ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Department of Classical Philology and Italian Studies

Second Cycle Degree in
Digital Humanities and Digital Knowledge

From Documents to Dialogue: Design, Implementation and Evaluation of a Question-Answering System for Geoportale Nazionale Archeologia

Dissertation in
Machine Learning for the Arts and Humanities

Supervisor
Prof. Giovanni Colavizza

Co-Supervisors
Prof. Paolo Bonora
Mario Caruso and Simone Persiani, BUP Solutions

Defended by
Lucrezia Pograri

Graduation Session II
Academic Year 2024/2025

Contents

Abstract

This dissertation presents the design and development of a question-answering (QA) system tailored to the Geoportale Nazionale Archeologia (GNA). The research addresses the challenge of extracting relevant information from archaeological documentation using Retrieval-augmented generation (RAG) amd natural language processing (NLP) techniques. The methodology combines transformer-based language models with domain-specific information extraction to enable intuitive, natural language querying of technical documentation related to archaeological data.

Keywords: Digital Humanities · Information Retrieval · Question-Answering Systems · Retrieval-Augmented Generation · Natural Language Processing · AI · Cultural Heritage.

Acknowledgments

List of Figures

List of Tables

Chapter 1

Introduction

At the swiftly evolving intersection of artificial intelligence (AI) and digital humanities (DH), computational methods have profoundly transformed access to and interpretation of cultural heritage resources. Question-answering systems (QASs), driven by advances in natural language processing (NLP) and retrieval-augmented generation (RAG), have become increasingly significant tools, offering new possibilities for engaging with extensive documentation and complex cultural repositories. This thesis emerges directly from an applied research experience conducted during an internship at BUP Solutions, aimed at exploring the realistic feasibility and effectiveness of these AI technologies in the context of cultural heritage. Specifically, the project focused on developing a specialized QAS for the Geoportale Nazionale Archeologia (GNA), Italy's primary repository for archaeological data.

The motivation for this study initially arose from a concrete, practical challenge: enabling efficient, intuitive, and contextually accurate access to the extensive and often fragmented archaeological documentation hosted by GNA. Archaeologists, heritage managers, and scholars regularly face difficulties in navigating vast volumes of technical reports, field notes, operational procedures, and geospatial data. In response, this project experimented with applying cutting-edge NLP and machine learning (ML) techniques – primarily transformer-based language models and advanced information retrieval methods – to dynamically retrieve and synthesize relevant information based on user queries expressed in natural language.

Central to the chosen methodology is retrieval-augmented generation (RAG), an approach that significantly enhances traditional QASs through the dynamic retrieval of domain-specific content, which augments the generative capabilities of language models. Instead of relying solely on internal model knowledge, RAG-based systems integrate external document retrieval with generative text production, resulting in greater reliability and contextually grounded responses – crucial qualities for scholarly and professional uses. While this approach inherently promises increased accuracy and reduced hallucinations compared to purely generative methods, it also involves several complexities and uncertainties, which were encountered firsthand during the development and evaluation phases, as will be discussed in the following chapters.

Rather than adopting a narrowly theoretical or idealized perspective, this study reflects the exploratory and evolving nature of hands-on experimentation, shaped by iterative cycles of trial-and-error, heuristic adjustments, and pragmatic responses to practical constraints such as computational limits, the absence of standardized evaluation benchmarks, and the linguistic complexity of the domain. This process brought to light the persistent tension between the ambitions of AI-driven solutions and the realities of applying them in intricate cultural contexts. In systems like the GNA's AI assistant, the focus necessarily shifts from abstract notions of understanding to measurable outcomes: the true test is not whether the system comprehends archaeology in any human sense, but whether it efficiently retrieves relevant information, handles the complexities of the domain, and supports users in making informed decisions. Against such backdrop, one might ask: how far can technical ingenuity take us before we run up against the unique subtleties of human knowledge and practice? It's in this spirit that I recall McDermott's classic essay, *Artificial Intelligence Meets Natural Stupidity*, which cautions against the lure of *wishful mnemonics* in AI and urges us to resist the temptation to label what our systems do with grand terms like "understand". Instead, McDermott advocates for a clear-eyed assessment and communication of what these systems actually accomplish – and where their true limits lie (McDermott, 1976).

In light of this reality, this study deliberately avoids overstating the system's semantic or interpretive capabilities. Instead, it foregrounds the project's exploratory nature, acknowledging both methodological achievements and encountered limitations. The outcome represents a pragmatic effort toward applying AI in the digital humanities, offering insights into the real-world challenges and possibilities of using retrieval-augmented generation in cultural heritage contexts.

This project remains, at its core, fundamentally hopeful. It demonstrates that even in the face of inherent methodological challenges, AI-driven tools such as RAG-based QAS hold substantial promise for enhancing access to cultural heritage information. Through a transparent presentation of both the strengths and shortcomings discovered during this internship experiment, this thesis seeks to contribute realistically yet optimistically to the ongoing dialogue between artificial intelligence and humanistic inquiry, offering a vision of AI's evolving role in supporting cultural heritage scholarship.

Chapter 2

The Evolution of Question-Answering Systems

This chapter introduces the foundations of question answering (QA) as both a computer science discipline and an applied technology. Before the emergence of large language models (LLMs),[1] Transformers,[2] and modern generative AI,[3] question-answering systems (QAS) progressed through distinct paradigms: from symbolic and rule-based architectures to classic information retrieval (IR) models and early neural networks approaches (Jurafsky and Martin, 2024; Antoniou and Bassiliades, 2022). Early systems depended on domain-specific adaptations, manually curated knowledge bases, keyword retrieval, and engineered features. In recent years, transformer-based language models such as BERT and GPT have significantly advanced the capabilities of QA systems by enabling both answer extraction and text generation. Unlike their predecessors, these models can generate or extract responses using deep contextual understanding derived from large-scale pretraining (Kaplan et al., 2020). However, they tend to exhibit factual inaccuracies, shallow contextual understanding in certain scenarios, and limited adaptability to new or evolving information. They also frequently hallucinate or generate outdated responses, constrained by their static training corpora (Harsh and Shobha, 2024).

The main stages in the evolution of QA systems, along with representative approaches and landmark examples, are summarized in Tab. 1 below.

[1]Large Language Models (LLMs) are advanced AI systems trained on massive text datasets to generate and understand human language. For an accessible overview, see *A Very Gentle Introduction to Large Language Models without the Hype* (Riedl, 2023).

[2]The Transformer is a neural network architecture introduced in 2017 that efficiently models sequential data using a self-attention mechanism. The original paper, *Attention Is All You Need* by Vaswani et al. (2017), provides a foundational outline.

[3]Generative AI refers to systems capable of producing new content, such as text, images, or audio, based on learned patterns. For more, see the *Stanford AI Index 2025 Report* (Maslej et al., 2025).

Models	QA Approach	Examples / Results
Symbolic / Rule-based (1960s–1980s)	Rule-based, domain-specific, handcrafted knowledge base	BASEBALL, LUNAR, SHRDLU
Early IR Approaches (1990s–mid-2010s)	Keyword retrieval, TF-IDF, BM25, open-domain ranking	TREC QA
Statistical / Seq2Seq (2000s–2018)	N-gram, embeddings, RNN/LSTM, statistical IR	Early neural QA, Reading comprehension in 2010s
Transformer-based	Pre-training, fine-tuning, self-attention	BERT (93% F1 on SQuAD), XLNet
Generative LLMs and agents	Prompting, retrieval-augmented generation, agentic reasoning	GPT-3, RAG pipelines

Table 1: Evolution of QA systems.

2.1 Pre-Transformer Era: Symbolic and Statistical Systems

The development of QAS prior to the rise of Transformers was shaped by several key methodological shifts and technological milestones. These earliest efforts prioritized manually curated knowledge bases and rules-based systems for precise but limited question matching. As the scope of QA expanded, techniques evolved to incorporate large-scale information retrieval methods, statistical modeling, and increasingly complex approaches to feature engineering and answer extraction. This trajectory ultimately set the stage for early neural models that leveraged word embeddings and sequence modeling, gradually moving the discipline toward data-driven architectures and deeper semantic representation. The following paragraphs trace these major trends, illustrating how each contributed to the capabilities and limitations of pre-Transformer QA systems.

2.1.1 Rule-Based Systems (1960s–1980s)

Early QAS relied on highly constrained, domain-specific approaches built around manually constructed knowledge bases. These systems operated within carefully delineated boundaries, matching user questions to a limited set of predefined templates and answer patterns. While this design enabled highly precise responses in their target domains, it also rendered the systems brittle and inflexible – minor variations in user queries or topics outside the encoded scope often resulted in failure to provide meaningful answers.

Expert systems from this era encoded explicit inference rules and logical representations of knowledge, enabling a form of automated reasoning that was fundamentally deterministic. However, these approaches struggled to address ambiguity or generalize beyond the hand-curated domain, and could not scale to larger, more dynamic information environments (*Question answering*, 2025; Jurafsky and Martin, 2024).

Seminal examples of early domain-specific QA systems include:

- **BASEBALL** (1960s): Hand-coded rules and database logic for Major League Baseball[4] questions (Green et al., 1961).
- **LUNAR** (1971): Pattern matching and restricted knowledge base for geological questions about Moon rocks (Woods et al., 1972).
- **SHRDLU**[5] (late 1960s): Symbolic reasoning for a blocks-world robot in a toy domain (Winograd, 1971).
- **Unix Consultant (UC)**[6] and **LILOG**[7] (1980s): Domain-specific QA via linguistic rules and expert knowledge; though both projects remained at the demonstration stage, they contributed to advanced research in computational linguistics.

These early QA systems demonstrated the potential of automated question answering but highlighted the central challenge of balancing precision with generality and scalability. Their

[4]Major League Baseball (MLB) is the leading professional baseball league in North America. It is regarded as the world's premier baseball competition.

[5]SHRDLU was developed at the MIT Computer Science and Artificial Intelligence Laboratory (CSAIL) between 1968–70. The software allowed users to interact conversationally with a program that could manipulate, describe, and answer questions about objects in a virtual `blocks world', a simplified environment containing various movable blocks. Read more about SHRDLU program here: https://hci.stanford.edu/winograd/shrdlu/.

[6]UC (QA) system, created at U.C. Berkeley (CA), answered queries about the Unix operating system using a hand-crafted knowledge base and could tailor responses to different user types (Wilensky et al., 1988).

[7]LILOG project was as a text-understanding system designed for tourism information in a German city (*Question answering*, 2025).

evolution would motivate the subsequent shift toward statistical and data-driven approaches (Jurafsky and Martin, 2024; Antoniou and Bassiliades, 2022).

2.1.2 Classic Information Retrieval Strategies (1990s–mid-2010s)

As the volume of unstructured web data grew, QA moved toward ranking text passages with IR techniques like TF-IDF[8] and BM25,[9] to locate relevant content within large text collections. Open-domain QA systems – such as those in TREC QA[10] (Hirschman and Gaizauskas, 2001) – shifted the focus from structured fact retrieval to returning ranked sentences or extracting answer spans from retrieved passages. These approaches made it possible to scale QA to a broad range of topics and data sources, yet they also introduced notable challenges. Lacking deep understanding of natural language, IR-based QA systems often failed to interpret nuances, synonyms, or complex phrasing, and frequently missed correct answers that did not explicitly match the user's query terms (Antoniou and Bassiliades, 2022; Caballero, 2021).

2.1.3 Statistical Models and Feature Engineering (2000s–2018)

During the 2000s and 2010s, the adoption of n-gram models and statistical IR approaches (cf. TF-IDF, BM25, probabilistic models[11]) enabled reasoning over large corpora, moving beyond hand-crafted rules and enabling automated extraction of candidate answers from vast, unstructured datasets (Manning et al., 2008). The introduction of word embeddings – e.g., Word2Vec, GloVe – marked a significant advancement by capturing semantic similarities between words, thereby allowing models to generalize beyond simple keyword matching. These dense vector representations supported the emergence of recurrent neural networks (RNNs) and long short-term memory networks (LSTMs), which facilitated more accurate modeling of sequence and context in reading comprehension and retrieval-based QA tasks (Jurafsky and Martin, 2024).

A major milestone in this era was IBM's *Watson* system, which achieved notable success by

[8] TF-IDF (Term Frequency–Inverse Document Frequency) is a statistical method for ranking how important a word is to a document in a collection.

[9] BM25 is a ranking function that improves information retrieval by considering term frequency, document length, and saturation effects.
For more details on TF-IDF and BM25, read *Introduction to Information Retrieval* (Manning et al., 2008).

[10] TREC QA refers to the Question Answering track of the Text REtrieval Conference (TREC), a long-running evaluation series that has set benchmarks for open-domain QA research since 1999. See https://trec.nist.gov/data/qa.html

[11] Language Models for IR (LMIR) – such as n-gram models – estimate the probability of a query being generated by a document's language model. They capture local word dependencies and were widely used in early QA, speech recognition, and spelling correction, (Ponte and Croft, 1998) but were later outperformed by models like RNNs, LSTMs, and Transformers due to their limited handling of long-range context.

winning the *Jeopardy!* challenge in 2011.[12] Watson's *DeepQA* architecture integrated hundreds of NLP, IR and ranking components, employing sophisticated pipelines to analyze and combine evidence from diverse sources (Ferrucci et al., 2011). However, despite its advanced design, *Watson* relied on non-generative methods; it synthesized and ranked candidate answers but did not generate free-form responses from scratch.

Simultaneously, semantic QA systems began to emerge, mapping natural language (NL) questions to structured queries (e.g. using SPARQL language) executed over knowledge bases like Freebase and DBpedia. These systems required advanced components for entity recognition, relation extraction, and reasoning over symbolic representations. Typical architectures included steps like question analysis, sentence mapping, disambiguation, and query building, enabling automatic translation of natural language into formal queries over RDF data sources. Thanks to the usage of ontology-mapping and linguistic resources – e.g., WordNet –, these approaches further bridged the gap between unstructured text and structured knowledge bases (Franco et al., 2020).

Throughout this period, feature engineering played a central role. Techniques such as conditional random fields (CRFs) and support vector machines (SVMs) enabled models to exploit hand-crafted features – including lexical overlap, question type, and answer patterns—to enhance answer extraction from retrieved texts. Hybrid QA systems appeared, combining keywords-based information retrieval methods for unstructured sources with knowledge-base querying for fact-based answers, thereby improving both coverage and precision (Antoniou and Bassiliades, 2022).

This period laid essential groundwork for the deep learning and neural approaches that would soon transform the QA landscape, highlighting the importance of both statistical modeling and intelligent feature design.

2.1.4 Early Neural and Generative Models (Late 2010s)

The late 2010s witnessed the adoption of neural architectures in question answering, building upon the foundational use of word embeddings and recurrent neural networks (RNNs). Embedding methods such as Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014)

[12]The *Jeopardy! challenge* was a high-profile test where IBM *Watson* competed on the American television quiz show *Jeopardy!* against two of the show's greatest human champions. Watson's victory demonstrated significant progress in machine comprehension and open-domain question answering (Wikipedia IBM Watson). In February 2013, IBM announced that *Watson*'s first commercial deployment would assist with utilization management decisions for lung cancer treatment at Memorial Sloan Kettering Cancer Center in New York City, in partnership with WellPoint (now Elevance Health) (Upbin, 2013).

allowed systems to capture deeper semantic relationships between words, providing a richer representation of both questions and candidate answers. RNNs, and their improved variants like long short-term memory networks (LSTMs) and gated recurrent units (GRUs), facilitated sequential modeling of language, enabling systems to better process and compare question and answer pairs based on their context within a sentence or passage.

Despite these advancements, early neural QA models still faced significant limitations. The reliance on RNNs restricted their ability to effectively model long-range dependencies in text, often resulting in incomplete understanding when questions required reasoning across multiple sentences or broader contexts. While neural models improved matching between questions and answers, their performance remained constrained by the size and variety of the training data.

Around this time, encoder-decoder architectures began to appear in QA research, drawing inspiration from their success in machine translation. These generative models aimed to produce answers by generating sequences of text, rather than simply extracting passages from a source document. However, early generative QA systems often struggled with factual consistency: they had a tendency to copy or paraphrase the input text rather than synthesizing novel, precise answers. Additionally, these models sometimes hallucinated information or failed to maintain logical coherence in their generated responses, limiting their reliability in open-domain settings (Caballero, 2021).

These developments set the stage for the subsequent breakthroughs brought about by attention mechanisms and transformer-based architectures, which dramatically improved the handling of context and factuality in generative QA.

2.2 Blind Spots and Bottlenecks: The Shortcomings of Early Approaches

Earlier approaches to question answering were hindered by several fundamental limitations. Most notably, symbolic and rule-based systems suffered from severe domain restrictions, as their performance relied on hand-crafted knowledge bases and rigid rules that did not generalize well to new or broader topics (Alqifari, 2019). The brittleness of these systems was further exposed by their heavy dependence on template matching, which frequently led to failures when users phrased questions in unanticipated ways or employed linguistic variations (Hirschman and Gaizauskas, 2001). Information retrieval (IR) and statistical models, while more scalable, continued to struggle with true semantic understanding and contextual reasoning, often retrieving only superficially relevant snippets rather than synthesizing comprehensive or contextually rich answers (Alanazi et al., 2021; Diefenbach et al., 2018). The answers these systems produced were typically shallow, extracted verbatim from source texts rather than generated or adapted to the user's specific information need (Hirschman and Gaizauskas, 2001; Alqifari, 2019).

Substantial manual effort was required to design, maintain, and update rules, features, and parsers, creating significant bottlenecks and making adaptation to new domains costly and time-consuming (Alanazi et al., 2021). In addition, IR and knowledge base approaches frequently exhibited incomplete coverage, missing relevant answers due to differences in phrasing or limitations in their underlying datasets (Diefenbach et al., 2018). Early neural models, despite improvements, were generally confined to handling short text spans and struggled with complex or multi-sentence reasoning tasks. Finally, all these methods exhibited a strong dependence on the quantity and quality of available training data and engineered features, resulting in inconsistent performance across different domains and question types (L. Liu et al., 2022; Alanazi et al., 2021; Alqifari, 2019; Diefenbach et al., 2018; Hirschman and Gaizauskas, 2001).

These cumulative factors left pre-generative QA systems largely inflexible and brittle, with limited ability to provide context-aware, nuanced, or creative responses to user queries.

2.3 Deep Learning Breakthroughs

The advent of the Transformer architecture fundamentally reshaped the field of deep learning and revolutionized neural QAS. Introduced by Vaswani et al. in 2017, Transformers replaced RNNs and LSTMs with a self-attention mechanism that could model relationships between words regardless of their distance in the input sequence. This innovation allowed for efficient parallelization during training and inference, dramatically improving the scalability and performance of language models on a range of NLP tasks, including QA.

One of the earliest and most influential transformer-based models was BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019). BERT employs a bidirectional attention mechanism and is pre-trained using a masked language modeling objective, allowing it to capture complex context from both directions in a sentence. When fine-tuned for QA benchmarks, such as SQuAD (Rajpurkar et al., 2016), BERT achieved unprecedented accuracy – reaching Exact Match and F1 scores above 85% and 87% respectively on the SQuAD 2.0 leaderboard –, surpassing previous neural models and establishing a new standard for QA (Li and Zhang, 2024).

Building on this foundation, subsequent models explored variations and enhancements of the Transformer paradigm. XLNet, for example, employed a permutation-based language modeling objective, enabling it to better capture bidirectional context and achieve state-of-the-art results on several QA benchmarks (Z. Yang et al., 2020). In specialized domains, models such as BioBERT extended the BERT architecture with additional pre-training on biomedical texts, achieving top performance on domain-specific challenges like the BioASQ question answering competition (Yoon et al., 2019). Parallel research into model architectures also produced frameworks such as Dynamic Coattention Networks (DCN), which fused question and context representations through attention mechanisms and iterative decoding, further improving accuracy on reading comprehension tasks (Xiong et al., 2018).

Surveys in literature underscore a pronounced move toward both extractive and generative QA pipelines, with each stage – tokenisation, embedding, retrieval, and answer generation – now being explicitly modeled and systematically optimized (Farea and Emmert-Streib, 2025). At the same time, interest in conversational and multi-turn QA has grown rapidly, as Transformers demonstrate substantial ability to manage dialogue context and maintain coherent, context-aware interactions with users (Yue, 2025; Antoniou and Bassiliades, 2022). Together, these advances have laid the foundation for generative AI systems and retrieval-augmented approaches that now dominate state-of-the-art QA research.

2.4 Large Language Models, Agents and Modular Pipelines

A clear distinction now emerges between "traditional" QA systems, primarily built upon general-purpose pre-trained language models – such as GPT, Llama, T5, etc. – and the new wave of modular approaches that dynamically retrieve external information sources. Traditional QA encompasses both extractive and generative paradigms, each defined by how they use the model's internal knowledge. Extractive QA models are designed to identify and extract exact answer spans directly from a provided text or document, making them highly effective for fact-based questions and reading comprehension tasks. Generative QA models, in contrast, use natural language generation (NLG) to produce answers, often synthesizing or paraphrasing responses in ways that may not appear verbatim in the original text. However, despite their success, both of these paradigms are fundamentally limited by the static nature of their training data: they may struggle with rare, rapidly changing, or domain-specific queries, and are prone to hallucinations and outdated information (Farea and Emmert-Streib, 2025).

The latest advances in question answering are characterized by the emergence of generative LLMs and retrieval-augmented generation (RAG) pipelines. In these systems, a retriever component dynamically accesses external knowledge bases, while a generator synthesizes fluent, grounded answers by conditioning on the retrieved information. This hybrid approach addresses many of the shortcomings of earlier transformer-based models by significantly enhancing factual accuracy, contextual relevance, and system adaptability. Generative LLMs within the RAG pipeline are able to incorporate real-time knowledge, thereby reducing hallucinated content and providing up-to-date responses, even as external data sources evolve (Yue, 2025; Lewis et al., 2020).

Furthermore, RAG-based QA systems offer practical advantages for scalability. Rather than requiring full model re-training to accommodate new information, they can simply update or expand the external document index or knowledge base (KB). This design allows for the integration of vast and dynamic data resources, enabling high coverage across domains and rapid adaptation to new information needs. At the same time, these benefits come with trade-offs. RAG architectures require more complex infrastructure, including document indexing and retrieval pipelines, which increase computational overhead and system latency compared to traditional, static QA models. As a result, deploying and maintaining RAG-based systems can be more challenging, especially at scale.

Recent research also explores QA agents that operate as multi-stage pipelines and highly modular systems, capable of orchestrating question understanding, document retrieval, reasoning, and answer synthesis in a coordinated workflow (Skarlinski et al., 2024). This evolution illustrates a profound transformation in how question answering is conceptualized, implemented, and applied across a wide range of domains.

Tab. 2 summarizes the functional differences between traditional and RAG-based QA systems, highlighting the shift toward dynamic, retrieval-augmented, and generative approaches that characterize the current state of the discipline.

Feature	**Traditional QAS** *(e.g., BERT, GPT-2/3)*	**RAG QAS** *(Retriever + Generator)*
Knowledge source	Fixed (training data)	Dynamic (external docs/databases)
Answer type	Extracted or generated	Retrieved + generated (synthesized)
Factual accuracy	Limited (can hallucinate or be outdated)	High (grounded in retrieved, up-to-date information)
Contextual depth	Limited	Comprehensive, nuanced
Scalability	Moderate	High (can update external data sources)
Computational cost	Lower	Higher (due to retrieval/generation)
Latency	Lower (faster for simple queries)	Higher (retrieval step adds time)
Complexity of setup	Simpler	More complex to maintain
Adaptability	Less adaptable to new domains	Highly adaptable via updated document index

Table 2: Comparison of traditional vs. RAG question-answering systems. Adapted from https://www.geeksforgeeks.org/nlp/rag-vs-traditional-qa/.

Chapter 3

State of the Art

In the landscape of AI, large language models (LLMs) have demonstrated remarkable results in text generation and understanding. Yet, when applied to real-world tasks such as question answering, these models still face significant limitations. As already mentioned in Chap. 2, LLMs are prone to hallucinations,[13] rely on static and often outdated training data, and offer limited transparency or traceability in their outputs. Additionally, they may struggle to incorporate domain-specific context or organizational knowledge (Vaibhav, 2025). These factors pose challenges for domains – like cultural heritage, GLAM (Galleries, Libraries, Archives and Museums) and archaeology – where reliability, provenance, and interpretive rigor are fundamental requirements (Di Marcantonio, 2024).

To address these concerns, retrieval-augmented generation (RAG)[14] has emerged as a crucial methodological advance. It improves the factual grounding and contextual relevance of generated answers, thorugh the integration of external and verifiable knowledge at inference time, thereby reducing the risk of generating fabricated or distorted information (Martineau, 2023). As discussed in Sec. 2.4, this approach represents a significant step beyond both traditional information retrieval and earlier neural QA models, which were often brittle, domain-dependent, or struggled to adapt to evolving information needs.chunks

The adoption of RAG in question answering reflects a broader evolution within the field: from early symbolic and rule-based systems, through statistical and information retrieval approaches, to today's transformer-based, generative architectures. This shift has transformed not only the technical capabilities of QA systems but also their applicability to complex, heterogeneous knowledge domains.

Although initially developed for open-domain question answering and enterprise search

[13] In the context of LLMs, hallucinations refer to outputs that are plausible-sounding but factually incorrect, fabricated, or unsupported by the underlying data or external sources (Harsh and Shobha, 2024).

[14] For more information about RAG technique, see https://en.wikipedia.org/wiki/Retrieval-augmented_generation.

(Akkiraju et al., 2024; Jiang et al., 2024; Packowski et al., 2024; R. Yang et al., 2025; Zhou et al., 2025), RAG pipelines are increasingly adopted in the humanities and cultural heritage contexts. In these sectors, where interpretive rigor, provenance, and information reliability are critical, RAG-enabled tools support scholars and professionals in navigating vast, fragmented knowledge repositories. While some initiatives employ RAG to analyze sensitive historical materials (Callaghan and Vieira, 2025; Ciletti, 2025; Sergeev et al., 2025; Fan et al., 2025), this thesis explores a distinct application: improving access to procedural and technical documentation, where clarity, consistency, and actionable guidance are the primary objectives.

This chapter therefore provides a comprehensive overview of the state of the art in retrieval-augmented generation, situates RAG within the current research landscape, outlines its core mechanisms, and examines its recent application in the digital humanities.

3.1 Foundations of Retrieval-Augmented Generation

Retrieval-augmented generation (RAG) is a hybrid approach designed to address several critical limitations of traditional LLMs, such as knowledge staleness, limited context awareness, and insufficient output traceability (Vaibhav, 2025; Y. Gao et al., 2024; Gupta et al., 2024). While LLMs excel at producing fluent, human-like text, they often falter when facing domain-specific queries or requests for information beyond their training cutoff. RAG directly addresses these challenges by integrating external information retrieval within the generation process, allowing outputs to be more factual, up-to-date, and grounded in verifiable sources (X. Wang et al., 2024).

At its core, a typical RAG workflow consists of two main stages: **retrieval** and **generation** (ODSC-Community, 2024). The process begins with preprocessing and indexing, where raw data is cleaned, extracted, segmented into manageable "chunks", and encoded into vector representations. These embeddings are then stored in a vector database (e.g., Milvus, Faiss, Qdrant)[15] to facilitate efficient similarity searches. When a user submits a query, it is encoded in the same vector space, and the system retrieves the top-k most relevant chunks from the indexed knowledge base. In the subsequent stage, these retrieved documents are passed as context to a generative language model – often based on Transformer architectures (Vaswani et al., 2017) – which synthesizes a response that blends the original query with external evidence, producing answers that are both coherent and contextually appropriate (Arslan et al.,

[15] Cf. glossary in Appendix B.

2024).

This modular design (Fig. 1) enables the continuous incorporation of domain-specific and current information, overcoming the constraints of static model parameters. Recent contributions have helped to formally systematize the RAG pipeline's, with frameworks delineating specific interdependent modules such as query classification, retrieval, reranking, and generation (X. Wang et al., 2024; Y. Gao et al., 2024).

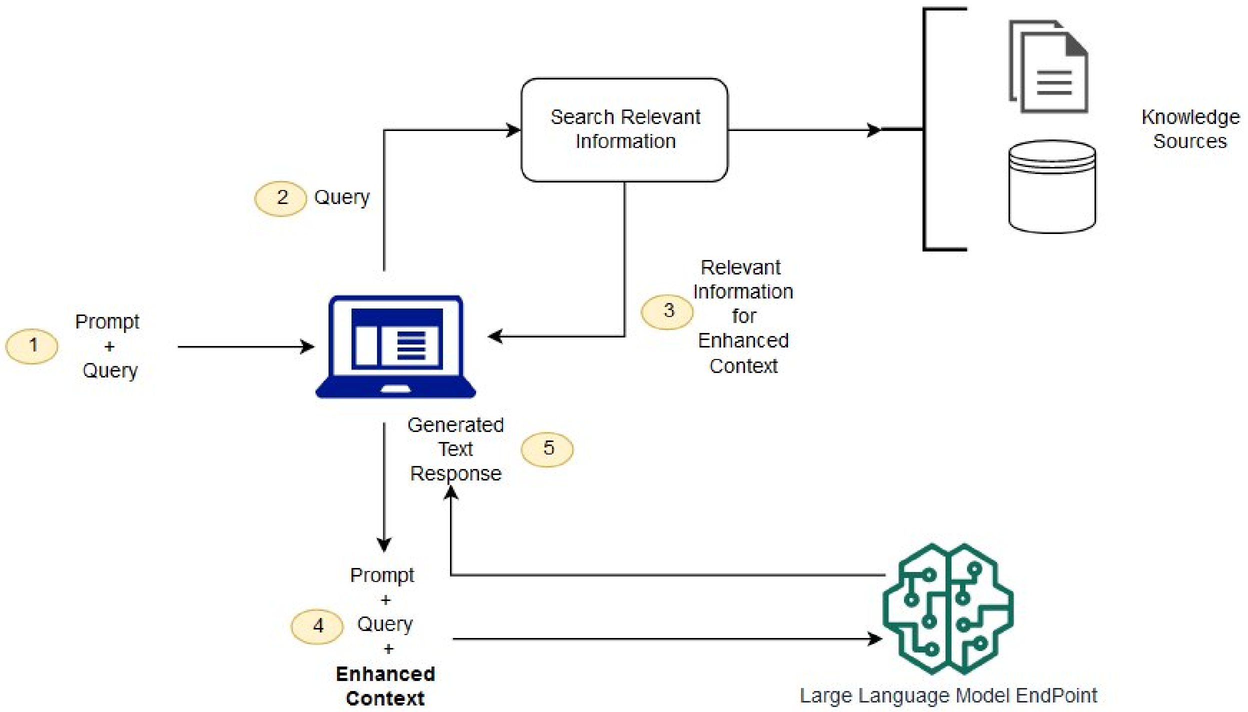

Figure 1: Typical RAG workflow.
Source: https://aws.amazon.com/de/what-is/retrieval-augmented-generation/.

3.1.1 Pipeline Design and Common Practices

The design of RAG pipelines can vary significantly based on the specific use case, domain, and available resources. However, several common practices have emerged that reflect the current state of the art in retrieval-augmented generation (Vaibhav, 2025; X. Wang et al., 2024; Arslan et al., 2024; Y. Gao et al., 2024; Gupta et al., 2024).

The standard workflow includes the following components:

- **Query understanding and classification:** Not all queries require retrieval from external sources. Advanced systems first analyse and classify incoming queries to determine whether retrieval is necessary or if the LLM alone suffices. This step leverages natural language understanding (NLU) techniques to extract key entities, relationships, and user intent, improving efficiency and reducing unnecessary retrieval latency.

- **Document indexing and chunking:** Raw data from source documents is preprocessed: cleaned, segmented into manageable chunks at token, sentence, or semantic level, and converted into dense vector representations (embeddings). Recent studies recommend dynamic or semantic chunking over simple fixed-size splitting, as it better preserves context and improves retrieval quality – especially in heterogeneous domains.

- **Embedding and Vector Database:** Both document chunks and user queries are embedded into a shared vector space using models fine-tuned for semantic similarity (e.g., BAAI/bge, LLM-Embedder, intfloat/e5). These vectors are stored in efficient vector databases (e.g., Milvus, Faiss, Qdrant), selected based on scalability, indexing strategies, and support for hybrid (vector plus keyword) search capabilities.

- **Retrieval and query transformation:** Upon receiving a user query, the system encodes it into a vector and retrieves the top-k most relevant chunks from the indexed knowledge base (KB) using similarity search. Robustness is enhanced through hybrid retrieval, which combines dense (vector-based, e.g., DPR, Contriever) and sparse (lexical, e.g., BM25) methods. Advanced query transformation techniques – including query rewriting, decomposition, or hypothetical document generation (e.g., HyDE)– can further improve retrieval effectiveness.

- **Reranking:** Initially retrieved candidates are often re-ranked based on relevance to the original query, using additional models (DLM-based) – e.g., cross-encoders like monoT5, monoBERT, or RankLLaMA, which jointly consider the query and each candidate – or more sophisticated algorithms through heuristics. This contextualization ensures that the most pertinent information is prioritized for the generative model.
  See Tab. 3 for a summary on reranking techniques.

- **Repacking and summarization:** In some cases, retrieved passages may be reorganized or summarized to distill key information, especially when dealing with lengthy corpora. This step can involve extractive summarization or abstractive (e.g., with Pegasus or T5) techniques to condense information and fit within the context window of the generator model.

- **Generation:** The generative model – usually a transformer-based LLM such as T5, BART, or GPT – synthesizes a response conditioned on both the original query and the retrieved context, integrating intrinsic model knowledge with external evidence to produce a coherent, accurate, and contextually grounded answer.

Tab. 4 lists the recommended best-performing methods for each module of a RAG workflow, employed empirically to achieve highest answer quality and accuracy. For balanced efficiency – i.e., achieving lower latency while maintaining good, but not maximal, accuracy – should be adjusted at the retrieval and reranking stages. Specifically, this involves replacing the Hybrid with HyDE retrieval method with the Hybrid search-only approach, which combines BM25 and dense retrieval without pseudo-document generation, and substituting "monoT5" with TILDEv2 for reranking, which offers much faster processing at a slight cost to answer quality (X. Wang et al., 2024).

Algorithm	Rationale
Cross-encoder rerankers with DLMs	DLMs (deep language models)-based rerankers (e.g., monoT5, monoBERT, RankLLaMA) jointly encode concatenated query-document pairs for fine-grained relevance scoring. These models are fine-tuned to classify relevance as "true" or "false", and at inference, documents are ranked by the predicted probability of the "true" label (X. Wang et al., 2024).
TILDE (Zhuang and Zuccon, 2021)	Token-level likelihoods for queries across a collection, allowing fast reranking by summing the probabilities of query tokens given each candidate passage.
Learning-to-Rank (LTR) (Gupta et al., 2024)	Traditional machine learning ranking approaches: **a) Pointwise:** predicts relevance score for each document independently; **b) Pairwise:** compares pairs of documents to learn relative relevance; **c) Listwise:** considers the entire ranked list at once.
HyDe (L. Gao et al., 2022)	Generates hypothetical documents from queries for dense retrieval.
Hybrid Search (sparse + dense scoring)	Blends scores from dense retrievers (semantic similarity – e.g., DPR, Contriever) and sparse methods (lexical overlap – e.g., BM25, TF-IDF) for robust ranking. Sometimes uses learnable weighting (X. Wang et al., 2024).
HyDE + Hybrid Search (2024)	Combines HyDE's hypothetical document generation with hybrid search for retrieval.
Graph-based (Han et al., 2025)	Constructs a graph of candidates (nodes) based on relationships (semantic, citation, or knowledge graph edges), then uses graph algorithms (e.g., PageRank, label propagation) to identify central passages.
Self-RAG (LLM-enhanced reranking) (Asai et al., 2023)	Uses LLMs directly to score or select the most relevant passages, sometimes via few-shot prompting or chain-of-thought reasoning.

Table 3: Algorithms for document retrieval and reranking in RAG pipelines.

Module	Method(s)	Functionality
Retrieval	Hybrid with HyDE	Combines Hybrid Search (BM25 + dense) and HyDE pseudo-documents.
Reranking	DLM w/ monoT5	Deep LM-based reranker (good balance of quality and speed).
Chunking	Small2big / Sliding Windows	Organizing chunk block relationships for context preservation.
Embedding	LLM-Embedder	Dense supervised retriever, best trade-off performance/size.
Vector Database	Milvus	Best coverage of index type, scalability, hybrid search, cloud-native.
Repacking	Reverse	Puts most relevant context close to the query.
Summarization	Recomp	Both extractive and abstractive methods tested; Recomp performs best.

Table 4: Best-performing RAG pipeline selections for maximizing performance w.r.t. answer quality and accuracy.

3.1.2 Evaluation and Benchmarking

Evaluating RAG systems poses unique challenges, as traditional metrics like BLEU or ROUGE may not fully capture the quality of generated responses, particularly regarding factual accuracy and contextual relevance. To address these shortcomings, researchers have introduced specialized benchmarks and frameworks. For example, the Retrieval-Augmented Generation Assessment System (RAGAS) specifically targets the evaluation of answer faithfulness and contextual alignment, offering a more nuanced perspective on RAG system performance (Es et al., 2023). Human-in-the-loop evaluations have also gained prominence, with expert judges assessing generated responses on criteria such as coherence, factual accuracy, and relevance to original queries – providing richer and more reliable quality assessments than automated metrics alone (Gupta et al., 2024).

Continuous evaluation against established benchmarks like BEIR (Thakur et al., 2021) and TREC (Voorhees et al., 2005) remains essential for tracking performance across diverse retrieval and generation tasks. Regular benchmarking ensures that systems maintain high standards for accuracy, relevance, and user satisfaction. Even after deployment, proactive monitoring is critical. Tracking system performance, user interactions, and error patterns allows developers to address issues quickly and keep RAG solutions reliable and effective as data and usage patterns evolve (Amershi et al., 2019).

Ethical considerations must be embedded throughout the lifecycle of RAG systems. Prioritizing data privacy, addressing algorithmic bias, and complying with regulations such as GDPR are fundamental. This means adopting technical measures like privacy by design, data minimization, and access control, in addition to committing to broader ethical principles widely recognized in international guidelines: transparency, justice and fairness, non-maleficence and responsibility (Jobin et al., 2019). Responsible AI use further requires explicit accountability for developers and organizations by clarifying roles and legal liabilities, supporting whistleblowing, and involving a diverse range of stakeholders in ongoing evaluation and governance. Regular audits and impact assessments are necessary to monitor bias, discrimination, and compliance with evolving ethical standards. Finally, these efforts should be seen not just as soft law, but as integral to bridging the gap between principles and practice, and to building trustworthy, socially beneficial RAG systems in line with both legal requirements and the broader values of justice and social inclusion (Ashery et al., 2025).

The trajectory of RAG systems points toward increasingly sophisticated applications that are deeply integrated into the workflows of research, industry, and cultural institutions. Innovations in evaluation frameworks, user interaction, and system scalability are steadily pushing the boundaries of what these models can achieve. As these technologies continue to mature, success will depend on the ability to combine robust benchmarking, user-centered feedback mechanisms, and adaptive optimization strategies. Addressing challenges related to factuality, scalability, and responsible deployment will be essential for building trustworthy systems capable of delivering high-quality, contextually relevant information. With continued progress, RAG systems are set to play a pivotal role in shaping the future of digital knowledge access and discovery. (X. Wang et al., 2024; Y. Gao et al., 2024).

3.2 New Frontiers Applications

RAG systems are increasingly deployed across diverse domains – spanning academia, enterprise, and product environments – to enhance data accessibility, support decision-making, and facilitate natural language interaction with complex knowledge bases. Recent surveys and empirical studies document a growing array of scholarly applications of RAG, including:

- Automated literature review tools and citation management – e.g., LitLLM; (Agarwal et al., 2025), KNIMEZoBot; (Alshammari et al., 2023);
- Generation of summaries for large corpora of academic papers;
- Field-specific knowledge extraction, including biomedical and legal research support.

In one experiment, a RAG system was developed to assist data scientists through a combination of GROBID library[16] for structured bibliographic extraction, fine-tuned embeddings, semantic chunking, and an abstract-first retrieval strategy. The system's performance, assessed using the Retrieval-augmented generation Assessment System (RAGAS), demonstrated improved faithfulness and context relevance in response generation (Aytar et al., 2024). A similar approach was explored in the context of academic library systems, where RAG was applied to improve contextual retrieval through semantic indexing of structured metadata (e.g., MARC/RDA standards) and multimodal resources. Additionally, the framework introduced conversational querying via a natural language interface, supporting complex interdisciplinary searches and significantly improving document discoverability by synthesizing citation-backed responses from diverse scholarly sources – including journals, datasets, and videos. This solution also addressed challenges such as copyright compliance and ethical AI transparency (Bevara et al., 2025). Collectively, these studies affirm RAG systems' efficacy in alleviating information overload and improving research workflow discoverability.

In parallel, the work of (Soman and Roychowdhury, 2024) provides further critical insights into the design of RAG systems for domain-specific and technical content, closely aligning with the methodological framework adopted in the GNA question-answering system. Using IEEE telecommunications engineering corpora (i.e., wireless LAN specifications and battery glossaries) as testbeds, their analysis highlights key factors influencing retrieval quality, which include chunk size, sentence-level similarity, and the strategic placement of domain-specific terms. These aspects are similarly addressed in the GNA RAG pipeline (Pograri, 2025), which

[16]GROBID is a machine learning library designed to extract, parse, and convert raw documents, like PDFs, structured XML/TEI encoded documents (*GROBID*, 2008–2025).

applies customized tailored chunking, semantic preprocessing, and contextual embedding strategies. Both studies advocate for more nuanced, context-aware approaches to enhance precision in technical and highly structured domains.

Numerous recent graduate-level research projects have provided substantive input into the implementation and evaluation of RAG systems:

- Antolini (2025) developed a custom RAG system for open-domain question answering using both traditional (BM25, PRF) and advanced retrieval strategies, integrated with local LLMs. A novel Parametric RAG (PRAG) approach was also explored, embedding context into model parameters for performance gains.
- Caramanna (2024) investigated conversational agent architectures, comparing various LLM types and retrieval configurations.
- Florio (2024) implemented a LangChain-based RAG chatbot for corporate documentation, evaluating multiple vector database technologies.
- Salcuni (2025) applied RAG to the medical domain, improving LLM responses in hypertension care. The study used RAGAS to assess quality and relevance, focusing on personalization and accuracy.
- Nicoletti (2025) developed Essence Coach, a chatbot that integrates LLMs with the Essence software engineering standard. This system significantly outperformed generic LLMs like GPT-4o in domain-specific reasoning tasks.

3.3 RAG in the Digital Humanities

A growing body of research is exploring RAG applications within the digital humanities. One such example is the *iREAL* project, which applied RAG to interpret archival records from Aboriginal schools in Australia, demonstrating a careful balance between cultural sensitivity and historical accuracy (Callaghan and Vieira, 2025). Another initiative, *ValuesRAG*, focuses on cultural alignment in LLMs by integrating societal and demographic knowledge through retrieval-augmented contextual learning, experimenting with the *World Values Survey* dataset (Seo et al., 2025). In another case, the *Foggia Occupator Dataset* project applied a RAG model to post-WWII Italian periodicals, extracting information on political figures and stylistic traits (Ciletti, 2025).

Among the technical approaches explored in recent experiments on generative AI for digital scholarly editions, RAG emerges as a promising method for addressing challenges such as entity linking (EL) and the integration of external knowledge sources. Notably, RAG is recognized for its ability to mitigate hallucinations in named entity recognition (NER) and to enable the enrichment of text with information from structured databases or knowledge graphs (Pollin et al., 2025). For example, an experiment with the Regesta Imperii project demonstrates how knowledge bases, including Neo4j graph databases, are leveraged within RAG pipelines to improve accuracy in information extraction, entity normalization, and semantic annotation (Kuczera and Armbruster, 2024). Similarly, the editorial workflow developed for the Hugo Schuchardt Archive outlines a process that combines prompt engineering, human-in-the-loop oversight, and RAG toolchains to enhance the generation of TEI-compliant XML, supporting more explainable and modular processing pipelines (Pollin et al., 2023). These and other experiments underscore the need for standardized workflows, robust evaluation protocols, and systematic research into both the strengths and weaknesses of LLMs and related tools in the editorial process, while also advocating for thoughtful engagement with advanced computational methods in the humanities (Pollin et al., 2024). As digital editions become increasingly complex and interconnected with broader knowledge infrastructures, the relevance and application of AI technologies – such as RAG – are both expected and desirable to grow accordingly.

RAG methodologies are being adopted within the GLAM sector as well. In archival contexts, a smart assistant developed for querying the *Prozhito* digital archive of personal diaries combines text-to-SQL filtering, hybrid search, and automatic query reformulation, proving especially effective for historians and anthropologists without prior knowledge of database query languages (Sergeev et al., 2025). Meanwhile, in museum settings, a comparative evaluation of RAG techniques versus direct large-context input approaches – i.e., feeding the entire context at once to a language model – for answering multimodal questions about artworks demonstrated that large-context models generally give more accurate answers than RAG, at least for this task and dataset. However, RAG remains useful when information exceeds context window limits or when efficiency is important (Ramos-Varela et al., 2025).

Innovations in graph-based retrieval are also gaining momentum. Techniques combining structured supervision and chain-of-thought prompting have been used to map character relationships in early modern English historiography, thereby reducing the manual workload typically associated with historical data annotation (Fan et al., 2025). Related directions are being explored within cultural heritage institutions, as seen in the *CAT-IA* initiative, which integrates ArCo knowledge graph (Carriero et al., 2019) within a RAG system for provenance tracking, AI explainability (XAI), and structured metadata extraction (Barbato, 2025). Designed to streamline and enrich user interactions with the General Catalogue of Cultural Heritage *(Catalogo generale dei beni culturali)*, *CAT-IA* marks a notable stride in applying advanced digital technologies to promote accessibility and valorization of cultural assets.

A complementary, conceptual perspective emerges in a critical mapping of the theoretical contours of RAG within the broader landscape of archives, libraries, and cultural heritage – articulating not only the potential for RAG-augmented LLMs to enhance the precision, accessibility, and contextualization of information retrieval, but also foregrounding the social and infrastructural challenges inherent in such integration. This viewpoint encourages the field to reflect on both the affordances and the epistemic and ethical complexities introduced by RAG systems in digital humanities contexts (Di Marcantonio, 2024).

Finally, efforts to advance access to fragmented digital repositories – such as web archives – have increasingly adopted RAG methodologies. An illustrative bespoke prototype transforms keyword-based search into semantically guided question answering (Davis, 2025), sharing architectural parallels with the GNA QA system presented in the context of this thesis. Both systems prioritize semantic retrieval over lexical matching using dense embeddings – e.g., *E5* variants (L. Wang et al., 2024) – to interpret queries in context, employ structured text processing pipelines to reduce noise in source materials, and apply optimized chunking strategies for retrieval accuracy. Crucially, these studies highlight RAG's potential to transform scattered and heterogeneous resources – whether web archives or catalographic procedures – into coherent, accessible knowledge through context-aware synthesis.

3.4 Future Directions

Ongoing research is rapidly pushing the frontiers of RAG, opening up new avenues that extend well beyond traditional information retrieval into domains such as scientific research and the digital humanities. Among the most promising innovations is the use of synthetic corpora to bolster the robustness and generalizability of RAG systems, particularly in low-resource or specialized domains where annotated data is scarce (Bor-Woei, 2024). This strategy improves retrieval accuracy while addressing longstanding issues of bias, coverage, and representativity in humanities corpora.

RAG is also at the core of a new wave of applications that automate and enhance scholarly practices. In scientific research, advanced RAG frameworks – including agentic systems like PaperQA (Lála et al., 2023) – are being leveraged to conduct systematic literature reviews, automate evidence synthesis, summarize emerging trends, and provide transparent citation recommendations. Particularly, these multi-stage architectures enable recursive reasoning and dynamic tool usage, often surpassing human-level performance in both retrieval and summary tasks (Skarlinski et al., 2024).

Despite these advances, several critical research challenges remain. There is an urgent need to develop domain-adapted and multilingual LLMs that can process not just text, but also multimodal data such as images, tables, and audiovisual materials – a key requirement for both scientific and cultural heritage applications. Future RAG systems should be able to retrieve and reason over heterogeneous, cross-domain sources, necessitating robust mechanisms for source evaluation, multimodal fusion, and trust calibration. The ongoing development of benchmarks and evaluation datasets, tailored to the peculiar needs of fields such as the digital humanities, is essential to guide progress and ensure methodological rigor (Yue, 2025).

Another major direction is the semantic enrichment of RAG pipelines through the integration of ontologies and knowledge graphs. Ontologies, as formal domain knowledge models, provide structured frameworks that enable more precise and explainable retrieval semantic coherence, and the inclusion of ethical dimensions in generative AI. Complementing this, knowledge graphs capture complex relationships and support context-aware multi-hop reasoning, improving accuracy, explainability, and cultural sensitivity of outputs. Current research and practical applications in this direction span a range of initiatives – from ontology-guided entity typing to the grounding of AI in explicit ethical and procedural knowledge, demonstrating that these semantic tools are essential for creating robust, context-aware, and transparent RAG systems, addressing challenges in fields as diverse as healthcare, engineering, scientific discovery, and enterprise knowledge management (Tiwari et al., 2025; Ludwig et al., 2025; Bran et al., 2024; Sharma et al., 2024; Xiao et al., 2024; Park et al., 2024; DeBellis, 2024; Franco et al., 2020).

In the specific context of the digital humanities, the accelerated adoption of AI is shaping a transformative future for scholarship, curation, and access to cultural heritage. The diverse case studies and technical innovations, discussed in Sec. 3.3, illustrate both the breadth of RAG's impact and the field's growing ambition. Across applications, from digital scholarly editions, to archival assistance and museum information systems, RAG is emerging as a pivotal enabler for addressing the limitations of traditional search and annotation by supporting context-aware, semantically rich, and explainable information access.

Looking forward, several converging trends and open challenges will define the evolution of RAG in the digital humanities. First, technical advances such as the integration of knowledge graphs, graph-based retrieval, and multimodal pipelines are driving improvements in semantic linking and annotation of historical, literary, and artistic materials. Second, the increasing complexity of digital scholarly editions and GLAM infrastructures is catalyzing demand for standardized, reproducible workflows, robust evaluation protocols, and domain-adapted benchmarks, ensuring that RAG methods are critically assessed and tuned for the nuanced needs of humanistic research.

At the same time, as digital repositories become ever more fragmented, the promise of RAG lies in its ability to synthesize heterogeneous, dispersed data – transforming scattered web archives, periodicals, and catalogues into accessible, contextualized knowledge spaces. Yet, this evolution also foregrounds critical conceptual and ethical questions. As highlighted by recent critical perspectives, it is essential to position RAG as an augmentative technology: one that enhances, but does not replace, established cataloguing, metadata, and interpretive practices. human interpretive oversight, transparency, and cultural sensitivity must remain central, particularly as RAG systems are increasingly relied upon for knowledge production and mediation in complex social and historical domains (Di Marcantonio, 2024).

In sum, the next phase of RAG's development in the digital humanities will require sustained interdisciplinary collaboration and critical reflection. Researchers and practitioners must continue to experiment with new strategies, but also engage deeply with the epistemic, social, and infrastructural complexities of integrating advanced AI into cultural knowledge management. Ultimately, RAG applications stand poised not only to offer improved access to information, but they also invite a reimagining of the relationship between artificial intelligence and cultural knowledge production, fostering tools that augment – not displace – human creativity and understanding.

Chapter 4

Case Study: Question-Answering System for GNA

4.1 Geoportale Nazionale per l'Archeologia (GNA)

Geoportale Nazionale per l'Archeologia (GNA) (Mic, 2019) serves as the central online hub for the collection, management, and dissemination of data generated by archaeological investigations carried out across Italy (Acconcia, 2023). Developed under the auspices of the Ministry of Culture (MiC), the project's primary goal is the creation of a dynamic archaeological map of the national territory, which is easily updatable over time, openly accessible, and designed for reuse and integration across multiple institutional and disciplinary contexts (Falcone et al., 2023).

The inception of the GNA traces back to a 2014 *Memorandum of Understanding* signed by the Ministero dei Beni e delle Attività Culturali e del Turismo (MiBACT) – specifically the Segretariato Generale, the Direzione Generale per le Antichità (DG-Ant), and the Consiglio Nazionale delle Ricerche (CNR). This agreement laid the groundwork for a national platform dedicated to the safeguarding and enhancement of cultural heritage through integrated digital infrastructure. However, it was the establishment of the Istituto Centrale per l'Archeologia (ICA) in 2016 that provided the structural and institutional foundation for the GNA. The ICA's mandate to define standards and promote digital archaeological databases gave renewed potential to the initiative, which culminated in the launch and formal presentation of the GNA at a ministerial venue in 2019 (Calandra, 2023).

Beyond being a data aggregator, the GNA serves as a dynamic knowledge base, collecting digital content from professional archaeologists – especially those engaged in preventive archaeology –, research groups, universities, and concession-holders. The platform also accommodates a variety of outputs, from QGIS-based vector data to reports, documentation packages, and datasets from academic and research contexts. Data publication in the GNA is managed with attention to quality standards, intellectual property, and open-access principles, supported by the assignment of DOIs and the use of Creative Commons licensing (CC-BY 4.0), ensuring both traceability and reusability (Acconcia, 2023; Falcone et al., 2023; Boi, 2023).

4.1.1 Purpose and Scope

As the official repository for all research activities in archaeology – particularly those related to public infrastructure projects – the GNA platform was established to provide a unified national access point to essential archaeological data gathered nationwide. This includes the interventions listed in Tab. 5, all conducted under the scientific supervision of the Italian Ministry of Culture (MiC) (Acconcia, 2023; Falcone et al., 2023).

Archaeological interventions	Description
Preventive archaeology reports	Data from excavations and surveys carried out ahead of construction projects (e.g., highways, railways, pipelines), often submitted by private firms or cultural heritage consultants.
Assisted scientific excavations records	Results from academic digs by universities or research institutions, including documentation of stratigraphy, finds, and site interpretation.
Accidental discoveries	Locations of fortuitous archaeological finds, such as during agricultural work or construction, reported to local heritage authorities. Typically include preliminary spatial data and descriptive reports.
Scheduled excavations	Long-term planned investigations, often at known heritage sites, including geospatial boundaries, uncovered structures, and findings.
Archaeological surveys	Surface survey data with GPS-tracked locations of finds, artifact scatters, and site features.
Cultural heritage GIS layers	External datasets from institutions (regional superintendencies, local governments, ICCD), e.g., maps of protected zones, risk maps, or site inventories.
Legacy data and digitized archives	Georeferenced digitizations of paper maps, notebooks, and archival records previously stored in non-digital formats, essential for integrating historical with current data.
Depository locations	Georeferenced storage locations of archaeological finds (museums, storerooms) associated with sites or interventions.
Remote sensing and aerial surveys	Drone imagery, LiDAR scans, or satellite data used to identify and map archaeological features not visible at ground level.
Paleontological sites	A specific level dedicated to paleontological sites is currently under study for future inclusion, aiming to protect this fragile heritage.

Table 5: Types of archaeological data sources integrated into the GNA.

These sources, once georeferenced and structured, are integrated into the GNA using standardized metadata and visualization protocols, to allow users to view, search, and analyze information in a spatially accurate and coherent manner (Boi, 2023; Acconcia, 2023).

4.1.2 Stakeholders and Intended Users

The development of the GNA saw significant acceleration during the COVID-19 pandemic, which provided both the urgency and institutional impetus toward the creation of a unified digital platform for managing archaeological data nationwide. This initiative built upon years of prior collaboration between key stakeholders, including the Istituto Centrale per l'Archeologia (ICA) and the Istituto Centrale per il Catalogo e la Documentazione (ICCD), who had already developed a cataloging structure to document archaeological assessments and identified sites within the Sistema Informativo Generale del Catalogo (SiGECweb) (Calandra, 2023; Boi, 2023). The pandemic underscored the limitations of purely textual cataloging and catalyzed a shift toward a more dynamic and geospatially grounded approach, leading to the adoption of a GIS-based framework better suited for preventive archaeology and territorial planning. The result was a consolidated national infrastructure designed not only to support compliance with cultural heritage protection regulations but also to enable data harmonization across previously fragmented practices (Acconcia, 2023).

The GNA is primarily intended for use by:

- Public administrators and government officials
- Professional archaeologists and cultural heritage consultants
- Stakeholders involved in public works, such as national infrastructure planners

For instance, major entities like TERNA (the national electricity grid operator), RFI (the Italian railway network), or the Milan Metro rely on the platform to assess archaeological constraints before launching construction projects. The platform helps them identify archaeological sites, deposits, and or protected areas that must be preserved. The GNA also supports compliance with European and Italian open data and transparency regulations, guaranteeing both civic access and the protection of intellectual property, as per national FOIA and EU directives[17] (Falcone et al., 2023).

Central to the system is a QGIS[18] template that standardizes data entry and visualization. This tool supports efficient integration of local information into the national infrastructure,

[17]The FOIA (Freedom of Information Act) Guidelines are documents issued by the Italian National Anti-Corruption Authority (ANAC) to clarify and guide the implementation of the right to generalized civic access in Italy. The guidelines – especially those from 2016 – define the limits and exclusions to access, as well as specify the publication and transparency obligations for public administrations.
See more at https://foia.gov.it/normativa.

[18]QGIS is a free, open-source Geographic Information System (GIS) software used for creating, managing, and analyzing geospatial data.

offering users a unified territorial overview. It enables the comparison of diverse archaeological records, improves the quality of evaluations, and promotes transparency across institutional workflows. Thanks to its open-source foundation and modular structure, the GNA continues to evolve based on user feedback, maintaining a shared national standard while accommodating diverse local contributions (Calandra, 2023; Boi, 2023).

4.1.3 User Manual and Operational Support

To guide users in correctly navigating the system, a collaboratively maintained user manual (*manuale operativo*) is made available online through a MediaWiki environment hosted on the GNA server (GNA, 2024). This living document offers structured instructions on all aspects of data input, visualization, and management within the GNA platform.

The manual offers step-by-step instructions for compiling and submitting data using the QGIS template, including the creation and editing of project modules (MOPR), the documentation of archaeological sites and events (MOSI), and the proper use of supporting layers such as risk maps or thematic overlays. Each section of the manual is designed to be accessible both to GIS beginners and to experienced professionals, offering annotated screenshots, workflow examples, and direct links to downloadable resources. A notable feature of the operational manual is its integration with the GNA QGIS plugin, which allows users to directly download standardized data layers – such as archaeological risk assessments, site boundaries, or previous project records – into their local GIS environment (Gabucci, 2023).

In addition to the written documentation, the GNA provides ongoing operational support through a dedicated Help Desk service, coordinated by Ada Gabucci.[19] Users encountering technical challenges or seeking clarification on data entry procedures can contact the Help Desk for personalized assistance. This direct support, together with the collaborative and evolving nature of the manual, fosters a strong community of practice, encouraging the sharing of expertise and continual improvement of the platform's tools and resources.

[19] Ada Gabucci is a specialist in Roman-period archaeology, with expertise in stratigraphic methods, northern Italian material culture, and the structuring of archaeological data. She has over thirty years of experience consulting for public institutions, including the Italian Ministry of Culture (ICCD, ICA, DG-ABAP), its regional branches, the Veneto Region, and several universities, including Trieste, Venice, Verona, Bologna, Genova, and Pisa. Her work also encompasses cultural heritage cataloguing, ministerial regulations, and the design of complex Geographic Information Systems.
See: https://web.archive.org/web/20250724081422/https://conf24.garr.it/it/speaker/ada-gabucci.

4.2 Proof of Concept

In response to the challenges users face in quickly locating relevant information when accessing and navigating the GNA operative manual, as well as the high volume of inquiries received by the Help Desk, a need emerged for a smarter and more efficient support solution. To address this, we developed an information system in the form of a question-answering system designed to assist users directly and reduce the Help Desk's workload. Based on the current state of AI, ML and DH methodologies – as discussed in Chap. 3 and especially Sec. 3.2 – RAG combined with NLP was chosen as the most effective approach. This technology enables the chatbot to dynamically retrieve relevant information, which serves as an augmented knowledge base, allowing it to generate precise, context-aware, and up-to-date answers tailored to user queries.

4.2.1 Functional Requirements

Functional requirements define what the system must do to deliver value to users and stakeholders:

- **Natural language understanding (NLU):** The system must interpret user queries phrased in natural language, supporting diverse question types (factoid, list, explanatory, etc.) and handling both simple and complex multi-part queries.
- **Information retrieval:** The system must retrieve relevant passages or document segments from the GNA knowledge base, using vector similarity search over chunked content.
- **Context-aware answer generation:** The system must synthesize coherent, context-aware answers using RAG, drawing from retrieved passages and maintaining reference to original sources.
- **Source attribution and citation:** Answers must include traceable citations (e.g., URLs) to ensure transparency and support verification.
- **Conversational memory:** The system must retain context from previous exchanges to handle follow-up questions and maintain dialogue continuity within a session.
- **Multilingual support:** The chatbot must process and generate responses in Italian, with potential extensibility to other languages.

- **User feedback collection:** The system must provide mechanisms for users to rate responses and submit qualitative feedback, enabling ongoing evaluation and improvement.

- **Interactive user interface:** Users must be able to input queries and view answers through an accessible web interface, including features such as clickable citations, feedback buttons, and session management.

4.2.2 Non-Functional Requirements

Non-functional requirements define how the system should operate to ensure quality, usability, and maintainability:

- **Accuracy and relevance:** Answers must be factually correct, directly address user queries, and reference up-to-date information.

- **Performance and scalability:** The system must deliver responses with low latency (target average retrieval and response time inferior to 1 second per query) and scale to support multiple concurrent users.

- **Robustness and reliability:** The system should gracefully handle invalid queries, errors, and resource constraints without crashing.

- **Transparency and traceability:** Every generated answer must cite its sources clearly. The underlying process for retrieval should be auditable.

- **Security and privacy:** The system must securely handle sensitive data. User interactions should be anonymized, and no personally identifiable information should be stored.

- **Maintainability and extensibility:** The architecture must support modular updates (e.g., changing retrieval strategies), and facilitate maintenance, debugging, and future enhancements.

- **Resource efficiency:** The solution must operate efficiently within the limits of available hardware, minimizing memory and compute consumption, especially for cloud deployment scenarios without GPU access.

- **User accessibility:** The web interface must be usable by non-technical users and meet accessibility standards (e.g., clear labeling, visual feedback, keyboard navigation).

- **Continuous evaluation:** The system must support automated and human-in-the-loop evaluation methodologies, generating reports on retrieval accuracy, answer quality, and user satisfaction over time.

(Abu Shawar and Atwell, 2007; Arslan et al., 2024; Gupta et al., 2024)

The following chapter details the methodological framework and practical steps undertaken during the development of the system, providng in-depth explanation of the design choices, technical architecture, data preparation, implementation and evaluation processes.

Chapter 5

Methodology

This chapter details the methodological workflow for designing and implementing the GNA QA system. The system leverages a RAG pipeline tailored to the Geoportale Nazionale per l'Archeologia (GNA) knowledge base (KB). It comprises modular components for data acquisition, preprocessing, retrieval, generation, feedback collection, and evaluation. The methodology evolved through iterative development: beginning with a prototype built on LangChain and advancing to a full-scale system with custom components optimised for resource efficiency and multilingual support.

5.1 Prototype

The initial prototype served as a proof-of-concept integrating core RAG elements using off-the-shelf tools (Mishra, 2024; Akkiraju et al., 2024). The pipeline combined:

- a CSV-based knowledge base,
- a FAISS vector store,
- the Mistral NeMo large language model,
- and a Streamlit-based interface.

The prototype uses LangChain for managing prompts, memory, and asynchronous streaming.[20] The interface allows users to input NL queries in Italian and receive fluent, context-aware responses.

Evaluation was conducted via a dual approach:

[20]LangChain is an open-source framework designed to simplify the development of applications powered by LLMs. For further details and practical examples, consult the official documentation at https://python.langchain.com/docs/introduction/.

1. **Human Assessment**, using a 5-point Likert scale to rate and annotate consistency, fluency, completeness, and relevance (Abeysinghe and Circi, 2024);

2. **LLM-as-a-Judge**, where GPT-3.5 is used to auto-evaluate responses via few-shot prompting.[21]

Challenges included limited scalability, resource inefficiency, lack of chunk-level metadata control and the absence of standardised evaluation methods and benchmarks. These findings informed the redesign of the full system.

5.2 Full-scale implementation

The full system was re-engineered from scratch to support dynamic, scalable document ingestion, contextual retrieval, and high-precision answer generation using open-source LLMs. All LangChain dependencies were removed in favor of custom Python implementations to improve modularity, debugging transparency, and flexibility in processing. The final architecture includes:

- a custom knowledge base construction module, which includes sitemap generation and web-crawling,
- semantic chunking and metadata enrichment,
- LLM-based vector embeddings,
- a FAISS vector store for retrieval,
- a generation module with open-source Mistral NeMo model,
- generative response with inline citation handling,
- a reactive front-end interface built on Streamlit,
- and a feedback management system.

[21] See also (Svikhnushina and Pu, 2023) for method inspiration.

5.3 Data Acquisition and Preprocessing

5.3.1 Sitemap Generation

The sitemap is constructed via a focused breadth-first crawler targeting the MediaWiki documentation (https://gna.cultura.gov.it/wiki) of the GNA (Mic, 2019). The crawler:

- starts at the root node (`Pagina_principale`);
- follows internal links matching `/wiki/index.php/`, excluding namespaces such as `Special:`, `User:`, or `Talk:`;
- removes query parameters to avoid duplicates;
- applies a polite crawling policy (1-second delay, custom user-agent header);
- imposes crawl depth (max 10) and page limits (max 200 pages);
- and generates a structured sitemap in XML format.

HTML is parsed using BeautifulSoup, isolating the main content via focusing on the `div` with `id="mw-content-text"`, and excluding sidebars and footers. The output is serialised into an XML file (`GNA__sitemap.xml`) including last-modified timestamps, priority, and change frequency.

This sitemap serves as the foundation for subsequent document crawling and chunking.

5.3.2 Document Crawling and Chunking

Crawling retrieves the URLs listed in the sitemap. The system fetches HTML content asynchronously, applying retries and concurrency limits. During parsing:

- extraneous HTML elements are stripped;
- headers (`h1-h6`), paragraphs, tables, and images are preserved in semantic order;
- a hierarchical structure is reconstructed to retain navigational breadcrumbs.

Content is then chunked using a sliding window strategy of max-512 characters per chunk and 128-character overlap. Each chunk includes metadata such as `source URL`, `page` `title`, `section headers`, `chunk ID`, `content type` (text, table, image), `keywords`, `named entities`.[22]

NER is performed using spaCy (`it_core_news_md`), while keywords are extracted with KeyBERT multilingual model (*paraphrase-multilingual-MiniLM-L12-v2*). Tables are chunked as standalone elements, and image references are retained for OCR via Tesseract when applicable.

This contributes to creating the knowledge base, which is stored in a JSON file (`dataknowledge_base.json`) for subsequent embedding, retrieval and generation tasks.

5.3.3 Vector Embeddings

Document chunks are converted into dense vector representations using the *intfloatmultilingual-e5-large* model from Sentence Transformers. This model was selected for its multilingual encoding capabilities and strong performance in semantic retrieval tasks, making it suitable for the predominantly Italian knowledge base while allowing also cross-lingual queries.

Text chunks are processed in batches and transformed into L2-normalised embeddings to ensure vector magnitudes are uniform. Embeddings are cached locally to avoid redundant computation across runs. The normalised vectors are stored in a FAISS `IndexFlatIP` index, which performs brute-force nearest neighbor search using the inner product (dot product) as metric. Alongside the vector index, a separate metadata store is maintained, linking each embedding to its corresponding chunk through a unique identifier. This separation enables efficient similarity search in FAISS while preserving quick access to rich metadata such as source URL, document structure, and content type for downstream processing.[23]

These embeddings and their associated metadata form the foundation for the retrieval stage, where user queries – also encoded with the same *multilingual-e5-large* model, to guarantee that both queries and documents share the same normalised vector space – are matched against the stored vectors to identify the most semantically relevant chunks for answer generation.

5.4 Candidates Retrieval

When a user submits a query, it is embedded using the same encoder to ensure vector space consistency. The FAISS index, configured for inner-product similarity, is queried to return the

[22] Output: 835 structured chunks saved in `data/chunks_memory.json`.
[23] Output: FAISS index stored in `.faiss_db`, linked with its metadata.

top-k candidate chunks.[24] Retrieval is executed entirely within the vector space to maximise speed and maintain consistent scoring across CPU-based deployments. The retrieved results are enriched with their stored metadata, which includes source URL, document title from the original web section, hierarchical section headings, and content type (text, table, image). Candidates are then grouped by provenance, ensuring that related chunks from the same source URL are passed together into the generation stage, thus improving contextual coherence, supporting inline citation, and reducing redundancy.

To further improve factual density, a lightweight filtering heuristic is applied to penalise very short or contextless chunks, deprioritising fragments that lack substantive information. The grouped and filtered candidates are returned as structured context objects, ready to be consumed by the answer generation module.

This retrieval framework serves as the baseline for subsequent ablation studies described in the evaluation phase (cf. subsection 5.4.1), where alternative retrieval strategies and scoring variations are tested against this reference implementation.

5.4.1 Experimental Setup for Ablation Studies

To systematically evaluate the contribution of various retrieval strategies, an experimental setup was implemented following an ablation logic. In this context, ablation refers to the process of selectively removing or isolating individual components to measure their specific impact on overall performance. None of the evaluated approaches was integrated into the main system; instead, each was tested independently to allow for a broader performance comparison, as detailed in Sec. 5.6.

The following retrieval configurations were tested:

- **Dense:** uses dense vector embeddings for retrieval. It wraps the FAISS vector database (via `VectorDatabaseWrapper`) and returns the top-k chunks ranked by embedding-based similarity scores. Queries are cached using a normalized MD5 hash to avoid recomputation, and batch querying is supported.

- **BM25:** employs the BM25 algorithm for traditional keyword-based retrieval. The index is built over concatenated fields – `title`, `keywords`, `headers_context`, and `document` – from the same metadata store. Text is preprocessed for Italian (stopword removal, stemming, and clitic/apocope handling), then tokenized. Batch mode is also supported.

[24] The default value of `k=5` was determined empirically to balance response quality and token constraints.

- **Hybrid:** combines the dense retriever and BM25 to balance semantic and lexical matching. We tested two fusion strategies:
  - **Weighted RRF:** aggregates ranks from both retrievers using Weighted Reciprocal Rank Fusion (RRF). The fusion score for document d is computed as

    $$\frac{w_{\text{dense}}}{k + \text{rank}_{\text{dense}}} + \frac{w_{\text{sparse}}}{k + \text{rank}_{\text{sparse}}}, \quad k = 60$$

    with default weights $w_{\text{dense}} = w_{\text{sparse}} = 1.0$, `candidate_k` $= 50$, and `top_k` $= 5$.
  - **Score-blend:** merges normalized scores from both retrievers using a custom blending function that allows for fine-tuning the influence of each method:

    $$S_{\text{norm},d} = \frac{S_d - \min(S_d)}{\max(S_d) - \min(S_d)}, \quad S_{\text{norm},s} = \frac{S_s - \min(S_s)}{\max(S_s) - \min(S_s)}$$

    $$S_h = S_{\text{norm},d} + \alpha \cdot S_{\text{norm},s}$$

    where α controls the influence of the sparse retriever ($w_d = w_s = 1$, $k = 60$). (X. Wang et al., 2024) Unlike RRF, this method requires compatible score scales between retrievers.[25]

What to pick?

The choice between Weighted RRF and Score-blend depends on the specific retrieval context. RRF is particularly suitable when score scales between retrievers are incompatible or unstable, as its rank-based aggregation is less sensitive to scale differences and prioritizes consensus across retrieval methods. Conversely, Score-blend is more appropriate when per-query scores are reliable, as it allows fine-grained control over the relative influence of dense and sparse components, enabling more tailored retrieval behaviour.

Additional ablation experiments included:

[25] Edge-case behaviors include: **a)** *Document appears in only one list*: in Score-blend, it keeps its normalized score (the missing side contributes zero), while in RRF, it receives only one rank term and typically ranks lower than documents appearing in both lists; **b)** *All scores equal in a list*: in Score-blend, normalization produces identical values, so BM25 adds little influence, while RRF still differentiates by rank order.

Query Rewrite

Query Rewrite is a retrieval enhancement technique designed to reformulate the user's input query in order to increase the likelihood of retrieving relevant documents. In the context of these experiments, query rewriting is realised as a multi-strategy process that generates alternative query variants through complementary transformations, each targeting different aspects of query understanding and expansion (Li et al., 2024). Specifically, the approach integrates:

- **Core Content Extraction (CCE)** – a sequence-to-sequence transformation, using the *it5-small model*, that rewrites the query to capture its essential informational content while removing peripheral terms.
- **Keyword Expansion (QE)** – key terms are extracted with KeyBERT and enriched with n-gram combinations and synonym substitutions to introduce semantically related expressions.
- **General Query Rewriting (GQR)** – this process is based on spaCy functionalities of lemmatization and stopword removal, producing a normalized lexical form of the query.
- **Pseudo-Relevance Feedback (PRF)** – top-ranked documents from an initial retrieval pass are analysed to extract additional high-frequency terms not present in the original query, which are then appended to form an expanded query.
- **Query Decomposition** – conjunctive or disjunctive queries are split into simpler sub-queries, each covering a distinct semantic aspect.

These strategies can be applied individually or in combination (`strategy="all"`), producing a set of reformulated queries. Each reformulated query is submitted to the base retriever (Dense, BM25, Hybrid variants) to have the resulting candidate documents.

Rerank

Rerank is a post-retrieval refinement process that reorders an initial set of candidate documents according to a more precise relevance estimation. In our implementation, this stage operates as a wrapper over a base retriever (Dense, BM25, or Hybrid) and uses a transformer-based cross-encoder model (*cross-encoder/ms-marco-MiniLM-L-6-v2*) to jointly encode the query and each candidate document, producing a contextual relevance score. Unlike the base retriever, which typically evaluates query–document similarity using independent embeddings or lexical term matching, the cross-encoder considers full cross-attention between query and document tokens, enabling a richer semantic alignment.

At runtime, the reranker receives the top-N candidates from the base retriever (with N controlled by `max_rerank_candidates`, set to 50), tokenizes each query–document pair, and performs inference in batches with mixed-precision support when available. The raw model outputs are interpreted as relevance scores, and candidates are sorted accordingly, producing a final top-k list with improved ordering accuracy.

5.5 Generation

The generation phase employs Mistral NeMo,[26] an open-source LLM accessible via a dedicated Mistral API and hosted independently. The choice of this model was driven by multiple factors:

- open-source availability and permissive license,
- strong performance on multilingual tasks,
- low latency and high throughput on CPU hardware,
- availability of an official API for direct deployment integration.
- ability to handle long context windows – e.g., up to 128 thousands tokens, which is sufficient for processing multiple retrieved chunks.

Among available open-source LLMs evaluated – e.g., LLaMA 3, OpenAI, Falcon –, only Mistral NeMo satisfied all criteria in terms of language coverage, response control, and reproducibility, while also providing infrastructure for model fine-tuning and evaluation in research contexts.[27]

The generation module is designed to produce fluent, context-aware answers with inline citations. It uses a custom prompt template that includes system instructions (see subsection 5.5.1), user query, top-k retrieved chunks (grouped and cited), chat history and memory (to support context-aware follow-up).

The API request includes temperature, top-p, and max token constraints, with defaults of:

- temperature = 0.3, to ensure factuality;

[26] https://web.archive.org/web/20250803120348/https://mistral.ai/news/mistral-nemo.

[27] For an overview between open-source and proprietary solutions, see: *Open Source vs. Proprietary LLMs: A Comprehensive Comparison* (2025).

- top-p = 0.9, to control diversity;
- max-tokens = 512, to limit response length.

Responses are post-processed to verify the inclusion of inline citations, language alignment for Italian, and basic formatting (e.g., numbered citations, paragraph boundaries).

5.5.1 Prompt Engineering Techniques

The system uses structured prompt engineering to ensure accurate, traceable, and contextually coherent answers. The prompt template is dynamically generated with the following components:

System Instructions, Boundaries and Constraints

A custom system message (Fig. 2) is injected at the top of the prompt to guide the model's behavior. This message instructs the system to enforce neutrality in its answers, prioritise relevant and verifiable information, and include inline citations in square brackets that correspond to metadata entries. It also explicitly discourages hallucinations and speculative responses.

```
system_content = """
    Sei un assistente virtuale incaricato di rispondere a domande sul
    ↪ manuale operativo del Geoportale Nazionale per l'Archeologia
    ↪ (GNA), gestito dall'Istituto Centrale per il Catalogo e la
    ↪ Documentazione (ICCD).

    Segui sempre queste regole:
    1. Non rispondere a una domanda con un'altra domanda.
    2. Rispondi **sempre** in italiano, indipendentemente dalla lingua
    ↪ della domanda, a meno che l'utente non richieda esplicitamente
    ↪ un'altra lingua.
    3. Cita le fonti utilizzando la notazione [numero] dove:
        - Le fonti sono fornite nel contesto della domanda e sono numerate
        ↪ in ordine crescente;
        - Usa numeri diversi per fonti diverse;
        - Non includere mai l'URL nel corpo della risposta;
    4. Alla fine della risposta, aggiungi un elenco di riferimenti con il
    ↪ seguente formato, su righe separate:
        [ID] URL_completo
    5. Se non hai informazioni sufficienti per rispondere, rispondi "Non
    ↪ ho informazioni sufficienti".

    Le tue risposte devono essere sempre:
    - Disponibili, professionali e naturali
    - Grammaticalmente corrette e coerenti
    - Espresse con frasi semplici, evitando formulazioni complesse o
    ↪ frammentate
    - Complete e chiare, evitando di lasciare domande senza risposta
    """
```

Figure 2: System prompt specifying assistant constraints and response instructions.

Dynamic Citation Handling

Each chunk passed to the LLM is numbered and grouped with its metadata (title, URL). When generating a response, Mistral is instructed to cite only the chunks used, ensuring traceability. Post-processing checks for unmatched citations or unreferenced metadata.

This modular prompting strategy proved crucial for maintaining factual consistency while supporting multilingual input and long-form reasoning.

5.6 Evaluation Protocol

Evaluation was conducted across two dimensions:

- **Quantitative**, using metrics such as Recall (R@), Mean Reciprocal Rank (MRR@), Normalized Discounted Cumulative Gain (nDCG@), Average Precision (AP@), and Latency to assess retrieval performance;
- **Qualitative**, gathering user feedback on response relevance, fluency, completeness, and usability.

This dual perspective follows the recognition that effective RAG-based chatbots require not only accurate retrieval and generation but also operational efficiency and adaptability (Akkiraju et al., 2024). The evaluation process is designed to be iterative, allowing for continuous refinement of the system based on user interactions and performance metrics.

The evaluation faced several structural limitations:

- **Absence of a golden standard:** there was no authoritative, verified set of responses to serve as an absolute accuracy benchmark.
- **No baseline system:** internal institutional tasks lacked comparable legacy solutions or predefined benchmarks.
- **Lack of real users or domain experts:** the system was initially developed without direct input from actual users, limiting the applicability of findings. Human evaluation by real end-users was integrated later in the process.
- **Limited applicability of traditional automated metrics:** common algorithmic measures such as BLEU, ROUGE, and METEOR are widely used in text generation but have been shown to be ineffective in dialogue and QA contexts (Deriu et al., 2020; C.-W. Liu et al., 2016).

Given these limitations, ... aligning with recommendations from recent RAG evaluation literature.

In this phase several challenges arose due to the absence of standardised evaluation methods and benchmarks. The lack of a golden standard – a verified set of responses representing objective truth – made it difficult to comprehensively assess accuracy. Additionally, there were no predefined benchmarks for internal institutional tasks, nor a baseline system for comparison, as real users or domain experts were unavailable for testing at this stage. While taxonomies for LLM evaluation exist (Guo et al., 2023), they are not universally applicable to chatbot and QA assessment. Moreover, commonly used algorithmic metrics such as BLEU, ROUGE, and METEOR are considered ineffective for dialogue systems (Deriu et al., 2019; Liu et al., 2016), reinforcing the need for human evaluation as a complementing approach (Mehri and Eskenazi, 2020; Abeysinghe, 2024).

Akkiraju (2024) fifteen RAG pipeline control points, empirical results on accuracy-latency tradeoffs between large and small LLMs.

5.6.1 Datasets

Two synthetic evaluation sets were created:

- **Single-hop dataset:** 508 queries designed to elicit single-document answers, each with a single gold document. This set tests the system's ability to retrieve and generate answers based on isolated chunks of information.
- **Combined dataset:** 400 additional queries that require multi-hop reasoning, where answers are derived from multiple documents (2-4 chunks), for a total of 908 queries entries. This set evaluates the system's capacity to integrate information from various sources and generate coherent, contextually rich responses.

Tasks:

5.6.2 Metrics

(2016) for not using automed metrics like BLEU, ROUGE, etc.

- R@5 (Recall at 5): Measures the fraction of relevant documents retrieved in the top 5 results.

- MRR (Mean Reciprocal Rank): Evaluates the rank position of the first relevant document.
- nDCG@5 (Normalised Discounted Cumulative Gain at 5): Measures ranking quality, emphasising higher placement of relevant results.
- AP@5 (Average Precision at 5): Computes the average of precision values at each relevant document within the top 5.
- Latency: Average retrieval time per query (in seconds).

5.6.3 Experimental Setup for Ablation Studies

Describe the retrieval configurations tested (Dense, BM25, Hybrid + Weighted RRF, Hybrid + Score-blend) and how they differ.

- Dense: Uses dense vector embeddings for retrieval.
- BM25: Employs the BM25 algorithm for traditional keyword-based retrieval.
- Hybrid + Weighted RRF: Combines dense and BM25 results using a weighted Reciprocal Rank Fusion (RRF) strategy.
- Hybrid + Score-blend: Merges scores from both methods using a custom blending function.
- Query Rewrite: Applies query rewriting to improve retrieval relevance.
- Rerank: Applies a reranking step to refine the top results based on relevance.

Ablations tested the effect of enabling/disabling query rewriting and reranking across both datasets, measuring R@5, MRR, nDCG@5, AP@5, and latency. The design aligns with Akkiraju et al.'s emphasis on control-point evaluation, ensuring that performance changes can be tied to specific retrieval and orchestration steps rather than observed only at the output level.

Note whether query rewriting and reranking were applied in each configuration, as shown in your table.

5.7 User Interface

The user interface (UI) is implemented using Streamlit, selected for its fast prototyping capabilities, built-in support for asynchronous processing, and integration with Python-based NLP components. The front end acts as the primary touchpoint between users and the GNA QA system, displaying generated answers with inline citations, and collecting user feedback.

Why the UI matters methodologically (collecting user feedback, enabling qualitative eval).

One screenshot max;

5.7.1 Streamlit User Experience

The Streamlit app is organised into three main areas:

1. **Sidebar:** Contains MiC refernce, institutional links to the GNA documentation, and contextual help describing the assistant's capabilities. It also provides functional controls including a `Clear Chat History` button to reset the session (`st.session_state.chat_history`) and a `Download Feedback` button for exporting user queries and system's responses.
2. **Main interface:** Provides a natural language input field (`st.chat_input`) for querying the assistant. It displays the chat history, including:
   - user messages,
   - assistant responses (formatted via `st.chat_message`),
   - feedback buttons (3 points Likert-scale) for each assistant reply.
3. **Session features**:
   - Chat history is limited to the most recent 10 exchanges (`MAX_HISTORY`), which are stored and updated in `st.session_state`;
   - Feedback from individual message indexes is stored as a set (`st.session_state.feedback_given`), allowing the system to prevent duplicate ratings and dynamically update UI feedback (e.g., "Valutazione registrata con successo.").

5.7.2 Session State Management

To ensure smooth, stateful interactions and to minimise computational overhead, the system makes extensive use of `st.session_state`. This approach allows chat memory for both user and assistant messages to persist, enables caching of API results – including responses generated by Mistral and citation mappings – and facilitates the tracking of feedback.[28] In addition, the system incorporates Python's `asyncio` and `concurrent.futures` modules to support non-blocking retrieval and generation. For each interaction with the language model, a new event loop is created to maintain compatibility with Streamlit's execution model, and the query to the model is run in a thread-safe environment using a `ThreadPoolExecutor`. This pattern guarantees UI responsiveness and provides a fluid user experience even under high network latency or API response times.

5.8 Feedback Loop

To support iterative improvement of the assistant and promote user engagement, the system integrates an interactive feedback module that allows users to rate each answer directly within the Streamlit interface. This design is intended to support continuous quality assessment and transparent evaluation of LLM-generated content.

5.8.1 Collection

Each assistant response is immediately followed by three clickable UI buttons in the form of a 3-point Likert scale, enabling users to provide a quick evaluation:

- 1 star $\star$ - Poor: The answer is incorrect, incomplete, or irrelevant.
- 2 stars $\star\star$ - Fair: The answer is partially correct but lacks clarity or depth.
- 3 stars $\star\star\star$ - Good: The answer is accurate, complete, and well-structured.

This mechanism is implemented using Streamlit's interactive widgets. Once a rating is submitted, the system prevents duplicate feedback using an in-memory tracking set (`st.session_state.feedback_given`). The interface then displays a confirmation message (e.g., "*Valutazione registrata con successo.*"), improving transparency.

5.8.2 Storage and Export

Feedback is stored locally in a SQLite database (`feedback.db`) with the following schema:

[28] For more details on Streamlit session state management, see https://docs.streamlit.io/.

```
CREATE TABLE feedback (
      id INTEGER PRIMARY KEY AUTOINCREMENT,
      timestamp TEXT NOT NULL,
      message_index INTEGER NOT NULL,
      question TEXT NOT NULL,
      answer TEXT NOT NULL,
      rating INTEGER NOT NULL
);
```

Figure 3: Schema of the feedback's SQL databse.

This structure supports reproducibility and traceability by maintaining a clear mapping between: the user's query string, the generated response, the rating score (1-3) and the associated timestamp (i.e., time of submission). All records are saved with minimal overhead, using parameterised SQL insertions and transaction-safe commits.

To ensure long-term preservation and collaborative accessibility of user feedback, the system implements a mechanism for periodic synchronization of collected feedback with a persistent repository. This setup enables version control over user interaction logs, supports iterative evaluation by external reviewers, and enables rollback and comparison across model updates.[29]

From the sidebar, users can export all feedback as a `.csv` file using the *"Esporta feedback"* button. This functionality is powered by the `export_feedbacks()` function, which queries the database and converts it to a downloadable format using Pandas.

This export can be used by researchers, developers, or project coordinators to assess the assistant's performance over time.

5.9 Resource and Deployment Constraints

Deploying a multilingual RAG workflow with different language models integrated and semantic search poses non-trivial challenges for environments lacking access to GPUs or large memory

[29]This implementation is intended for controlled research use only. For production environments, secure alternatives such as authenticated APIs and hardened database infrastructures are recommended to ensure data safety and compliance with privacy standards.

allocations. To ensure the GNA QA system remains responsive and cost-efficient, especially for open usage, several optimization strategies were integrated into the pipeline.

5.9.1 Memory Management

To reduce RAM usage during embedding, retrieval, and generation, the following practices were adopted:

- **Garbage collection routines:** Explicit calls to Python's garbage collector (`gc.collect()`) were introduced to free unused memory between embedding and response generation steps.
- **Batch processing:** Document chunks are processed in batches to minimise memory overhead, especially during retrieval.
- **Lazy loading:** All embeddings are computed once and stored to disk. On app startup, only metadata is loaded, and the FAISS index is memory-mapped to reduce RAM footprint.
- **Asynchronous processing:** Streamlit's async capabilities allow the UI to remain responsive while background tasks are executed, preventing memory spikes during long-running operations.
- **Cache clearing policies:** `st.session_state` objects are pruned after each session or on manual reset by the user to prevent memory bloating during prolonged use.

5.9.2 Computational Constraints Mitigation

Given the constraint of CPU-only deployments on platforms like Streamlit Cloud, the following strategies were implemented:

- **Model selection:** The use of *intfloat/multilingual-e5-large* for embedding provides a trade-off between semantic accuracy and compute efficiency, even without GPU acceleration.
- **API offloading:** Offloading generative tasks to an external API prevents the local system from being overloaded and allows scaling independently of frontend performance.

- **Timeout and fallback handlers:** If the generation request exceeds 10 seconds or fails – e.g., due to API rate limits –, the app returns a graceful fallback response, allowing users to retry or simplify their query without crashing the app.
- **Asynchronous I/O:** For embedding, retrieval, and response generation, asynchronous requests reduce UI freezing and ensure smoother user experience even under high latency conditions.

5.10 Ethics and Data Governance

The GNA QA system is designed with a strong emphasis on ethical considerations and data governance, particularly in the context of cultural heritage and public information. Key principles include:

- **Transparency:** The system provides clear information about its data sources, methodologies, and limitations, ensuring users understand how answers are generated and the provenance of information.
- **Privacy:** The system avoids processing personally identifiable information (PII) and ensures that all data used is publicly available or explicitly licensed for use in research and educational contexts.
- **Licensing:** All components, including the knowledge base, models, and software libraries, are selected based on permissive licenses that allow for academic and non-commercial use, ensuring compliance with legal and ethical standards.
- **Auditability:** The system maintains logs of user interactions, feedback, and system performance, enabling ongoing evaluation and improvement while respecting user privacy and data protection regulations.

These principles guide the development and deployment of the GNA QA system, ensuring it serves as a responsible and ethical tool for cultural heritage public engagement. Provenance, PII avoidance, licensing, auditability.

Chapter 6

Results

6.1 Ablation Studies

			SINGLE-HOP					**COMBINED (single+multi-hop)**				
Method	**Query rewrite**	**Rerank**	R@5	MRR	nDCG@5	AP@5	Latency	R@5	MRR	nDCG@5	AP@5	Latency
Dense	✗	✗	67.51	47.04	52.18	47.04	0.29	50.78	48.21	53.70	34.98	0.19
	✓	✗	58.07	36.76	42.09	36.76	5.98	42.64	35.41	41.32	26.24	3.10
	✗	✓	45.47	25.41	30.36	25.41	1.30	34.57	27.71	32.90	19.48	0.82
	✓	✓	52.55	31.66	36.87	31.66	4.76	38.16	30.45	36.0	22.51	3.46
BM25	✗	✗	65.15	43.56	48.98	43.56	**0.001**	53.09	**51.23**	**57.13**	35.50	**0.001**
	✓	✗	57.87	37.37	42.50	37.37	1.98	46.65	42.15	48.33	29.38	1.20
	✗	✓	45.47	25.41	30.36	25.41	1.30	34.57	27.71	32.90	19.48	0.82
	✓	✓	43.89	25.87	30.35	25.87	10.02	35.18	30.33	35.68	20.59	4.96
Hybrid												
+ *Weighted RRF*	✗	✗	69.68	46.72	52.49	46.72	0.33	**53.98**	50.93	56.92	36.15	0.32
	✓	✗	57.48	37.48	42.48	37.48	4.38	43.52	38.41	44.10	27.68	3.21
	✗	✓	43.50	24.70	29.33	24.70	1.67	33.06	26.36	31.26	18.70	0.87
	✓	✓	38.58	21.14	25.47	21.14	6.51	29.98	23.95	28.66	16.44	6.75
+ *Score-blend*	✗	✗	**70.27**	**48.59**	**54.02**	**48.59**	0.55	53.35	50.84	56.48	**36.69**	0.45
	✓	✗	57.67	37.17	42.31	37.17	4.57	43.61	38.16	43.87	27.52	3.99
	✗	✓	43.70	24.89	29.53	24.89	2.64	33.14	26.21	31.14	18.70	1.22
	✓	✓	38.58	21.47	25.72	21.47	6.44	30.13	23.98	28.82	16.55	4.8

Table 6: Results for different retrieval methods on test datasets. Best per column is bold and the second-best is underlined. The latency is measured in seconds per query.

Provide a short interpretation — e.g., which configuration excels in which scenario, trade-offs between speed and accuracy, and performance differences between single-hop and combined datasets.

6.2 Qualitative analysis

Results showed that:

65% of answers were rated as “Relevant”,

25% as “Partially relevant”,

10% as “Not relevant”.

Feedback indicated that relevance dropped when:

the query used ambiguous phrasing,

or too few document chunks were retrieved due to vector sparsity.

Users appreciated:

the traceability of answers via citations,

the lightweight UI,

and the multilingual support.

Chapter 7

Discussion

7.1 Further Development

specializzazione sul dominio archeologia

Chapter 8

Conclusion

Appendices

Appendix A

Implementation details

In this study, all the experiments have been executed in Python 3.11.9 on a system equipped with an Intel Core i7-1185G7 CPU at 3.00 GHz, 16 GB of RAM, and integrated Intel Iris Xe Graphics with 128 MB of VRAM.

The ablation studies were conducted on the *singlehop* dataset, comprising 508 queries, and on the *combined* dataset, comprising 908 queries in total (singlehop plus multi-hop questions). The retrieval index, implemented using FAISS, contained 801 document chunks with a length of 687 words (average: 91.83 words). Evaluation was carried out in batch mode with a batch size of 32, retrieving up to `candidate_k` = 50 documents per query before applying top-k selection ($k = 5$). For hybrid retrieval configurations, the RRF parameter was set to $k = 60$, with dense and sparse weights both equal to 1.0, while $\alpha = 0.3$.

Additionally, the proposed approach was implemented using the following libraries: PyTorch, Hugging Face Transformers, SentenceTransformers, PEFT, spaCy, KeyBERT, NLTK, FAISS, Rank-BM25, and scikit-learn. These were complemented by supporting and utility packages such as NumPy, Pandas, Matplotlib, Streamlit, Pillow (PIL), BeautifulSoup (bs4), httpx, and the MistralAI API, as well as standard Python libraries including asyncio, concurrent.futures, collections, functools, and urllib.

Appendix B

Abbreviations and Glossary

Table 7: Abbreviations and acronyms with their full forms and definitions used in this thesis.

Term	Full form	Glossary definition
AI	Artificial intelligence	The field of computer science dedicated to creating systems capable of performing tasks that typically require human intelligence, such as reasoning, learning, and problem-solving.
DH	Digital humanities	An interdisciplinary field that applies computational methods and tools to humanities research, analysis, and dissemination.
QA	Question answering	A technology or task in which a system provides precise answers to questions posed in natural language.
QAS	Question-answering system	A system designed to answer questions automatically by processing natural language input, often using methods from IR and NLP.
RAG	Retrieval-augmented generation	An approach combining information retrieval with generative models, allowing AI to reference external data sources when generating answers.
LLM	Large language model	A neural network trained on massive text corpora to generate or understand human language, such as GPT or BERT.
IR	Information retrieval	The process of searching, retrieving, and ranking relevant documents or data from large collections based on user queries.
GNA	Geoportale Nazionale Archeologia	Italy's institutional repository for archaeological data, hosting extensive documentation and resources related to the country's cultural heritage.
MiC	Ministero della Cultura	The Italian Ministry of Culture, responsible for the preservation and promotion of Italy's cultural heritage.
MiBACT	Ministero dei Beni e delle Attività Culturali e del Turismo	The former name of the Italian Ministry of Culture, which was responsible for cultural heritage and tourism before its reorganization in 2021.
CNR	Consiglio Nazionale delle Ricerche	The Italian National Research Council, a major public research institution that conducts scientific research across various disciplines, including cultural heritage.

Continued on next page

Term	Full form	Glossary definition
DG-Ant	Direzione Generale Archeologia, Belle Arti e Paesaggio	The Directorate General for Archaeology, Fine Arts, and Landscape within the Italian Ministry of Culture, overseeing archaeological heritage and cultural sites.
ICA	Istituto Centrale per l'Archeologia	The Central Institute for Archaeology in Italy, established in 2016 as part of the Ministry of Culture, responsible for archaeological research and documentation.
ICCD	Istituto Centrale per il Catalogo e la Documentazione	The Central Institute for Cataloging and Documentation, part of the Italian Ministry of Culture, responsible for cataloging cultural heritage assets and proposing best practices.
SiGECweb	Sistema Informativo Generale del Catalogo	A web platform that handles every stage of cultural heritage cataloguing, from standard creation and code assignment to cataloguing diverse assets and publishing records online for public access.
GIS	Geographic information system	A computer system, including software and hardware, designed to capture, store, manipulate, analyze, manage, and present spatial or geographic data, often used in archaeology for mapping and spatial analysis.
QGIS		A particular GIS software that is free and open-source.
GLAM	Galleries, Libraries, Archives and Museums	A collective term for institutions that preserve and provide access to cultural heritage in the public interest.
KB	Knowledge base	A structured collection of information or data, often used to support reasoning, search, or retrieval in AI systems.
NLP	Natural language processing	The area of AI focused on enabling computers to understand, interpret, and generate human language.
NLG	Natural language generation	The process of automatically generating human-like text from structured data or models, often used in chatbots and content creation.
NL	Natural language	The everyday language used by humans for communication, which NLP systems aim to understand and generate.
ML	Machine learning	A subset of AI that involves training algorithms to recognize patterns and make decisions based on data.
NER	Named entity recognition	A subtask of NLP that identifies and classifies named entities (e.g., people, organizations, locations) in text.
EL	Entity linking	The process of connecting named entities in text to their corresponding entries in a knowledge base, enhancing understanding and retrieval.
TF-IDF	Term Frequency-Inverse Document Frequency	A statistical measure used in IR to evaluate how important a word is to a document relative to a corpus, balancing term frequency and document rarity.
BM25	Best match 25	A ranking function used in IR to estimate the relevance of documents to a given search query, based on term frequency and document length normalization.
PRF	Precision-Recall-F1	Metrics used to evaluate the performance of classification models, where precision measures the accuracy of positive predictions, recall measures the ability to find all relevant instances, and F1 is the harmonic mean of precision and recall.

Continued on next page

Term	Full form	Glossary definition
RDF	Resource Description Framework	A standard model for data interchange on the web, allowing structured representation of information about resources in a machine-readable format.
SQL	Structured Query Language	A standard programming language used for managing and manipulating relational databases, allowing users to query, insert, update, and delete data.
SPARQL		A query language and protocol used to retrieve and manipulate data stored in Resource Description Framework (RDF) format, commonly used for querying knowledge graphs.
ontology		A formal representation of a set of concepts within a domain and the relationships between those concepts, used to enable knowledge extraction, sharing and reuse.
JSON	JavaScript Object Notation	A lightweight data interchange format that is easy for humans to read and write, and easy for machines to parse and generate, often used for data exchange in web applications.
TREC	Text REtrieval Conference	An ongoing series of workshops and evaluations focused on advancing research in text retrieval and related tasks.
LMIR	Language model information retrieval	A method of using language models to improve the effectiveness of information retrieval systems by leveraging their understanding of language and context.
RNN	Recurrent Neural Network	A type of neural network architecture designed to process sequential data by maintaining a form of memory of previous inputs.
LSTM	Long Short-Term Memory	A special kind of RNN capable of learning long-range dependencies, often used for tasks like language modeling or time series prediction.
CRF	Conditional Random Field	A probabilistic graphical model used for structured prediction, especially in NLP tasks such as sequence labeling.
SVM	Support Vector Machine	A supervised machine learning algorithm used for classification and regression, which finds the optimal boundary between classes in the feature space.
Word2Vec	Word to Vector	A technique for representing words as vectors in a continuous vector space, capturing semantic relationships between words based on their context in large text corpora.
GloVe	Global Vectors for Word Representation	An unsupervised learning algorithm for obtaining vector representations of words, which captures global statistical information from a corpus.
BERT	Bidirectional Encoder Representations from Transformers	A pre-trained language model that uses the Transformer architecture to understand the context of words in a sentence by considering both left and right contexts simultaneously.
GPT	Generative Pre-Trained Transformer	A family of LLMs based on the Transformer architecture.
ChatGPT	Generative Pre-trained Transformer	An application of the GPT architecture developed by OpenAI, fine-tuned for conversational interaction and instruction following, and released to the public in November 2022.
T5	Text-to-Text Transfer Transformer	T5 is a series of LLMs developed by Google AI and introduced in 2019.

Continued on next page

Term	Full form	Glossary definition
E5	Embedding-based model for information retrieval	A family of models designed to generate high-quality embeddings for text, improving retrieval performance in IR tasks.
BGE	BAAI general embedding	A family of models designed to generate high-quality embeddings for text, improving retrieval performance in IR tasks.
intfloat	intfloat/e5	A specific implementation of the E5 model, optimized for generating embeddings for information retrieval tasks.
LLM-embedder		A model designed to generate embeddings for text using large language models, enhancing the quality of semantic representations for retrieval tasks.
embeddings		Dense vector representations of text that capture semantic meaning, used in various NLP tasks including retrieval and classification.
chunking		The process of breaking down text into smaller, manageable pieces or "chunks" to facilitate processing and analysis in NLP tasks.
vector database		A specialized database designed to store and retrieve high-dimensional vectors efficiently, often used in RAG systems for managing embeddings.
retriever		A component of a system responsible for searching and retrieving relevant documents or information from a database or corpus based on user queries.
ranking function		A mathematical function used to score and order documents based on their relevance to a given query, often employed in IR systems.
XML	eXtensible Markup Language	A markup language used to encode documents in a format that is both human-readable and machine-readable, often used for data interchange.
TEI	Text Encoding Initiative	A set of guidelines for encoding literary and linguistic texts in XML, providing a standardized way to represent complex textual structures.
MARC/RDA	Machine-Readable Cataloging / Resource Description and Access	Standards for encoding bibliographic information in a machine-readable format, widely used in libraries and information systems.
GROBID	GeneRation Of BIbliographic Data	A machine learning library for extracting and structuring bibliographic information from scholarly documents, often used in academic publishing and research.
Milvus	Milvus Vector Database	An open-source vector database designed for efficient storage, indexing, and retrieval of high-dimensional vectors, commonly used in RAG systems.
Faiss	Facebook AI Similarity Search	A library for efficient similarity search and clustering of dense vectors, widely used in RAG systems for indexing and searching large datasets.
Qdrant	Qdrant Vector Database	An open-source vector database that provides efficient storage and retrieval of high-dimensional vectors, supporting hybrid search capabilities.

Continued on next page

Term	Full form	Glossary definition
DLM reranking	Deep language model reranking	Deep language model-based reranking uses fine-tuned models that jointly encode query-document pairs and classify their relevance as “true” or “false”. At inference, documents are ranked by the probability of being labeled “true”.
HyDE	Hypothetical Document Embeddings	A method that generates a brief, plausible answer to the query first, then embeds that “hypothetical doc” for retrieval. This richer proxy query improves vector search recall/precision in RAG context, especially for vague or underspecified queries.
Hybrid Search		A search approach that combines vector-based retrieval with traditional keyword search, allowing for more comprehensive and context-aware results in RAG systems.
TILDE		A framework designed to facilitate the development and deployment of RAG systems, providing tools for data preparation, indexing, and retrieval.
TILDEv2		An updated version of the TILDE framework, incorporating improvements in efficiency and performance.
LTR	Learning-to-Rank	A machine learning approach used to optimize the ranking of search results based on user interactions and relevance feedback, improving the quality of retrieved documents in RAG systems.
Self-RAG	Self-Retrieval-Augmented Generation	A variant of RAG where the system retrieves relevant information from its own generated content, enhancing the context and accuracy of responses.
RAGAS	Retrieval-Augmented Generation with Adaptive Sampling	An advanced RAG approach that dynamically selects and retrieves the most relevant information based on the context of the query, improving the efficiency and accuracy of responses.
XAI	Explainable Artificial Intelligence	A field of AI focused on rendering the decision-making processes of AI systems transparent and understandable to humans, often used to build trust and accountability in AI applications.
RAG-Chain	Retrieval-Augmented Generation Chain	A method that links multiple RAG components in a sequence.
ArCo	Italian Cultural Heritage knowledge graph	A knowledge graph representing Italian cultural heritage, providing structured information about historical sites, artifacts, and related entities.

Bibliography

Abeysinghe, Bhashithe, and Ruhan Circi. 2024. *The Challenges of Evaluating LLM Applications: An Analysis of Automated, Human, and LLM-Based Approaches.* ArXiv:2406.03339, June. Accessed August 3, 2025. DOI: 10.48550/arXiv.2406.03339. http://arxiv.org/abs/2406.03339.

Abu Shawar, Bayan, and Eric Atwell. 2007. "Chatbots: Are they Really Useful?" *Journal for Language Technology and Computational Linguistics* 22 (1): 29–49. ISSN: 2190-6858, accessed June 19, 2025. DOI: 10.21248/jlcl.22.2007.88. https://jlcl.org/article/view/88.

Acconcia, Valeria. 2023. "LA PUBBLICAZIONE DEI DATI NEL GEOPORTALE NAZIONALE PER L'ARCHEOLOGIA" [in ita]. In *GEOPORTALE NAZIONALE PER L'ARCHEOLOGIA,* 1. IT: Ministero della Cultura - Istituto Centrale per l'Archeologia, July. Accessed July 24, 2025. https://doi.org/10.60974/GNA_03.

Ada Gabucci. 2025, July. Accessed July 24, 2025. https://web.archive.org/web/20250724081422/https://conf24.garr.it/it/speaker/ada-gabucci.

Agarwal, Shubham, Gaurav Sahu, Abhay Puri, Issam H. Laradji, Krishnamurthy DJ Dvijotham, Jason Stanley, Laurent Charlin, and Christopher Pal. 2025. *LitLLM: A Toolkit for Scientific Literature Review.* ArXiv:2402.01788, March. Accessed July 23, 2025. DOI: 10.48550/arXiv.2402.01788. http://arxiv.org/abs/2402.01788.

Akkiraju, Rama, Anbang Xu, Deepak Bora, Tan Yu, Lu An, Vishal Seth, Aaditya Shukla, et al. 2024. *FACTS About Building Retrieval Augmented Generation-based Chatbots.* ArXiv:2407.07858, July. Accessed July 23, 2025. DOI: 10.48550/arXiv.2407.07858. http://arxiv.org/abs/2407.07858.

Alanazi, Sarah Saad, Nazar Elfadil, Mutsam Jarajreh, and Saad Algarni. 2021. "Question Answering Systems: A Systematic Literature Review" [in en]. *International Journal of Advanced Computer Science and Applications* 12 (3). ISSN: 21565570, 2158107X, accessed July 23, 2025. DOI: 10.14569/IJACSA.2021.0120359. http://thesai.org/Publications/ViewPaper?Volume=12&Issue=3&Code=IJACSA&SerialNo=59.

Alqifari, Reem. 2019. "Question Answering Systems Approaches and Challenges." In *Proceedings of the Student Research Workshop Associated with RANLP 2019,* edited by Venelin Kovatchev, Irina Temnikova, Branislava Šandrih, and Ivelina Nikolova, 69–75. Varna, Bulgaria: INCOMA Ltd., September. Accessed July 26, 2025. DOI: 10.26615/issn.2603-2821.2019_011. https://aclanthology.org/R19-2011/.

Alshammari, Suad, Lama Basalelah, Walaa Abu Rukbah, Ali Alsuhibani, and Dayanjan S. Wijesinghe. 2023. *KNIMEZoBot: Enhancing Literature Review with Zotero and KNIME OpenAI Integration using Retrieval-Augmented Generation.* ArXiv:2311.04310, November. Accessed July 21, 2025. DOI: 10.48550/arXiv.2311.04310. http://arxiv.org/abs/2311.04310.

Was ist RAG? – Retrieval Augmented Generation erklärt – AWS [in de-DE]. Accessed July 24, 2025. https://aws.amazon.com/de/what-is/retrieval-augmented-generation/.

Amershi, Saleema, Andrew Begel, Christian Bird, Robert DeLine, Harald Gall, Ece Kamar, Nachiappan Nagappan, Besmira Nushi, and Thomas Zimmermann. 2019. "Software Engineering for Machine Learning: A Case Study." In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP),* 291–300. Montreal, QC, Canada: IEEE, May. ISBN: 9781728117607, accessed July 28, 2025. DOI: 10.1109/ICSE-SEIP.2019.00042. https://ieeexplore.ieee.org/document/8804457/.

Antolini, Gianluca. 2025. "Experimental Study on Retrieval-Augmented Generation: Engineering and Evaluation of a Custom RAG system for Open-Domain QA" [in en]. Master's thesis, Computer Engineering, University of Padova, July. Accessed July 21, 2025. https://thesis.unipd.it/handle/20.500.12608/86949.

Antoniou, Christina, and Nick Bassiliades. 2022. "A survey on semantic question answering systems" [in en]. *The Knowledge Engineering Review* 37:e2. ISSN: 0269-8889, 1469-8005, accessed July 26, 2025. DOI: 10.1017/S0269888921000138. https://www.cambridge.org/core/product/identifier/S0269888921000138/type/journal_article.

Arslan, Muhammad, Hussam Ghanem, Saba Munawar, and Christophe Cruz. 2024. "A Survey on RAG with LLMs" [in en]. *Procedia Computer Science* 246:3781–3790. ISSN: 18770509, accessed June 19, 2025. DOI: 10.1016/j.procs.2024.09.178. https://linkinghub.elsevier.com/retrieve/pii/S1877050924021860.

Asai, Akari, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. *Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection.* ArXiv:2310.11511, October. Accessed July 27, 2025. DOI: 10.48550/arXiv.2310.11511. http://arxiv.org/abs/2310.11511.

Ashery, Ariel Flint, Luca Maria Aiello, and Andrea Baronchelli. 2025. "Emergent social conventions and collective bias in LLM populations" [in en]. *Science Advances* 11, no. 20 (May): eadu9368. ISSN: 2375-2548, accessed July 29, 2025. DOI: 10.1126/sciadv.adu9368. https://www.science.org/doi/10.1126/sciadv.adu9368.

Aytar, Ahmet Yasin, Kemal Kilic, and Kamer Kaya. 2024. *A Retrieval-Augmented Generation Framework for Academic Literature Navigation in Data Science.* ArXiv:2412.15404, December. Accessed July 21, 2025. DOI: 10.48550/arXiv.2412.15404. http://arxiv.org/abs/2412.15404.

Barbato, Fabrizio. 2025. *Nasce Cat-IA, l'agente conversazionale che semplifica la consultazione del Catalogo generale dei beni culturali* [in it-IT], April. Accessed July 21, 2025. https://digitallibrary.cultura.gov.it/notizie/nasce-cat-ia/.

Bevara, Ravi Varma Kumar, Brady D. Lund, Nishith Reddy Mannuru, Sai Pranathi Karedla, Yara Mohammed, Sai Tulasi Kolapudi, and Aashrith Mannuru. 2025. "Prospects of Retrieval Augmented Generation (RAG) for Academic Library Search and Retrieval." *Information Technology and Libraries* 44, no. 2 (June). ISSN: 2163-5226, 0730-9295, accessed July 19, 2025. DOI: 10.5860/ital.v44i2.17361. https://ital.corejournals.org/index.php/ital/article/view/17361.

Boi, Valeria. 2023. "IL GEOPORTALE NAZIONALE PER L'ARCHEOLOGIA: STANDARDIZZAZIONE E APERTURA DEI DATI" [in ita]. In *GEOPORTALE NAZIONALE PER L'ARCHEOLOGIA,* 1. IT: Ministero della Cultura - Istituto Centrale per l'Archeologia, July. Accessed July 24, 2025. https://doi.org/10.60974/GNA_02.

Bor-Woei, Huang. 2024. *Generative Large Language Models Augmented Hybrid Retrieval System for Biomedical Question Answering.* Technical report. Grenoble, France: University of Padova, Italy. https://ceur-ws.org/Vol-3740/paper-12.pdf.

Bran, Andres M., Alexandru Oarga, Matthew Hart, Magdalena Lederbauer, and Philippe Schwaller. 2024. "Ontology-Retrieval Augmented Generation for Scientific Discovery" [in en] (October). Accessed July 23, 2025. https://openreview.net/forum?id=DbZDbg2z9q.

BUP Solutions. *BUP Solutions.* Accessed July 25, 2025. https://www.bupsolutions.com/en/home_en/.

Caballero, Michael. 2021. "A Brief Survey of Question Answering Systems." *International Journal of Artificial Intelligence & Applications* 12, no. 5 (September): 01–07. ISSN: 09762191, accessed July 26, 2025. DOI: 10.5121/ijaia.2021.12501. https://aircconline.com/ijaia/V12N5/12521ijaia01.pdf.

Calandra, Elena. 2023. "IL GEOPORTALE NAZIONALE PER L'ARCHEOLOGIA (GNA). UN'INTRODUZIONE" [in ita]. In *GEOPORTALE NAZIONALE PER L'ARCHEOLOGIA,* 1. IT: Ministero della Cultura - Istituto Centrale per l'Archeologia, July. Accessed July 24, 2025. https://doi.org/10.60974/GNA_01.

Callaghan, Samantha, and Miguel Vieira. 2025. *Prototyping a RAG System for Digital Humanities: Exploring AI/ML with Indigenous Data | King's Digital Lab* [in en], January. Accessed July 19, 2025. https://kdl.kcl.ac.uk/blog/ireal-rag/.

Caramanna, Gianluigi. 2024. "Progettazione e sviluppo di un chatbot basato su tecniche di Intelligenza Artificiale Generativa" [in it]. Master's thesis, Ingegneria Informatica, Università di Bologna, October. Accessed July 21, 2025. https://amslaurea.unibo.it/id/eprint/32820/.

Carriero, Valentina Anita, Aldo Gangemi, Maria Letizia Mancinelli, Ludovica Marinucci, Andrea Giovanni Nuzzolese, Valentina Presutti, and Chiara Veninata. 2019. *ArCo: the Italian Cultural Heritage Knowledge Graph.* ArXiv:1905.02840, May. Accessed July 23, 2025. DOI: 10.48550/arXiv.1905.02840. http://arxiv.org/abs/1905.02840.

Ciletti, Michele. 2025. "Retrieval-Augmented Generation systems for enhanced access to digital archives" [in en]. In *Diversità, Equità e Inclusione: Sfide e Opportunità per l'Informatica Umanistica nell'Era dell'Intelligenza Artificiale, Proceedings del XIV Convegno Annuale AIUCD2025,* 663. Quaderni di Umanistica Digitale. Verona: AIUCD, June. ISBN: 978-88-942535-9-7. DOI: 10.6092/unibo/amsacta/8380. https://amsacta.unibo.it/id/eprint/8380/.

Davis, Corey. 2025. "Unlocking web archives: LLMs, RAG, and the future of digital preservation" [in en] (February). Accessed July 19, 2025. https://hdl.handle.net/1828/21379.

DeBellis, Michael. 2024. *Integrating Large Language Models (LLM) and ontologies to Implement Retrieval Augmented Generation* [in en], July. Accessed July 23, 2025. https://www.michaeldebellis.com/post/integrating-llms-and-ontologies.

Deriu, Jan, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak. 2020. *Survey on Evaluation Methods for Dialogue Systems.* ArXiv:1905.04071, June. Accessed August 3, 2025. DOI: 10.48550/arXiv.1905.04071. http://arxiv.org/abs/1905.04071.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* ArXiv:1810.04805, May. Accessed July 26, 2025. DOI: 10.48550/arXiv.1810.04805. http://arxiv.org/abs/1810.04805.

Di Marcantonio, Giorgia. 2024. "Intelligenza artificiale, Large Language Models (LLMs) e Retrieval-Augmented Generation (RAG). Nuovi strumenti per l'accesso alle risorse archivistiche e bibliografiche" [in it]. *Bibliothecae.it* 13, no. 1 (July): 146–173. Accessed July 28, 2025. DOI: 10.6092/ISSN.2283-9364/19982. https://bibliothecae.unibo.it/article/view/19982.

Diefenbach, Dennis, Vanessa Lopez, Kamal Singh, and Pierre Maret. 2018. "Core techniques of question answering systems over knowledge bases: a survey" [in en]. *Knowledge and Information Systems* 55, no. 3 (June): 529–569. ISSN: 0219-1377, 0219-3116, accessed July 26, 2025. DOI: 10.1007/s10115-017-1100-y. http://link.springer.com/10.1007/s10115-017-1100-y.

Es, Shahul, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. 2023. *Ragas: Automated Evaluation of Retrieval Augmented Generation.* Accessed June 19, 2025. DOI: 10.48550/ARXIV.2309.15217. https://arxiv.org/abs/2309.15217.

Falcone, Annalisa, Elena Calandra, Valeria Boi, Annalisa Falcone, and Valeria Acconcia. 2023. *DEMATERIALIZZAZIONE E CONDIVISIONE IN RETE DEI DATI DELLE INDAGINI ARCHEOLOGICHE SVOLTE IN REGIME DI CONCESSIONE. PROMOZIONE E CONDIVISIONE DEI DATI PRODOTTI DALLE MISSIONI ARCHEOLOGICHE ITALIANE ALL'ESTERO* [in ita]. Technical report. IT: Ministero della Cultura - Istituto Centrale per l'Archeologia, July. Accessed August 3, 2025. https://doi.org/10.60974/GNA_04.

Fan, Yang, Zhang Qi, Xing Wenqian, Liu Chang, and Liu Liu. 2025. *Research on Graph-Retrieval Augmented Generation Based on Historical Text Knowledge Graphs.* ArXiv:2506.15241, June. Accessed July 19, 2025. DOI: 10.48550/arXiv.2506.15241. http://arxiv.org/abs/2506.15241.

Farea, Amer, and Frank Emmert-Streib. 2025. "Understanding question-answering systems: Evolution, applications, trends, and challenges" [in en]. *Engineering Applications of Artificial Intelligence* 156 (September): 110997. ISSN: 09521976, accessed July 26, 2025. DOI: 10.1016/j.engappai.2025.110997. https://linkinghub.elsevier.com/retrieve/pii/S0952197625009972.

Ferrucci, David, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A. Kalyanpur, Adam Lally, et al. 2011. "Building watson: An overview of the deepQA project" [in en-US]. *AI Magazine* (January). ISSN: 07384602, accessed July 27, 2025. DOI: 10.1609/aimag.v31i3.2303. https://research.ibm.com/publications/building-watson-an-overview-of-the-deepqa-project.

Florio, Michelangelo. 2024. "Progettazione e implementazione di un chatbot intelligente tramite piattaforma LangChain: studio e valutazione dei VectorDB" [in it]. Master's thesis, August. Accessed July 21, 2025. https://amslaurea.unibo.it/id/eprint/32282/.

Franco, Wellington, Caio Viktor, Artur Oliveira, Gilvan Maia, Angelo Brayner, V. Vidal, Fernando Carvalho, and V. Pequeno. 2020. "Ontology-based Question Answering Systems over Knowledge Bases: A Survey." In *Proceedings of the 22nd International Conference on Enterprise Information Systems,* 532–539. Prague, Czech Republic: SCITEPRESS - Science / Technology Publications. ISBN: 9789897584237, accessed July 26, 2025. DOI: 10.5220/0009392205320539. http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0009392205320539.

Gabucci, Ada. 2023. "UN TEMPLATE QGIS AL SERVIZIO DEL GEOPORTALE NAZIONALE PER L'ARCHEOLOGIA" [in ita]. In *GEOPORTALE NAZIONALE PER L'ARCHEOLOGIA,* 1. IT: Ministero della Cultura - Istituto Centrale per l'Archeologia, July. Accessed July 24, 2025. https://doi.org/10.60974/GNA_05.

Gao, Luyu, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2022. *Precise Zero-Shot Dense Retrieval without Relevance Labels.* ArXiv:2212.10496, December. Accessed August 3, 2025. DOI: 10.48550/arXiv.2212.10496. http://arxiv.org/abs/2212.10496.

Gao, Yunfan, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. *Retrieval-Augmented Generation for Large Language Models: A Survey.* ArXiv:2312.10997. Accessed July 10, 2025. DOI: 10.48550/arXiv.2312.10997. http://arxiv.org/abs/2312.10997.

GNA, MiC. 2024. *Wiki GNA Manuale Operativo v.1.4,* February. Accessed July 24, 2025. https://gna.cultura.gov.it/wiki/index.php/Pagina_principale.

Green, Bert F., Alice K. Wolf, Carol Chomsky, and Kenneth Laughery. 1961. "Baseball: an automatic question-answerer" [in en]. In *Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference on - IRE-AIEE-ACM '61 (Western),* 219. Los Angeles, California: ACM Press. Accessed July 27, 2025. DOI: 10.1145/1460690.1460714. http://portal.acm.org/citation.cfm?doid=1460690.1460714.

GROBID. 2008–2025. https://github.com/kermitt2/grobid. swh: 1:dir:dab86b296e3c3216e2241968f0d63b68e8209d3c.

Gupta, Shailja, Rajesh Ranjan, and Surya Narayan Singh. 2024. *A Comprehensive Survey of Retrieval-Augmented Generation (RAG): Evolution, Current Landscape and Future Directions.* ArXiv:2410.12837. Accessed June 19, 2025. DOI: 10.48550/arXiv.2410.12837. http://arxiv.org/abs/2410.12837.

Han, Haoyu, Yu Wang, Harry Shomer, Kai Guo, Jiayuan Ding, Yongjia Lei, Mahantesh Halappanavar, et al. 2025. *Retrieval-Augmented Generation with Graphs (GraphRAG).* ArXiv:2501.00309, January. Accessed July 27, 2025. DOI: 10.48550/arXiv.2501.00309. http://arxiv.org/abs/2501.00309.

Harsh and T. Shobha. 2024. "Comprehending and Reducing LLM Hallucinations" [in en]. *International Journal of Innovative Science and Research Technology (IJISRT),* 1222–1227. ISSN: 2456-2165, accessed June 19, 2025. DOI: 10.38124/ijisrt/IJISRT24JUL882. https://www.ijisrt.com/comprehending-and-reducing-llm-hallucinations.

Hirschman, L., and R. Gaizauskas. 2001. "Natural language question answering: the view from here" [in en]. *Natural Language Engineering* 7, no. 4 (December): 275–300. ISSN: 1351-3249, 1469-8110, accessed July 26, 2025. DOI: 10.1017/S1351324901002807. https://www.cambridge.org/core/product/identifier/S1351324901002807/type/journal_article.

Home : Regesta Imperii. Accessed July 29, 2025. https://www.regesta-imperii.de/en/home.html.

Open Source vs. Proprietary LLMs: A Comprehensive Comparison. 2025, January. Accessed June 19, 2025. https://www.hostcomm.co.uk/blogs/open-source-vs-proprietary-llms-a-comprehensive-comparison.

Jiang, Zhuoxuan, Tianyang Zhang, Shengguang Bai, Lin Lin, Haotian Zhang, Yinong Xun, Jiawei Ren, Wen Si, and Shaohua Zhang. 2024. *Towards Enterprise-Specific Question-Answering for it Operations and Maintenance Based on Retrieval-Augmented Generation Mechanism.* Accessed July 23, 2025. DOI: 10.2139/ssrn.5069318. https://www.ssrn.com/abstract=5069318.

Jobin, Anna, Marcello Ienca, and Effy Vayena. 2019. "The global landscape of AI ethics guidelines" [in en]. *Nature Machine Intelligence* 1, no. 9 (September): 389–399. ISSN: 2522-5839, accessed July 28, 2025. DOI: 10.1038/s42256-019-0088-2. https://www.nature.com/articles/s42256-019-0088-2.

Jurafsky, Daniel, and James H. Martin. 2024. "Chapter 14: Question Answering, Information Retrieval, and RetrievalAugmented Generation." In *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models,* 2nd ed. Accessed July 26, 2025. https://web.stanford.edu/~jurafsky/slp3/.

Kaplan, Jared, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. *Scaling Laws for Neural Language Models.* ArXiv:2001.08361, January. Accessed July 29, 2025. DOI: 10.48550/arXiv.2001.08361. http://arxiv.org/abs/2001.08361.

Kuczera, Andreas, and Stephan Armbruster. 2024. *ChatGPT und generative KI in der mediävistischen Grundlagenforschung | H-Soz-Kult. Kommunikation und Fachinformation für die Geschichtswissenschaften | Geschichte im Netz | History in the web* [in de], July. Accessed July 28, 2025. https://www.hsozkult.de/event/id/event-142284.

Lála, Jakub, Odhran O'Donoghue, Aleksandar Shtedritski, Sam Cox, Samuel G. Rodriques, and Andrew D. White. 2023. *PaperQA: Retrieval-Augmented Generative Agent for Scientific Research.* ArXiv:2312.07559, December. Accessed July 25, 2025. DOI: 10.48550/arXiv.2312.07559. http://arxiv.org/abs/2312.07559.

LangChain Documentation v0.3 [in en]. 2024. Accessed June 19, 2025. https://python.langchain.com/docs/introduction/.

Lewis, Patrick, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, et al. 2020. *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.* Accessed June 19, 2025. DOI: 10.48550/ARXIV.2005.11401. https://arxiv.org/abs/2005.11401.

Li, Jiawei, and Yue Zhang. 2024. *The Death of Feature Engineering? BERT with Linguistic Features on SQuAD 2.0.* ArXiv:2404.03184, April. Accessed July 28, 2025. DOI: 10.48550/arXiv.2404.03184. http://arxiv.org/abs/2404.03184.

Li, Zhicong, Jiahao Wang, Zhishu Jiang, Hangyu Mao, Zhongxia Chen, Jiazhen Du, Yuanxing Zhang, Fuzheng Zhang, Di Zhang, and Yong Liu. 2024. *DMQR-RAG: Diverse Multi-Query Rewriting for RAG.* ArXiv:2411.13154, November. Accessed August 9, 2025. DOI: 10.48550/arXiv.2411.13154. http://arxiv.org/abs/2411.13154.

Liu, Chia-Wei, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. "How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation." In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing,* edited by Jian Su, Kevin Duh, and Xavier Carreras, 2122–2132. Austin, Texas: Association for Computational Linguistics, November. Accessed August 3, 2025. DOI: 10.18653/v1/D16-1230. https://aclanthology.org/D16-1230/.

Liu, Linqing, Patrick Lewis, Sebastian Riedel, and Pontus Stenetorp. 2022. *Challenges in Generalization in Open Domain Question Answering.* ArXiv:2109.01156, May. Accessed July 26, 2025. DOI: 10.48550/arXiv.2109.01156. http://arxiv.org/abs/2109.01156.

Ludwig, Heiner, Thorsten Schmidt, and Mathias Kühn. 2025. "An ontology-based retrieval augmented generation procedure for a voice-controlled maintenance assistant" [in en]. *Computers in Industry* 169 (August): 104289. ISSN: 01663615, accessed July 23, 2025. DOI: 10.1016/j.compind.2025.104289. https://linkinghub.elsevier.com/retrieve/pii/S0166361525000545.

Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval.* New York: Cambridge University Press. ISBN: 9780521865715.

Martineau, Kim. 2023. *What is retrieval-augmented generation (RAG)?* [In en-US]. Accessed July 19, 2025. https://research.ibm.com/blog/retrieval-augmented-generation-RAG.

Maslej, Nestor, Loredana Fattorini, Raymond Perrault, Yolanda Gil, Vanessa Parli, Njenga Kariuki, Emily Capstick, et al. 2025. *Artificial Intelligence Index Report 2025.* ArXiv:2504.07139, July. Accessed July 26, 2025. DOI: 10.48550/arXiv.2504.07139. http://arxiv.org/abs/2504.07139.

McDermott, Drew. 1976. "Artificial intelligence meets natural stupidity" [in en]. *ACM SIGART Bulletin,* no. 57 (April): 4–9. ISSN: 0163-5719, accessed July 25, 2025. DOI: 10.1145/1045339.1045340. https://dl.acm.org/doi/10.1145/1045339.1045340.

Mic, GNA. 2019. *MiC GNA Geoportale Nazionale Archeologia* [in it], September. Accessed July 24, 2025. https://gna.cultura.gov.it.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. *Efficient Estimation of Word Representations in Vector Space.* ArXiv:1301.3781, September. Accessed July 29, 2025. DOI: 10.48550/arXiv.1301.3781. http://arxiv.org/abs/1301.3781.

Mishra, Onkar. 2024. *Using LangChain for Question Answering on own data* [in en]. Accessed June 19, 2025. https://medium.com/@onkarmishra/using-langchain-for-question-answering-on-own-data-3af0a82789ed.

Mistral NeMo | Mistral AI. 2025, August. Accessed August 3, 2025. https://web.archive.org/web/20250803120348/https://mistral.ai/news/mistral-nemo.

Nicoletti, Sonia. 2025. "LLMs and Essence: Developing a Chatbot to Support Software Engineering Practices" [in en]. Master's thesis, Computer Science, University of Bologna, February. Accessed July 21, 2025. https://amslaurea.unibo.it/id/eprint/34197/.

Normativa - FOIA [in it-it]. 2016. Accessed August 3, 2025. https://foia.gov.it/normativa.

ODSC-Community. 2024. *Retrieval-Augmented Generation (RAG): A Synergistic Approach to NLU and NLG* [in en-US], July. Accessed July 19, 2025. https://opendatascience.com/retrieval-augmented-generation-rag-a-synergistic-approach-to-nlu-and-nlg/.

Packowski, Sarah, Inge Halilovic, Jenifer Schlotfeldt, and Trish Smith. 2024. “Optimizing and Evaluating Enterprise Retrieval-Augmented Generation (RAG): A Content Design Perspective” [in en]. In *Proceedings of the 2024 8th International Conference on Advances in Artificial Intelligence,* 162–167. London United Kingdom: ACM, October. ISBN: 9798400718014, accessed July 23, 2025. DOI: 10.1145/3704137.3704181. https://dl.acm.org/doi/10.1145/3704137.3704181.

Park, Yeun, Paul Witherell, Nowrin Akter Surovi, and Hyunbo Cho. 2024. “Ontology-based Retrieval Augmented Generation (RAG) for GenAI-supported Additive Manufacturing” [in en]. *NIST* (August). Accessed July 23, 2025. https://www.nist.gov/publications/ontology-based-retrieval-augmented-generation-rag-genai-supported-additive.

Pennington, Jeffrey, Richard Socher, and Christopher Manning. 2014. “GloVe: Global Vectors for Word Representation.” In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP),* edited by Alessandro Moschitti, Bo Pang, and Walter Daelemans, 1532–1543. Doha, Qatar: Association for Computational Linguistics, October. Accessed July 29, 2025. DOI: 10.3115/v1/D14-1162. https://aclanthology.org/D14-1162/.

Pograri, Lucrezia. 2025. *Question-Answering AI Assistant for Geoportale Nazionale Archeologia (GNA),* July. Accessed July 21, 2025. DOI: 10.5281/zenodo.16259759. https://zenodo.org/records/16259759.

Pollin, Christopher, Franz Fischer, Patrick Sahle, Martina Scholger, and Georg Vogeler. 2025. “When it was 2024 – Generative AI in the field of digital scholarly editions” [in en]. *Zeitschrift für digitale Geisteswissenschaften* 10. ISSN: 2510-1358, accessed July 28, 2025. DOI: 10.17175/2025_008. https://zfdg.de/2025_008.

Pollin, Christopher, Martina Scholger, Patrick Sahle, Georg Vogeler, Torsten Schaßan, Stefan Dumont, Franz Fischer, et al. 2024. “Workshop Generative KI, LLMs und GPT bei digitalen Editionen” [in deu] (March). Accessed July 28, 2025. https://zenodo.org/records/10893761.

Pollin, Christopher, Christian Steiner, and Constantin Zach. 2023. “New Ways of Creating Research Data: Conversion of Unstructured Text to TEI XML using GPT on the Correspondence of Hugo Schuchardt with a Web Prototype for Prompt Engineering. FORGE 2023. Tübingen” (October). Accessed July 28, 2025. DOI: 10.5281/ZENODO.8425162. https://zenodo.org/record/8425162.

Ponte, Jay M., and W. Bruce Croft. 1998. “A language modeling approach to information retrieval” [in en]. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval,* 275–281. Melbourne Australia: ACM, August. ISBN: 9781581130157, accessed July 27, 2025. DOI: 10.1145/290941.291008. https://dl.acm.org/doi/10.1145/290941.291008.

RAG vs Traditional QA - GeeksforGeeks. 2025, July. Accessed July 28, 2025. https://web.archive.org/web/20250728131605/https://www.geeksforgeeks.org/nlp/rag-vs-traditional-qa/.

Rajpurkar, Pranav, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. *SQuAD: 100,000+ Questions for Machine Comprehension of Text.* ArXiv:1606.05250, October. Accessed July 28, 2025. DOI: 10.48550/arXiv.1606.05250. http://arxiv.org/abs/1606.05250.

Ramos-Varela, Samuel, Jaime Bellver-Soler, Marcos Estecha-Garitagoitia, and Luis Fernando D'Haro. 2025. "Context or Retrieval? Evaluating RAG Methods for Art and Museum QA System." In *Proceedings of the 15th International Workshop on Spoken Dialogue Systems Technology,* edited by Maria Ines Torres, Yuki Matsuda, Zoraida Callejas, Arantza del Pozo, and Luis Fernando D'Haro, 129–136. Bilbao, Spain: Association for Computational Linguistics, May. ISBN: 9798891762480, accessed July 19, 2025. https://aclanthology.org/2025.iwsds-1.10/.

Riedl, Mark. 2023. *A Very Gentle Introduction to Large Language Models without the Hype* [in en], May. Accessed July 26, 2025. https://mark-riedl.medium.com/a-very-gentle-introduction-to-large-language-models-without-the-hype-5f67941fa59e.

Salcuni, Giuseppe Pio. 2025. "Utilizzo di tecniche RAG per la Valutazione e Comparazione dei Modelli LLM in ambito medico" [in it]. Master's thesis, Ingegneria Informatica, Università di Bologna, February. Accessed July 21, 2025. https://amslaurea.unibo.it/id/eprint/33387/.

Seo, Wonduk, Zonghao Yuan, and Yi Bu. 2025. *ValuesRAG: Enhancing Cultural Alignment Through Retrieval-Augmented Contextual Learning.* ArXiv:2501.01031, May. Accessed July 23, 2025. DOI: 10.48550/arXiv.2501.01031. http://arxiv.org/abs/2501.01031.

Sergeev, Alexander, Valeriya Goloviznina, Mikhail Melnichenko, and Evgeny Kotelnikov. 2025. *Talking to Data: Designing Smart Assistants for Humanities Databases.* ArXiv:2506.00986, June. Accessed July 19, 2025. DOI: 10.48550/arXiv.2506.00986. http://arxiv.org/abs/2506.00986.

Sharma, Kartik, Peeyush Kumar, and Yunqing Li. 2024. *OG-RAG: Ontology-Grounded Retrieval-Augmented Generation For Large Language Models.* ArXiv:2412.15235, December. Accessed July 23, 2025. DOI: 10.48550/arXiv.2412.15235. http://arxiv.org/abs/2412.15235.

Skarlinski, Michael D., Sam Cox, Jon M. Laurent, James D. Braza, Michaela Hinks, Michael J. Hammerling, Manvitha Ponnapati, Samuel G. Rodriques, and Andrew D. White. 2024. *Language agents achieve superhuman synthesis of scientific knowledge.* Accessed June 19, 2025. DOI: 10.48550/ARXIV.2409.13740. https://arxiv.org/abs/2409.13740.

Soman, Sumit, and Sujoy Roychowdhury. 2024. *Observations on Building RAG Systems for Technical Documents.* ArXiv:2404.00657, March. Accessed July 19, 2025. DOI: 10.48550/arXiv.2404.00657. http://arxiv.org/abs/2404.00657.

Streamlit Documentation v1.47.0. 2025, July. Accessed June 19, 2025. https://docs.streamlit.io/.

Svikhnushina, Ekaterina, and Pearl Pu. 2023. *Approximating Online Human Evaluation of Social Chatbots with Prompting.* ArXiv:2304.05253, August. Accessed August 3, 2025. DOI: 10.48550/arXiv.2304.05253. http://arxiv.org/abs/2304.05253.

Thakur, Nandan, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. *BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models.* ArXiv:2104.08663, October. Accessed July 28, 2025. DOI: 10.48550/arXiv.2104.08663. http://arxiv.org/abs/2104.08663.

Tiwari, Yash, Owais Ahmad Lone, and Mayukha Pal. 2025. *OntoRAG: Enhancing Question-Answering through Automated Ontology Derivation from Unstructured Knowledge Bases.* Accessed July 23, 2025. DOI: 10.48550/ARXIV.2506.00664. https://arxiv.org/abs/2506.00664.

Text REtrieval Conference (TREC) QA Data. Accessed July 26, 2025. https://trec.nist.gov/data/qa.html.

Upbin, Bruce. 2013. *IBM's Watson Gets Its First Piece Of Business In Healthcare* [in en], February. Accessed July 27, 2025. https://www.forbes.com/sites/bruceupbin/2013/02/08/ibms-watson-gets-its-first-piece-of-business-in-healthcare/.

Vaibhav, Fanindra Mahajan. 2025. "Retrieval-augmented generation: The technical foundation of intelligent AI Chatbots." *World Journal of Advanced Research and Reviews* 26, no. 1 (April): 4093–4099. ISSN: 25819615, accessed July 23, 2025. DOI: 10.30574/wjarr.2025.26.1.1571. https://journalwjarr.com/node/1453.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. *Attention Is All You Need.* Accessed June 19, 2025. DOI: 10.48550/ARXIV.1706.03762. https://arxiv.org/abs/1706.03762.

Voorhees, E., D. K. Harman, and National Institute of Standards and Technology (U.S.), eds. 2005. *TREC: experiment and evaluation in information retrieval.* Digital libraries and electronic publishing. Cambridge, Mass: MIT Press. ISBN: 9780262220736.

Wang, Liang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024. *Text Embeddings by Weakly-Supervised Contrastive Pre-training.* ArXiv:2212.03533, February. Accessed July 25, 2025. DOI: 10.48550/arXiv.2212.03533. http://arxiv.org/abs/2212.03533.

Wang, Xiaohua, Zhenghua Wang, Xuan Gao, Feiran Zhang, Yixin Wu, Zhibo Xu, Tianyuan Shi, et al. 2024. *Searching for Best Practices in Retrieval-Augmented Generation.* ArXiv:2407.01219, July. Accessed July 18, 2025. DOI: 10.48550/arXiv.2407.01219. http://arxiv.org/abs/2407.01219.

Wiki GNA. 2025, August. Accessed August 3, 2025. https://web.archive.org/web/20250803092155/https://gna.cultura.gov.it/wiki/index.php/Pagina_principale.

Question answering [in en]. 2025. Page Version ID: 1293783575, June. Accessed July 26, 2025. https://en.wikipedia.org/w/index.php?title=Question_answering&oldid=1293783575.

Wilensky, Robert, David N. Chin, Marc Luria, James Martin, James Mayfield, and Dekai Wu. 1988. "The berkeley UNIX consultant project." *Comput. Linguist.* (Cambridge, MA, USA) 14, no. 4 (December): 35–84. ISSN: 0891-2017.

Woods, William, Ronald M. Kaplan, and Bonnie L. Webber. 1972. *The Lunar Sciences Natural Language Information System: Final Report* [in en]. Technical report 2378. Cambridge, Massachusetts: Bolt Beranek and Newman Inc., June. https://www.researchgate.net/publication/24285293_The_Lunar_Sciences_Natural_Language_Information_System.

Xiao, Jinfeng, Linyi Ding, James Barry, Mohab Elkaref, Geeth De Mel, and Jiawei Han. 2024. "ORAG: Ontology-Guided Retrieval-Augmented Generation for Theme-Specific Entity Typing" [in en]. August. Accessed July 23, 2025. https://openreview.net/forum?id=cKBmZ2PZ6c.

Xiong, Caiming, Victor Zhong, and Richard Socher. 2018. *Dynamic Coattention Networks For Question Answering.* ArXiv:1611.01604, March. Accessed July 26, 2025. DOI: 10.48550/arXiv.1611.01604. http://arxiv.org/abs/1611.01604.

Yang, Rui, Michael Fu, Chakkrit Tantithamthavorn, Chetan Arora, Lisa Vandenhurk, and Joey Chua. 2025. "RAGVA: Engineering retrieval augmented generation-based virtual assistants in practice" [in en]. *Journal of Systems and Software* 226 (August): 112436. ISSN: 01641212, accessed July 23, 2025. DOI: 10.1016/j.jss.2025.112436. https://linkinghub.elsevier.com/retrieve/pii/S0164121225001049.

Yang, Zhilin, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2020. *XLNet: Generalized Autoregressive Pretraining for Language Understanding.* ArXiv:1906.08237, January. Accessed July 26, 2025. DOI: 10.48550/arXiv.1906.08237. http://arxiv.org/abs/1906.08237.

Yoon, Wonjin, Jinhyuk Lee, Donghyeon Kim, Minbyul Jeong, and Jaewoo Kang. 2019. *Pretrained Language Model for Biomedical Question Answering.* Accessed July 26, 2025. DOI: 10.48550/ARXIV.1909.08229. https://arxiv.org/abs/1909.08229.

Yue, Murong. 2025. *A Survey of Large Language Model Agents for Question Answering.* ArXiv:2503.19213, March. Accessed July 26, 2025. DOI: 10.48550/arXiv.2503.19213. http://arxiv.org/abs/2503.19213.

Zhou, Tianyu, Yuwei Wan, Ying Liu, and Maneesh Kumar. 2025. *Enabling interactive AI in industry 5.0 with RAG-enhanced GenAI Chatbots* [in en]. conference. Valencia, Spain. Accessed July 23, 2025. https://orca.cardiff.ac.uk/id/eprint/178617/.

Zhuang, Shengyao, and Guido Zuccon. 2021. "TILDE: Term Independent Likelihood moDEl for Passage Re-ranking" [in en]. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval,* 1483–1492. Virtual Event Canada: ACM, July. ISBN: 9781450380379, accessed July 27, 2025. DOI: 10.1145/3404835.3462922. https://dl.acm.org/doi/10.1145/3404835.3462922.