp”: “02-07-2024 22:03”,
“text”: “La Bella Napoli never disappoints! Had a fantastic dinner with Julian tonight. The pasta was to die for 🍝👌 #GoodFood #LaBellaNapoli”
```

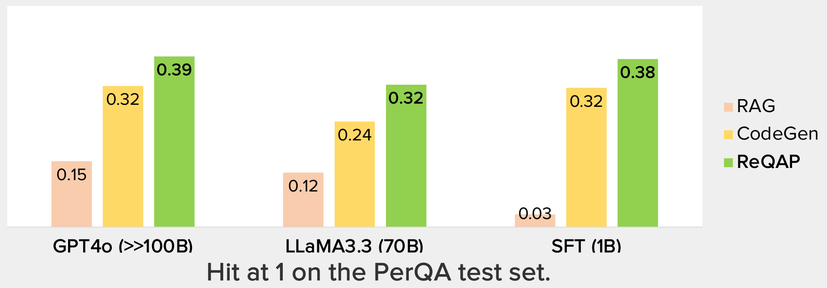
COMPLEXITY TYPES IN PerQA

(together with the number of questions they appear in, and an example)

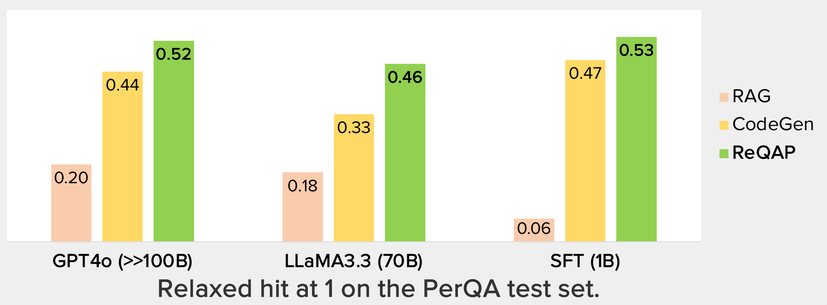
Ordering (1,236)	“*Which song did I listen to the most during my last trip?*”
Grouping (931)	“*which four restaurants do I go to the most?*”
Temporal (1,945)	“*How many times did I visit the doctor in the last 3 years?*”
Aggregation (1,790)	“*How much did I spend on online purchases last year?*”
Join (369)	“*How many times did I watch a movie after a run?*”
Multi-source (1,710)	“*which of my trips was the shortest?*”

EXPERIMENTS ON PerQA

★ **Baselines:** RAG and code generation (CodeGen)
★ **Models:** GPT4o (ICL), LLaMA3.3-70B (ICL), LLaMA3.3-1B (SFT)
★ **Metrics**: hit at 1, relaxed hit at 1 (allows +/- 10% for numerical answers)



Hit at 1 on the PerQA test set.



Relaxed hit at 1 on the PerQA test set.

ABLATION OF MODEL SIZES

	QUD → Operators ↓	XS (135M)	S (360M)	M (1B)	L (3B)
XS	`RETRIEVE` (4M) & `EXTRACT` (16M)	0.167	0.185	0.193	0.193
S	`RETRIEVE` (16M) & `EXTRACT` (31M)	0.240	0.287	0.302	0.302
M	`RETRIEVE` (23M) & `EXTRACT` (70M)	0.331	0.353	0.378	0.389
L	`RETRIEVE` (33M) & `EXTRACT` (139M)	0.356	0.364	0.396	0.400

Hit at 1 on the PerQA dev set.

USER STUDY EVALUATION

★ **Participants: 20** local **students**

★ **ReQAP** run **on** their **devices** and user **data**

★ **Each student** generated **100 questions**
(all 2000 questions publicly available)

★ Comparison with **benchmark questions**

	Answer correct	Answer acceptable
Real user questions on real user data	28 %	41 %
Benchmark questions on real user data	45 %	60 %

Saarland Informatics Campus

Website: reqap.mpi-inf.mpg.de

Demo: reqap.mpi-inf.mpg.de/demo

ReQAP-Website

MAX PLANCK INSTITUTE FOR INFORMATICS