ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

A Multi-Agent LLM System for Natural Language Querying of Operational Knowledge Graphs in Satellite Ground Stations

Fosco Eugenio Quadri, Filippo Bianchini *

Telespazio SpA - AI Competence Center,
Via Tiburtina 965, Rome (RO), Italy

Abstract. Satellite ground-station maintenance generates vast operational data, yet traditional query interfaces limit discoverability and slow time-critical decision making. We present a multi-agent system deployed at Fucino Space Centre that combines Large Language Models with knowledge graphs and Retrieval-Augmented Generation (RAG) to support operators in troubleshooting by exploiting 40,000 historical maintenance tickets. Specialized agents collaborate on intent mapping, multi-hop reasoning, and explainable synthesis. This work brings the following contributions: (1) an explainable architecture for conversational retrieval, (2) a domain knowledge graph operationalizing antenna-system context, and (3) integration lessons for operator-in-the-loop. Our novel approach demonstrates how agentic AI enhances transparency and operational reliability in aerospace ground operations.

1 Introduction

Ground-station maintenance teams at the Fucino Space Centre manage 40,000 historical incident and OM tickets that encode operational knowledge about antenna subsystems, failure modes, and remedies. We design and deploy a multi-agent assistant that enables operators to pose natural-language questions (e.g., "What caused antenna misalignment last month?"). The system retrieves evidence-grounded, explainable answers from a knowledge graph built on this ticket corpus. Our contributions include: (1) a modular multi-agent architecture with specialized agents for classification, query generation, refinement, and explanation; (2) RAG-enhanced text-to-nGQL translation using retrieved query examples as few-shot demonstrations for accurate nGQL generation; (3) conversational query refinement mechanisms to manage multi-turn dialogues by merging related consecutive queries; and (4) explainable AI integration providing understandable interpretations of technical query results. To the best of our knowledge, this is the first system enabling natural language querying over operational knowledge graphs in satellite ground station environments.

*This work was supported by Telespazio SpA.

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

2 Related Work

Recent approaches to NL-to-graph query translation include fine-tuned models [1], GPT-4 prompting [2], and application frameworks [3]. We adopt a RAG-based approach, retrieving similar NL-nGQL pairs as few-shot demonstrations rather than fine-tuning. Following established RAG practices [4] we use cosine similarity over a curated 133-pair library to select exemplars, blending in-context few-shot learning [5] with selective retrieval to improve faithfulness to schema constraints. Beyond single-model prompting, multi-agent LLM systems decompose tasks and can outperform monolithic setups [6, 7]; accordingly, we adopt a pipeline of classification, translation, refinement, and explanation to enhance modularity and failure containment. Knowledge graphs are gaining traction in aerospace from NASA case studies on critical-data discovery to GSOC knowledge management practices [8, 9]; we extend this line to operational ticket graphs with natural language access, addressing gaps highlighted in aerospace support-system KM literature [10]. Maintaining conversational intent across turns remains a challenge in conversational search [11]; query rewriting and contextualization offer practical remedies [12], which we implement by merging consecutive queries. Operational deployment demands explainability: XAI chatbots improve user trust and decision support [13]. Our XAI module synthesizes KG evidence into cited, concise report to support operator decision-making.

3 System Architecture

3.1 Knowledge Graph Construction

We construct a domain-specific knowledge graph using NebulaGraph. The graph schema consists of three node types: *Antenna* (properties: name, type, location, country), *Ticket* (ID, antenna, type, description, circuit_id, technical_closure, notice_date, end_date, duration_min, effect, root_cause, activity, status), and *Note* (ID, text, timestamp, author, work duration). Two relationship types connect entities: *documents* (Note→Ticket) and *related_to* (Antenna→Ticket). Data preprocessing involved filtering relevant tickets, normalizing temporal fields, cleaning text descriptions, and validating referential integrity.

3.2 Multi-Agent Pipeline

Our system implements a pipeline-based multi-agent architecture where LLM-powered agents handle distinct subtasks. Agents use Qwen3 8B as language model, running locally via Ollama with flash attention optimization for reduced inference latency. The Multi-agent Pipeline works as follows(see Figure 1): firstly, the Router Agent classifies operator queries into two categories: temporal and non-temporal. Temporal queries explicitly reference time periods (e.g., "Show me tickets from the last three months"), while non-temporal queries focus on content without time constraints. Classification uses few-shot prompting with example queries. For temporal queries, the router extracts the temporal expres-

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

sion and forwards it to the Temporal Query Processor. For non-temporal queries, the system proceeds directly to the Text-to-nGQL Agent. The Temporal Query Processor performs two operations: (1) extraction of the temporal expression from the natural language input, and (2) normalization to ISO 8601 standard format. For example, "Last 3 months" (requested on November 7, 2025) is normalized to 2025-08-07 to 2025-11-07 and passed to the Text-to-nGQL agent. This agent constitutes the core translation component, converting natural language queries to executable nGQL statements. The translation process leverages RAG to select examples from a curated dataset of 133 natural language-nGQL query pairs. The RAG pipeline operates as follows: (1) The natural language question is embedded using nomic-embed-text. (2) Example Retrieval: Cosine similarity is computed between the query embedding and pre-embedded examples in the dataset. The top 5 most similar examples are retrieved. (3) Few-Shot Prompting: Retrieved examples are formatted as few-shot demonstrations and included in the LLM prompt alongside the graph schema definition and the target query. Finally, the LLM generates an nGQL query following the syntax patterns observed in the examples and adhering to the schema constraints. To support natural multi-turn conversations, the Query Refinement Node analyzes consecutive queries to determine if they are contextually related. Using the LLM with conversational history as context, the node identifies cases where follow-up questions reference previous queries (e.g., "What about Antenna Y?" following "Show me Antenna X problems"). When queries are deemed related, the system merges them into a single comprehensive query. After nGQL execution returns structured data (tickets, notes, antenna relationships), the XAI module performs result synthesis, aggregating query results into summaries and identifying key patterns such as frequent failure modes. It uses LLM to generate explanations in natural language that contextualize the findings for operational decision-making. This approach bridges the gap between technical graph query outputs and actionable operational insights, which is important for operators without deep database expertise.

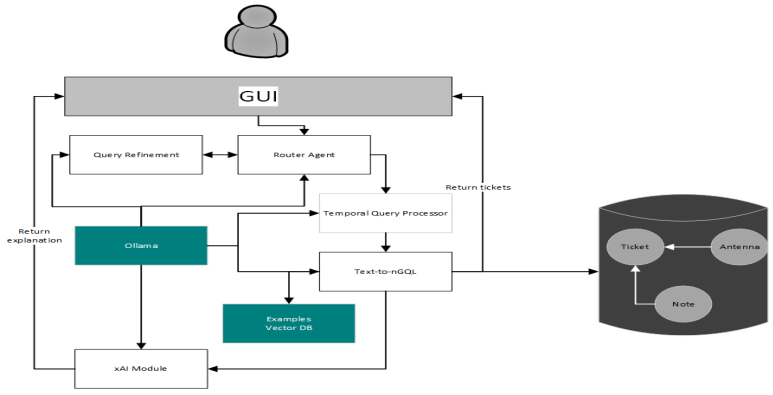

Fig. 1: Overview of the agentic pipeline

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

4 Evaluation

We evaluate nGQL generation quality and explanation quality separately, as standard text-to-text metrics (BLEU, ROUGE) are unsuitable for our structured query generation and synthesis pipeline. We used Claude 4.5 Sonnet to synthesize 100 test triples from non-sensitive data: (1) a natural language question, (2) a corresponding ground-truth nGQL query, and (3) a ground-truth explanation.

4.1 nGQL Generation Quality

We evaluated the Text-to-nGQL agent on two metrics, given the rigorous and domain-specific nature of nGQL on which LLMs are not extensively pre-trained:

- **Execution Accuracy (EA):** A binary metric (pass/fail) indicating whether the generated nGQL query executed successfully against the knowledge graph schema without syntax errors.
- **Semantic Similarity (SS):** We measured the cosine similarity between the `nomic-embed-txt` embeddings of the generated nGQL and the ground-truth nGQL. This assesses semantic correctness, capturing cases where a query is logically equivalent but syntactically different, a common practice for evaluating semantic textual similarity.

4.2 Explanation Quality

We employed an "LLM-as-a-Judge" methodology [14] to score the quality of the explanation. Due to the sensitive nature of the operational data hosted in our KG, we used a locally-hosted 70B-parameter model (DeepSeek-R1-70B) as the judge, preventing data exposure to cloud APIs. The judge was prompted to provide a JSON output scoring the model's explanation on four criteria: Relevance (faithfulness to intent), Completeness (coverage of the query), Faithfulness (absence of hallucination) and Coherence (logical flow).

4.3 Results

The agentic pipeline demonstrated strong performance on our test set. The main results are summarized in Table 1.
The multi-agent pipeline with RAG achieved Execution Accuracy of 100% and average Semantic Similarity of 0.8579 to ground-truth queries. However, ablation studies (Table 2) reveal that EA alone is insufficient for assessing practical utility. The single-agent baseline without RAG achieved 88% EA but only 51% of queries returned non-zero results, indicating syntactically valid but semantically incorrect queries. These failures typically involved queries with wrong field values, non-existent entity IDs, or overly restrictive filters that matched the schema but did not retrieve meaningful data. Adding RAG to the single-agent improved non-zero results to 86%, demonstrating that few-shot examples help

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

Table 1: System Evaluation Results

Metric	**Score**
nGQL Generation Quality	
Execution Accuracy (EA)	**100%**
Semantic Similarity (SS)	**0.8579**
Explanation Quality (LLM-as-a-Judge)	
Relevance	[3.75/5.0]
Completeness	[2.92/5.0]
Faithfulness	[3.73/5.0]
Coherence	[3.27/5.0]

ground the model in valid entity references and schema patterns. The multi-agent architecture without RAG achieved 84% EA and 75% non-zero results, showing that while the refinement step helps catch some errors, it cannot fully compensate for the lack of concrete query exemplars. The multi-agent+RAG configuration achieved the best performance (100% EA, 98% non-zero results), suggesting that RAG exemplars and multi-agent structure are complementary: RAG provides syntactic and semantic grounding, while the multi-agent structure catches errors through specialized task decomposition and iterative refinement.

Table 2: Ablation Study: Impact on Result Retrieval

Configuration	**EA**	**Non-Zero Results**
Single-Agent No-RAG	88%	51%
Single-Agent + RAG	96%	86%
Multi-Agent No-RAG	84%	75%
Multi-Agent + RAG	**100%**	**98%**

5 Discussion

The Execution Accuracy (100%) underscores the effectiveness of our RAG-based Text-to-nGQL approach. Using a curated 133-pair library as few-shot examples proved more reliable than zero-shot prompting for a syntax like nGQL. The primary challenge was ensuring data security. The "LLM-as-a-Judge" approach addresses this but may lack ground-truth information due to LLM stochasticity. The XAI module's explanations, evaluated by the local LLM-as-a-Judge, achieved an average score of 3.41/5.0, indicating good overall quality in generating operator-oriented interpretations. The moderate Completeness scores are partially attributable to the judge model's strictness and to differences in information retrieval between language models.

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

6 Conclusions and Future Work

This work presents the first multi-agent LLM system for natural language querying of operational knowledge graphs in satellite ground station environments. By combining knowledge graph technology with modern LLMs, RAG-enhanced query translation, and explainable AI, we significantly reduce the barrier to accessing critical operational data at the Fucino Space Centre. Our contributions include a modular multi-agent architecture handling diverse query types, a RAG-based text-to-nGQL approach that achieves high translation accuracy, and XAI-driven result interpretation that builds operator trust. This system represents the first step in a broader predictive maintenance project, which will focus on integrating predictive algorithms for the Fucino antennas. Future work on the conversational agent itself will focus on two primary directions. First, we plan to explore the fine-tuning of a specialized model on nGQL to move beyond RAG and optimize translation accuracy. Second, we will implement automated ticket creation functionality, designed to be triggered directly by the outputs of the predictive algorithms. Alongside these new objectives, we will continue to implement evaluation metrics, explore advanced prompting, and investigate active learning mechanisms to continuously improve the RAG dataset.

References

[1] Y. Chen et al., NAT-NL2GQL: A Novel Multi-Agent Framework for Translating Natural Language to Graph Query Language, *arXiv preprint arXiv:2412.10434*, 2024.

[2] T. Bratanic, Generating Cypher Queries With ChatGPT 4 on Any Graph Schema, *Neo4j Developer Blog*, May 2023.

[3] LangChain Documentation, Natural Language Querying of GraphDB in LangChain, *https://python.langchain.com*, 2023.

[4] Pinecone, Retrieval-Augmented Generation (RAG), *https://www.pinecone.io/learn*, 2024.

[5] T. B. Brown et al., Language Models are Few-Shot Learners, *Neural Information Processing Systems (NeurIPS)*, volume 35, pages 24824-24837, 2020.

[6] T. Guo et al., Large Language Model based Multi-Agents: A Survey of Progress and Challenges, *arXiv preprint arXiv:2402.01680*, 2024.

[7] C. Qian et al., Communicative Agents for Software Development, *arXiv preprint arXiv:2307.07924*, 2023.

[8] NASA, How NASA Finds Critical Data through a Knowledge Graph, *Neo4j Blog*, 2019.

[9] SpaceOps 2025, A Quality Approach to Knowledge Management - the experience of Telespazio Germany in GSOC, *Paper ID 86*, 2025.

[10] K. Li et al., Research on knowledge management of operational support system for aerospace manufacturers, *Procedia CIRP*, volume 83, pages 161-165, 2019.

[11] Z. Abbasiantaeb et al., Query Understanding in LLM-based Conversational Information Seeking, *ACM Web Conference Tutorial*, 2025.

[12] J. Wang et al., Contextualizing Search Queries: In-Context Learning for Conversational Rewriting with LLMs, *arXiv preprint arXiv:2502.15009*, 2025.

[13] A. Johnson et al., Explainable AI chatbots towards XAI ChatGPT: A review, *Heliyon*, volume 11, 2025.

[14] L. Zheng, C. Chiang, Y. Sheng, S. Lin, S. Zhuang, Z. Zhang, Y. Dong, Y. Li, Z. Liu, M. Stoica, Judging LLM-as-a-judge with MT-Bench and Chatbot Arena, *arXiv preprint arXiv:2306.05685*, 2023.