

Chatbot for Alice Run3 simulation/analysis tasks

SUMMER STUDENT: TJAŠ AJDOVEC (UNIVERSITY OF LJUBLJANA)

SUPERVISORS: M. CONCAS, M. GIACALONE, S. WENZEL

What is the project about?

- Want to develop a chatbot with knowledge about the ALICE O2 software framework.
- It should be able to **use the existing documentation in order to answer basic questions** and provide user support.
- Available as a Mattermost user @askalicebeta
- It can help with reading documentation and reduce the load on experts.
- Based on a minimal prototype system.

Why not use ChatGPT?

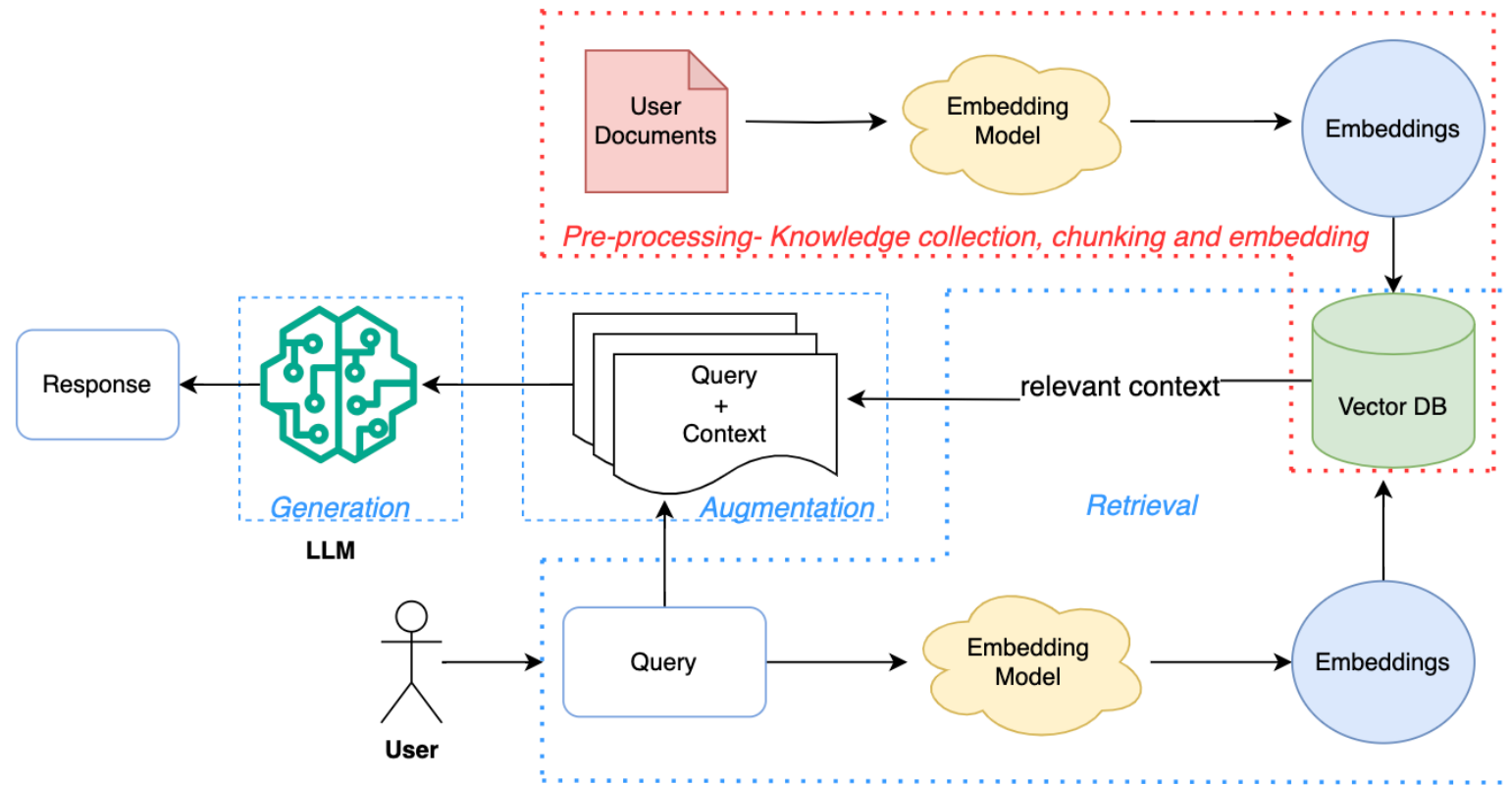
- Generally pretrained on public web data but not specialized.
- It has a **knowledge cutoff point** and gets updated only ~ once a year.
- Does not know about our **private repositories**.
- Not open-source and expensive.

Retrieval augmented generation (RAG)

- A technique to **adapt any generative large language model (LLM)** to our knowledge base without retraining it.
- It can be updated daily or even on documentation change.
- Cheap in terms of computing and works well for $O(1000)$ documents.
- Required components:
 1. Documents
 2. Vector database
 3. Text embedding and reranking models
 4. LLM
 5. User interface



RAG: Architecture



Building LLM applications using RAG - by Sagar Gandhi

Knowledge base

CHATBOT DATA

- **GitHub scraper** for simulation, data processing and analysis framework repositories
- Parsed presentation PDFs
- Transcribed talks from experts
- Total of ~ 400 documents = 3500 chunks

OTHER

- Collected real user questions from O2 simulation Mattermost channel
- **Evaluation dataset** of 35 expert question-answer pairs

LLM setup

- Popular models are available for free on HuggingFace.
- Named something like: <user>/<model>-<size>:<quantization>
- For example, unsloth/gpt-oss-20b-GGUF:Q8_0
- Served in a containerized environment on AMD MI100 GPUs with 32 GB of VRAM.
- Enough to run **medium-sized (20 to 30 B parameter) models**. For reference, DeepSeek-R1 uses 700B and ChatGPT-4 is speculated to be 1T+.
- UI and OpenAI compatible API hosted at <http://pc-alice-ph01:809X/> for each model.



Hugging Face

RAG evaluation

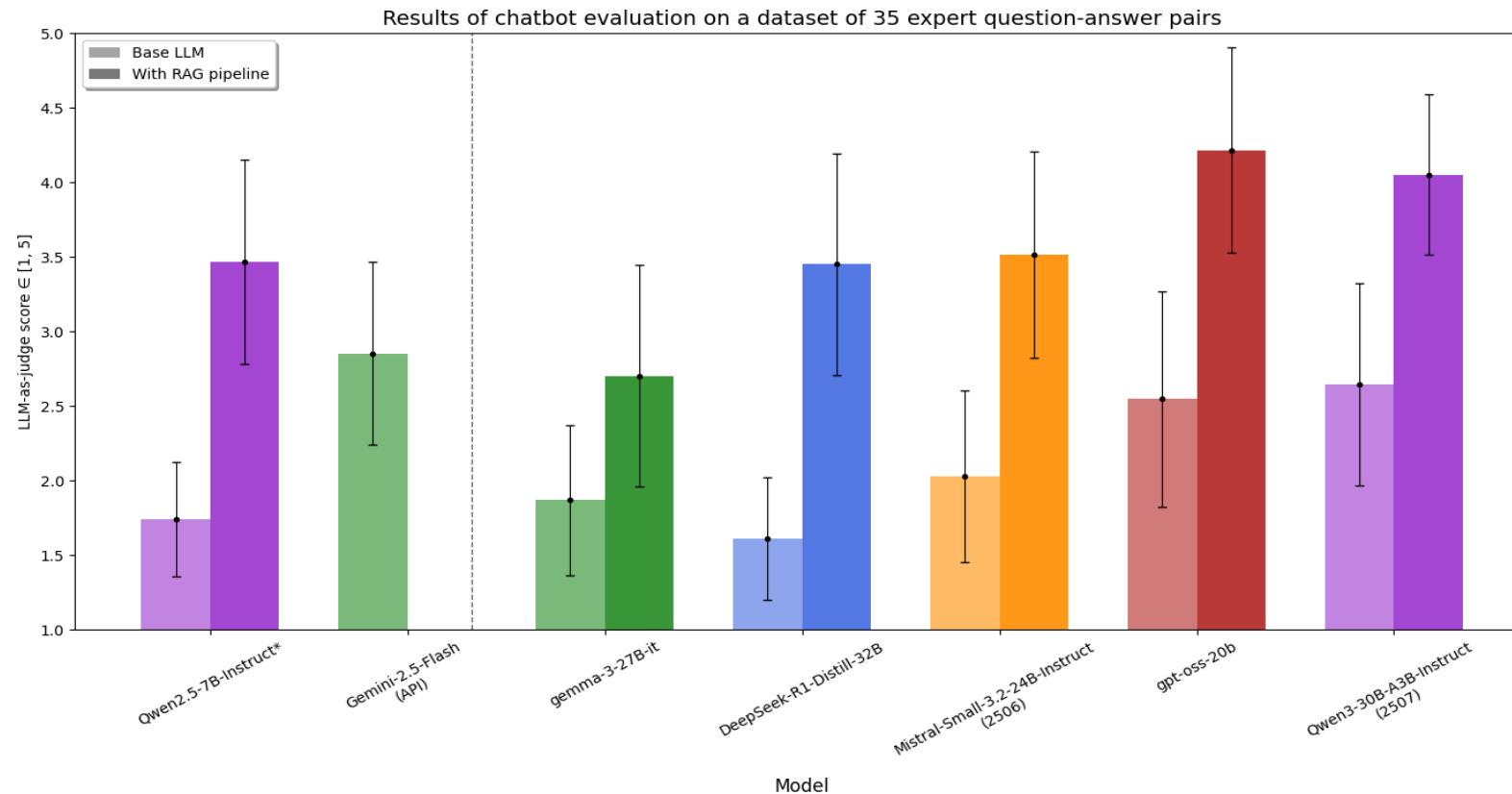
- To select the **best performing models and database retrieval parameters**.
- Used the 35 expert labeled question-answer pairs.
- For example: “How do you create and use event pools?”
 1. Choose a text similarity metric
 2. Prompt the chatbot to generate an answer
 3. Calculate similarity between the generated and correct answer
 4. Calculate mean of similarity scores across all questions
- But the problem is that LLMs are not deterministic!

RAG evaluation

- To select the **best performing models and database retrieval parameters**.
- Used the 35 expert labeled question-answer pairs.
- For example: “How do you create and use event pools?”
 1. Choose a text similarity metric
 2. Choose a sample size ($N = 5$)
 3. Prompt the chatbot to generate an answer N times
 4. Calculate similarity between the generated and correct answer N times
 5. Calculate mean and std of similarity scores across all generated answers
 6. Calculate mean and std of similarity scores across all questions
- We can plot the sample standard deviations as error bars.

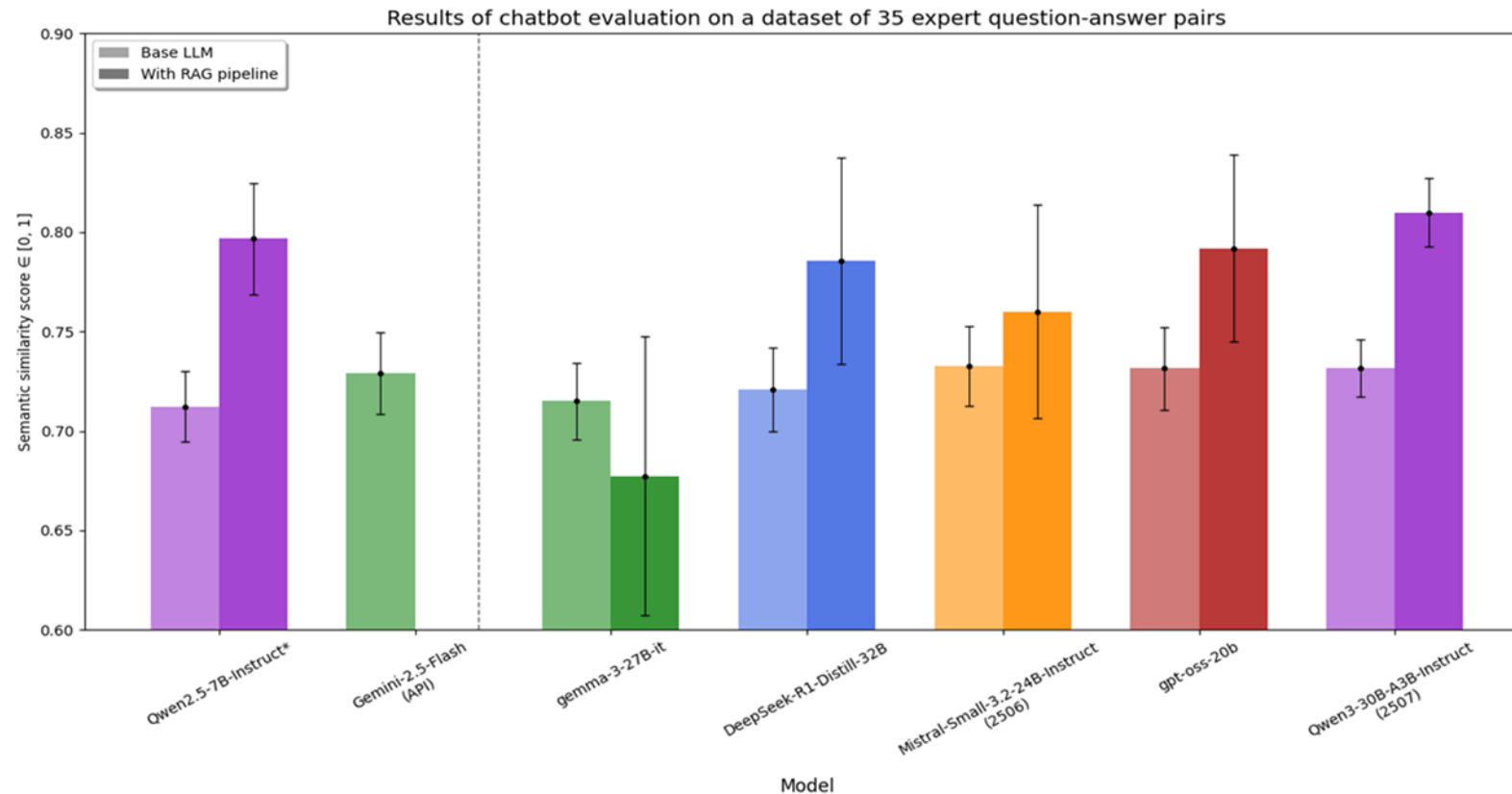
Model comparison

LLM-as-judge score: Ask another LLM to grade the generated answer against the correct one



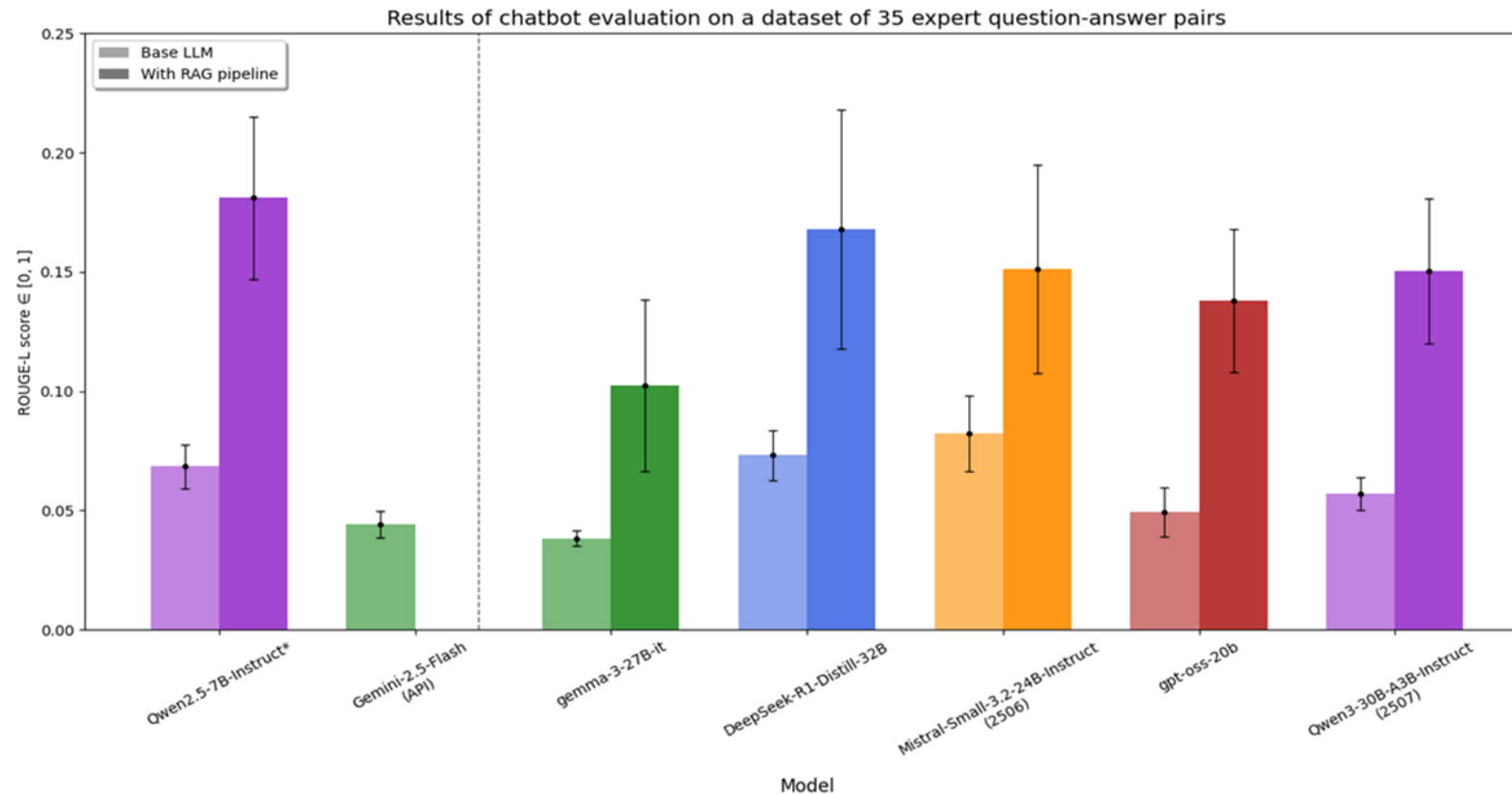
Model comparison

Semantic similarity score: Cosine distance between answer embeddings (\in vector space)



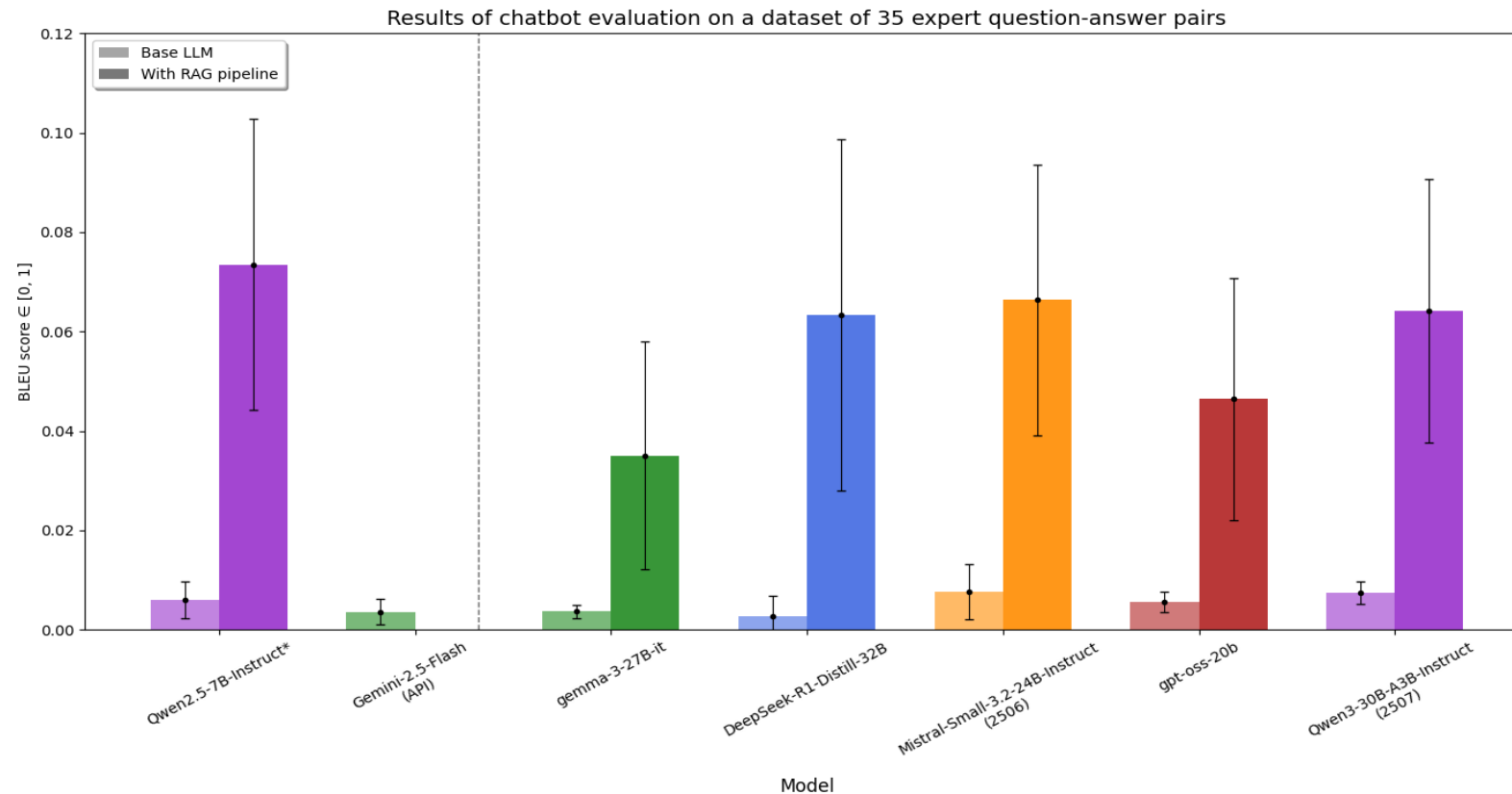
Model comparison

ROUGE-L score: Recall based longest sequence overlap between the generated and correct answer



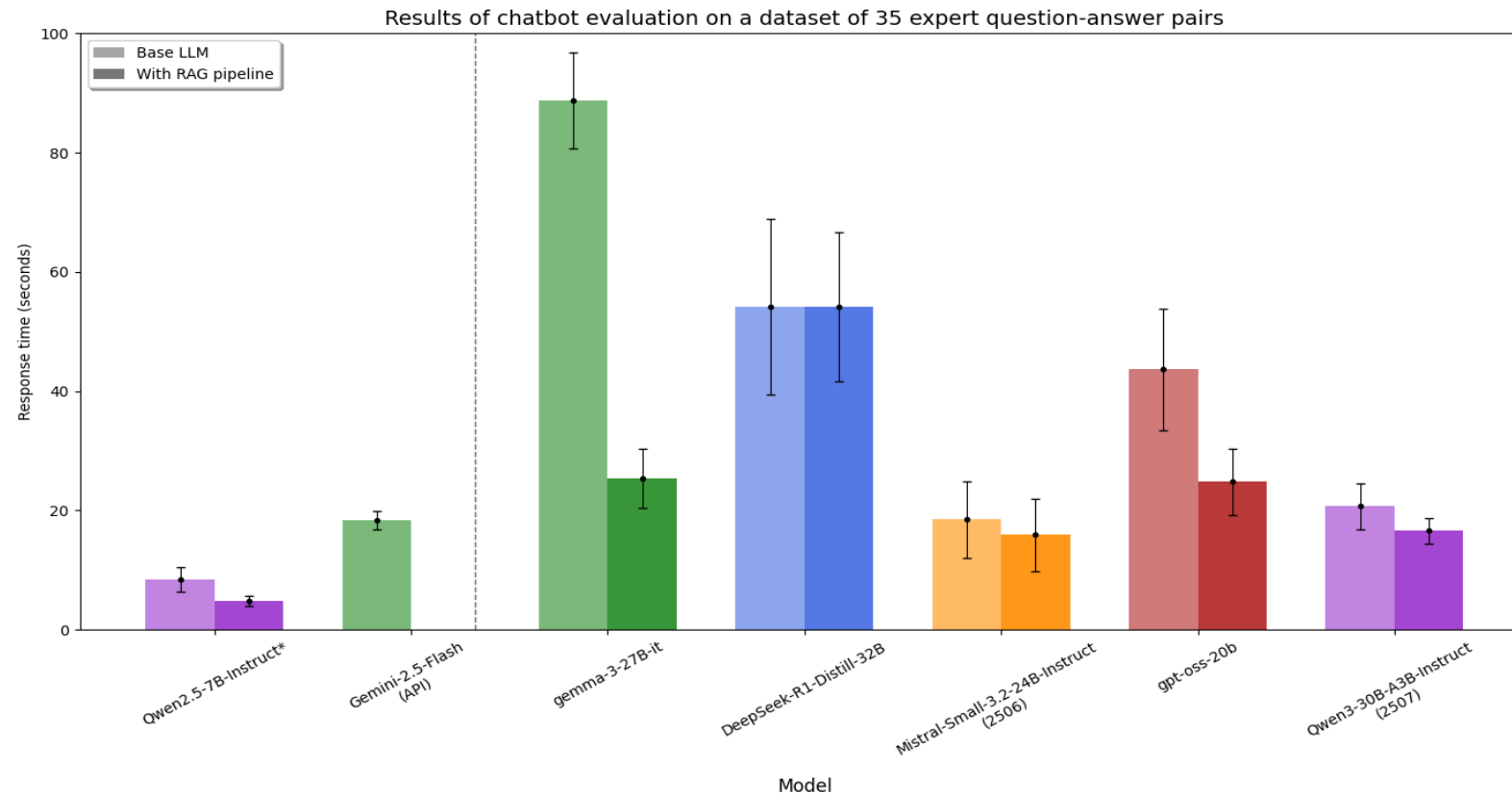
Model comparison

BLEU score: Precision based n-gram overlap between the generated and correct answer



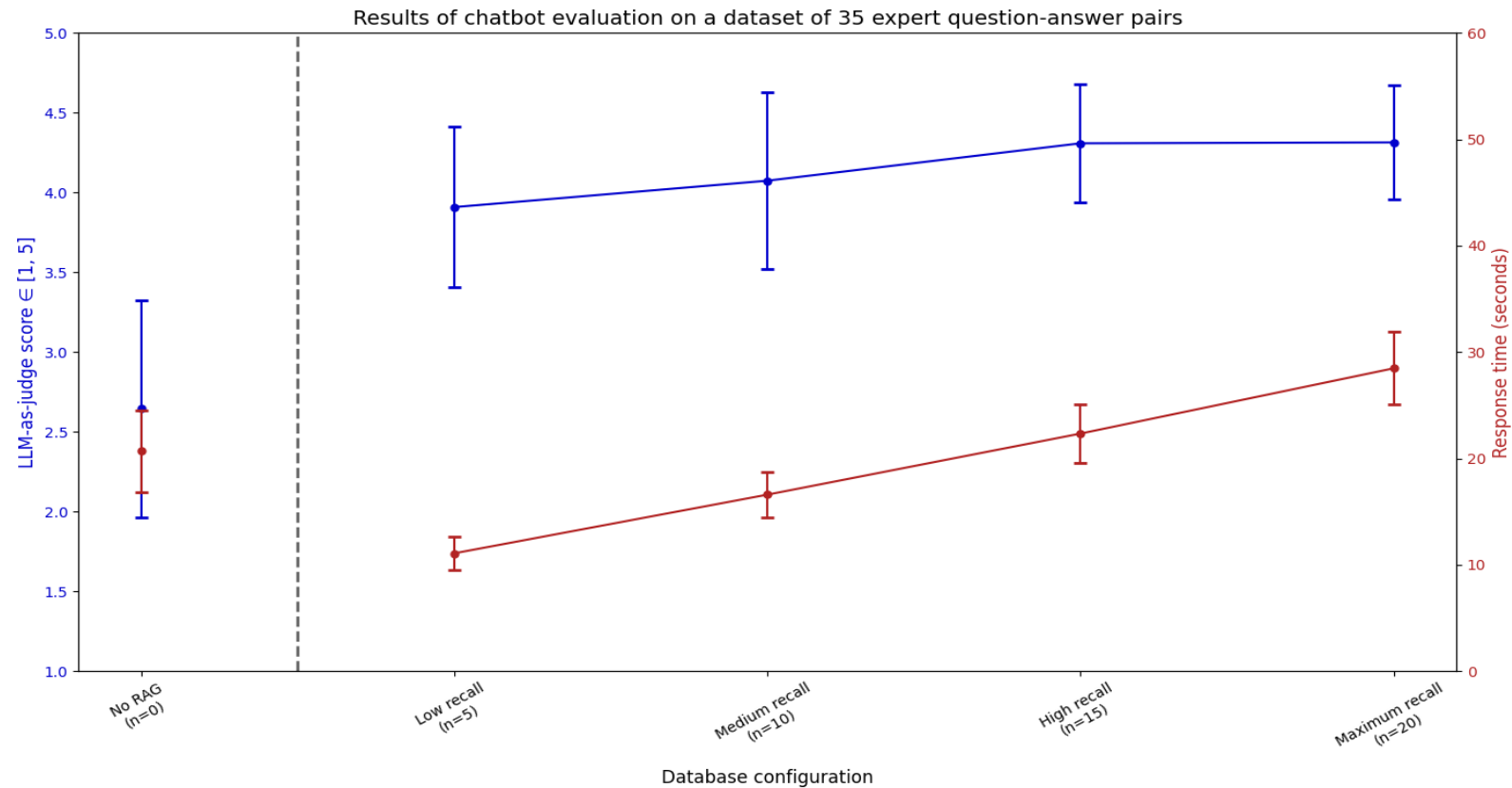
Model comparison

Response time: From sending a question to receiving the answer



Database configuration comparison

Response time scales linearly with the number of most relevant documents to retrieve.



How do you create and use event pools?

askALICE LLM chatbot 11:50 AM

To create and use event pools in ALICE O2 simulations:

1. Create an event pool:

Use the `--make-evtpool` flag in the `o2dpg_sim_workflow.py` script to generate a workflow for event pool creation. This skips transport steps and merges kinematic files (`kine.root`) from all timeframes into a single `evtpool.root` file. Example command:

```
1 | ${O2DPG_ROOT}/MC/bin/o2dpg_sim_workflow.py -gen <generator> -ecm <emc energy [GeV]> -tf <nTFs> --ns <nEvents> --make-evtpool
```

For instance, using Pythia8:

```
1 | ${O2DPG_ROOT}/MC/bin/o2dpg_sim_workflow.py -ecm 14000 -col pp -gen pythia8 -proc cdiff -tf 2 -ns 5000 --make-evtpool -seed 546 -interactionRate 500000 -productionTag "evtpoolcreation" -o evtpool
```

2. Run the workflow:

Execute the generated workflow using `o2dpg_workflow_runner.py`:

```
1 | ${O2DPG_ROOT}/MC/bin/o2dpg_workflow_runner.py -f evtpool.json -tt pool
```

3. Use the event pool:

Once created, the `evtpool.root` file can be used as input for subsequent simulations by providing it via the `-i` or `--input` argument in the workflow script. The input can be a local file or an Alien path. Example usage in a simulation workflow:

```
1 | ./event_pool.sh --make # to create the pool
2 | ./event_pool.sh -i evtpool.root # to use the pool
```

For detailed examples, refer to:

- [O2DPG event pool example script](#)
- [O2DPG workflow documentation](#)

Sources:

1. <https://github.com/AliceO2Group/simulation/blob/main/docs/o2dpgworkflow/README.md>
2. https://github.com/AliceO2Group/O2DPG/blob/master/MC/run/examples/event_pool.sh
3. https://github.com/AliceO2Group/simulation/blob/main/additional_resources/talks/O2_AnalysisTutorial_Nov2024/MCTutorial4Giacalone.pdf
4. https://github.com/ta5946/alice-rag/blob/master/data/knowledge_base/presentations/ALICE-simulation-tutorial.pptx-1.pdf
5. https://github.com/AliceO2Group/simulation/blob/main/additional_resources/talks/O2_AnalysisTutorial_April2023/ALICE-Run3-MC-HowTo.pdf
6. <https://github.com/AliceO2Group/simulation/blob/main/docs/o2dpgworkflow/anchored.md>
7. https://github.com/AliceO2Group/simulation/blob/main/additional_resources/talks/O2_AnalysisTutorial_April2023/ALICE-Run3-MC-HowTo_Transcript.md

This response used Qwen3-30B-A3B-Instruct-2507 model and a medium recall database.

Help us improve the askALICE chatbot by providing your feedback - React to this message with 👍 if the answer was helpful or 👎 if it was not.

 Edited


Example #1: How to create and use event pools?










Note that threads are treated as conversations.




Chatbot configuration

- By default, it uses the best performing LLM Qwen3-30B-A3B-Instruct and a medium recall database configuration.
- This can be changed by providing **user flags** like: #param:value

Parameter	Description	Values	Default
model	Which LLM to use for the response.	gpt , qwen , mistral , gemma , deepseek	Qwen
db	How many documents to retrieve from the vector database (recall).	low , med , high , max	Medium

#db:max Which generators are natively included in AliGenO2?

Aa ^   



Tjas Ajdovec Update your status 1:31 PM

#db:max Which generators are natively included in AliGenO2?



askALICE LLM chatbot 1:31 PM

The generators natively included in AliGenO2 are:

- `pythia8` (the default generator for ALICE Run3, with pre-configured options for pp and PbPb collisions)
- `pythia8pp` (pre-configured Pythia8 for proton-proton collisions)
- `pythia8hi` (pre-configured Pythia8 for heavy-ion collisions)
- `boxgen` (a simple mono-PDG particle generator that produces particles with uniform distributions in p , η , and ϕ)
- `extkin02` (used to read external kinematics files, e.g., generated in a previous simulation step)
- `hepmc` (reads events from HepMC files)

These generators can be directly specified using the `-g` option in `o2-sim`.

For more information, see:

<https://github.com/AliceO2Group/simulation/blob/main/docs/generators/generators2.md>

Sources:

1. https://github.com/AliceO2Group/simulation/blob/main/additional_resources/talks/O2_AnalysisTutorial_Nov2024/MCTutorial4Giacalone.pdf
2. https://github.com/ta5946/alice-rag/blob/master/data/knowledge_base/presentations/ALICE-simulation-tutorial.pptx-1.pdf
3. https://github.com/AliceO2Group/simulation/blob/main/additional_resources/talks/O2_AnalysisTutorial_April2023/ALICE-Run3-MC-HowTo.pdf
4. <https://github.com/AliceO2Group/simulation/blob/main/docs/generators/generators2.md>
5. https://github.com/ta5946/alice-rag/blob/master/data/knowledge_base/presentations/MC_Tutorials.pdf
6. https://github.com/AliceO2Group/AliceO2/blob/dev/run/SimExamples/AliRoot_AMPT/README.md
7. https://github.com/AliceO2Group/simulation/blob/main/additional_resources/talks/O2_AnalysisTutorial_April2023/ALICE-Run3-MC-HowTo_Transcript.md
8. https://github.com/AliceO2Group/AliceO2/blob/dev/run/SimExamples/AliRoot_Hijing/run.sh
9. https://github.com/AliceO2Group/AliceO2/blob/dev/run/SimExamples/AliRoot_Hijing/README.md
10. <https://github.com/AliceO2Group/simulation/blob/main/docs/generators/generatorscustom.md>

This response used Qwen3-30B-A3B-Instruct-2507 model and a maximum recall database.

Help us improve the askALICE chatbot by providing your feedback - React to this message with 👍 if the answer was helpful or 👎 if it was not.

Edited

Example #2: Which generators are included in AliGenO2?



Tjas Ajdovec

Update your status 11:45 AM

#model:gpt What is the easiest way to access MC tracks given a MC label?



askALICE LLM chatbot 11:45 AM

The quickest way is to use the `MCKinematicsReader` helper class. Create a reader from the digitization-context file and then call its `getTrack()` (or `getTracks(event)` and look up the label).

For example:

```

1  #include <o2/steer/MCKinematicsReader.h>
2
3  o2::steer::MCKinematicsReader reader("collisioncontext.root"); // or "o2sim" for a kinematics file
4  o2::MCTrack track = reader.getTrack(mcLabel);                // mcLabel is the MC label you have

```

`MCKinematicsReader` is specifically designed to fetch MC tracks directly by their label (or by event), so it is the easiest way to access a track once you know its MC label.

References:

- “MCKinematicsReader – Class to easily read and retrieve tracks for a given event or a Monte Carlo label” (GitHub: <https://github.com/AliceO2Group/simulation/blob/main/docs/transport/mckine.md>)

Sources:

1. <https://github.com/AliceO2Group/simulation/blob/main/docs/transport/mckine.md>
2. https://github.com/AliceO2Group/simulation/blob/main/additional_resources/talks/O2_AnalysisTutorial_Nov2024/MCTutorial4Giacalone.pdf
3. https://github.com/ta5946/alice-rag/blob/master/data/knowledge_base/presentations/ALICE-simulation-tutorial.pptx-1.pdf
4. <https://github.com/AliceO2Group/AliceO2/blob/dev/doc/DetectorSimulation.md>
5. https://github.com/AliceO2Group/simulation/blob/main/additional_resources/talks/O2_AnalysisTutorial_April2023/ALICE-Run3-MC-HowTo_Transcript.md
6. https://github.com/AliceO2Group/simulation/blob/main/additional_resources/talks/O2_AnalysisTutorial_April2023/ALICE-Run3-MC-HowTo.pdf
7. <https://github.com/AliceO2Group/analysis-framework/blob/master/docs/tutorials/workingWithMCdata.md>

This response used gpt-oss-20b model and a medium recall database.

Help us improve the askALICE chatbot by providing your feedback - React to this message with 👍 if the answer was helpful or 👎 if it was not.

Edited

Example #3: How to access MC tracks given a label?

How can I run a script on the ALICE GRID?

askALICE LLM chatbot 11:40 AM

To run a script on the ALICE GRID, follow these steps:

1. Create a shell script (e.g., `my_script.sh`) that defines the simulation workflow:

```
1  #!/usr/bin/env bash
2  # Workflow creation: step 1
3  ${O2DPG_ROOT}/MC/bin/o2dpg_sim_workflow.py -eCM 13600 -col pp -gen pythia8 -proc cdiff -tf 1 -ns 200 -e TGeant4 -interactionRate 500000
4  # Workflow execution: step 2
5  ${O2DPG_ROOT}/MC/bin/o2dpg_workflow_runner.py -f workflow.json -tt aod --cpu-limit 8
```

2. Submit the script to the GRID using the provided `grid_submit.sh` tool:

```
1  ${O2DPG_ROOT}/GRID/utils/grid_submit.sh \
2  --script my_script.sh \
3  --jobname test \
4  --outputspec "*.log@disk=1","*.root@disk=2" \
5  --packagespec "VO_ALICE@O2sim:v20241014-1" \
6  --wait \
7  --fetch-output
```

- `--jobname` : Assigns a name visible on MonALISA.
- `--outputspec` : Specifies output files to save; `@disk=2` means two replicas are stored for security.
- `--packagespec` : Defines the software environment (e.g., `VO_ALICE@O2sim:v20241014-1`).
- `--wait` : Waits for job completion.
- `--fetch-output` : Downloads output files after execution.

🗨️ ✅ Ensure you have a valid GRID token and the `O2sim` environment loaded via `alienv`.

📖 For more details, see: [O2DPG GRID Submission Tool](#) and [ALICE GRID Documentation](#).

Sources:

1. https://github.com/AliceO2Group/simulation/blob/main/additional_resources/talks/O2_AnalysisTutorial_Nov2024/MCTutorial4Giacalone.pdf
2. https://github.com/ta5946/alice-rag/blob/master/data/knowledge_base/presentations/ALICE-simulation-tutorial.pptx-1.pdf
3. https://github.com/ta5946/alice-rag/blob/master/data/knowledge_base/presentations/MC_Tutorials.pdf
4. <https://github.com/AliceO2Group/simulation/blob/main/docs/gridproductions/README.md>
5. <https://github.com/AliceO2Group/simulation/blob/main/docs/o2dpgworkflow/README.md>
6. <https://github.com/AliceO2Group/analysis-framework/blob/master/docs/tutorials/analysisistask.md>
7. https://github.com/AliceO2Group/simulation/blob/main/additional_resources/talks/O2_AnalysisTutorial_April2023/ALICE-Run3-MC-HowTo.pdf

This response used Qwen3-30B-A3B-Instruct-2507 model and a medium recall database.

Help us improve the askALICE chatbot by providing your feedback - React to this message with 👍 if the answer was helpful or 👎 if it was not.

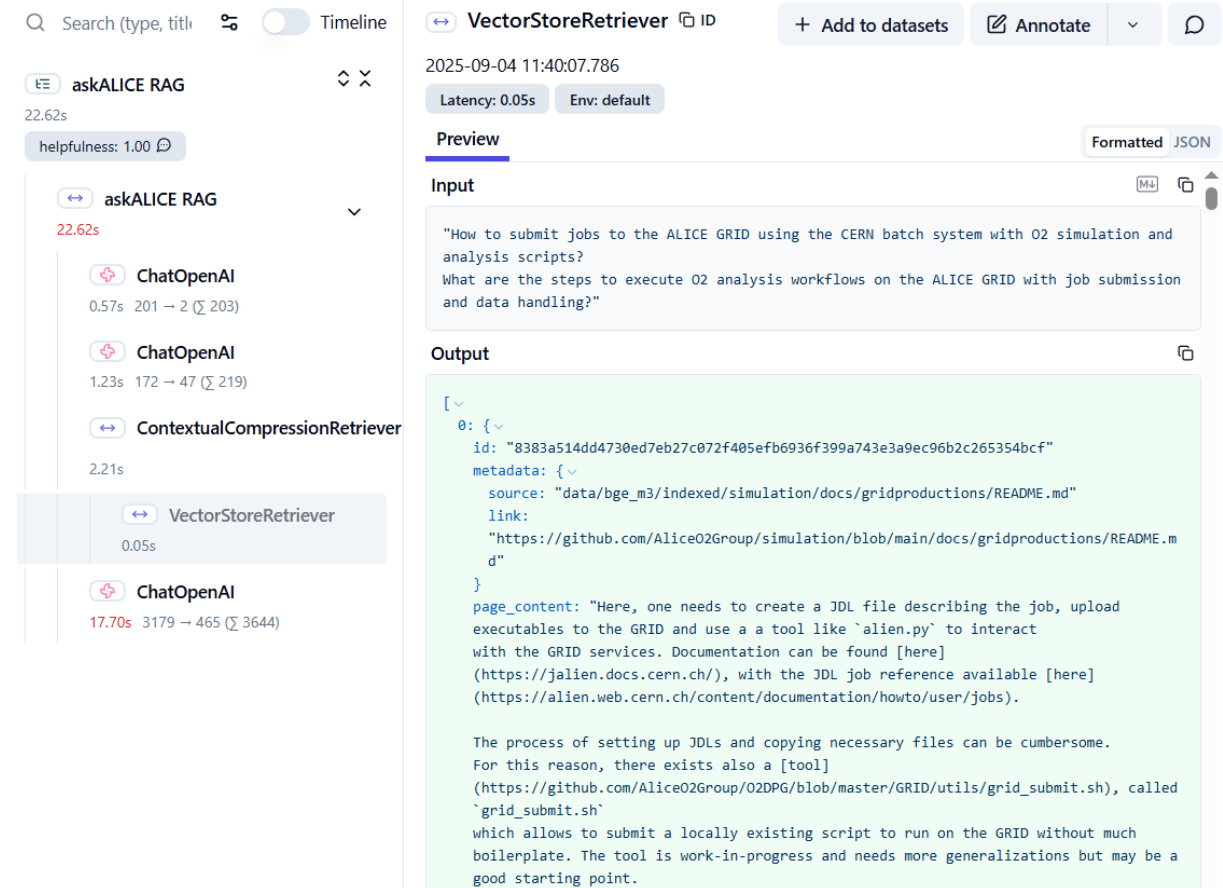
✎ Edited

Example #4: How to run a script on ALICE GRID?

Call tracing



- RAG consists of multiple steps:
 1. Question classification
 2. Search query generation
 3. Initial document retrieval
 4. Document reranking
 5. Answer generation
- Each chatbot call is stored with the used LLM, database configuration, response time (and user feedback).
- UI hosted at <http://pc-alice-ph01:3000/>
- This is valuable for error, performance analysis and creating new datasets.



The screenshot displays the Langfuse interface for a specific chatbot call. On the left, a vertical timeline shows the sequence of steps: 'askALICE RAG' (22.62s), 'askALICE RAG' (22.62s), 'ChatOpenAI' (0.57s), 'ChatOpenAI' (1.23s), 'ContextualCompressionRetriever' (2.21s), 'VectorStoreRetriever' (0.05s), and 'ChatOpenAI' (17.70s). The main panel on the right provides a detailed view of the 'VectorStoreRetriever' step, including its input and output. The input is a query about submitting jobs to the ALICE GRID. The output is a JSON object containing a document ID, metadata (source, link), and page content. The page content describes the process of setting up JDLs and submitting scripts to the GRID, mentioning a tool called 'grid_submit.sh'.

Search (type, title) Timeline

askALICE RAG 22.62s helpfulness: 1.00

askALICE RAG 22.62s

ChatOpenAI 0.57s 201 → 2 (Σ 203)

ChatOpenAI 1.23s 172 → 47 (Σ 219)

ContextualCompressionRetriever 2.21s

VectorStoreRetriever 0.05s

ChatOpenAI 17.70s 3179 → 465 (Σ 3644)

VectorStoreRetriever ID

+ Add to datasets Annotate

2025-09-04 11:40:07.786

Latency: 0.05s Env: default

Preview Formatted JSON

Input

"How to submit jobs to the ALICE GRID using the CERN batch system with O2 simulation and analysis scripts?
What are the steps to execute O2 analysis workflows on the ALICE GRID with job submission and data handling?"

Output

```
[
  {
    id: "8383a514dd4730ed7eb27c072f405efb6936f399a743e3a9ec96b2c265354bcf"
    metadata: {
      source: "data/bge_m3/indexed/simulation/docs/gridproductions/README.md"
      link: "https://github.com/AliceO2Group/simulation/blob/main/docs/gridproductions/README.md"
    }
    page_content: "Here, one needs to create a JDL file describing the job, upload executables to the GRID and use a tool like 'alien.py' to interact with the GRID services. Documentation can be found [here] (https://alien.docs.cern.ch/), with the JDL job reference available [here] (https://alien.web.cern.ch/content/documentation/howto/user/jobs).

    The process of setting up JDLs and copying necessary files can be cumbersome. For this reason, there exists also a [tool] (https://github.com/AliceO2Group/O2DPG/blob/master/GRID/utis/grid_submit.sh), called 'grid_submit.sh' which allows to submit a locally existing script to run on the GRID without much boilerplate. The tool is work-in-progress and needs more generalizations but may be a good starting point."
  }
]
```

Key findings

- **Even a smaller (7B) LLM can handle a simple RAG application** and outperform state-of-the-art API models.
- RAG prompt engineering can reduce hallucinations and latency.
- Scaling up the models showed a moderate performance improvement.
- Multiple data sources can be stored in a single vector database.
- RAG reaches its limit at a certain number of documents to retrieve (top n).
- A capable LLM judge \neq good conversationalist.
- Generating artificial training data is not trivial.

What comes next?

- **Try out the chatbot** and leave a feedback.
- Write **more documentation** and let everyone contribute, even if it is not exact.
- Continuously update the RAG database with GitHub actions.
- Regularly evaluate new open-source models for improvements.
- Can extend the chatbot to related use cases:
 1. Analyze output files
 2. Monitor Mattermost channels (O2 simulation)
 3. Expert support for shifters / on-callers
- Or try to automate and improve the chatbot with larger LLMs and agentic AI.

Questions?

CONTACT: TJAS.AJDOVEC@GMAIL.COM