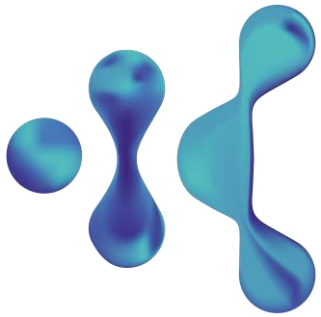


How well can "small" Language Models learn SPARQL w.r.t. a target KG?

Leveraging small language models for Text2SPARQL tasks to improve the resilience of AI assistance

Felix Brei, Johannes Frey, and Lars-Peter Meyer

Institute for Applied Informatics & Leipzig University

Supported by grants
FKZ: 13XP5116B FKZ: 01MK21007A
FKZ: 13XP5119F FKZ: 1MK22001A

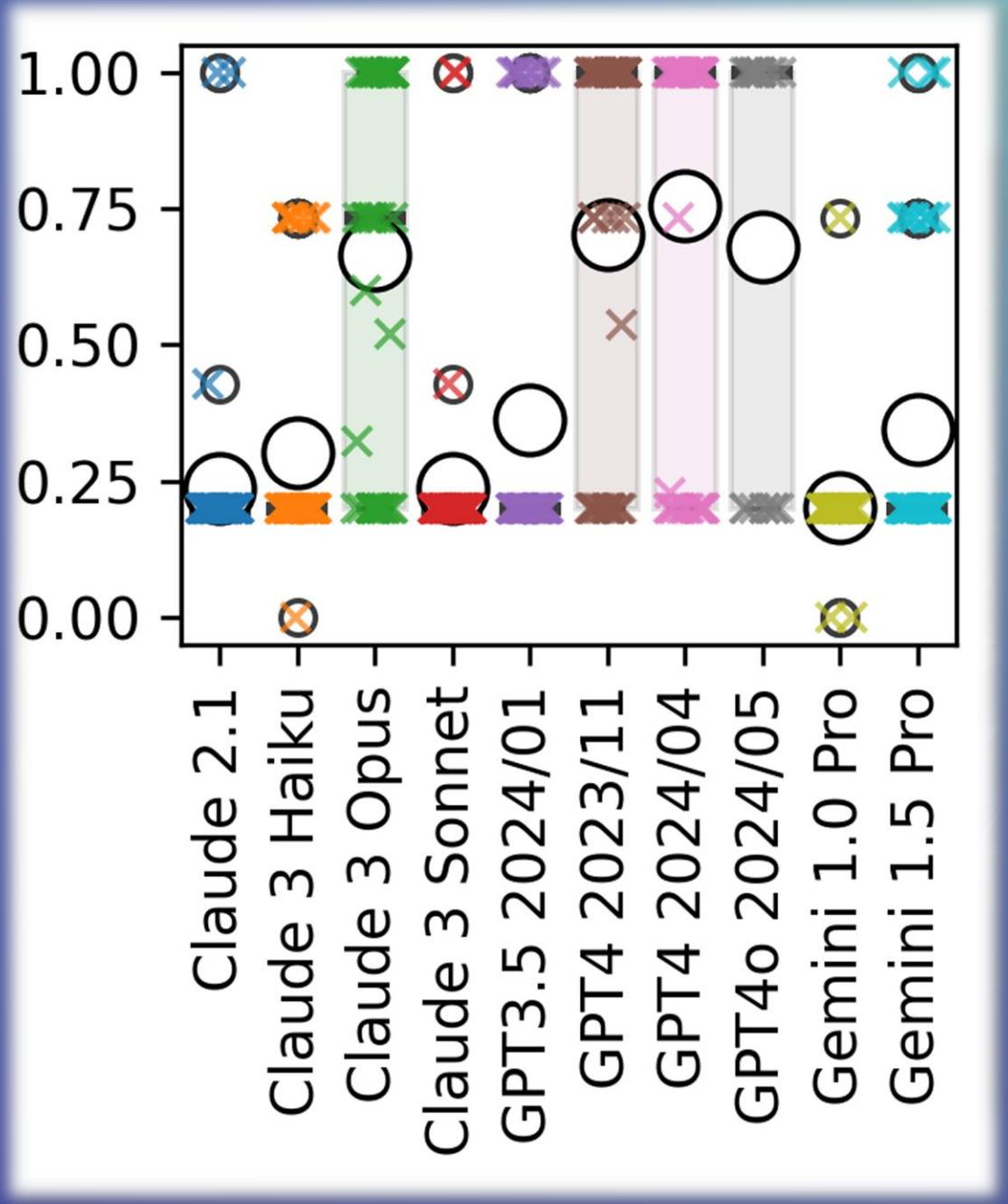
KISS
KI-GESTÜTZTES RAPID SUPPLY NETWORK

How to get Knowledge out of a Knowledge Graph?

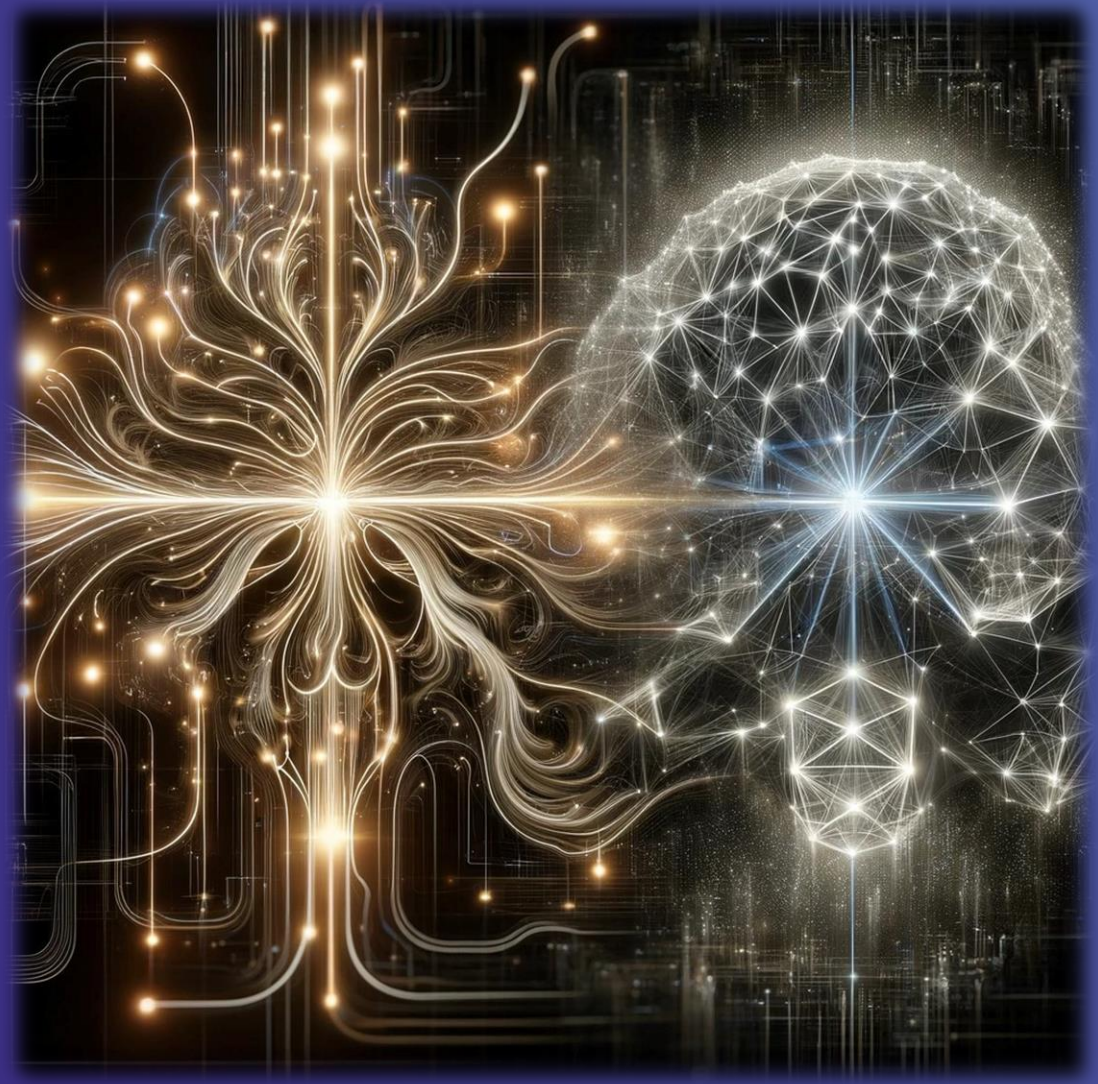
Writing a correct SPARQL query to answer a question requires knowledge about:

- Syntax/Semantics of SPARQL features
- Semantics of the classes, relations specific to one knowledge graph

LLMs can assist with that! → Chatbot Interfaces for non-technical users



KISS
KI-GESTÜTZTES RAPID SUPPLY NETWORK



So what's the drawback?

Currently the models that perform best out-of-the box are commercial

→ hosted by 3rd parties:

- Data protection is an issue
  - Need to provide access to schema or other APIs (e.g. entity lookup)
- Availability risks
  - Service at capacity / too slow
  - Network issues
  - Breaking updates
  - Service discontinuation
  - Sanctions, regulations, wars
- Costs (longterm?)
  - pricing policy could change anytime

KISS
KI-GESTÜTZTES RAPID SUPPLY NETWORK

Motivation

Empower small businesses or research facilities
to use Text2SPARQL with “**small** & **local**” AI

- Hosting models of comparable size to GPT, Gemini, Claude, etc. can be prohibitively expensive due to infrastructure/deployment costs
- Don't need an AI assistant that can do anything, but one that does one thing really good (UNIX approach)
  → After training, a model should be able to translate from natural language to SPARQL for one specific graph (only)
- Lots of open source language models are available for free and fit on “consumer-grade” hardware (8GB VRAM)

Step 1: Selecting language models

- According to a survey by STEAM, about 2/3 of their users have at least 8GB of VRAM available
- This is enough to hold a model with up to **1B parameters** and some training data
- Crawling through Huggingface gave us the following model families for our task

Family name	Parameter range in millions
T5	60.5 - 738
FLAN-T5	77 – 783
BART	139 – 611
M2M100	418 - 600
MREBEL	484 - 611

Step 2: Selecting datasets / target KGs

Aimed at three levels of difficulty: easy, medium, and hard

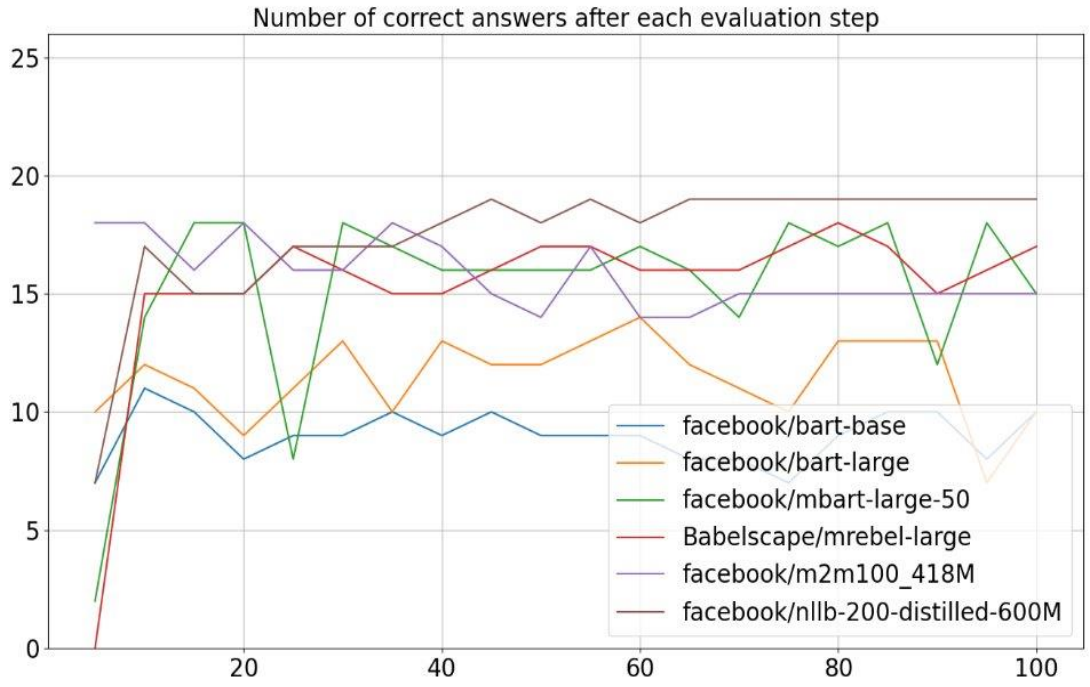
- Easy: Organizational graph
  - Well defined mapping between an IRI and the label of the object it points to (no cryptic identifiers)
  - Small, so only a few datapoints are needed to cover the full graph
  - Only well-known vocabularies (rdfs, owl, foaf, vcard, org)
- Medium: CoyPu mini graph
  - Real world example, subset of the knowledge graph from CoyPu project
  - larger than first one, about the size of one context window of ChatGPT

Step 2: Selecting datasets (cont.)

- Hard: Wikidata KG / QALD dataset
  - Based on Wikidata (numeric identifiers)
  - Very large knowledge graph, LM must learn the structure of the graph only from the Question-SPARQL-pairs provided during training
- QA Datasets for Org & Coypu:
  - Pairs of natural language question and corresponding SPARQL were generated by ChatGPT, along with expected query result
  - All queries were executed on the resp. graph and the results compared with the expected answer to filter out wrong queries

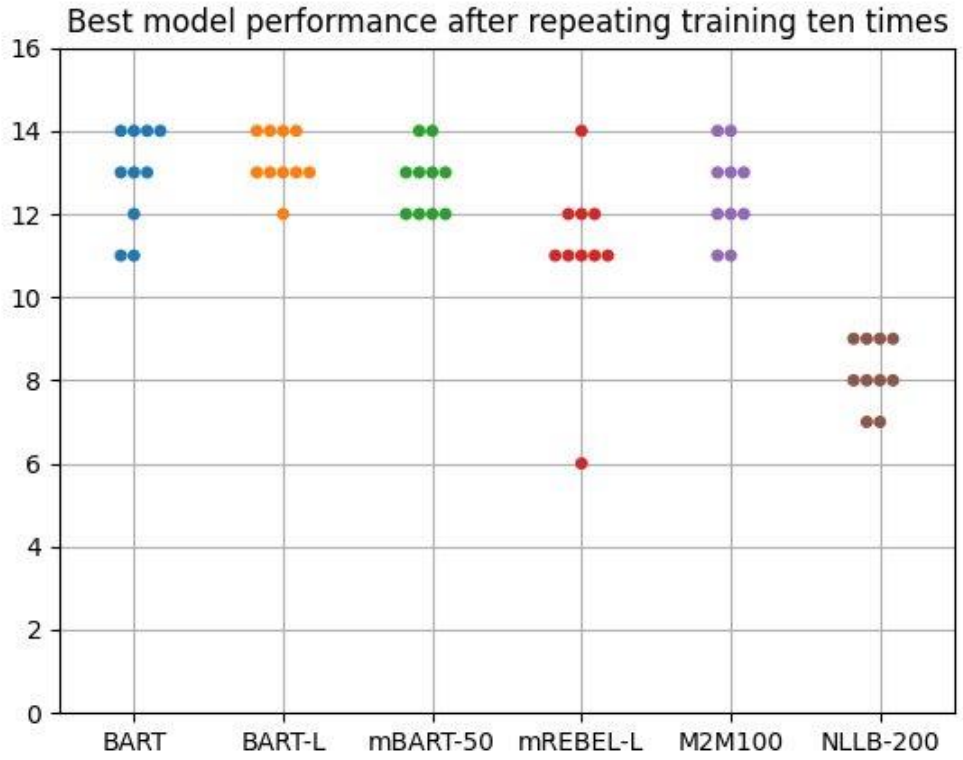
Step 3: Running the training / fine-tuning

- For each dataset we did the following things 10 times:
  - Shuffle the training data with a deterministic random seed
  - Train each of the models for 100 epochs
  - Run against validation dataset every 5 epochs
- Results on the right are for a single run to illustrate how the performance fluctuates



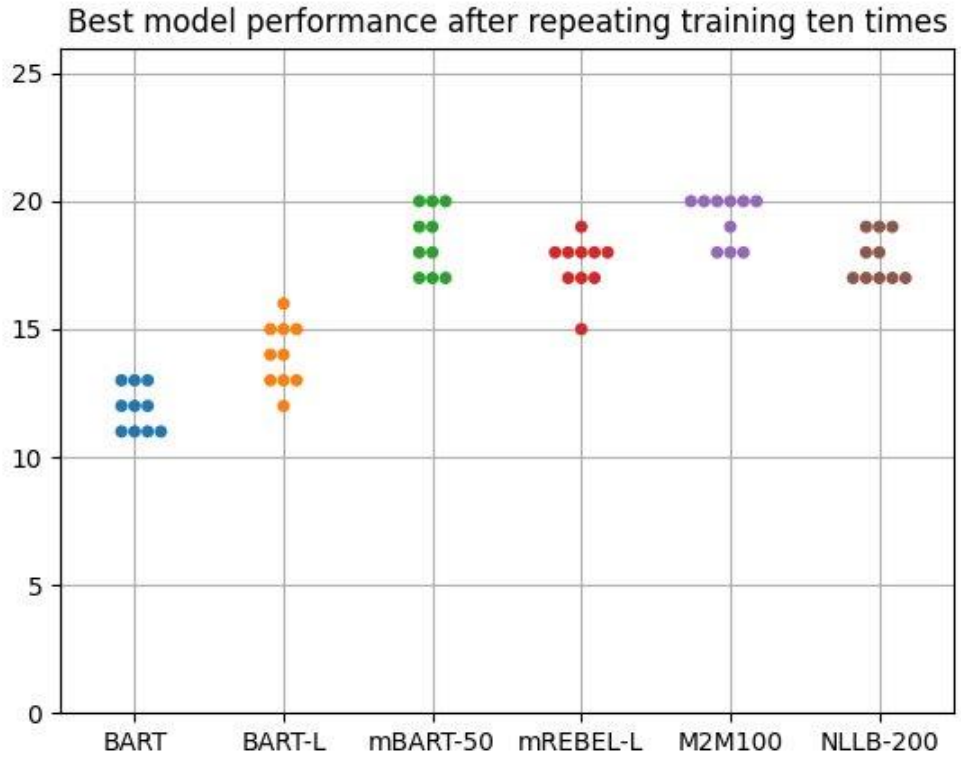
Results (Organizational)

- T5 family produced no correct query
- Other LMs manage to generate up to 14/16 correct SPARQL queries
- Outliers are present, rerunning the training after shuffling improved performance
- No clear winner, but NLLB-200 performs worst



Results (Coypu)

- Slightly different picture for CoyPu mini graph (medium difficulty)
- Esp. the models that are pretrained on multilingual data perform well
- Performance hits ceiling at 20/26 correct SPARQL queries



Results (Wikidata/QALD)

- LMs did not produce in a single correct answer
- 104 out of 394 queries parsed
- 51/104 queries empty result
- 50/104 COUNT with 0 as result

- IRI identifiers and prefixes are a problem

Selected Findings & Conclusions & FW

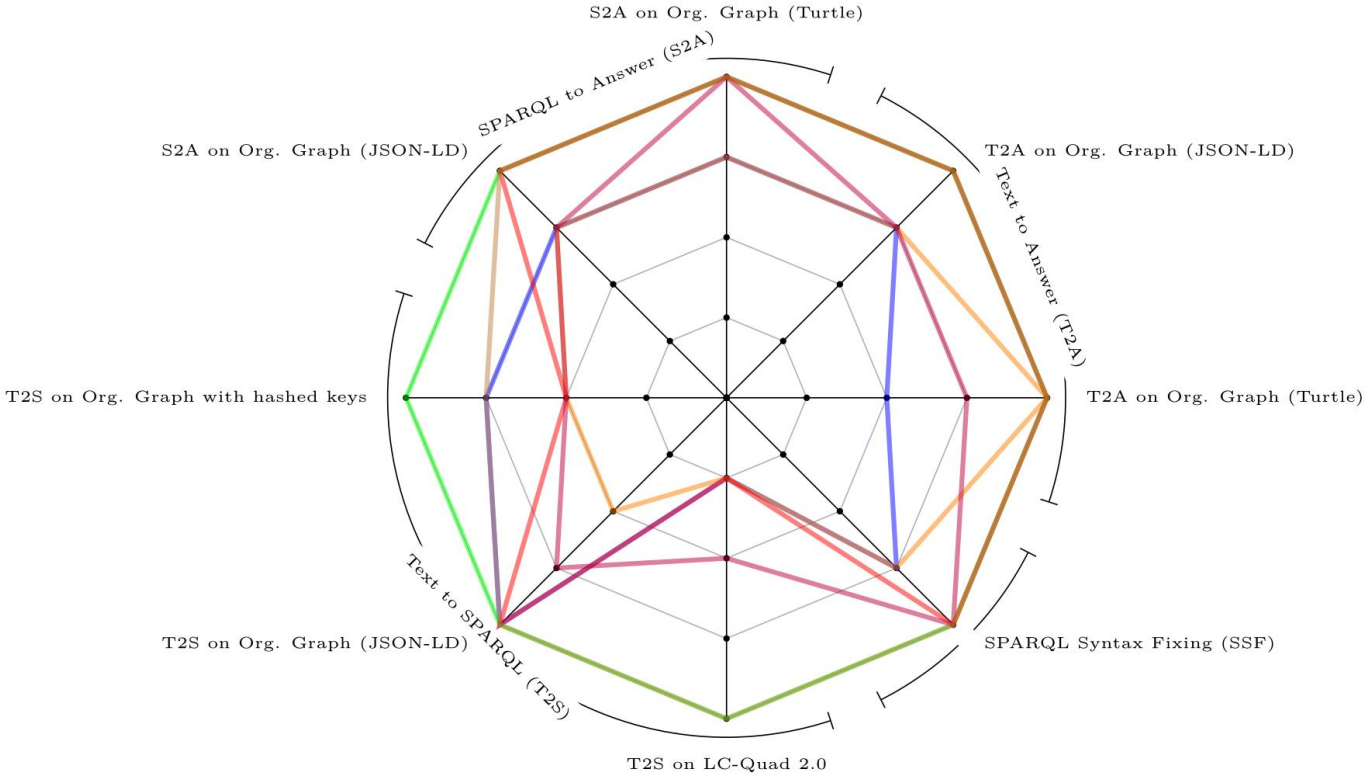
- Fine-tuned LMs can generate well-formed SPARQL queries and also meaningful queries with little training data for KGs (with human readable edges)
  - → Generating high quality training data for arbitrary knowledge graphs is an open issue
- Varying performance ranking across different KGs shows that there is not one single model that handles this task best
  - →experimenting with different models is encouraged and viable
- It is still under investigation, which properties of a graph favor which model architecture
  - →more fine-grained analysis especially with our custom graphs

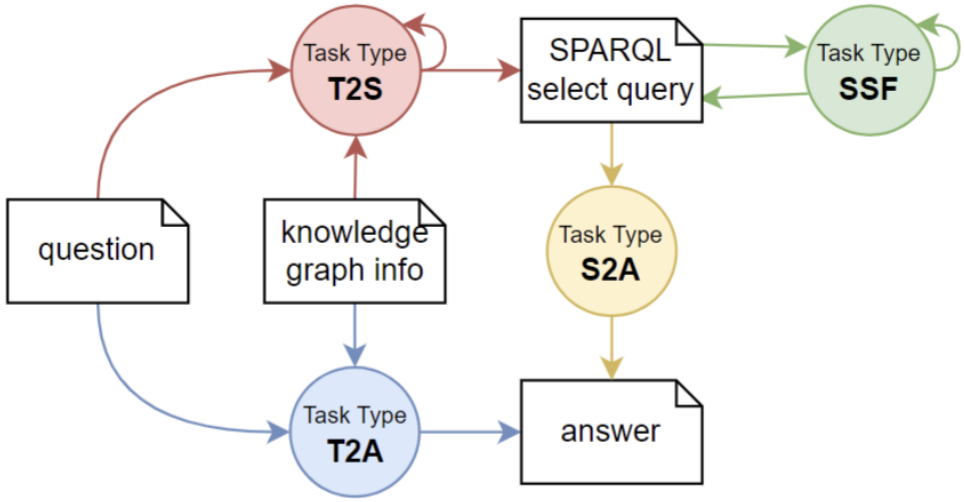
Future & Ongoing Work

Integrate/Align work into our LLM-KG-bench framework to assess fine-tuning efficiency in-depth

- Target KGs with slightly different IRI characteristics (e.g. numeric vs. human-readable)
- Iterative dialogs with feedback (syntax error, empty result set)

Different serialization formats (JSON-LD vs. Turtle)





Thank you

CONTACT

Felix Brei

brei@infai.org

Johannes Frey

frey@infai.org

Lars-Peter Meyer

lpmeyer@infai.org

KISS

KI-GESTÜTZTES RAPID SUPPLY NETWORK

Gefördert durch:

Bundesministerium für Wirtschaft und Klimaschutz

aufgrund eines Beschlusses des Deutschen Bundestages