RAG on Multimodal Databases: Orchestrating Textual, Vector, and Graph-based Retrieval

Otávio Calaça Xavier[1], **Anderson da Silva Soares**[1]

[1]Instituto de Informática – Universidade Federal de Goiás (UFG)
Alameda Palmeiras, Quadra D, Câmpus Samambaia – 74690-900 – Goiânia – GO – Brazil

{otaviocx,andersonsoares}@ufg.br

Abstract. *This tutorial explores contemporary Information Retrieval (IR) techniques for building RAG systems from a multimodal database perspective. We cover the implementation of textual retrieval (e.g., Full-Text Search), the rise of vector search with native extensions (e.g., pg_vector, ChromaDB), and the use of Knowledge Graphs with Cypher/GQL. Focusing on the challenge of hybrid search, the course presents evaluation metrics (e.g., Recall@K, MRR, NDCG@K) and relevance fusion techniques such as Reciprocal Rank Fusion (RRF). Finally, we demonstrate the construction of an end-to-end RAG pipeline that orchestrates these multiple data sources to augment an LLM. Participants will learn how to design and implement hybrid retrieval systems to enrich text generation with relevant, structured, and verifiable data.*

1. Introduction and Justification

The Retrieval-Augmented Generation (RAG) architecture [Lewis et al. 2020, Gao et al. 2023] represents a fundamental synergy between Large Language Models (LLMs) and data management systems. By grounding text generation in evidence retrieved from external sources, RAG enhances the accuracy and verifiability of responses. However, the effectiveness of a RAG system is fundamentally determined by the performance of its retrieval module. Consequently, optimizing Information Retrieval (IR) has become a driving force for innovation in database technologies.

For the SBBD audience, this transformation is particularly relevant. We are witnessing the evolution of traditional DBMSs into **multimodal databases**, designed to natively and efficiently manage and query lexical, vector, and graph data. Classic lexical techniques, such as BM25 [Robertson et al. 1995], once the domain of search engines, are now optimized within relational DBMSs through *Full-Text* indexes like PostgreSQL's GIN/GiST.

The rise of *embeddings* [Mikolov et al. 2013, Reimers and Gurevych 2019] has driven the search for vector similarity, creating a demand for new indexing and query mechanisms. In response, the database community has developed extensions like `pg_vector` and integrated approximate nearest neighbor (ANN) search indexes, such as HNSW [Malkov and Yashunin 2020], directly into the core of DBMSs. Meanwhile, Knowledge Graphs, managed by graph-native databases like Neo4j, have become crucial for modeling complex relationships, with the standardization of the GQL query language [iso 2024] signaling its maturity and importance to the industry.

The exploration of advanced RAG implementations, such as RAG on Knowledge Graphs—the topic of our tutorial at SBBD 2024

[Xavier and da Silva Soares 2024]—demonstrates the immense potential of structured retrieval for complex queries. However, the discussions and feedback from that event made it clear that, for most practical applications, the central challenge is hybrid in nature: how can a system query and fuse evidence from a textual index, a vector similarity search, and a structured query on a knowledge graph? The answer to this question, which motivated the elaboration of this tutorial, lies in the intelligent orchestration of different database paradigms.

This tutorial addresses this convergence precisely. Structured as a 2-hour session, it will enable participants to: (i) Understand the differences and synergies between lexical, vector, and graph retrieval in the context of modern DBMSs; (ii) Implement and evaluate each search type using standard metrics; (iii) Apply fusion techniques, such as *Reciprocal Rank Fusion (RRF)*, to combine results from different data sources; (iv) Build an end-to-end RAG pipeline that orchestrates queries across multiple database systems (relational/text, vector, and graph) to augment an LLM.

The content directly aligns with the SBBD tracks: *Information Retrieval*; *Management of semi-structured, network, and graph data*; *Knowledge bases, knowledge graphs, and modeling*; *Machine learning, AI, data management, and data systems*; and *Data science applications/pipelines*. The tutorial prepares participants to design the next generation of intelligent applications centered on the orchestration of multimodal databases.

1.1. Target Audience

This tutorial is designed for a broad range of participants interested in the intersection of databases and artificial intelligence. The audience includes:

- **Undergraduate and Graduate Students** in Computer Science, Engineering, and related fields, who wish to understand how modern database systems support AI applications.
- **IT Professionals, Data Engineers, and Researchers** seeking to update their knowledge on how to integrate lexical, vector, and graph search to build robust RAG systems.

Participants should have a basic knowledge of databases (SQL) and Python programming. Familiarity with AI concepts is helpful, but not required.

2. Tutorial Program (Duration: 2 hours)

1. **Lexical Retrieval, Vector Search and Database Extensions** (≈ 40 min)
   - Principles of TF-IDF and BM25.
   - The inverted index and its application in databases.
   - From Word2Vec to Sentence-BERT (SBERT): The evolution of embeddings.
   - Approximate vector indexing: HNSW and IVF.
   - Evaluation: Metrics such as Precision@K, Recall@K, MRR, and NDCG@K.
   - Hands-on: Using fulltext search and vector engines with DBMS extensions.

2. **Knowledge Graphs and Structured Search** (≈ 25 min)
   - Knowledge Graph modeling: Entities, Relations, Properties.
   - Query languages: Cypher and the new GQL standard.
   - Hands-on: Querying a knowledge graph in Neo4j to find relevant subgraphs.
3. **Building a Hybrid RAG Pipeline with LangChain** (≈ 40 min)
   - End-to-end flow: Query → Hybrid Retrieval (Lexical + Vector + Graph) → Fusion (RRF) → Injection into the LLM Prompt.
   - Demo in an executable notebook (Google Colab) integrating PostgreSQL, ChromaDB, and Neo4j.
4. **Discussion, Challenges, and Q&A** (≈ 15 min)
   - New approaches in RAG: LightRAG [Guo et al. 2024], PathRAG [Chen et al. 2025], among others.
   - Questions and answers.

3. Core Concepts in Retrieval-Augmented Generation

Lexical and Vector Retrieval

Modern information retrieval is characterized by a blend of traditional lexical search and contemporary vector-based methods. Lexical techniques, such as BM25 [Robertson et al. 1995], rely on inverted indexes within databases to perform efficient keyword matching, a feature now highly optimized in systems like PostgreSQL. Complementing this is the rise of vector search, driven by the evolution of dense embeddings from models like Sentence-BERT (SBERT) [Reimers and Gurevych 2019]. This approach enables semantic similarity search, which is operationalized in databases through extensions like `pg_vector` and approximate nearest neighbor (ANN) indexes such as HNSW [Malkov and Yashunin 2020]. The effectiveness of any retrieval system is contingent on rigorous evaluation, utilizing standard metrics like Precision@K, Recall@K, Mean Reciprocal Rank (MRR), and NDCG@K to measure the quality of the returned results.

Knowledge Graphs for Structured Search

For domains with complex and interconnected information, Knowledge Graphs (KGs) offer a powerful paradigm for structured data representation. KGs model knowledge by defining entities, their descriptive properties, and the explicit relations that connect them, creating a rich, machine-readable semantic network [Xavier and da Silva Soares 2024]. This structure is ideal for complex queries that go beyond simple keyword or semantic matching. Interrogating these graphs is accomplished through specialized query languages like Cypher or the new ISO standard, GQL [iso 2024, Xavier and da Silva Soares 2024]. Using these languages, one can traverse the graph to find and extract precise, contextually relevant subgraphs, which serve as a highly structured and verifiable source of information for augmenting language models.

Hybrid RAG Pipelines

A hybrid Retrieval-Augmented Generation (RAG) pipeline represents a sophisticated architecture that orchestrates multiple retrieval strategies to generate more accurate and context-aware responses [Gao et al. 2023, Xavier and da Silva Soares 2024]. The end-to-end flow begins with a user query, which triggers parallel searches across heterogeneous data sources: a lexical search for keyword relevance, a vector search for semantic similarity, and a graph query for structured relationships. A crucial challenge in this process is the fusion of results from these distinct sources. Techniques such as Reciprocal Rank Fusion (RRF) are employed to intelligently combine the relevance scores and produce a single, ranked list of the most pertinent information [Xavier and da Silva Soares 2024]. This final, synthesized context is then injected into the prompt of a Large Language Model (LLM) to ground its generation in verifiable, multi-faceted evidence.

The Evolving Landscape of Graph RAG

The RAG field is rapidly evolving, with significant innovations in graph-based retrieval. Advanced techniques are moving beyond simple entity fetching to be more efficient and context-aware. For instance, LightRAG employs a dual-level retrieval system to capture both specific and broad themes [Guo et al. 2024], while PathRAG prunes redundant information by focusing on key relational paths between nodes [Chen et al. 2025]. These methods exemplify a trend towards more intelligent and interpretable RAG systems.

4. Additional Information

4.1. About the Authors

Otávio Calaça Xavier is a professor at UFG and IFG, and is currently a PhD candidate in Computer Science, focusing on *Graph Neural Networks* and Retrieval-Augmented Generation (RAG). He holds a Master's degree in Computer Science from UFG (2011), with an emphasis on Semantic Web and Linked Data. With more than 20 years of experience in Web Application Development and Database Administration, he also has 15 years of experience in Software Architecture and Team Leadership. Since 2007, he has been a speaker at over 100 technology events in cities such as Goiânia, Brasília, São Paulo, Foz do Iguaçu, Rio de Janeiro, and Porto Alegre. He has been teaching for nearly 15 years in undergraduate, graduate, and professional training courses. He also works as a consultant in Machine Learning, Data Science, and Software Engineering and Architecture. Otávio will be the author presenting the tutorial.

Anderson da Silva Soares is a professor at the Institute of Informatics at the Federal University of Goiás, where he is a permanent member of the Master's and PhD programs in Computer Science. He was the vice-coordinator of the PhD program from 2015 to 2016 and served as an associate editor for the Journal of Computer Science from 2015 to 2018. He is a renowned researcher in the areas of machine learning, deep learning, and optimization with heuristics. As the founder of the Deep Learning Brasil laboratory, Anderson is also the chairman of the creation committee and the current coordinator of the Bachelor's program in Artificial Intelligence at UFG. As a Brazilian representative at the *Global Partnership on Artificial Intelligence (GPAI)*, he has secured significant R&D funding of over 200 million BRL, providing scholarships for students in projects with companies like Data-H, Copel Distribuição, iFood, among others. He was the founder and general director of the Center of Excellence in Artificial Intelligence of Goiás (an Embrapii unit), where he currently serves as the scientific coordinator. Several of his R&D initiatives have generated spin-off startups. As an outreach activity, he coordinated the Brazilian Robotics Olympiad in Goiás and the Federal District, and continues to volunteer with the organization. Additionally, he is an enthusiast and maintainer of the Pequi Mecânico robotics group, which is focused on the university level.

4.2. Required Computational and Audiovisual Resources

The tutorial is centered around a practical and self-contained notebook. To ensure the best experience and engagement, it is recommended that participants have access to a computer with an internet connection. The practical materials will be provided as a Google Colab notebook, which can be run directly in a web browser without the need for local software installation. The presenter will require a standard audiovisual projector and internet access.

References

(2024). Information technology – database languages – gql. Standard ISO/IEC 39075:2024, International Organization for Standardization (ISO), Geneva, CH.

Chen, B., Guo, Z., Yang, Z., Chen, Y., Chen, J., Liu, Z., Shi, C., and Yang, C. (2025). Pathrag: Pruning graph-based retrieval augmented generation with relational paths. *arXiv preprint arXiv:2502.14902*.

Gao, Y., Xiong, Y., Gao, X., and et al. (2023). Retrieval-augmented generation for large language models: A survey. *arXiv preprint*, arXiv:2312.10997.

Guo, Z., Xia, L., Yu, Y., Ao, T., and Huang, C. (2024). Lightrag: Simple and fast retrieval-augmented generation.

Lewis, P., Perez, E., Piktus, A., and et al. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 9459–9474.

Malkov, Y. A. and Yashunin, D. A. (2020). Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):824–836.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint*, arXiv:1301.3781.

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3982–3992.

Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M., and Gatford, M. (1995). Okapi at TREC-3. In *Proceedings of the Third Text REtrieval Conference (TREC-3)*, pages 109–126. NIST Special Publication.

Xavier, O. C. and da Silva Soares, A. (2024). Geração com recuperação aumentada (rag) em grafos de conhecimento. In da Silva Monteiro Filho, J. M., Razente, H., and dos Santos Mello, R., editors, *Tópicos em Gerenciamento de Dados e Informações: Minicursos do SBBD 2024*. Sociedade Brasileira de Computação, São Paulo, Brazil.