MultiAgent-Constructed Knowledge Bases for Long-Tail Microbial Question Answering

Sandro Gepiro Contaldo[1,*], Giuseppe Mammolo[1], Sergio Rabellino[1], Giovanna Cristina Varese[2], Marco Aldinucci[1], Rossana Damiano[1] and Marco Beccuti[1]

[1]Department of Computer Science, University of Torino, Turin, Italy.
[2]Department of Life Sciences and Systems Biology, Mycotheca Universitatis Taurinensis, University of Torino, Turin, Italy.

Abstract

Long-tail scientific knowledge is often poorly represented in general-purpose large language models and cannot always be reliably recovered using conventional retrieval-augmented generation. We present a multi-agent framework that organizes heterogeneous scientific metadata into a persistent, document-centric knowledge base of interconnected entities. A construction agent processes the source records during ingestion, while a querying agent navigates the resulting knowledge base to answer factual and reasoning questions. The framework was evaluated on the Turin University Culture Collection, a specialized scientific resource characterized by rare microbial strains, poorly represented species, sparsely populated metadata fields, and evidence distributed across heterogeneous catalogue records, using 77 SQL-like queries and 15 reasoning questions assessed by domain experts. Gemma-4:31B correctly answered 76 of the 77 factual queries, outperforming GPT-OSS:120B and GPT-OSS:20B. Under the same experimental conditions, the conventional RAG baseline did not produce any exact match according to the strict factual benchmark. Expert evaluation showed good performance for questions directly supported by consistent records, while reliability decreased for questions requiring the interpretation of heterogeneous evidence. Overall, the results indicate that organizing entities and relationships during ingestion can improve the completeness, traceability, and reliability of long-tail scientific question answering.

Keywords

Microbial collection, Multiagent system, Knowledge base, LLM wiki

1. Introduction

Large language models (LLMs) are increasingly used not only for text generation, but also as components of more complex systems that can access external information, perform multi-step reasoning, and interact with structured and unstructured data sources [1]. Among these approaches, retrieval-augmented generation (RAG) has become one of the most common strategies for providing LLMs with external, domain-specific knowledge [2, 3].

RAG systems retrieve documents or data fragments that are considered relevant to a query and provide them to the model as additional context. This can improve factual accuracy and make specialized information available to general-purpose LLMs, but the result still depends on what is retrieved. The context has to be selected again for every request, which may increase both the computational cost and the variability of the answers. In addition, relationships spread across several records are not always recovered correctly. If the available evidence is incomplete, ambiguous, or poorly organized, the model may still produce conclusions that seem plausible but not supported by the data [4].

This problem is common in scientific domains such as biology, medicine, and chemistry. In these areas, data are usually accompanied by heterogeneous metadata describing, for example, their origin, experimental conditions, acquisition procedures, classifications, processing steps, and analytical workflows. Answering questions over this data, therefore, requires more than the retrieval of isolated passages and involves entities that models may have encountered only rarely during training. The

The 25th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2026) - Satellite Events (Workshops, Tutorials, Demos, Posters
*Corresponding author.
0009-0007-3889-8644 (S. G. Contaldo); 0000-0002-1757-2000 (S. Rabellino); 0000-0002-1455-6208 (G. C. Varese); 0000-0001-8788-0829 (M. Aldinucci); 0000-0001-9866-2843 (R. Damiano); 0000-0001-6125-9460 (M. Beccuti)
© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-11-26

CEUR-WS.org/Vol-4340/paper19.pdf

relationships among entities, attributes, procedures, and domain concepts must also be preserved.

In this work, we present a multi-agent knowledge-based framework inspired by the principles of LLM-Wiki [5], which is described in detail in the methods section. The framework transforms heterogeneous scientific metadata into a persistent knowledge base that can be navigated and queried by LLM agents. Instead of reconstructing the domain context directly from raw data at every interaction, the proposed approach organizes knowledge during the ingestion phase into interconnected entities and relationships. At query time, the agent selectively accesses the information identified as relevant to each request, thereby reducing the amount of irrelevant context provided to the model and limiting the need to infer implicit relationships from unstructured inputs.

The framework addresses two complementary objectives. First, it aims to improve the reliability, consistency, and traceability of agent responses by grounding them in an explicit representation of domain-specific knowledge. Second, by separating knowledge construction from knowledge access and reducing the amount of context processed for each query, it investigates whether effective interaction can also be achieved using smaller LLMs and more limited computational infrastructures.

We evaluate the proposed approach through a realistic case study based on the microbiological dataset of the Turin University Culture Collection (TUCC) [6, 7]. The dataset includes heterogeneous and interconnected information on microbial strains, their origins, cultivation conditions, physiological characteristics, habitats, and taxonomy. Since this type of specialized knowledge is unlikely to be fully represented in the training data of general-purpose LLMs, TUCC provides a suitable case study for evaluating how different agents access, retrieve, and combine information from the same persistent knowledge base, and for comparing the proposed framework with a conventional RAG pipeline under the same experimental conditions.

2. Methods

The proposed framework is based on two agents with different tasks. The first one reads the source records and builds the knowledge base by identifying the relevant entities and the links among them. The extracted information is stored in a set of structured *pages*. The second agent uses these pages to answer queries. As a result, the source material is processed during ingestion, and the same representation is then used throughout the querying phase.

In this work, we use the term long-tail scientific knowledge to refer to specialized information associated with entities that are sparsely represented in general-purpose language-model training data and whose relevant evidence is distributed across heterogeneous domain-specific records.

Knowledge-base construction The knowledge base was developed following the *LLM-Wiki* approach [5], which structures knowledge as an interconnected collection of textual documents. In this approach, the source information is not stored as a collection of independent text fragments. Instead, the agent creates a set of linked pages, each representing an entity of the domain. Each page contains the available attributes, the links to related entities, and references to the corresponding source records. The construction agent operated according to a fixed set of instructions defining the directory structure, the ingestion workflow, the entity model, the permitted operations, and the rules used to prevent unsupported additions.

The resulting knowledge base follows an open information extraction approach. Entities correspond to identifiable objects, concepts, records, or other elements described in the source material, without being restricted to a predefined domain-specific schema. Entity types and properties are therefore induced from the input records and can expand as new content is ingested. Relations capture evidence-supported links between entities, in a role similar to hyperlinks between Web pages, rather than necessarily encoding predefined semantic relation types. However, this open extraction process remains provenance-constrained: the agent may only add information supported by the source records.

The workspace was organized into three directories. The `raw/` directory contained the original source records and was treated as read-only. The `wiki/` directory contained the pages generated by the agent

and could be modified only during the construction phase. The `output/` directory was used to store query results, reports, validation files, and other derived material. This organization ensured a clear separation between the original evidence, the generated knowledge base, and the outputs produced during querying.

During ingestion, the agent inspected the source records, extracted the available attributes, and associated them with the corresponding entities. Before creating a new page, it searched the existing knowledge base for possible matches in order to reduce duplication. The knowledge base used in the experiments was generated during a single initial ingestion phase and was kept unchanged during validation. However, the same procedure also supports incremental updates, since new source records can be added without rebuilding the complete knowledge base. The agent was not allowed to introduce information that was not supported by the sources. Properties were added only when they were explicitly reported in the original records or in knowledge-base content with documented provenance. Missing values were left empty or marked as unavailable. The absence of a value was not interpreted as evidence that the corresponding property was absent.

Knowledge-base querying The second agent worked on the pages stored in `wiki/` and did not access the original records. Given a query, it searched for the entities involved, followed the available links, and collected the information needed for the answer.

Some queries concern a single entity and can be answered from one page. Others require the agent to follow links or combine information found in several pages. Comparative and analytical questions may also require values to be collected from different parts of the knowledge base.

The answer was generated from the information retrieved during this navigation. If the requested information was distributed across multiple pages, the agent combined their contents while preserving the relationships defined during ingestion. Observe that the agent was instructed not to add missing information based on unsupported assumptions.

The output could take the form of a textual answer, a list, a summary table, or a report. These results were written to `output/` and did not alter the knowledge base.

Figure 1 shows the proposed approach (top) in comparison with a conventional RAG pipeline (bottom). Briefly, in the RAG approach, relevant text fragments are retrieved at query time, and the relationships among records must be reconstructed from them. In our framework, these relationships are instead identified and explicitly represented during knowledge-base construction.

3. Results

The framework was evaluated using data from the Turin University Culture Collection (TUCC), a microbial culture collection maintained by the University of Turin that comprises 7,227 prokaryotic and eukaryotic strains, including filamentous fungi and yeasts. TUCC was selected because its information system contains heterogeneous and strongly interconnected records, making it a suitable case study for evaluating both knowledge-base construction and subsequent querying.

The dataset included information on the microbial strains preserved in the collection, the people involved in their isolation, collection, or deposition, the associated culture media and growth conditions, their habitats and environmental contexts, and the corresponding taxonomic and collection metadata.

All computational activities, including knowledge-base construction, agent-based querying, execution of the conventional RAG baseline, and validation, were carried out using the computational resources of the *HPC4AI* [8] datacenter at the Department of Computer Science, University of Turin. More specifically, we used two *AArch64* compute nodes of the *HPC4AI* infrastructure. The first was equipped with an 80-core Ampere Altra Q80-30 processor at 3.0 GHz, 512 GB of RAM, two NVIDIA A100 GPUs, two NVIDIA BlueField-2 DPUs, and a 100 Gbps Ethernet connection. The second was equipped with a 72-core NVIDIA Grace A02 processor at 3.4 GHz, 480 GB of RAM, one NVIDIA H100 GPU, and a 100 Gbps Ethernet connection.

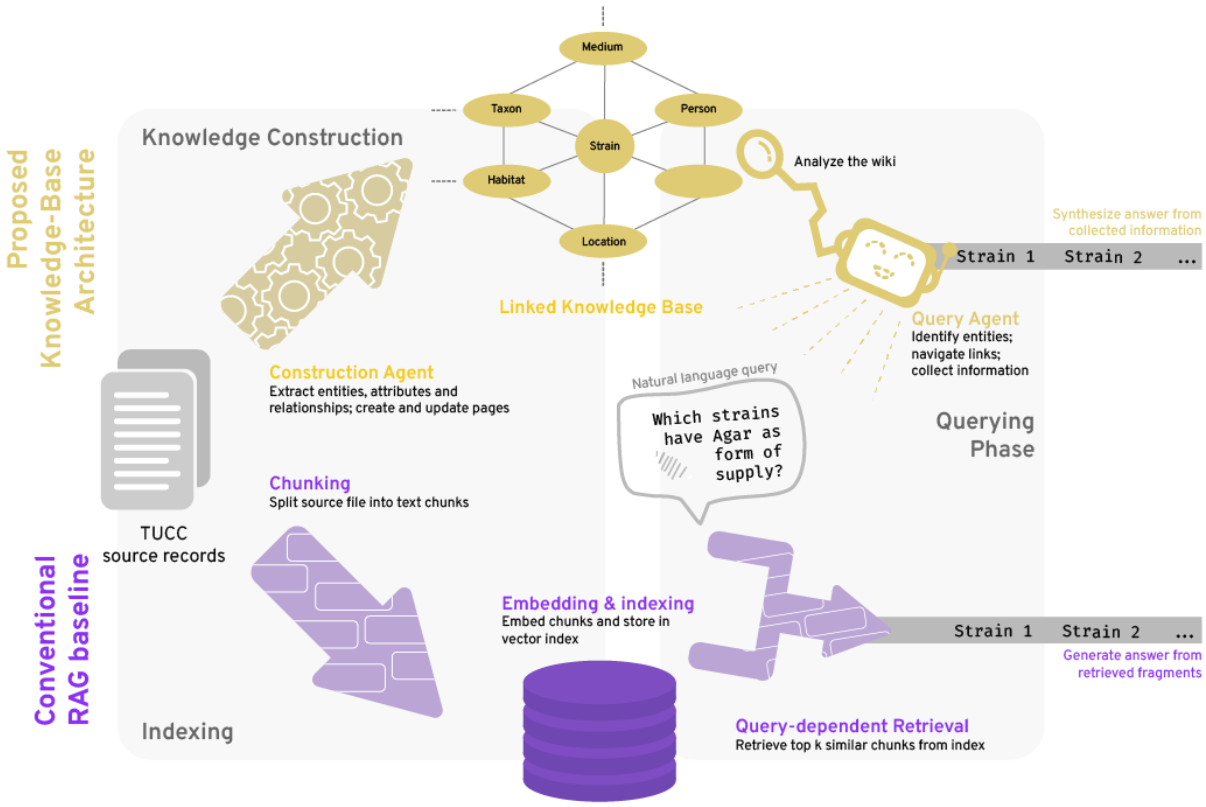


Figure 1: Comparison of our framework (top) with a conventional RAG system (bottom).

To support reproducibility, the raw data, the agent instructions, the validation scripts, and the experimental results are available at https://github.com/qBioTurin/MIRRIKnowledgeBase.

Knowledge-base generation The TUCC knowledge base was generated using *Gemma-4:31B* as the primary construction model. Since the source records were provided in structured JSON format, the main task of the construction agent was to parse and reorganize the available information according to the instructions defined in the `AGENTS.md` file.

The construction process transformed the source records into a persistent domain representation composed of interconnected entities rather than independent text fragments. The resulting knowledge base was kept unchanged throughout validation and constituted the common information layer used by all models during the querying phase. Consequently, the differences observed among the evaluated models mainly reflect their ability to identify relevant entities, navigate their relationships, and synthesize the available information, rather than differences in the underlying data representation.

In the TUCC knowledge base, the entities generated during construction represented the main concepts and metadata elements occurring in the source data. The resulting entity categories included availability information, datasets, denominations, fields, growth media, habitats, markers, methods, microorganisms, OntoBiotope terms, organism types, people, regulatory information, risk groups, sources, and taxonomy. Each entity was represented by a dedicated wiki page and could be connected to other entities through evidence-supported links derived from the original records. These links did not necessarily correspond to a predefined set of semantic relation types, but primarily expressed associations between related entities in the knowledge base.

Multi-model deployment The evaluated LLMs were made available through *UTOPIA*, a local *Open WebUI* [9] instance deployed on the *HPC4AI* infrastructure. *OpenCode* [10] was used as the agent execution environment and as the interface between the domain-specific agents and the knowledge-base workspace. This configuration provided a controlled environment in which all models operated under the same domain-specific instructions and epistemic constraints. All three models were first evaluated using the proposed knowledge-based framework. The best-performing model was then used to evaluate the conventional RAG baseline on the same benchmark.

Validation benchmark The benchmark was designed to cover different forms of interaction with the knowledge base, ranging from direct factual retrieval to questions requiring the integration of information distributed across multiple related entities. It comprised 77 SQL-like queries and 15 reasoning queries.

The SQL-like queries represented structured searches that could, in principle, be executed directly against the underlying TUCC relational database. They were derived from *frequently performed searches within the Italian National Catalogue of Microbial Resources*, developed in the context of the MIRRI-IT research infrastructure. For each query, the complete expected result set could therefore be obtained directly from the authoritative TUCC source data, providing an exact reference for evaluation. The queries included searches based on taxonomic classification, isolation source, geographic origin, culture medium, growth conditions, and combinations of attributes distributed across related records. In contrast, the reasoning queries were formulated as more open-ended natural-language questions for which the answer could not be obtained through a single structured lookup. Answering these questions required the model to identify relevant entities, combine information distributed across multiple records, and interpret the available evidence to formulate a response.

> **Query:** Give me all strains from the genus *Diplodia*.
>
> **Expected answer:** A list of seven strains (i.e. *TUCC00000386*, *TUCC00000387*, *TUCC00000733*, *TUCC00000734*, *TUCC00000735*, *TUCC00000865*, *TUCC00000867*)

The SQL-like answers were evaluated using a strict content-based comparison. An answer was considered correct only when it contained all and only the microbial strains included in the reference result. It was therefore classified as incorrect if it omitted at least one expected strain or included at least one additional strain not present in the reference set.

The reasoning queries were proposed by *TUCC domain experts and reflected scientific or operational questions that may be submitted by users of the microbial collection.* Unlike the SQL-like queries, these questions required the synthesis and interpretation of information distributed across different entities and could not be evaluated through exact set matching. The reasoning queries included, for example, questions about suitable cultivation conditions, comparisons among microorganisms with related characteristics, and the identification of strains satisfying combinations of biological and operational requirements.

> **Query:** What growth conditions would you recommend for working with strains of Scedosporium dehoogii?
>
> **Reasoning information:** The reasoning was based on an examination of the nine Scedosporium dehoogii strains listed in the catalogue, all of which were reported to grow on malt extract agar (MEA) at 25 °C.

The answers to the reasoning queries were independently assessed by domain experts using a satisfaction score ranging from 0 to 5. Higher scores indicated answers that were more complete, accurate, supported by the collection data, and useful from a scientific or operational perspective.

For every query, the relevant information was independently retrieved through the proposed agent-based system. The same benchmark was subsequently submitted to the conventional RAG baseline using the best-performing model, allowing the two knowledge-access strategies to be compared while keeping the language model, source information, and evaluation criteria unchanged.

Table 1
Validation results obtained by the evaluated language models on the benchmark of 77 SQL-like queries and 15 reasoning queries.

Model	Correct	Incorrect	Accuracy (%)	Time (s)
SQL-like queries				
GPT-OSS:120B	67	10	87.0	141
Gemma-4:31B	76	1	98.7	220
GPT-OSS:20B	51	26	66.2	102
	Mean	Standard Deviation	Median	Time (s)
Reasoning queries				
GPT-OSS:120B	2.85	1.52	3.60	54
Gemma-4:31B	3.47	1.28	4.60	201
GPT-OSS:20B	2.00	1.66	2.60	17

Quantitative results across language models Table 1 reports the results obtained by the evaluated models using the proposed framework.

Gemma-4:31B achieved the highest accuracy, correctly answering 76 of the 77 SQL-like queries, corresponding to an accuracy of 98.7%. GPT-OSS:120B correctly answered 67 queries, achieving an accuracy of 87.0%, whereas GPT-OSS:20B correctly answered 51 queries, with an accuracy of 66.2%. These results show that model size alone did not determine performance. Despite having fewer parameters than GPT-OSS:120B, Gemma-4:31B was more effective in locating relevant entities, navigating their relationships, and reporting the information represented in the knowledge base. This suggests that, in this setting, the ability to follow the agent instructions and exploit the structure of the knowledge base may be more important than parameter count.

The execution times suggest a trade-off between accuracy and computational cost. GPT-OSS:20B was the fastest model, with an average execution time of 102 seconds per query, but also achieved the lowest accuracy. Gemma-4:31B produced the most accurate results, with an average execution time of 220 seconds per query, compared with 141 seconds per query for GPT-OSS:120B.

A preliminary inspection of the incorrect answers suggests that the most frequent errors involved incomplete result sets or the inclusion of records that did not satisfy all the constraints expressed in the query. In some cases, the models also appeared to interpret the relationships among entities incorrectly.

These errors may arise at different stages of the querying process, including entity identification, navigation across linked pages, application of query constraints, and generation of the final answer. A more detailed analysis will be required to determine which of these stages has the greatest impact on performance and to identify possible improvements to the querying procedure.

Comparison with the conventional RAG baseline After comparing the language models within the proposed framework, the same benchmark was evaluated using a conventional RAG baseline. The RAG system was implemented using the knowledge-base functionality provided by *Open WebUI* and was populated with the original source files used to construct the persistent knowledge base. Both approaches therefore relied on the same underlying TUCC information.

According to the strict content-based evaluation, the conventional RAG system did not correctly answer any of the 77 SQL-like queries. In most cases, it returned only a subset of the strains contained in the expected result set. This limitation was particularly evident for queries requiring the integration of information distributed across multiple records.

The observed difference reflects the way relationships are handled by the two approaches. In the RAG baseline, relevant text fragments are selected independently for each query, and the relationships among records must be reconstructed from the retrieved context. In the proposed framework, these relationships are identified during ingestion and are directly available to the querying agent.

Query: Give me all strains from the genus *Coniochaeta*.

Expected answer: A list of 9 strains (i.e. *MUT00006665*, *MUT00007135*, *MUT00007136*, *MUT00007138*, *MUT00007139*, *MUT00007163*, *MUT00007165*, *MUT00007167*, *MUT00007396*)

RAG answer: A list of 86 strains

A further difference concerned queries for which the requested information was not available in the source records. In these cases, the conventional RAG baseline could still generate a plausible answer despite the absence of supporting evidence. By contrast, the proposed framework returned no result or explicitly indicated that the requested information was unavailable.

Query: Which strains have Deposit date different from Inclusion date?

RAG answer: A list of more than 50 strains

Proposed framework answer: An empty list (there are no strains with different deposit and inclusion date in the collection)

These results indicate that explicitly organizing domain relationships during ingestion improves the completeness of answers requiring information from multiple records. Moreover, the distinction between supported and missing information encoded in the knowledge base appears to reduce the generation of unsupported responses.

Reasoning-query results The expert validation was conducted by the five experts from the microbial culture collection, and the overall results are reported in Table 1. These two examples were selected from the validation of *Gemma-4:31B*, our best model, to illustrate one of the highest-scoring and one of the lowest-scoring responses. The response concerning the growth conditions of *Scedosporium dehoogii* received a mean score of 4.50 out of 5, indicating that the recommendation was considered highly accurate, relevant, and adequately supported by the available information. The answer correctly identified Malt Extract Agar at 25°C as the recommended cultivation condition and presented the information clearly and concisely.

Query: What growth conditions would you recommend for working with strains of *Scedosporium dehoogii*?

Generated answer: For strains of *Scedosporium dehoogii*, the recommended growth conditions are Malt Extract Agar (MEA) at 25°C

Mean expert score: 4.50 for *Gemma-4:31B*

By contrast, the answer recommending MUT00006172 (*Curvularia* sp.) for the bioremediation of contaminated industrial soil received a substantially lower mean score of 1.75. Although the response identified a potentially relevant strain and reported several characteristics associated with hydrocarbon degradation, the low score suggests that the evidence provided was not considered sufficient to support the recommendation confidently. In particular, the answer required the integration and interpretation of information from multiple related records, increasing the risk of overgeneralization, inaccurate associations, or unsupported conclusions.

Query: If a team wants to test bioremediation of contaminated industrial soil, which strain would you recommend starting with and what evidence makes it suitable?

Generated answer: I recommend starting with MUT00006172 (Curvularia sp.), as it was isolated from industrial soil contaminated by PHAs, BTEX, and alkanes. It is suitable because it uses crude oil, alkanes (paraffin oil and heptadecane), and PHAs (pyrene) as sole carbon sources and produces biosurfactants.

Mean expert score: 1.75 for *Gemma-4:31B*

These examples indicate that the model performs well when the requested information can be retrieved directly from consistent records, but its reliability decreases for questions requiring the comparison of candidates and the interpretation of heterogeneous evidence.

4. Limitations

The present study has several limitations. First, the evaluation was conducted on a single case study and on English-language metadata, limiting the generalizability of the results to other scientific domains and multilingual settings. Second, although the experiments included three open-weight LLMs, a broader comparison involving additional architectures and proprietary models would provide a more comprehensive assessment of the framework. Finally, the proposed architecture represents an intermediate paradigm between conventional RAG and Graph-RAG: while it explicitly organizes entities and relationships during ingestion, it relies on an LLM-maintained document-centric knowledge base rather than on a formal graph representation. Further work will therefore be needed to compare this approach directly with state-of-the-art Graph-RAG systems.

5. Conclusions and Discussion

This paper presented a multi-agent framework for constructing and querying persistent domain-specific knowledge bases inspired by the LLM-Wiki paradigm. By explicitly organizing entities and their relationships during the ingestion phase, the proposed approach separates knowledge construction from access and provides a structured alternative to conventional retrieval-based pipelines. The experimental evaluation on the TUCC microbial collection showed that this organization substantially improves the completeness and reliability of complex information retrieval compared with a conventional RAG baseline while enabling effective querying by multiple LLMs.

Future work will extend the evaluation to additional case studies to assess the generalizability of the framework across scientific domains. A broader experimental validation involving a larger and more diverse set of language models, including both open- and closed-weight systems, will also be conducted. We will compare the framework with Graph-RAG approaches [11] and explore scalable hybrid solutions combining LLM-maintained entity pages with graph-processing libraries such as graph-tool. We will also investigate whether explicit provenance and evidence-aware answers can help domain users critically assess the reliability and limitations of generated responses. Finally, a systematic error analysis will be performed to better characterize failures in complex reasoning queries and determine which stages of the pipeline—entity identification, relationship navigation, or answer synthesis—have the greatest impact on performance.

Acknowledgments

We would like to thank Ilario Ferrocino, Francesco Venice, Narcisa Mandras, and Federica Spina for their valuable contribution to the evaluation of the models. We also thank Simone Ramello for his support in preparing the figure. We acknowledge the contribution and support from the Italian national Node (MIRRI-IT) of the European Research Infrastructure MIRRI-ERIC.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT (OpenAI) for grammar and spelling checks.

References

[1] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, J. Gao, Large language models: A survey, 2025. URL: https://arxiv.org/abs/2402.06196. arXiv:2402.06196.

[2] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive nlp

tasks, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20, Curran Associates Inc., Red Hook, NY, USA, 2020.

[3] P. Zhao, H. Zhang, Q. Yu, Z. Wang, Y. Geng, F. Fu, L. Yang, W. Zhang, J. Jiang, B. Cui, Retrieval-augmented generation for ai-generated content: A survey, Data Science and Engineering 11 (2026) 1–29. URL: https://doi.org/10.1007/s41019-025-00335-5. doi:10.1007/s41019-025-00335-5.

[4] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, T. Liu, A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions, ACM Trans. Inf. Syst. 43 (2025). URL: https://doi.org/10.1145/3703155. doi:10.1145/3703155.

[5] LLM Wiki contributors, Llm wiki: A pattern for building personal knowledge bases using large language models, 2026. Unpublished methodological document.

[6] L. Cocolin, P. Gonthier, A. e. a. Laera, The turin university culture collection (TUCC): A successful networking initiative of microbial resources, in: ECCO 2019 Conference Proceedings, ECCO, Turin, Italy, 2019, pp. 64–65. XXXVIII Annual Meeting of the European Culture Collections' Organisation, 12–14 June 2019.

[7] Turin University Culture Collection, TUCC: Turin university culture collection, 2026. URL: https://tucc.unito.it.

[8] M. Aldinucci, S. Rabellino, P. et al., Hpc4ai: an ai-on-demand federated platform endeavour, in: Proceedings of the 15th ACM Int. Conf. on Computing Frontiers, CF '18, ACM, New York, NY, USA, 2018, p. 279–286.

[9] Open WebUI, Open webui, 2026. URL: https://github.com/open-webui/open-webui, self-hosted AI platform and source code repository.

[10] Anomaly, Opencode: The open source ai coding agent, 2026. URL: https://github.com/anomalyco/opencode, software repository; year refers to the access date.

[11] X. Zhu, Y. Xie, Y. Liu, Y. Li, W. Hu, Knowledge graph-guided retrieval augmented generation, in: NAACL, 2025.