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: A Question-Answering System for Geoportale Nazionale Archeologia with Retrieval-Augmented Generation

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

At the confluence of artificial intelligence and digital humanities, this thesis explores the deployment of retrieval-augmented generation (RAG) to facilitate access to the *Geoportale Nazionale Archeologia (GNA)*, the Italian central repository of archaeological data. The study presents the design, implementation, and assessment of a dedicated question-answering system which integrates semantic embeddings, hybrid retrieval mechanisms, transformer-based language models, and user feedback loops into a modular pipeline.

Evaluation combined quantitative benchmarking with qualitative analysis by expert users, yielding results that underscore both the promise and the vulnerabilities of RAG in a cultural heritage context. The system improved access to procedural guidelines and technical documentation, accompanied by a reduction in misleading or extraneous information. Nonetheless, experiments revealed sensitivity to document structure and inconsistencies in provenance tracking, together with the challenge of balancing computational efficiency against contextual fidelity.

Far from claiming semantic comprehension, the system positions itself as a mediating tool that orients archaeologists and heritage professionals within vast textual corpora, surfacing relevant passages and easing interpretive navigation. Beyond the archaeological domain, its broader significance emerges in demonstrating how technical innovation intersects with infrastructural limitations and ethical imperatives. In this light, AI appears not as a surrogate for scholarly judgment, but as a means of extending humanistic inquiry and fostering renewed modes of interpretation and engagement across the digital humanities.

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

Acknowledgments

As I reach the end of this master's path, I find myself filled with a mixture of emotions. There is gratitude for all I have learned, admiration for those who have inspired me, joy for the friendships I have made, and, at the same time, some (*read:* a lot of!) fear of what comes next. These two years have been transformative, and I am deeply thankful for the experiences that have broadened my perspective.

My genuine appreciation is reserved to the professors and researchers whose work and teaching were inspirational, for their knowledge has encouraged me to think critically and grow with curiosity. In particular, I would like to express my gratitude to my thesis supervisor, Prof. Giovanni Colavizza, whose advice throughout this work was pivotal in shaping its outcome, and to my co-supervisor, Prof. Paolo Bonora, for his generous guidance. I am equally grateful to Mario Caruso and Simone Persiani from BUP Solutions, whose supervision during my internship was invaluable. I thank them for their precious mentorship, collaboration and patience – from all the calls and technical discussions to the debugging sessions I had the opportunity to take part in –, for sparking the very beginning of this project and making it possible, and above all for enriching my learning experience. My thanks also go to Ada Gabucci from the GNA for her kind availability during the evaluation stage of the final system – which by now feels less like a chatbot and more like a creature I've raised.

I am deeply grateful to my family for their constant encouragement and steadfast love. To *mamma* Serenella, and to Gabriele: thank you for being my safe harbour, a place of care and steadiness. To my father Fausto, whose loving memory always remains as a lighthouse along my way, I owe a source of inspiration that silently sustains my path. I wish to extend my heartfelt gratitude to the other relatives, spread across the Pianura Padana and Toscana, whose support has accompanied me along the way too.

I would like to create some space as well to say thank you to my peers and to all the wonderful people I have met along the way at DHDK. The friendships formed across countries and continents are among the truest gifts of this journey, and I cherish them with affection. To the friends from the P&R House, the command center of our countless projects and study sessions, and our harvest ground when we simply needed to have fun together and forget for a while – thank you for making these moments brighter. Thanks are due also to all the many collaborators in the projects we developed together, for the energy, creativity, and fellowship you brought to every step.

Of course, to Maicol, for being such a true and dedicated friend of the rarest kind, your faithfulness is a gift beyond measure. My renewed thanks to both old and new friends, near and far, who have shared in laughter, motivation, and companionship throughout this venture. Each of you has enriched these years in ways I will always carry with me.

Finally, I want to acknowledge the quiet but powerful lessons that come with facing uncertainty. Stepping into programming, coding, and even revisiting maths after years immersed in Art History was anything but easy; it must be said, trading brushstrokes and artists biographies for algorithms and tensors felt, to say the least, as an uphill run. In my personal opinion, these challenges are at the core of the hurdles many humanities students experience when approaching the digital humanities – a field deeply interdisciplinary in nature, yet one where technical skills often give an advantage to those coming from STEM backgrounds. And yet, we persevere. I will always remember the hesitation we felt when modelling our first knowledge graph, as well as the gratification of printing our first simple "Hello world". While the future can still feel daunting, this journey has given me tools, courage, and a true sense of belonging, enough to embrace what lies ahead with hope and ambition.

List of Figures

List of Tables

List of Code Examples

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. Among these, 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 of engaging with extensive documentation and complex repositories. This thesis arises directly from an applied research experience conducted during an internship at BUP Solutions, aimed at exploring the realistic feasibility and effectiveness of AI technologies in the context of cultural heritage. Specifically, the project focused on the desing, implementation and evaluation of a specialised QAS for the *Geoportale Nazionale Archeologia (GNA)*, Italy's primary repository of archaeological data under the auspices of ministerial authorities. For clarity, throughout this work the implemented system will be referred to interchangeably as the "GNA QA system" or the "GNA AI assistant".

The motivation of the present study stemmed from a practical challenge: facilitating efficient, intuitive, and accurate access to the extensive and often fragmented body of archaeological documentation hosted by the GNA. Archaeologists, heritage professionals and scholars working with this resource frequently face difficulties in navigating vast volumes of intricate technical reports, field notes, procedural guidelines, and complex geospatial data. In response, the project experimented with applying cutting-edge NLP and machine learning (ML) techniques – primarily transformer-based language models combined with advanced retrieval methods – to dynamically locate and synthesise relevant information based on user queries expressed in natural language.

Central to the chosen methodology is 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, systems grounded in RAG integrate external document retrieval with generative text production, resulting in greater reliability and outputs tethered in evidentiary contextual material – 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 idealised 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 resolutions to practical constraints such as computational limits, the absence of standardised evaluation benchmarks, and the structural complexity of the domain. This process brought to light the persistent tension between the ambitions of AI 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 propel us before we run up against the unique subtleties of human knowledge and practice? Here, McDermott's essay *Artificial Intelligence Meets Natural Stupidity* offers a timely reminder, warning against the lure of *wishful mnemonics* in AI and urging us to resist this inflationary language and 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 an equally frank acknowledgment of where their true limits lie. Only through such intellectual honesty can the field avoid self-delusion and maintain its credibility (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 work remains, at its heart, fundamentally hopeful. It demonstrates that even in the face of inherent methodological challenges, AI-driven tools carry genuine promise for enhancing access to cultural heritage resources. By presenting both the achievements and the limitations encountered along the way with transparency, this thesis seeks to contribute to the ongoing dialogue between AI and the humanities, offering a vision of AI's evolving role as a catalyst for new forms of stewardship, interpretation, and engagement with cultural heritage.

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 task. 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).

[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).

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 prioritised 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 direction 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.

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 generalise 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).

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

- **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 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).

[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).

[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

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

During the 2000s and early 2010s, QA began to move beyond brittle rule-based systems. Instead of relying on hand-crafted heuristics alone, researchers increasingly turned to statistical methods capable of reasoning over large corpora. N-gram models and statistical IR techniques – e.g., TF-IDF, BM25 and probabilistic models[11] – provided the first real capacity to navigate and rank massive text collections with some measure of relevance. By weighting terms according to their frequency and informativeness, these models made it possible to automatically surface candidate passages from unstructured data, a crucial step in scaling QA systems to the size of the web, large repositories and archives (Manning et al., 2008).

A major milestone of this period was IBM's *Watson* system, which achieved notable success by winning the *Jeopardy!* challenge in 2011.[12] Watson's *DeepQA* architecture integrated hundreds of NLP, IR and ranking components, employing sophisticated pipelines to analyse and combine evidence from diverse sources (Ferrucci et al., 2011). However, despite its advanced design, *Watson* relied on non-generative methods; it synthesised and ranked candidate answers but did not generate free-form responses from scratch.

Simultaneously, semantic QA systems also matured, mapping natural language (NL) questions to structured queries – e.g., using SPARQL – 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 NL into formal queries over RDF data sources. Thanks to the usage of ontology-mapping and linguistic resources – e.g., WordNet (Miller, 1992) and BabelNet (Navigli et al., 2021) –, these approaches further bridged the gap between unstructured text and structured knowledge bases (Franco et al., 2020).

Throughout this period, feature engineering was the beating heart of QA. 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

[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

[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).

enhance answer extraction from retrieved texts. Hybrid QA systems also appeared, combining keywords-based IR methods for unstructured sources with knowledge-base querying for fact-based answers, thereby improving both coverage and precision (Antoniou and Bassiliades, 2022).

This period, although still extractive and feature-dependent, set the stage for what followed. It demonstrated that scaling QA required both statistical reasoning over large corpora and semantic mapping into structured resources. All the while, it highlighted the bottlenecks of hand-engineered systems: they were labour-intensive to build, brittle across domains, and ultimately limited in their ability to capture deeper semantic relations. The gradual introduction of distributed word representations toward the end of this period hinted at a new trail, one that would come fully into focus with the neural architectures of the late 2010s.

2.1.4 Early Neural and Generative Models (Late 2010s)

The late 2010s marked a profound transition, as QA systems began to absorb the lessons of neural representation learning. The introduction of distributed word embeddings – Word2Vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014) and similar models – shifted the paradigm from sparse statistical features to dense, continuous vector spaces. Instead of simply counting word overlaps, systems could now measure semantic proximity between terms, enabling them to recognise that "excavation" and "dig" refer to related concepts. This advance laid the groundwork for capturing meaning beyond surface forms, improving both retrieval and answer matching (Jurafsky and Martin, 2024).

Embedding representations enabled the rise of recurrent architectures, particularly recurrent neural networks (RNNs), long short-term memory (LSTM) networks, and gated recurrent units (GRUs), which for the first time allowed systems to process language as sequences rather than bags of words. These models could in principle carry information across multiple tokens, making them attractive for reading comprehension tasks where the relation between question and passage unfolds over several sentences. Benchmarking datasets such as SQuAD (Rajpurkar et al., 2016) and Natural Questions (Kwiatkowski et al., 2019) became testing grounds for these methods, with LSTM-based encoders achieving state-of-the-art results by aligning question and context representations. Yet, the limitations quickly became evident: RNNs were notoriously poor at handling long-range dependencies, leading to failures when reasoning was required across multiple sentences, paragraphs, or documents (Jurafsky and Martin, 2024).

Around this time, researchers also began to experiment with generative models for QA, drawing inspiration from machine translation. Encoder-decoder architectures offered the tantalising possibility of producing answers as free-form text instead of merely extracting spans from source documents. These early generative QA systems demonstrated that models could synthesise responses in natural language, opening the door to more conversational applications. However, their outputs were often unreliable. Many simply rephrased the input passage, hallucinated details not grounded in evidence, or failed to maintain coherence when stitching together information from multiple contexts (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 generalise 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 in place of synthesising 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 (KB) 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 systems largely inflexible and frail for QA purposes, 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 revolutionised neural QA. 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, greatly 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 – e.g., SQuAD –, BERT achieved unprecedented accuracy, reaching Exact Match and F1 scores above 85% and 87% respectively on the SQuAD 2.0 leaderboard, thus 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 specialised 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 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).

These breakthroughs ushered in a new research culture. QASs became systematically optimised at every stage, from tokenisation and embedding to retrieval and answer extraction (Farea and Emmert-Streib, 2025). At the same time, the flexibility of Transformers encouraged exploration into conversational QA, multi-turn dialogue, and domain-specific fine-tuning (Yue, 2025). Yet, for all their impact, transformer models still relied primarily on *parametric memory*: their knowledge remained bounded by the data seen during pre-training. This limitation set the stage for a new class of approaches designed to bridge the gap between static models and dynamic, real-world information needs.

2.4 Large Language Models, Agents and Modular Pipelines

Currently, a clear distinction emerges between "traditional" QA systems, primarily built upon general-purpose pretrained language models, 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 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, typically synthesising 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, fast-changing, or domain-specific queries, and are prone to hallucinations[13] and outdated information (Farea and Emmert-Streib, 2025).

Recent advances in question answering are characterised by the emergence of retrieval-augmented generation (RAG). In these pipelines, a retriever component dynamically accesses external knowledge bases, while a generator conditions on the retrieved information to produce grounded answers. This approach addresses many of the shortcomings of earlier transformer-based models and significantly enhances 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). Benchmarks show that RAG-enhanced models significantly outperform standard LLMs in factual QA, particularly in do-

[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).

mains demanding precise recall or up-to-date knowledge. For instance, enterprise evaluations demonstrate up to 30-40% improvement in incorporating domain-specific terminology compared to standalone models, while user trust increases substantially when source citations are included (Vaibhav, 2025).

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 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. However, these benefits come with trade-offs. RAG architectures require more complex infrastructures, including document indexing and retrieval mechanisms, which increase operational overhead and latency compared to traditional, static QA systems. As a result, deploying and maintaining RAG-based systems can be more challenging, especially at scale.

Beyond RAG, a parallel evolution is visible in the emergence of LLM-based agents. Unlike monolithic models, these agents operate as orchestrators of multi-stage reasoning, combining planning, question understanding, retrieval, reasoning, and answer generation in an iterative loop. Architectures typically integrate memory to retain conversational context, planning modules to decide on next actions, and reasoning modules to balance internal "thinking" with external interactions, such as calling APIs, querying databases, or consulting humans (Yue, 2025). It overcomes the rigidity of earlier pipelines, which relied on static submodules trained in isolation, and the limitations of naive LLM QA, which lacked external grounding and dynamic adaptability. Agents thus introduce a form of controlled autonomy: they not only retrieve information but also decide *when* and *how* to engage tools, creating more flexible and resilient QA systems.

Current research highlights that these modular, agentic pipelines offer more than incremental improvements. They introduce transparency through source attribution, factual grounding, and explainability – qualities increasingly demanded in high-stakes domains such as law, medicine, and cultural heritage. At the same time, promising directions include multimodal modes – retrieving from text, images, or audio –, though cross-modal alignment remains an open challenge (Vaibhav, 2025); hybrid retrieval that combines sparse lexical methods with dense search; and adaptive systems that dynamically tune retrieval and reasoning strategies based on query type and context (Yue, 2025; Vaibhav, 2025). Taken together, these advances point toward a decisive shift: from static models locked within their parametric memory to dynamic, agentic systems capable of interacting with and reasoning over the evolving universe of human knowledge.

Tab. 1 summarises the functional differences between traditional and RAG-based QA systems, highlighting the shift toward dynamic, retrieval-augmented, and generative approaches that characterise 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 (synthesised)
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 1: Comparison of traditional vs. retrieval-augmented generation question-answering systems.
Adapted from https://www.geeksforgeeks.org/nlp/rag-vs-traditional-qa/
.

The main stages in the evolution of QA systems, along with representative approaches and landmark examples, are summarised in Tab. 2.

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 2: Evolution of question-answering systems

.

Chapter 3

State of the Art in Retrieval-Augmented Generation

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 discussed in Chap. 2, LLMs are prone to hallucinations, 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 organisational knowledge (Vaibhav, 2025), posing challenges for domains like cultural heritage, GLAM (Galleries, Libraries, Archives and Museums) and archaeology, where reliability, provenance, and interpretive consistency 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, through the integration of external and verifiable knowledge at inference time, thereby reducing the risk of generating fabricated or distorted information (Martineau, 2023). This approach marks a clear progression beyond both traditional IR techniques and earlier neural QA models, which were often brittle, domain-dependent, or struggled to adapt to evolving information needs.

Although initially conceived for open-domain question answering and enterprise search (Akkiraju et al., 2024; Jiang et al., 2024; Packowski et al., 2024; R. Yang et al., 2025; Zhou et al., 2025), RAG pipelines are are now finding growing resonance in the humanities and cultural heritage domains. In these settings, where interpretive rigour, provenance, and

[14]The terminology "retrieval-augmented generation" was introduced by Lewis et al. (2020) in their influential paper *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. Since then, the term has come to designate a broad family of methods and design patterns that combine retrieval with generative techniques, unifying diverse approaches to knowledge-grounded text generation.
For more information about RAG technique, see https://en.wikipedia.org/wiki/Retrieval-augmented_generation.

reliability of information are critical, they serve as valuable instruments to support scholars and professionals in navigating vast, fragmented knowledge repositories. Recent initiatives have begun to experiment with RAG for the analysis of sensitive historical materials (Callaghan and Vieira, 2025; Ciletti, 2025; Sergeev et al., 2025; Fan et al., 2025), underscoring its potential to support critical scholarly practices. However, the present work explores a distinct application: improving access to procedural and technical documentation, where clarity, consistency, and actionable guidance are the primary objectives.

This chapter provides a comprehensive account of the state of the art in retrieval-augmented generation, situating it within the broader research landscape, clarifying its core mechanisms, and tracing its emerging applications in the digital humanities.

3.1 Foundations of the Technique

Retrieval-augmented generation (RAG) has emerged as a hybrid paradigm that tackles some of the most persistent shortcomings of LLMs, such as knowledge staleness, narrow scope of their context windows, and difficulty of tracing outputs back to their sources (Vaibhav, 2025; Y. Gao et al., 2024; Gupta et al., 2024). Although LLMs excel at producing fluent, human-like text, they often falter when facing specialised queries or requests for information that falls 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 (Wang, Wang et al., 2024).

At its core, a typical RAG pipeline 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 dedicated database 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[15] – often based on Transformer architectures (Vaswani et al., 2017) – which synthesises

[15]In the domain of LLMs, *context* carries a dual meaning. On one hand, it denotes the context window, i.e. the maximum sequence length of tokens that the model can process at once, which determines how much information can be considered simultaneously. On the other hand, it refers to the context prompt, i.e. the specific contextual material provided to the model at inference time, including retrieved evidence, instructions, and user queries. Both are critical in RAG pipelines, since the size of the context window constrains how many retrieved chunks can be incorporated, while the design of the context prompt influences how effectively the

a response that blends the original query with external evidence, producing answers that are both coherent and contextually appropriate (Arslan et al., 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 systematise the RAG pipeline, with frameworks delineating specific interdependent modules such as query classification, retrieval, reranking, and generation (Wang, Wang et al., 2024; Y. Gao et al., 2024).

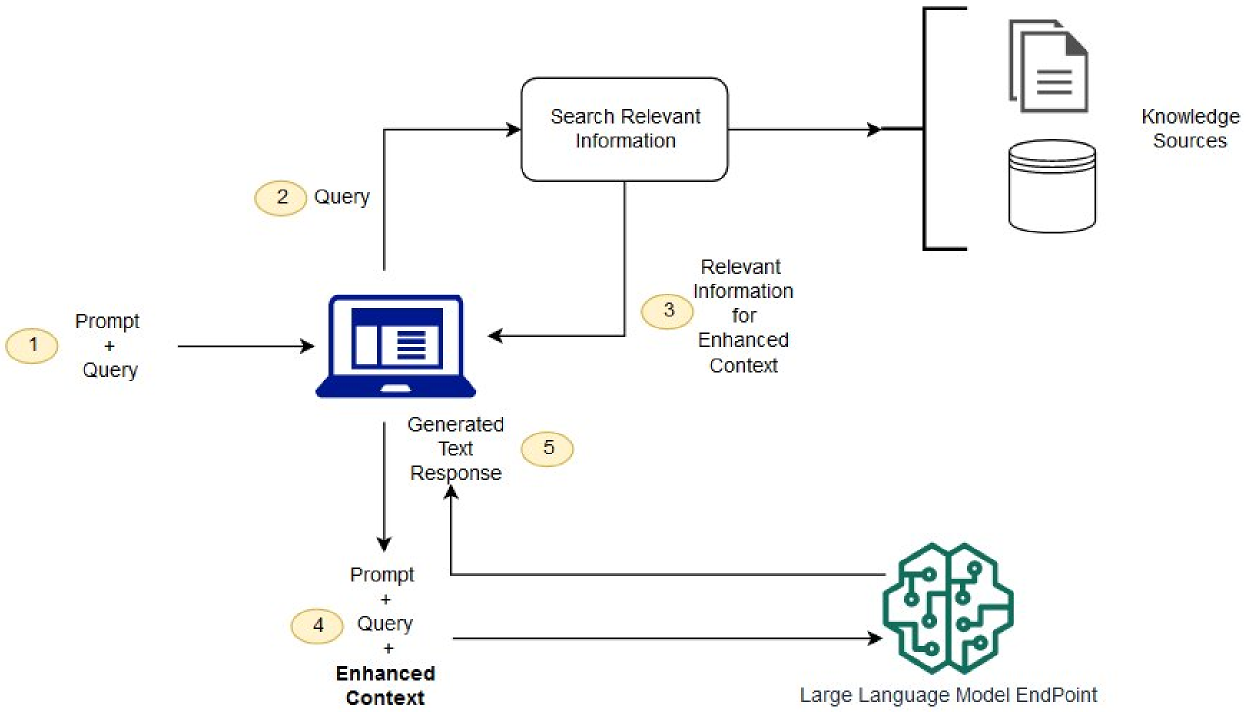


Figure 1: Retrieval-augmented generation pipeline combining user queries with external knowledge to produce context-aware responses.
Source: https://aws.amazon.com/de/what-is/retrieval-augmented-generation/.

3.1.1 Pipeline Components and Common Practices

The design of RAG pipelines can vary considerably depending on the specific use case, domain of application, and resources available. Still, a number of recurring practices have gradually crystallised into what can be regarded as the current state of the art in retrieval-augmented generation (Vaibhav, 2025; Wang, Wang et al., 2024; Arslan et al., 2024; Y. Gao et al., 2024; Gupta et al., 2024). These practices often serve as reference blueprints rather than rigid prescriptions, since not every component needs to be implemented in every system. Instead, they represent a modular design space, where specific strategies can be combined, adapted or

model grounds its outputs in external knowledge.

omitted to suit particular tasks.

In its most typical configuration, a RAG pipeline comprises the following modules:

- **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 language model alone suffices. This step relies on natural language understanding (NLU) techniques to extract key entities, relationships, and user intent, thereby improving efficiency and reducing unnecessary retrieval latency.

- **Document indexing and chunking.** Raw data from source documents is preprocessed, cleaned, segmented into smaller chunks at token, sentence, or semantic level, and converted into dense vector representations called 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 (L. Gao et al., 2022).

- **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 vector databases – e.g., Milvus, Faiss, Qdrant –,[16] selected based on scalability, indexing strategy, and support for specific search capabilities.

- **Retrieval and query transformation.** When a user submits a query, it is first encoded into a vector representation and used to retrieve the top-k relevant chunks from the indexed KB via similarity search. To make it more robust, the pipeline can adopt a hybrid approach based on dense retrieval – embedding-based methods such as DPR or Contriever – combined with sparse retrieval – lexical methods such as BM25. Retrieval can be further improved through query transformation strategies, including rewriting, decomposition, or the generation of hypothetical supporting documents – e.g., HyDE.

- **Reranking.** the initial candidate set can be reordered to emphasise relevance to the original query. This is frequently achieved with cross-encoder models – such as monoT5,

[16] Vector databases such as Milvus and Qdrant extend similarity search with native metadata management and distributed scalability. Milvus, built for large-scale applications, focuses on cloud-native deployment and support for multiple indexing strategies, making it suitable for enterprise environments (Wang et al., 2021; Guo et al., 2022). Qdrant, written in Rust, is designed for high-performance real-time search and offers particularly strong metadata filtering capabilities (Qdrant,) Faiss (Facebook AI Similarity Search) is a high-performance library developed by Meta AI that focuses on efficient vector indexing and similarity search without built-in metadata handling (Douze et al., 2024). In research contexts, these characteristics can be advantageous: Faiss offers speed, flexibility, and a wide choice of indexing methods, while metadata can be managed externally (e.g., through a separate mapping file), allowing full control over the experimental pipeline.

monoBERT, or RankLLaMA –, which jointly consider the query and each candidate, or with more sophisticated algorithms through heuristics. This contextualization ensures that the most pertinent information is prioritised for the generative model.
See Tab. 3 for a summary on reranking techniques.

- **Repacking and summarization.** In some cases, retrieved passages may be reorganised or summarised to distil key information, especially when dealing with lengthy corpora. This step can involve extractive summarization or abstractive – e.g., with Pegasus – 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 – synthesises 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 presents an overview of the methods most consistently reported as high-performing for each module of a RAG pipeline. When aiming for balanced efficiency – i.e., reducing latency while maintaining good, but not maximal, accuracy – adjustments are typically made at the retrieval and reranking stages. In practice, this involves replacing the Hybrid + HyDE retrieval method with a standard Hybrid search approach, which combines BM25 and dense retrieval without pseudo-document generation, and substituting monoT5 with TILDEv2 for reranking, which delivers faster processing at the cost of a modest reduction in answer quality (Wang, Wang et al., 2024).

Algorithm	**Rationale**
Cross-encoder rerankers	Jointly encode concatenated query-document pairs to produce fine-grained relevance scores. These models – e.g., monoT5, monoBERT, RankLLaMA – are fine-tuned to classify relevance as "true" or "false", and at inference, documents are ranked by the predicted probability of the "true" label (Wang, 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) Point wise:** predicts relevance score for each document independently; **b) Pair wise:** compares pairs of documents to learn relative relevance; **c) List wise:** 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 (Wang, 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 LLM-based reranker (good balance of quality and speed).
Chunking	Small2big / Sliding Windows	Organising 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 configurations across modules, selected for maximising performance w.r.t. answer quality and accuracy.

Fig. 2 provides a broader overview of the RAG ecosystem. The paradigm evolves from naive to modular RAG, incorporating techniques for improving retrieval and generation – e.g., chunk optimization, adaptive retrieval, dual fine-tuning –, as well as the key issues of when, what, and how to retrieve. RAG also faces open challenges, including robustness, scaling laws, production readiness, alongside modality extensions to image, audio, video, code, and ecosystem directions like customization, specialization. Finally, current evaluation targets and frameworks distinguish between retrieval quality and generation quality, together with their assessment aspects such as answer relevance, context relevance, faithfulness, robustness, and integration (Y. Gao et al., 2024).

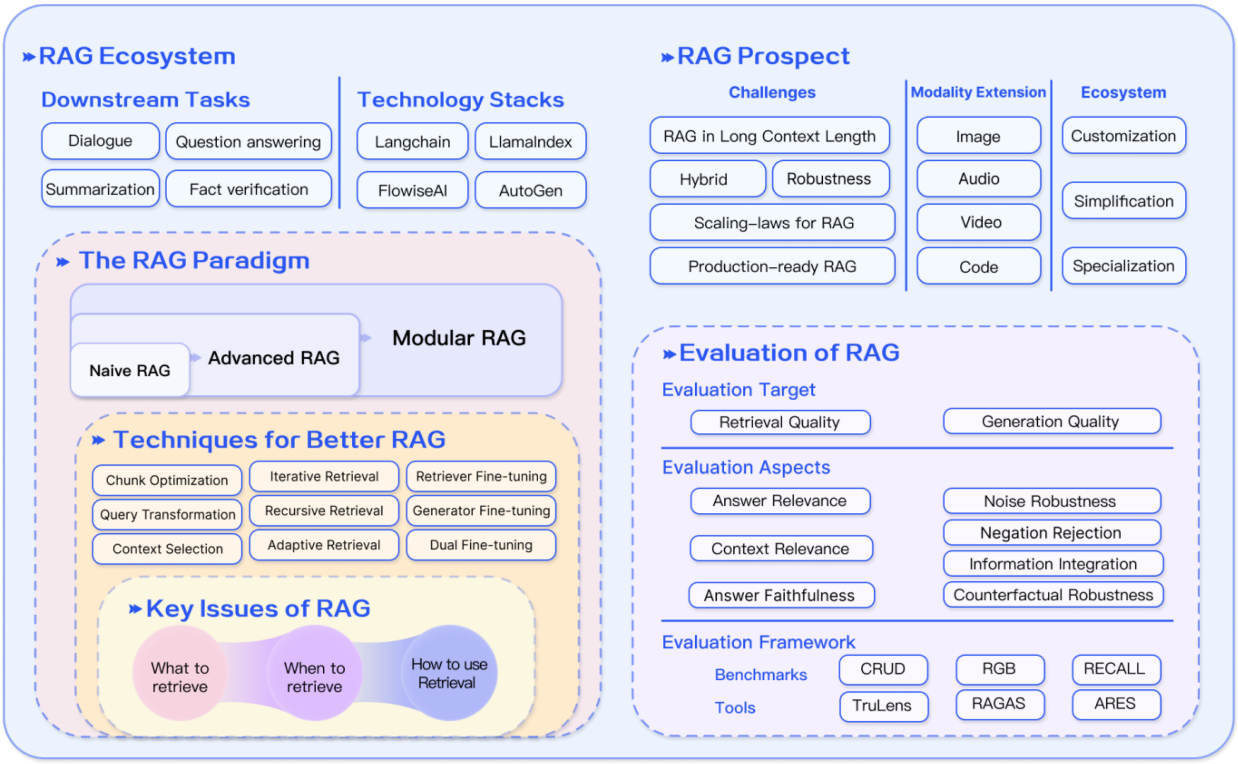


Figure 2: Summary overview of the RAG ecosystem.
Source: Y. Gao et al., 2024.

3.1.2 Evaluation and Benchmarking

Evaluating RAG systems poses unique challenges, as traditional metrics like BLEU, ROUGE or METEOR may not fully capture the quality of generated responses, particularly in terms of factual accuracy and contextual relevance (Deriu et al., 2020). In conversational QA research, this limitation has long been acknowledged, as multi-turn settings require models to answer a single query and to maintain consistency, resolve co-references, and adapt to evolving conversational context (Zaib et al., 2022). Consequently, evaluation must go beyond surface level overlap and incorporate measures that reflect contextual appropriateness and understanding at discourse level. Moreover, assessing RAG systems requires attention to both retrieval and generation quality, since failures in either stage directly impact the final response (Abeysinghe and Circi, 2024).

Surveys of the field increasingly stress the need for frameworks capable of measuring multiple dimensions of RAG. Despite the rapid advancements in retrieval, generation, and augmentation, evaluation methods remain underdeveloped, with persistent challenges in capturing retrieval quality, hallucination rates, and faithfulness of generated content in a systematic way (Y. Gao et al., 2024). Large-scale benchmarks such as TREC (Voorhees et al., 2005), MS MARCO

(Bajaj et al., 2018) and BEIR (Thakur et al., 2021) continue to be standard for retrieval, but generation quality requires different approaches. Newer frameworks like the *Retrieval-Augmented Generation Assessment System (RAGAS)* have therefore been introduced to capture aspects such as contextual alignment, answer faithfulness, and pipeline-level performance (Es et al., 2023).

Experimental studies of RAG pipelines within LLM-based applications further demonstrate the complexity of the task. For example, a recent work compares automated metrics, human evaluation, and LLM-based evaluation in the context of *EdTalk*, a RAG-powered chatbot built to navigate educational reports. Their findings show that automated metrics such as BLEURT are useful for rapid iteration but often misalign with human judgments, while factored human evaluation – structured around criteria including correctness, informativeness, relevance, clarity, and hallucination – provides richer insights into system performance. At the same time, LLM-based evaluators show promise for scalable assessment but risk inflating scores, especially when the same model is used for both generation and evaluation. This stresses the need for hybrid and carefully designed evaluation pipelines when deploying RAG in real-world contexts (Abeysinghe and Circi, 2024).

Human-in-the-loop assessment remains indispensable, with expert judges assessing criteria such as factual accuracy, coherence, and domain relevance – offering richer and often more reliable quality assessments than quantitative metrics alone. In conversational contexts, this is especially important, as metrics must reflect user satisfaction and interaction quality rather than isolated response correctness (Gupta et al., 2024).

Alongside automated and human-centred metrics, evaluation taxonomies are moving toward a more fine-grained view of system quality. Dimensions such as answer relevance, context relevance, and faithfulness directly address hallucination and grounding, while robustness dimensions – including resilience to noise, negation, counterfactual scenarios, and multi-source information integration – test whether systems remain reliable under real-world conditions. To operationalise these dimensions, new benchmarks such as CRUD, RGB, and RECALL extend beyond traditional IR settings by jointly assessing retrieval and generation. Complementary tools make evaluation continuous and developer-friendly: RAGAS targets faithfulness and contextual alignment, ARES offers flexible automated pipelines, and TruLens integrates performance monitoring into deployed workflows (Y. Gao et al., 2024).

Furthermore, ethical considerations must be embedded throughout the lifecycle of RAG pipelines. Protecting data privacy, mitigating algorithmic bias, and complying with regulations such as GDPR are fundamental. This means adopting technical measures like privacy by design, data minimisation, and access control, in addition to committing to broader ethical principles widely recognised in international guidelines: transparency, justice and fairness, non-maleficence and responsibility (Jobin et al., 2019). Ongoing audits, stakeholder involvement, and mechanisms for accountability, including whistleblowing and legal clarity, help bridge the gap between abstract principles and operational practice, ensuring systems remain trustworthy and socially beneficial (Ashery et al., 2025).

Overall, the line of evolution 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-centred feedback mechanisms, and adaptive optimisation strategies. Overcoming challenges related to factuality, scalability, and responsible deployment will be essential for building trustworthy systems capable of delivering high-quality information in context-sensitive settings. Looking ahead, continued advances in RAG are set to play a pivotal role in shaping the future of digital knowledge access and discovery, and establish it as a cornerstone technology (Zaib et al., 2022; Wang, Wang et al., 2024; Y. Gao et al., 2024).

3.2 New Frontiers Applications

RAG systems are increasingly deployed across a wide spectrum of contexts – spanning academic research, enterprise infrastructures, and real-world product environments – where they serve to enhance data accessibility, support decision-making, and facilitate natural language interaction with complex KBs. Recent surveys and empirical studies trace a rapidly expanding set of scholarly applications. In the context of academic support, for instance, retrieval-augmented pipelines power automated literature review tools and citation management platforms such as *LitLLM* (Agarwal et al., 2025), and *KNIMEZoBot* (Alshammari et al., 2023). In conversational AI, advanced retrieval enables grounded multi-turn dialogue, exemplified by *Wizard of Wikipedia (WoW)* (Dinan et al., 2019), which can be read as an early instantiation of what is now formalised as RAG-based dialogue modelling. The paradigm is also leveraged for large-scale summarisation, distilling insights across vast corpora of scholarly papers, and for fact verification tasks, as in resources like *PubHealth*, increasingly adopted to counter misinformation in sensitive domains such as health communication (Kotonya and Toni, 2020). In this regard, RAG has proven especially valuable in domain-specific knowledge extraction, most notably in biomedical and legal research, where retrieval mechanisms act as an extension of expert knowledge.

In one experiment, a RAG system was developed to assist data scientists through a combination of GROBID library[17] for structured bibliographic extraction, fine-tuned embeddings, semantic chunking, and an abstract-first retrieval strategy. The system's performance, assessed using 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 compound interdisciplinary searches and significantly improving document discoverability by synthesising 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 the efficacy of RAG systems in alleviating information overload and improving research workflow discoverability.

In parallel, further critical insights into the design of RAG systems for domain-specific and technical content have been highlighted in recent work (Soman and Roychowdhury, 2024), which closely aligns 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, 2025b), which applies customised chunking, semantic preprocessing, and contextual embedding strategies. Both studies advocate for more nuanced and contextually informed 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

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

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 Retrieval-Augmented Generation 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 (DSEs), RAG emerges as a promising method for addressing challenges such as entity linking (EL) and the integration of external knowledge sources. Notably, RAG is recognised 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 tool chains 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 standardised 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 the potential for RAG-augmented LLMs to enhance the precision, accessibility, and contextualization of information retrieval, and 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 prioritise semantic retrieval over lexical matching using dense embeddings – e.g., *E5* variants (Wang, Yang, Huang, Jiao et al., 2024) – to interpret queries in context, employ structured text processing pipelines to reduce noise in source materials, and apply optimised 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 contextual 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 specialised 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, summarise 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 evaluation, source citation, 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 rigour (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 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 DSEs, 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 contextually grounded, 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 catalysing demand for standardised, 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 synthesise heterogeneous, dispersed data, transforming scattered web archives, periodicals, and catalogues into knowledge spaces that are accessible and meaningfully structured. 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 and disciplines. 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: A 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).

Far from being a mere data aggregator, the GNA serves as a dynamic knowledge base, collecting digital contributions from professional archaeologists – especially those active in preventive archaeology – as well as from research groups, universities, and concession-holders. Its scope encompasses a wide spectrum of outputs, ranging from vector data based on QGIS[18] to reports, documentation packages, and datasets from academic and research projects. Data publication within the GNA is managed with attention to quality standards, intellectual property rights, and open-access principles, supported by the assignment of DOIs and distribution under Creative Commons licensing (CC-BY 4.0), ensuring both traceability and reusability (Acconcia, 2023; Falcone et al., 2023; Boi, 2023). The platform is also aligned with European and Italian open data and transparency regulations, fulfilling requirements of national FOIA provisions and EU directives[19] (Falcone et al., 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).

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

[19] 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 generalised 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.
Read more at https://foia.gov.it/normativa.

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 digitised 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 Geoportale Nazionale per l'Archeologia.

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

What makes this material especially demanding is the combination of heterogeneity in format, sources and regulatory frameworks. The GNA brings together preventive reports, excavation records, surveys, legacy archives, GIS layers, and remote sensing outputs, each with its own structure, level of detail, and standards of documentation. Their integration is further conditioned by legal and procedural frameworks, from FOIA transparency obligations to Creative Commons licensing and DOI assignment, which ensure accessibility but add additional layers of compliance. Equally significant are the operative guidelines and technical instructions that regulate how this information is produced, structured, and uploaded into the platform. For example, the MOPR (Modulo di Progetto) section provides step-by-step guidance for consultants preparing preventive archaeology reports, from the structuring of stratigraphic descriptions to the encoding of metadata fields (cf. Compilare il MOPR). Similarly, technical notes (cf. Brevi note su QGIS) illustrate the proper use of QGIS software in generating and validating shapefiles prior to submission, underscoring the centrality of geospatial data in contemporary archaeological practice.

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 cataloguing and sparked 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).

Today, the GNA serves as a centralised platform for a broad community of users: public administrators and government officials, who rely on it for regulatory oversight; professional archaeologists and cultural heritage consultants, who use it for research and field documentation; and stakeholders involved in public works, including national infrastructure planners, for whom it facilitates informed decision-making within the constraints of heritage protection.

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.

Central to the platform is a QGIS template, which standardises data entry and visualisation. This tool supports collaborative integration of local information into the national infrastructure, 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 consultants 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,[20] which allows users to directly download standardised 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 offers continuous operational support through a dedicated Help Desk service, coordinated by Ada Gabucci.[21] Users encountering technical challenges or seeking clarification on data entry procedures can contact the Help

[20] The GNA Plugin enables interaction with the platform to directly load data related to a specific Project Module (MOPR) into QGIS and to submit the Project Module back to GNA, making it quickly available to everyone. The address for the official repository is: https://gna.cultura.gov.it/qgis/plugins/plugins.xml.

[21] 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.
Source: https://web.archive.org/web/20250724081422/https://conf24.garr.it/it/speaker/ada-gabucci.

Desk for personalised assistance. Combined with the collaborative and evolving character of the manual, the Help Desk sustains a genuine community of practice, promoting the exchange of expertise, and nurtures the ongoing refinement of the platform's tools and resources.

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, the need emerged for a more intelligent and scalable support solution. To meet this demand, we developed an AI-powered information system in the form of a question-answering assistant, designed both to assist users directly and to alleviate the workload of the Help Desk. Drawing on the current state of AI, ML and DH methodologies – as discussed in Chap. 3 and especially Sec. 3.2 –, RAG was chosen as the most effective approach. This technology equips the GNA AI assistant to dynamically access the GNA corpus and produce answers that are precise, contextually grounded, and tailored to user needs.

4.2.1 Functional Requirements

Functional requirements specify the concrete capabilities the system must provide in order to meet the needs of its users and stakeholders, outlining the core actions through which it delivers value. These features are detailed as follows:

- **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 multipart queries.
- **Information retrieval:** the system must retrieve relevant passages or document segments from the GNA knowledge base, using vector similarity search over chunked content.
- **Answer generation:** the system must synthesise 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 anonymised, 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, minimising 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 labelling, 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.

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 (Pograri, 2025a) and advancing to a full-scale system with custom components optimised for resource efficiency (2025b).[22] In doing so, the chapter outlines the broader methodological framework and the concrete steps undertaken, offering an in-depth account of the design choices, technical architecture, data preparation, implementation, and evaluation processes.

5.1 Prototype

The initial prototype served as a proof-of-concept, developed to validate the feasibility of applying RAG to the case study and to identify early challenges. The system integrated a minimal pipeline built with LangChain[23] – an off-the-shelf tool widely adopted for coordinating retrieval and generation tasks (Mishra, 2024; Akkiraju et al., 2024) – to combine vector-based retrieval

[22] All the source code is available on GitHub.
Prototype repository:https://github.com/Asemica-me/chatw-GNA.
Full-scale system repository: https://github.com/Asemica-me/GNAvigator.

[23] LangChain is an open-source and Python-centric 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/.

and LLM generation within an interactive environment.

5.1.1 System Design

The prototype architecture (Fig. 3) consisted of:

- **knowledge base**: a CSV dataset created from the web-based operational manual of the Geoportale Nazionale dell'Archeologia (GNA). Content was preprocessed into document chunks enriched with metadata (titles, URLs, descriptions);
- **vector store**: semantic embeddings of chunks were computed using MistralAI API and then stored in a Faiss database;
- **generation language model**: the Mistral *NeMo* LLM was accessed via the same Mistral API for answer generation;
- **retrieval mechanism**: queries were embedded and matched against FAISS; all retrieved chunks were concatenated into a single context and passed to the Mistral model;
- **user interface**: a front-end implemented via Streamlit (Fig. 4) allowed users to input queries in natural language and visualise answers in a clean, accessible format.

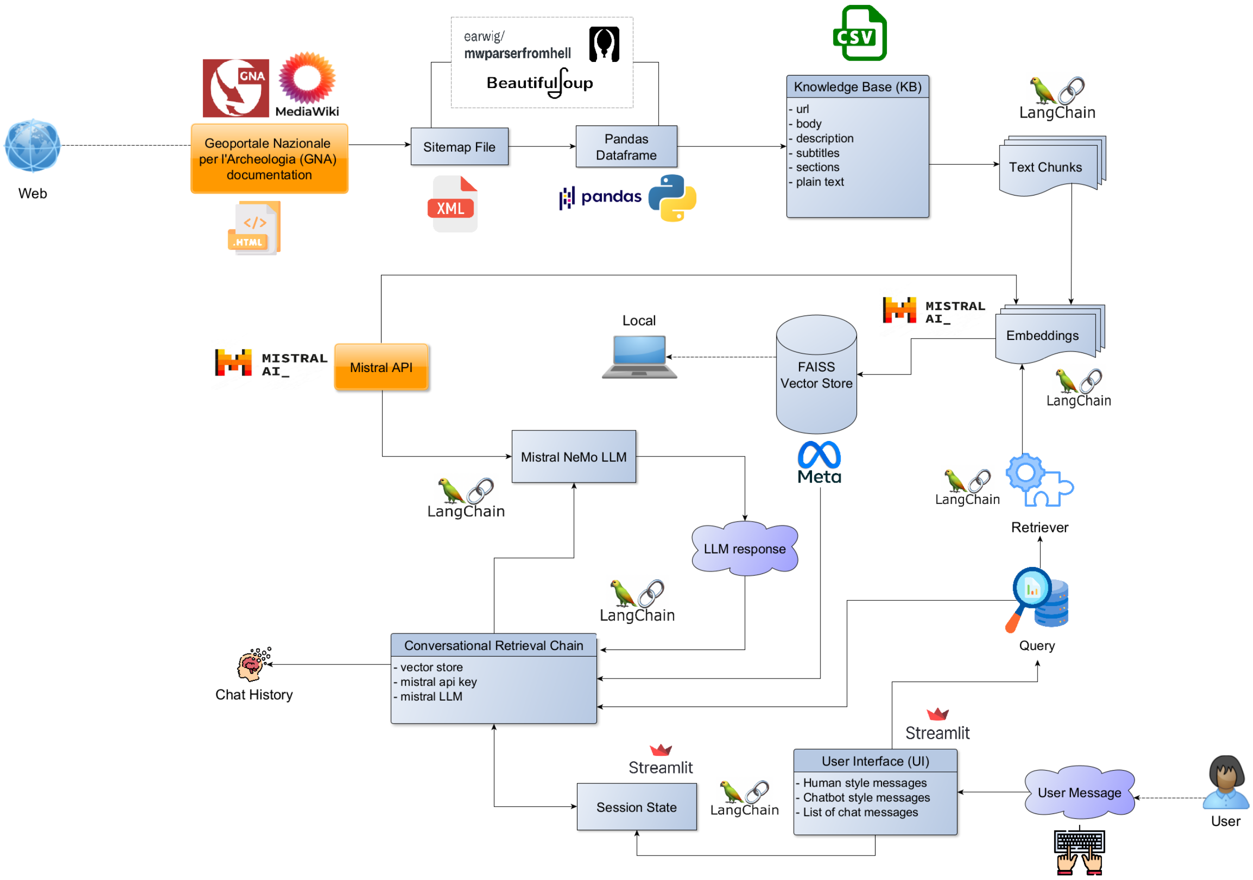


Figure 3: System architecture of the GNA AI assistant prototype.

Figure 4: User interface of the GNA QA system prototype deployed on Streamlit Community Cloud.

LangChain served as the orchestration layer, providing ready-made abstractions that connected data ingestion, retrieval, and generation into a single workflow (Fig. 5). Concretely, it handled prompt templating for the Mistral *NeMo* model, mediated the `Conversational Retrieval` flow, and maintained short-term dialogue state through `ConversationBufferMemory`. The retriever was based on a Faiss vector store, which indexed chunked representations of documents and returned the top-k semantically similar chunks given a user query. Retrieved chunks were then combined using LangChain's *stuff* method[24] (Listing 5.1), in which all documents are concatenated into a single prompt template and passed to the LLM.

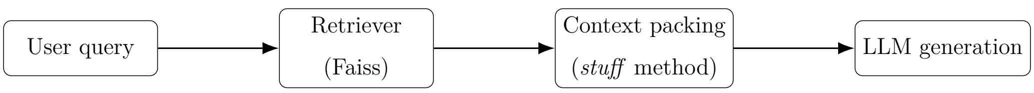


Figure 5: Prototype retrieval pipeline: from user query to generation via Faiss retrieval with LangChain *stuff* method and context packing.

The appeal, methodologically, privileged rapid deployment over optimisation, with the

[24] In LangChain, the stuff method is a particular abstraction for context packing. It concatenates the top-k retrieved documents and inserts the combined block into a single prompt template passed to the LLM. While computationally lightweight, it is limited by the model's context window. LangChain also provides alternative strategies such as *map-reduce*, *refine*, and *map-rerank* (Topsakal and Akinci, 2023).

primary aim of validating the pipeline. A single Pythonic framework glued together embedding creation, vector search, prompt assembly, and LLM calls, enabling a minimal viable product that validated feasibility before investing efforts in bespoke infrastructure.

```python
from mistralai import Mistral
from langchain_mistralai import ChatMistralAI
from langchain.memory import ConversationBufferMemory
from langchain.chains import ConversationalRetrievalChain
from langchain.prompts import (
    ChatPromptTemplate,
    HumanMessagePromptTemplate,
    SystemMessagePromptTemplate,
)

def get_conversation_chain(vector_store, api_key: str, model_name:
    str, system_message:str, human_message:str):
    """
    Create a conversational retrieval chain with a Mistral LLM.

    The chain uses LangChain's ConversationalRetrievalChain with the
    default "stuff" document combination strategy, concatenating the
    retrieved chunks into a prompt template.
    A ConversationBufferMemory maintains chat history across turns
    for stateful interaction.

    Args:
      vector_store: The vector database (Faiss) used for retrieval.
      api_key (str): API key for the Mistral LLM.
      model_name (str): Identifier of the Mistral model to be used.
      system_message (str): The system-level instructions (prompt).
      human_message (str): The user-facing prompt template (query).

    Returns:
      conversation_chain: A ConversationalRetrievalChain instance
      ready for query-answering with memory support.
      """

    llm = create_mistral_llm(api_key, model_name)

    # Configure memory for conversation history
    memory = ConversationBufferMemory(memory_key="chat_history",
       return_messages=True)

    conversation_chain = ConversationalRetrievalChain.from_llm(
        llm=llm,
        retriever=vector_store.as_retriever(),
```

```
        memory=memory,
        rephrase_question=False,
        combine_docs_chain_kwargs={
            "prompt": ChatPromptTemplate.from_messages(
                [
                    system_message,
                    human_message,
                ]
            ),
        },
    )
    return conversation_chain
```

Listing 5.1: Usage example in Python of the stuff method from LangChain.

At the same time, the prototyping phase exposed several framework frictions that informed the later redesign. One issue concerned redundant model calls within the LangChain stack – most notably the automatic query rephrasing in `ConversationalRetrievalChain` – which doubled LLM invocations and increased the risk of hitting API rate limits. This was mitigated through a simple rate-limiting guard with a one-second delay, but the solution remained suboptimal. A second problem arose with the retry logic in `MistralAIEmbeddings`, which failed to handle `HTTP 429` *Too Many Requests*[25] responses correctly due to a bug in LangChain's client error handling. This reduced robustness under bursty traffic until an upstream patch was introduced later on,[26] but the episode highlighted the limited control over low-level failure modes. Finally, reliance on predefined chains restricted experimentability by limiting chunk-level metadata control, and constraining the ability to vary context-packing strategies or introduce custom pre- and post-processing filters without abandoning the abstraction layer altogether.

These findings, which emerged during implementation and debugging, motivated a shift toward a more modular pipeline. In this redesign, chunking, vectorisation, and retrieval, were configured as distinct, observable stages, evaluated in isolation rather than hidden within predefined chains. This approach granted finer control over performance-critical components and made their effects more transparent.

[25]The HTTP 429 *Too Many Requests* status code signals that the client has exceeded the allowable number of requests within a specified timeframe. This response enforces what is commonly known as *rate limiting*, instructing the client to reduce its request frequency.

[26]Pull request addressing the incorrect exception handling for rate limiting in `MistralAIEmbeddings` emerged in LangChain PR#29242 and it was released in the subsequent version (January 2025).

5.1.2 Evaluation

The evaluation of the system's prototype followed a pronged path:

1. **Human assessment**: responses were manually annotated on a 5-point Likert scale for consistency, fluency, completeness, and relevance, following best practices in human-centred evaluation of dialogue systems (Abeysinghe and Circi, 2024; Lee et al., 2021);

2. **LLM-as-a-judge**: an external model – namely, OpenAI's GPT-3.5 – was prompted in a few-shot setting to automatically score responses across the same criteria, producing an additional layer of intrinsic evaluation (Svikhnushina and Pu, 2023).

This custom approach reflected the broader methodological challenges in evaluating RAG systems, where traditional automatic metrics – e.g., BLEU, ROUGE, METEOR, etc. – have shown limited reliability in dialogue contexts (cf. Sec. 3.1.2).

5.1.3 Insights for System Redesign

Prototyping revealed several challenges that guided the design of the subsequent system:

- **Scalability:** reliance on CSV storage and naive concatenation of retrieved chunks limited efficiency and made the system unsuitable for larger-scale deployments.

- **Resource efficiency:** redundant LLM calls (e.g., for query processing in LangChain) caused unnecessary latency and risked exceeding API rate limits.

- **Metadata control:** the absence of fine-grained chunk-level metadata management constrained retrieval flexibility.

- **Evaluation scope:** intrinsic evaluation without a gold standard or baseline system restricted comparability with other solutions; user and domain-expert feedback was absent at this stage.

- **Dependency on third-party frameworks:** reliance on LangChain introduced constraints in terms of customization and debugging transparency, limiting the control over specific components.

- **Limited multilingual support:** initial prototype focused primarily on Italian, with limited capabilities for handling cross-lingual queries or documents.

- **Simplistic retrieval:** the use of basic dense retrieval without exploring hybrid or advanced reranking techniques may have constrained retrieval effectiveness.
- **Lack of user feedback integration:** the prototype did not incorporate mechanisms for collecting and utilising user feedback to iteratively improve system performance.
- **Narrow evaluation metrics:** assessment relied mainly on subjective human ratings and LLM-based scoring, without the inclusion of broader quantitative measures.
- **Absence of real-world testing:** the prototype was not deployed in a live environment, preventing the observation of actual user interaction patterns and limiting insights into robustness under realistic conditions.

These limitations underscored the need for a more scalable and systematically evaluated architecture, motivating the full system design presented in Sec. 5.2. In parallel, the prototype confirmed the viability of the RAG approach, offering crucial insights into system bottlenecks that directly informed the subsequent design choices.

5.2 Full-Scale Implementation

The full system was re-engineered from the ground to support dynamic, scalable document ingestion, contextual retrieval, and answer generation using open-source language models. All LangChain dependencies were removed in favour of custom Python implementations to improve modularity, debugging transparency, and flexibility in processing. The final architecture is organised around the following processing stages (see Fig. 6):

- a custom KB construction module, which integrates sitemap generation and web crawling,
- semantic chunking and metadata enrichment,
- MistralAI vector embeddings,
- a Faiss vector store for retrieval,
- a generation module with open-source Mistral *NeMo* model,
- generative responses with inline citation handling,
- a reactive front-end Streamlit interface,
- and a feedback management system.

Together, these modules amount to a complete end-to-end pipeline. Web documents are ingested, cleaned, and segmented; semantic metadata and embeddings are created; retrieval is performed over a dedicated vector store; and answers are generated, presented through an interactive interface, and enriched by user feedback mechanisms. The outcome is not a loose set of tools, but a coherent workflow that moves seamlessly from source material to user-facing response.

A fundamental element of this design is the `RAG Orchestrator`, which acts as the central coordinator of the pipeline. It manages the end-to-end flow of information across modules by issuing retrieval requests to the vector database, aggregating and grouping the retrieved chunks into relevant context, and constructing the final prompt. This prompt combines the contextual evidence, the user's query, and the dialogue history, and is then passed to the LLM for answer generation. In doing so, the orchestrator centralises control, ensures consistency in context construction, and abstracts the complexity of the underlying retrieval and generation stages from the user-facing interface.

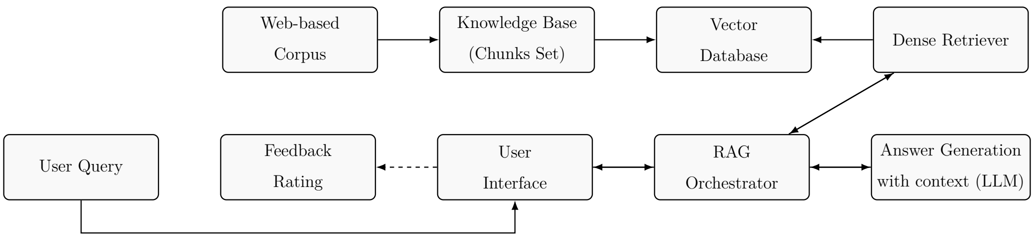


Figure 6: Dataflow and macro modules of the re-engineered GNA QA system. The *RAG Orchestrator* mediates between the UI, the vector database, the retriever, and the LLM: it issues retrieval, builds the context prompt and returns the final answer. Solid arrows show the main data flow; the dashed arrow indicates user feedback.

It is worth noting that Fig. 6 may be read both as a dataflow diagram and as a representation of the system architecture. In this case, the two perspectives converge: the logical architecture of the system is itself defined by the pipeline of data transformations, with each module corresponding directly to a deployed component. The orchestrator sits at the core of this dual view, embodying both the control logic of the architecture and the connective tissue of the dataflow.

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 – accessible at https://gna.cultura.gov.it/wiki – which hosts the official user manual of the GNA (Mic, 2019).

The crawler is configured to:

- begin at the root node (`Pagina_principale`);
- follow only internal links matching `/wiki/index.php/`, excluding irrelevant namespaces such as `Special:`, `User:`, or `Talk:`;
- normalise URLs by removing query parameters to prevent duplicates;
- apply a polite crawling policy, including a one-second delay between requests and a custom user-agent header;
- limit traversal with a maximum crawl depth[27] of 10 and page limit[28] set to 200 pages.

The output is a structured sitemap serialised as an XML file (`GNA__sitemap.xml`), annotated with metadata such as last-modified timestamps, priority, and update frequency. This sitemap then provides the foundation for all subsequent document harvesting and processing steps.

5.3.2 Document Crawling

The next stage systematically collects the contents of all URLs listed in the sitemap to ensure comprehensive coverage of the manual. Pages are fetched asynchronously with retry logic, exponential back off, and controlled concurrency to withstand transient network issues or throttling. Requests are interleaved with pauses to avoid overwhelming the server and failed URLs are logged for reprocessing.

Once retrieved, the raw HTML is parsed using BeautifulSoup (Hajba, 2018), focusing on the main content (`deiv#mw-content-text`) and stripped of extraneous elements such as navigation

[27] Crawl depth is intended as the maximum number of link "level" away from the starting url that the crawler will follow.

[28] Page limit parameter refers to the maximum number of pages the crawler will visit and include in the sitemap, regardless of depth.

bars, footers, and other layout components that do not contribute semantic content. The remaining material is processed preserving the logical reading order. Structural features are carefully retained: section headers (`h1-h6`) are used to reconstruct a hierarchical outline of the page; paragraphs, tables, lists, and images are preserved as discrete items, each linked back to its contextual breadcrumb trail. This ensures that the captured content is situated within its original navigational hierarchy, which later supports more precise retrieval.

The outcome of this step is a collection of structured representations of the user manual's pages, where meaningful content is disentangled from noise and the logical organisation of the source is preserved.

5.3.3 Chunking

After parsing, documents are segmented into smaller and more manageable units. To this end, the system employs a sliding-window strategy with a maximum span of 512 characters and an overlap of 128, a configuration recommended to favour dense retrieval while mitigating excessive fragmentation of context (Wang, Wang et al., 2024).

Chunks are not treated as inert slices of text, but as semantically enriched units that carry a set of contextual signals. Each one is assigned a unique identifier – a SHA-256 hash of its source and position – and is accompanied by the source URL, page title, and the chain of section headers that serve as navigational breadcrumbs. Content is further annotated with keywords extracted through the KeyBERT method[29] and named entities identified with the model *it_core_news_md* from spaCy.[30] The metadata also records the content type – whether a passage of text, a table, a list, or an image – ensuring that the heterogeneous nature of the original source remains visible at retrieval time.

Tables. Tables are preserved as autonomous chunks and formatted in Markdown, maintaining both their structure and interpretability.

Lists. Lists are likewise extracted as discrete units, safeguarding their enumerative character.

Images. Images are handled with particular care, since they frequently consist of illustrative diagrams, tables embedded as graphics (e.g., Fig. 7), or screenshots of the GNA QGIS plugin (e.g., Fig. 8) that can be of value for the retrieval and answer generation stages. To unlock this layer of information, the system applies an OCR pipeline built on Tesseract (*Tesseract OCR,*

[29] *KeyBERT* is a keyword extraction technique which uses BERT embeddings to generate the keywords and keyphrases most closely aligned with a document (Grootendorst, 2025).
[30] Reference: *Release explosion/spacy-models it_core_news_sm-3.8.0* (2024)

2025), enhanced with a series of preprocessing steps. Each image is first converted to greyscale and rescaled through bicubic interpolation to sharpen textual contours, then contrast-limited adaptive histogram equalisation (CLAHE) is applied to reduce noise and improve legibility across varied backgrounds. The processed image is passed to Tesseract with a configuration optimised for Italian language recognition and extended character support, allowing the extraction of structured textual surrogates. These OCR outputs are integrated into the retrieval space as additional content, making usable otherwise inaccessible visual information.

Code. A further challenge is posed by the presence of code snippets in the operative manual, expressed in the form of HTML tags. These typically provide instructions for formatting reports or embedding elements within QGIS layouts. For instance, in the section on compiling the MOPR, users are instructed to insert HTML tags (e.g., Fig. 9) to generate two-column layouts when descriptive text exceeds 5000 characters, or to employ `<br>` to insert line breaks. In the chunking phase, these snippets are preserved as discrete "code" chunks, tagged accordingly in the metadata. This ensures that when a user queries the assistant for formatting or technical guidance, the relevant HTML examples are retrievable in their proper context.

TABELLA 1 – POTENZIALE ARCHEOLOGICO

VALORE	POTENZIALE ALTO	POTENZIALE MEDIO	POTENZIALE BASSO	POTENZIALE NULLO	POTENZIALE NON VALUTABILE
Contesto archeologico	Aree in cui la frequentazione in età antica è da ritenersi ragionevolmente certa, sulla base sia di indagini stratigrafiche, sia di indagini indirette	Aree in cui la frequentazione in età antica è da ritenersi probabile, anche sulla base dello stato di conoscenze nelle aree limitrofe o in presenza di dubbi sulla esatta collocazione dei resti	Aree connotate da scarsi elementi concreti di frequentazione antica	Aree per le quali non è documentata alcuna frequentazione antropica	Scarsa o nulla conoscenza del contesto
Contesto geomorfologico e ambientale in epoca antica	**E/O** Aree connotate in antico da caratteri geomorfologici e ambientali favorevoli all'insediamento umano	**E/O** Aree connotate in antico da caratteri geomorfologici e ambientali favorevoli all'insediamento umano	**E/O** Aree connotate in antico da caratteri geomorfologici e ambientali favorevoli all'insediamento umano	**E/O** Aree nella quale è certa la presenza esclusiva di livelli geologici (substrato geologico naturale, strati alluvionali) privi di tracce/materiali archeologici	**E/O** Scarsa o nulla conoscenza del contesto
Visibilità dell'area	**E/O** Aree con buona visibilità al suolo, connotate dalla presenza di materiali conservati *in situ*	**E/O** Aree con buona visibilità al suolo, connotate dalla presenza di materiali conservati prevalentemente *in situ*	**E/O** Aree con buona visibilità al suolo, connotate dall'assenza di tracce archeologiche o dalla presenza di scarsi elementi materiali, prevalentemente non *in situ*	**E/O** Aree con buona visibilità al suolo, connotate dalla totale assenza di materiali di origine antropica	**E/O** Aree non accessibili o aree connotate da nulla o scarsa visibilità al suolo
Contesto geomorfologico e ambientale in età post-antica	**E** Certezza/alta probabilità che le eventuali trasformazioni naturali o antropiche dell'età *post* antica non abbiano asportato in maniera significativa la stratificazione archeologica	**E** Probabilità che le eventuali trasformazioni naturali o antropiche dell'età *post* antica non abbiano asportato in maniera significativa la stratificazione archeologica	**E** Possibilità che le eventuali trasformazioni naturali o antropiche dell'età *post* antica non abbiano asportato in maniera significativa la stratificazione archeologica	**E** Certezza che le trasformazioni naturali o antropiche dell'età *post* antica abbiano asportato totalmente l'eventuale stratificazione archeologica preesistente	**E** Scarse informazioni in merito alle trasformazioni dell'area in età *post* antica

```
<div class="floatnone">
  <a href="/wiki/index.php/File:Tabella_del_potenziale.jpg"
   class="image">
    <img alt="Tabella del potenziale.jpg" src="/wiki/images/
    5/50/Tabella_del_potenziale.jpg" decoding="async" width=
    "1151" height="558">
  </a>
</div>
```

Figure 7: Example of a table embedded as an image, accompanied by the corresponding HTML snippet used for its integration in the GNA user manual website.

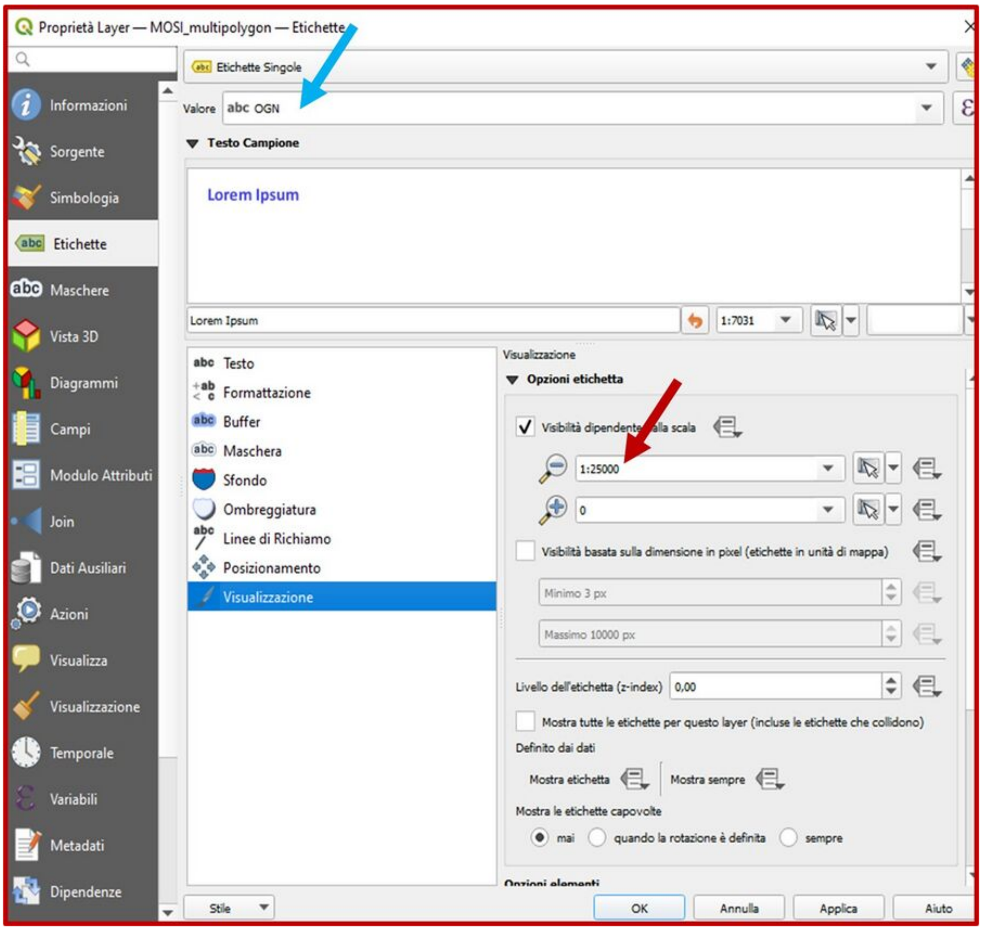
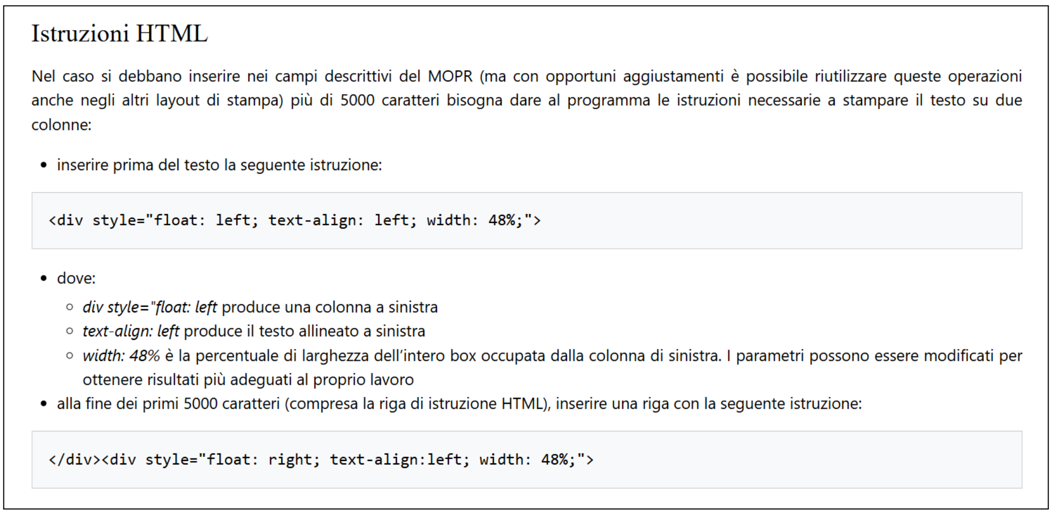


Figure 8: Screenshot of the GNA QGIS plugin interface, as presented in the user manual, illustrating configuration options to support data entry and visualisation.



Istruzioni HTML

Nel caso si debbano inserire nei campi descrittivi del MOPR (ma con opportuni aggiustamenti è possibile riutilizzare queste operazioni anche negli altri layout di stampa) più di 5000 caratteri bisogna dare al programma le istruzioni necessarie a stampare il testo su due colonne:

- inserire prima del testo la seguente istruzione:

```
<div style="float: left; text-align: left; width: 48%;">
```

- dove:
  - *div style="float: left* produce una colonna a sinistra
  - *text-align: left* produce il testo allineato a sinistra
  - *width: 48%* è la percentuale di larghezza dell'intero box occupata dalla colonna di sinistra. I parametri possono essere modificati per ottenere risultati più adeguati al proprio lavoro
- alla fine dei primi 5000 caratteri (compresa la riga di istruzione HTML), inserire una riga con la seguente istruzione:

```
</div><div style="float: right; text-align:left; width: 48%;">
```

Figure 9: Example of HTML snippet instructions from the GNA user manual, illustrating how to format long descriptive fields.

The overarching principle guiding this design is that chunking must adapt to the structure of the document rather than follow a rigid, uniform rule (see Fig. 10). Accordingly, the approach combines sentence-level segmentation for textual passages – further constrained by configurable character-based windowing and overlap – with boundary-based segmentation for structural elements such as title headers, tables, lists, and images, which are treated as independent chunks, acting as natural delimiters. This custom strategy avoids the weaknesses of relying exclusively on one paradigm, as purely sentence-based methods risk breaking semantic continuity, while purely boundary-based approaches tend to produce uneven or oversized segments (Microsoft Learn, 2025).

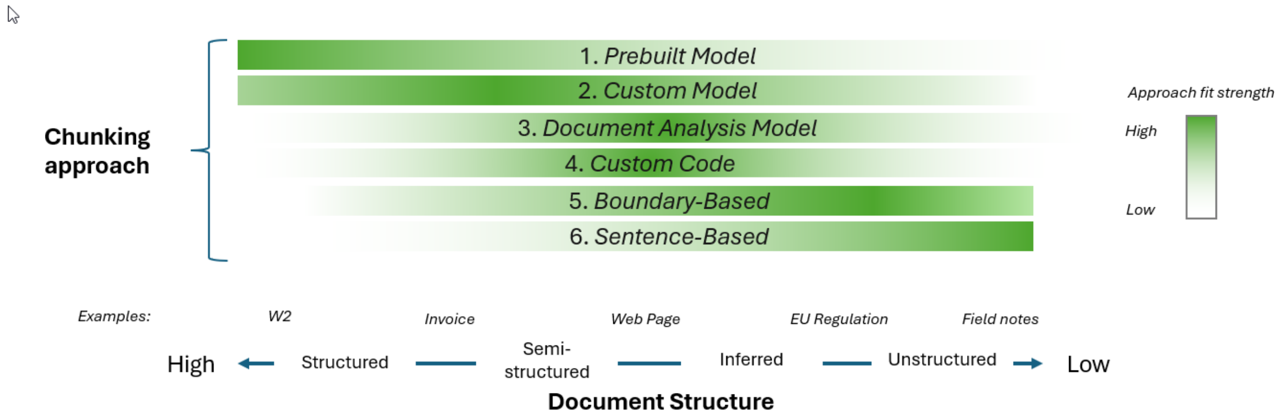


Figure 10: Different chunking strategies according to document structure.
Source: Microsot Learn.

Such tailoring proves especially important in the archaeological domain, where textual exposition often coexists with structured metadata, tabular references, and visual documentation. The resulting knowledge base is one that strikes a careful balance: granular enough to be tractable for retrieval, yet faithful enough to preserve the integrity of the original material. In practice, this balance enhances retrieval precision and ensures that the content of the GNA user manual – complex, layered, and semi-structured – remains interpretable in downstream processing.

Finally, all enriched chunks are serialised into a JSON file,[31] which constitutes the data source for embedding, retrieval, and generation tasks. This modular format ensures transparency and traceability of the preprocessing pipeline and enables flexibility to experiment with alternative retrievers, query rewriting techniques, and reranking strategies as outlined in Sec. 5.4.1.

[31] Output: 835 structured chunks saved in `data/chunks_memory.json`.

5.3.4 Vector Embeddings

Document chunks are converted into dense vector representations using the *intfloat/multilingual-e5-large* model from Sentence Transformers (Wang, Yang, Huang, Yang et al., 2024). This model was selected for its multilingual encoding capabilities and strong performance in semantic retrieval tasks, making it suitable for the predominantly Italian KB 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 neighbour search using the inner product – i.e., dot product – as metric, computed as:

$$\cos(\theta) = \frac{q \cdot x}{\|q\|_2 \, \|x\|_2}.$$

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 while preserving quick access to metadata such as source URL, document structure, and content type for downstream processing.[32]

Faiss was retained for the full-scale implementation because its lightweight yet high-performance design matched the project's computational constraints and the requirement for retrieval methods that are both controllable and replicable. Unlike Milvus or Qdrant, which offer integrated metadata management and distributed scalability but at the cost of additional infrastructure, this choice enabled the system to remain simple and efficient within the deployment setting, which consists of a single-node architecture. In particular, the application runs on Streamlit Community Cloud, a self-contained environment where introducing external database services would have added unnecessary complexity, with the risk of reduced performance and higher latency. With Faiss, the index type, similarity metric, and vector normalization could be explicitly defined, keeping the retrieval process understandable and explainable. Furthermore, storing the vector index in a `.faiss` file and maintaining metadata separately in a `.pkl` file guaranteed stable and consistent results across sessions, different runs and redeployments. The availability of Python wrappers further contributed to making the integration both straightforward and efficient.

These embeddings and their associated metadata form the foundation for the retrieval stage,

[32]Output: Faiss index stored in `.faiss_db/`, linked with its metadata.

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.

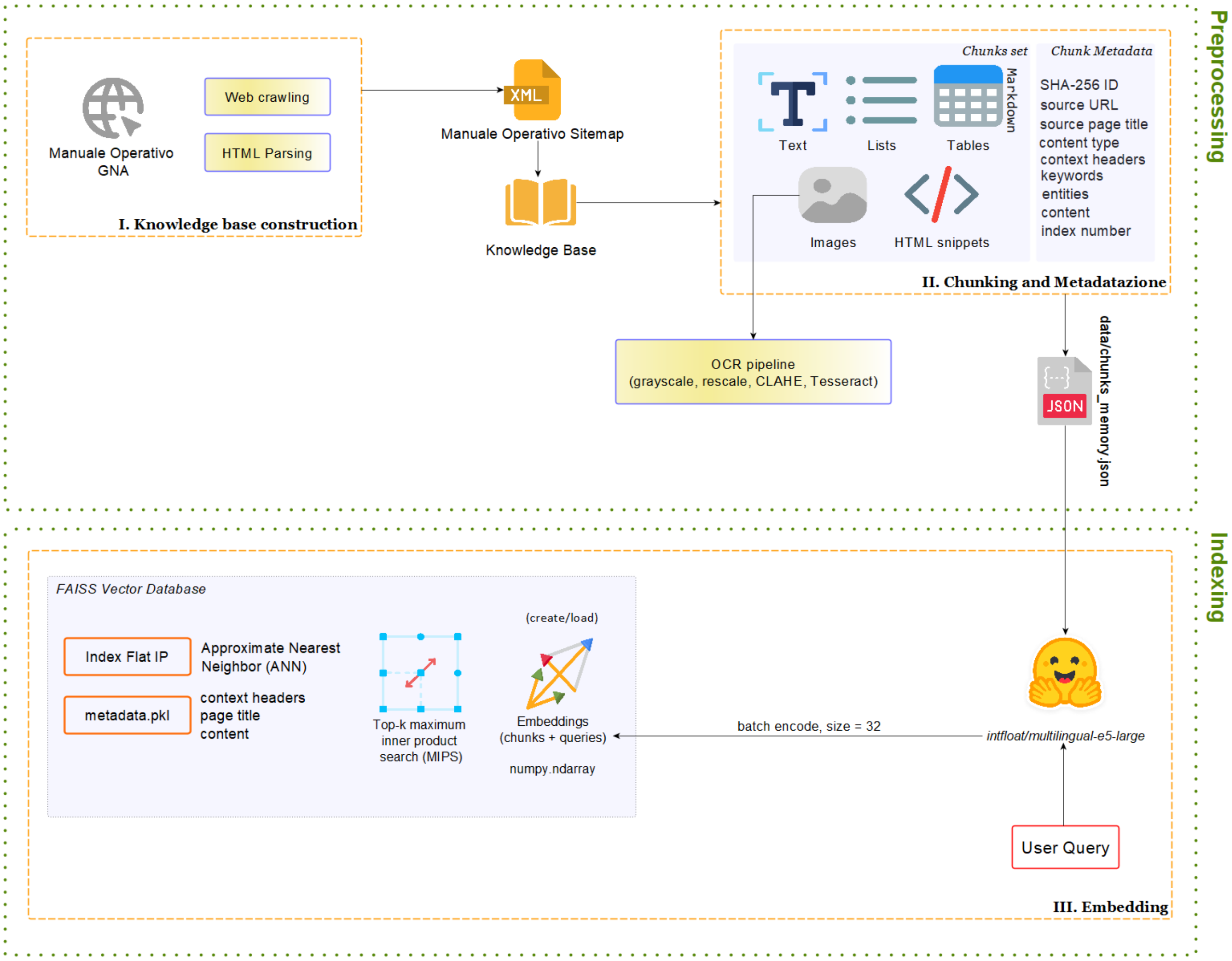


Figure 11: First stages of the GNA QA pipeline: from knowledge base construction and chunking with metadata to embedding into the Faiss index for subsequent retrieval.

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 top-k candidate chunks.[33] 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

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

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, where alternative retrieval strategies and scoring variations are tested against this reference dense implementation.

5.4.1 Experimental Setup for Ablation Studies

To systematically evaluate the contribution of different retrieval strategies, we conducted a series of ablation experiments (see Fig. 12). In this context, *ablation* refers to the systematic removal, isolation, or modification of system components in order to evaluate their specific effect on overall performance. Importantly, none of the tested approaches was permanently integrated into the main pipeline; instead, each configuration was evaluated independently to allow for a broader performance comparison. The results were then compared to the baseline retrieval outcomes.

The following retrieval configurations were implemented and evaluated:

- **Dense retrieval.**

  This method employs dense vector embeddings for document representation and similarity search. It relies on Faiss as the underlying vector database, wrapped via the `VectorDatabaseWrapper` module. Queries are encoded into embeddings and compared with pre-computed document embeddings, returning the top-k results ranked by inner-product similarity. Query embeddings are cached using a normalised MD5 hash to avoid redundant computations, and batch querying is supported.

- **BM25 retrieval.**

  A traditional sparse retriever based on the BM25 algorithm. The index was constructed over concatenated metadata fields – `title`, `keywords`, `headers_context`, and `document` – from the same chunks metadata store. Preprocessing was applied specifically for Italian, including stop word removal, stemming, and handling of clitics and apocope forms. The tokenised corpus was then indexed for lexical matching. As with dense retrieval, a batch mode was implemented. The default cutoff parameter (k = 5) was selected empirically to balance response quality against tokenisation and latency constraints.

- **Hybrid retrieval.**
  To combine semantic similarity with lexical matching, hybrid retrieval strategies were explored. Two fusion techniques were implemented:

  – **Weighted Reciprocal Rank Fusion (RRF):** ranks from dense and sparse retrievers are aggregated using RRF. The fusion score for document d is computed as:

  $$\frac{w_{\text{dense}}}{k + \text{rank}_{\text{dense}}} + \frac{w_{\text{sparse}}}{k + \text{rank}_{\text{sparse}}};$$

  where the default weights parameters were $w_{\text{dense}} = w_{\text{sparse}} = 1.0$, candidates set size $= 50$, and $\texttt{k} = 60$.

  – **Score-blend fusion:** here, normalised scores from the two retrievers are merged 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 $S_{\text{norm},d}$ and $S_{\text{norm},s}$ are the min-max normalised scores of the dense and sparse retrievers, respectively, and α controls the relative weight of the sparse component ($w_d = w_s = 1$, $k = 60$). Unlike RRF, this approach requires that scores from different retrievers are defined on comparable scales (Wang, Wang et al., 2024).

What to pick?

The choice between 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 prioritises consensus across retrieval methods. Conversely, Score-blend is more appropriate when per-query scores are reliable, as it allows finer control over the relative influence of dense and sparse components. Edge-case behaviours include:

(a) *Document appears in only one list*: RRF ranks it lower, while Score-blend assigns it a normalised score from the contributing retriever;

(b) *All scores equal in a list*: Score-blend reduces the contribution of that retriever, whereas RRF still differentiates documents by rank.

Query Rewrite

To sharpen retrieval quality, we experimented with *query rewriting*, i.e. reformulating user queries to increase the likelihood of retrieving relevant documents. Within these experiments, query rewriting was conceived as a multi-strategy process that generates alternative query variants through complementary transformations, each targeting different aspects of query understanding and manipulation (Li et al., 2024). Specifically, the approach integrated:

- **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):** extraction of key terms with KeyBERT, followed by enrichment through n-gram combinations and synonym substitutions to introduce semantically related expressions.
- **General Query Rewriting (GQR):** linguistic normalisation based on spaCy lemmatisation and stop word removal, yielding canonical query forms.
- **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 the original query to form an expanded version.
- **Query Decomposition:** conjunctive or disjunctive queries are split into simpler subqueries, each addressing a distinct semantic aspect.

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

Reranking

Finally, a *reranking* stage was introduced to refine retrieval outcomes.[34] In our implementation, it operates as a wrapper around a base retriever (Dense, BM25, or Hybrid) and uses

[34] Reranking is a post-retrieval process that reorders an initial set of candidate documents based on a more precise estimation of their relevance.

a transformer-based cross-encoder model (*cross-encoder/ms-marco-MiniLM-L-6-v2*) to jointly encode the query and each candidate document, assigning 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, thereby capturing finer semantic relationships.

At runtime, the reranker receives the top-N candidates (with $N = 50$) from the base retriever, tokenises each query-document pair, and performs inference in batches with mixed-precision support when available. The final output consists of a top-k ranked list reordered by cross-encoder scores. This stage aims at providing a more precise estimation of relevance, attempting to compensate for weaknesses in both dense and sparse retrieval alone.

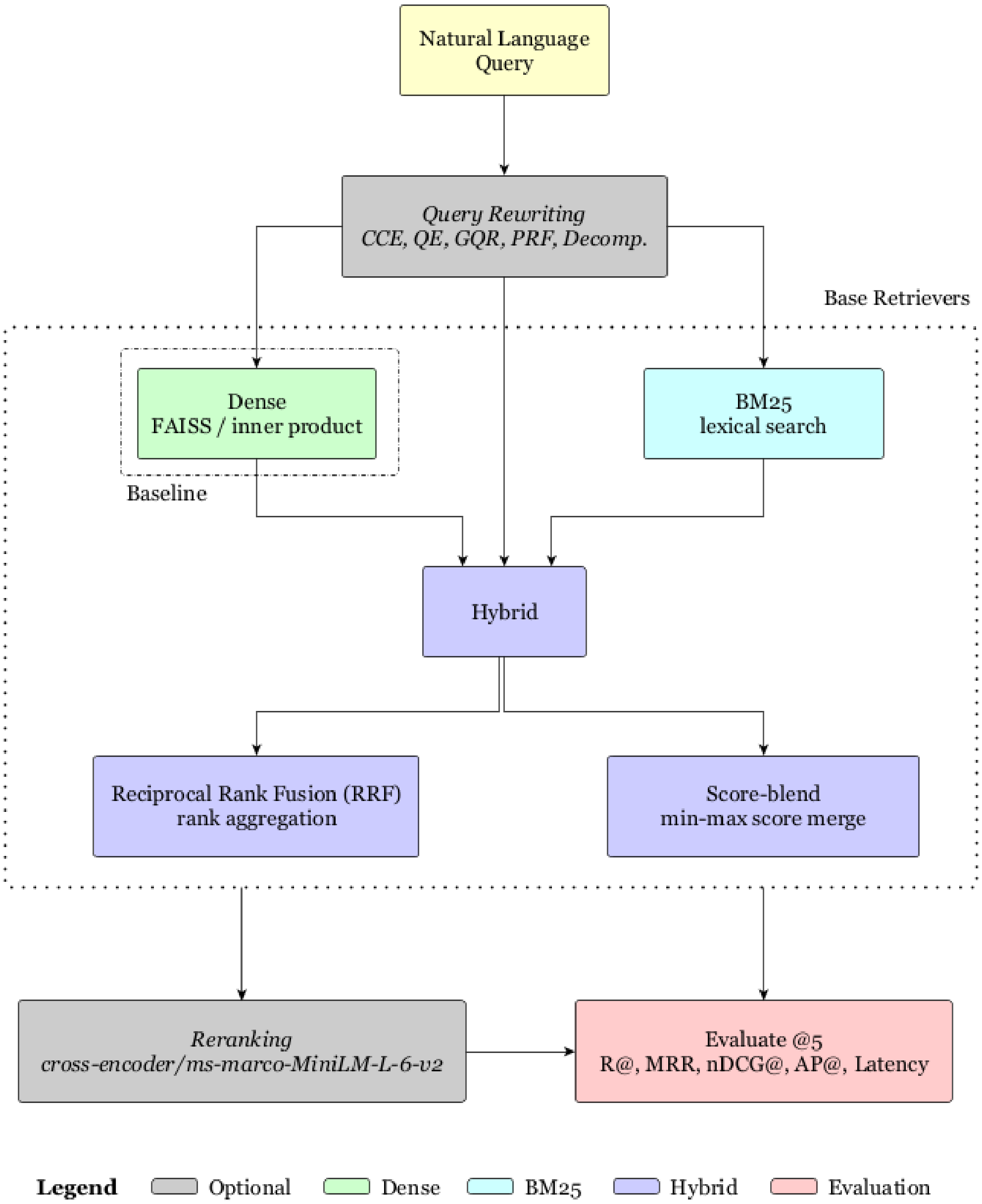


Figure 12: Ablation setup for retrieval strategies. Each configuration (Dense, BM25, Hybrid via RRF or Score-blend), with or without Query Rewriting and/or Reranking, is executed *independently*. Outputs are evaluated at the @5 cutoff (R@5, MRR, nDCG@5, AP@5, Latency) and compared to Dense retriever baseline.

5.5 Generation

Central to the generation phase is the `RAG Orchestrator`, the component responsible for coordinating the entire workflow between retrieval and answer synthesis. When a user query is received, the orchestrator first queries the vector database to obtain the most relevant chunks. These chunks are then grouped by provenance – i.e., their original URLs –, ensuring that related fragments are presented together and assigned consistent citation identifiers. Next, the orchestrator constructs the retrieval-augmented prompt by combining the grouped context with the user's question and any previous dialogue history. This structured prompt, together with the decoding parameters, is forwarded to the Mistral model through an asynchronous streaming interface. As the model generates its response, the orchestrator performs post-processing by validating and formatting in-answer citations and appending the corresponding reference list.

The Mistral model employed is *NeMo* 12B,[35] an open-source LLM accessible via a dedicated API and hosted independently. Its selection was guided by a convergence of methodological and practical requirements. First, open-source availability and a permissive licence ensured transparency, reproducibility, and the possibility of adaptation without the restrictions of proprietary services. Second, the model's strong performance on multilingual benchmarks, including robust handling of Italian,[36] made it particularly suitable for a system intended to operate in a cultural heritage setting where linguistic specificity is paramount. Third, Mistral *NeMo* demonstrated competitive efficiency, with low latency and high throughput even when deployed on modest hardware, a quality that enabled real-time responsiveness without requiring more powerful computing infrastructures. In addition, the availability of an official API greatly facilitated seamless integration into the retrieval-augmented generation pipeline, while its ability to support extended context windows – up to 128k tokens – allowed the system to process multiple retrieved passages in a single prompt without truncation or loss of coherence.

While different open-source alternatives were considered too – such as LLaMA, Falcon, Google Gemma and OpenAI models –, none aligned as closely with the system's combined requirements. LLaMA, for instance, offers state-of-the-art performance and strong community support, but their licensing terms restrict certain deployment scenarios and context windows are more limited in practice. Falcon models, though efficient, have shown more variability in multilingual performance, particularly outside English. Proprietary APIs, while powerful, introduce cost barriers and potential vendor lock-in, compromising the long-term reproducibility of the research workflow, as in the case of OpenAI models. By contrast, Mistral *NeMo* offered a balanced compromise: multilingual coverage, scalable deployment options, and infrastructure for fine-tuning embeddings, made it a natural fit for both the prototyping stage and the full-

[35]NeMo 12B, released in July 2024 by Mistral, surpasses comparable open-source models with 83.5% accuracy on HellaSwag, 73.8% on TriviaQA, and 68% on MMLU, while offering a 128k context window – vs. 8k for LLaMA 3 and Gemma 2. On instruction-following, it achieves 7.84 on MT Bench and 42.57 on WildBench, demonstrating stronger reasoning and alignment in multi-turn QA. Source: *Mistral NeMo | Mistral AI,* 2025.

[36]On multilingual tasks, Mistral model consistently outperforms LLaMA 3 (8B). For Italian specifically, accuracy reaches about 72% on HellaSwag, 49% on ARC Challenge, and 62% on MMLU. The model is able to tokenizes Italian efficiently (compression ratio 1.28) and also sustains strong accuracy across reasoning, commonsense, and knowledge-intensive benchmarks. Ibid.

scale system.[37]

As mentioned earlier, the generation module (see Fig. 14) itself is designed to deliver fluent and context-aware answers with citations, ensuring that responses remain both interpretable and verifiable. Prompts are constructed through a structured template combining system-level instructions (see Sec. 5.5.1), the user query, top-k retrieved grouped chunks, and the conversational history maintained in memory to sustain continuity across follow-ups. API calls are issued with a combination of decoding parameters: a temperature of 0.3[38] to prioritise factual accuracy, a top-*p* of 0.9[39] to maintain lexical diversity without drifting off-topic, and a maximum of 512 tokens to keep answers concise. Finally, responses undergo lightweight post-processing to enforce inline citation formatting, guarantee Italian output, and preserve readability through numbered references and 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 constructed as follows, based on specific instructions, boundaries and constraints.

A custom system message (Listing 5.2) is injected at the top of the prompt to guide the model's behaviour. This message instructs the system to enforce neutrality in its answers, prioritise relevant and verifiable information, and include inline citations 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.
```

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

[38] The parameter temperature set to 0.3 sharpens the model's probability distribution, favouring high-likelihood tokens and suppressing unlikely alternatives, thereby producing more deterministic and factual outputs.

[39] The parameter top-*p*, also called *nucleus sampling*, restricts generation to the smallest set of candidate tokens whose combined probability mass reaches a chosen threshold (here 90%), ensuring that the model considers diverse but still plausible continuations while discarding unlikely options.

```
    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
"""
```

Listing 5.2: System prompt specifying assistant constraints and response instructions.

Each chunk passed to the LLM is numbered and grouped by 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.

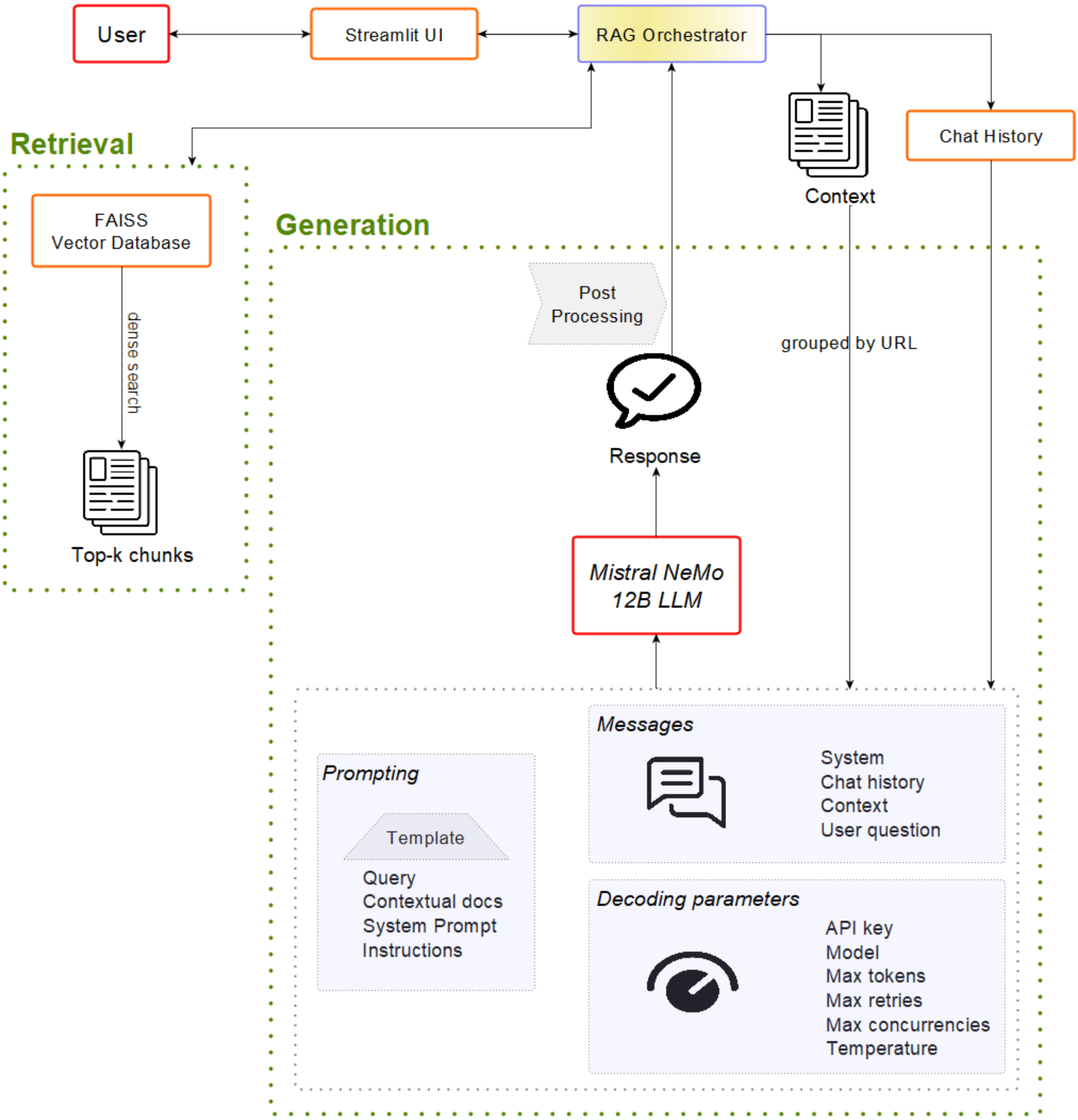


Figure 13: Generation and retrieval stages of the GNA QA pipeline: components interaction.

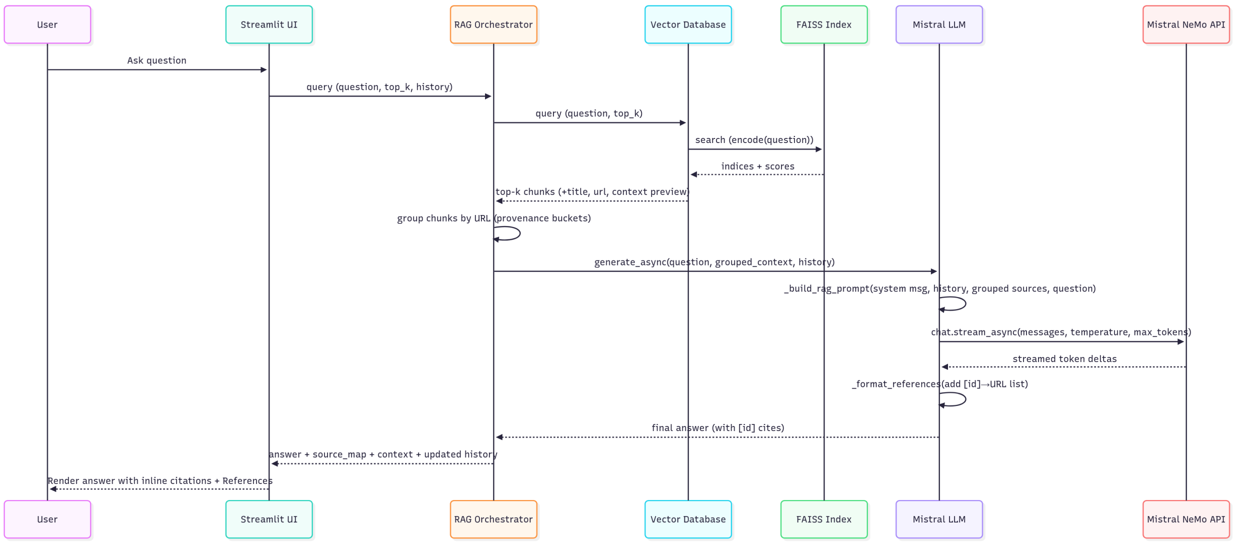


Figure 14: Sequence diagram of the generation stage from the GNA QA pipeline: a user query is sent to the RAG orchestrator, the Faiss vector database and combined with context and chat history by the orchestrator before being sent to the Mistral NeMo 12B model, which generates a response that is post-processed and returned to the user.

5.6 Evaluation Protocol

Evaluation was conducted across two complementary dimensions:

- **Quantitative**, focusing on retrieval performance through metrics such as Recall (R@), Mean Reciprocal Rank (MRR@), Normalised Discounted Cumulative Gain (nDCG@), Average Precision (AP@), and Latency to assess retrieval performance;
- **Qualitative**, assessing the perceived quality and usability of responses through human feedback.

This dual perspective reflects the understanding that effective RAG-based QASs require not only accurate retrieval and generation but also operational efficiency and adaptability in real-world contexts (Akkiraju et al., 2024). To support continuous refinement, the evaluation was designed as an iterative process, with quantitative results informing system optimisation and qualitative insights guiding user-centred adjustments.

All retrieval configurations (Dense, BM25, Hybrid, Query Rewriting, and Reranking variants) were evaluated under the same protocol, ensuring comparability of results across ablation studies. While this framework provided a coherent structure for assessment, its application also revealed several limitations inherent to the experimental setting and the available resources. These included:

- **Absence of a gold standard:** there was no authoritative or verified set of annotated responses to serve as a benchmark of correctness.
- **No baseline system:** internal institutional tasks had no legacy solutions or established benchmarks for direct comparison in the archaeological domain.
- **Limited availability of domain experts:** during the early phases, development was conducted without input from real users or expert annotators; human feedback was integrated only at a later stage.
- **Limited applicability of automated generation metrics:** common algorithmic measures, although widely applied in text generation, have been shown to be ineffective for dialogue and QA evaluation (Deriu et al., 2020; C.-W. Liu et al., 2016).

These constraints mirror challenges identified in recent RAG evaluation literature, which emphasises the lack of standardised protocols and the importance of balancing intrinsic metrics with human-centred evaluation (Abeysinghe and Circi, 2024), as noted in Sec. 3.1.2. Akkiraju et al. (2024) further highlight the need to assess both accuracy and efficiency, pointing to critical control points in the RAG chain where trade-offs between retrieval quality and latency emerge.

5.6.1 Test Datasets

Widely used QA benchmarks such as SQuAD and Natural Questions or multi-hop datasets like HotpotQA (Z. Yang et al., 2018) and 2WikiMultiHopQA (Ho et al., 2020) are indispensable for advancing general-domain research but are misaligned with the objectives of this study. These datasets are predominantly built around encyclopaedic or factoid queries in English, often centred on entity recognition and short-span extraction. By contrast, the GNA QA system targets procedural, explanatory, and domain-specific questions expressed in Italian, where answers are not simple factual snippets but often involve step-by-step guidance, cross-references to interface components, or the interpretation of heterogeneous content.

Equally important, while KBs such as Wikipedia do consists of text that coexists with tables and lists, their role is largely illustrative or descriptive. In the GNA manual, instead, such structures are integral to the semantics of the documentation. Tables encode data entry examples or parameter constraints, lists enumerate workflows, and images convey key instructional content. HTML snippets add yet another layer, neither plain text nor program code in the strict sense, but procedural directives embedded within reporting practices. These formats are not ancillary but essential to the information needs of end-users, requiring reasoning that goes beyond the scope of existing benchmarks.

Evaluating on standard datasets would therefore risk producing misleadingly inflated scores that fail to capture the genuine challenges of domain-specific, multimodal documentation. Given these considerations, a custom evaluation set was constructed and sampled directly from the GNA KB. This ensured ecological validity, testing the QA system on the exact materials and information needs it is designed to serve.

To support systematic analysis, two synthetic evaluation sets were created:

- **Single-hop dataset:** containing 508 queries designed to elicit single-document answers, each with a single gold document (see Listing 5.3). This dataset tested the system's ability to retrieve and generate answers based on isolated chunks of information.
- **Combined dataset:** containing 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 (see Listing 5.4). This dataset assessed the system's capacity to integrate information from various sources and generate coherent contextual responses.

```
{
  "question": "<Italian question>",
  "relevant_docs": ["<chunk_id>"],
  "document_content": "<chunk text>"
}
```

Listing 5.3: JSON output format for single-hop dataset items.

```
{
  "question": "<Italian question>",
  "relevant_docs": ["<chunk_id_1>", "<chunk_id_2>",
       "..."],
  "document_content": ["<text_1>", "<text_2>",
      "..."],
  "is_multihop": false|true,
  "num_docs": 1|2|3|4
}
```

Listing 5.4: JSON output format for combined dataset items, including single-hop and multi-hop questions.

Together, these custom test sets provided balanced coverage of both simple and complex retrieval scenarios. In both cases, questions were generated directly from the chunk corpus using Mistral *NeMo* (cf. Sec. 5.5). For the single-hop dataset, each question was elicited from an individual chunk through a prompt that constrained the model to formulate queries strictly grounded in the given text, ensuring that answers could only be retrieved from the associated gold document. In contrast, the multi-hop dataset required a more demanding construction: subsets of two to four chunks were sampled at random, concatenated, and provided as input to the LLM with explicit instructions to generate questions resolvable only by combining information across all documents in the set. To avoid superficial overlaps, in this case the prompt enforced specificity and disallowed formulations answerable from a single source.

5.6.2 Metrics

To evaluate retrieval in a consistent and transparent manner, the system was assessed using a set of standard IR metrics, complemented by latency measurements to capture operational performance (Wang, Wang et al., 2024). Rerank, MRR, nDCG, and AP were computed with a cutoff at five candidates, a choice that strikes a balance between evaluation depth and efficiency by focusing on the documents most likely to provide relevant context for generation.

- R@5 (Recall at 5): proportion of relevant documents successfully retrieved within the top 5 results, reflecting coverage of the retrieval step;
- MRR (Mean Reciprocal Rank): average of reciprocal ranks of the first relevant document retrieved, rewarding systems that place the correct answer as early as possible;
- nDCG@5 (Normalised Discounted Cumulative Gain at 5): evaluates ranking quality by weighting relevant documents higher when they appear near the top of the result list;
- AP@5 (Average Precision at 5): computes the average of precision values at each point a relevant document is retrieved within the top 5, providing a balance of recall and precision across ranks;
- Latency: mean retrieval time per query (in seconds).

Each evaluation run stores a machine-readable report (JSON) capturing dataset name (single-hop / multi-hop), creation timestamp (UTC), orchestrator and model identifiers (Mistral API model name), retrieval parameters (top-k and candidate-k), batch size, and device configuration. These artefacts ensure precise traceability of conditions across ablation experiments (cf. Sec. 6.2).

5.6.3 Qualitative Assessment

To complement intrinsic metrics, qualitative evaluation focused on dimensions more directly linked to user experience (Bronsdon, 2025):

- **Relevance:** whether the generated answer addressed the query meaningfully.
- **Fluency:** linguistic naturalness and readability of responses.
- **Completeness:** coverage of the key information needed to satisfy the query.
- **Usability:** perceived usefulness of the system as an interactive tool.

Human feedback was through a lightweight 3-point Likert scale – Poor, Fair, Good –, later exported as structured datasets for analysis. Although limited in scale, this qualitative perspective shed light on aspects of response quality – such as user trust and usability – that purely algorithmic metrics could not capture, particularly in relation to user trust and system transparency.

5.7 User Interface

The user interface (UI) was implemented using Streamlit (*Streamlit Documentation v1.47.0*, 2025), chosen for its capacity to support rapid prototyping and its native integration with Python NLP pipelines. The framework also provides built-in support for asynchronous processes, which is particularly advantageous in a RAG setting where retrieval and generation stages may exhibit variable latency. Functioning as the primary interaction layer, the UI enables users to submit queries in natural language and obtain generated answers in real time through a clear and accessible design (see Fig. 15). Supporting evidence is presented through in-answer citations linked to the corresponding source documents, enabling users to verify the information, assess the reliability of the response, and explore the broader context from which the answer was derived.

Beyond serving as a functional entry point to the system, the UI assumes a crucial methodological role. It establishes a channel for gathering explicit user feedback, thereby complementing quantitative retrieval metrics with qualitative evaluations. Through the interface, users can annotate system responses according to perceived relevance, fluency, completeness, and usability by selecting an option on a three-point Likert scale. These ratings generate valuable data that inform iterative refinement of the system. As previously noted, this human-centred dimension is particularly important in dialogue and QA contexts, where conventional automated metrics often fail to capture the subtleties of interaction quality. The interface therefore does more than displaying outputs; it serves as an evaluation instrument in its own right, supporting the triangulation of system performance across computational indicators and human judgements.

Figure 15: Streamlit user interface of the GNA AI Assistant.

The application UI is organised into three main areas:

1. **Sidebar:** contains MiC reference, 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 ("*Cancella cronologia chat*") to reset the Streamlit session (`st.session_state.chat_history`) and a "*Download feedback*" button for exporting user queries and system's responses.

2. **Main section:** provides a natural language input field for querying the assistant. It displays the chat history, including user messages, assistant responses and feedback buttons for each system answer.

3. **Session features**:

- Chat history is limited to the most recent ten exchanges, which are stored and updated in the 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.

To guarantee smooth interactions and to minimise computational overhead, the system makes extensive use of `st.session_state`, a built-in Streamlit object designed for persisting variables across user interactions. Unlike ordinary Python variables, which are re-initialised each time Streamlit re-runs a script, `st.session_state` retains values for the duration of a session. This functionality is critical in a conversational application, as it allows chat memory to persist between turns. Both user queries and model responses are stored, so that continuity is preserved across multi-turn dialogues.[40]

Beyond the bounds of memory management, `st.session_state` is also used to cache API outcomings, including responses generated by Mistral and their associated citation mappings. This prevents redundant API calls when users revisit or re-evaluate queries, at the advantage of efficiency and costs mitigation. In addition, the same mechanism is employed to track user feedback, linking annotations directly to the relevant conversational context.

To ensure a responsive user experience, the system integrates Python's `asyncio` and `concurrent.futures` modules. Each call to the language model is executed within a new event loop, ensuring compatibility with Streamlit's execution environment, while a `ThreadPoolExecutor` provides thread-safe parallel execution. This design enables non-blocking retrieval and generation: the interface remains responsive even under conditions of high latency or slow API responses, thus preserving a fluid user experience.

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.

[40]For more details on Streamlit session state management, see the official API documentation at https://docs.streamlit.io/.

5.8.1 Collection

As noted earlier, each assistant response is immediately followed by three clickable UI buttons based on a three-point Likert scale – i.e., "*Valuta questa risposta:*" (see Fig. 16):

- 1 point - Poor: the answer is incorrect, incomplete, or irrelevant.
- 2 points - Fair: the answer is partially correct but lacks clarity or depth.
- 3 points - 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 upon submission ("*Valutazione registrata*").

Figure 16: Feedback mechanism embedded in the user interface via interactive rating buttons.

5.8.2 Storage and Export

Feedback is stored locally in a SQLite (Team, 2025) database (`feedback.db`) with each record capturing the timestamp, user query, generated answer, rating, and message index (Listing 5.5).

This schema ensures reproducibility and traceability by maintaining a transparent mapping between input, output, and evaluation. Records are inserted with minimal overhead through parameterised SQL statements and transaction-safe commits.

```sql
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
);
```

Listing 5.5: SQL schema of the feedback database.

For long-term preservation and collaborative access, the system implements a mechanism for periodic synchronization of the local database 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 system updates.[41]

From the UI sidebar, users can also export the full feedback dataset as a `.csv` file using the *"Esporta feedback"* functionality. This latter is powered by the `export_feedbacks()` function, which queries the database and converts records into a downloadable format using Pandas.

Collected feedback thus serves as a valuable resource for researchers, developers, and project coordinators to monitor performance, identify weaknesses, and guide system refinement over time.

5.9 Resources and Deployment

The GNA QA system is deployed on Streamlit Community Cloud, a platform that allows developers to publicly host interactive Python applications directly from a linked GitHub repository. The service abstracts most of the infrastructure management: each push to the repository automatically triggers a build and deployment, packaging dependencies specified in configuration files and exposing the application as a live web service via a public URL. This arrangement facilitated rapid iteration and made the system easily accessible to end users,

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

without the need to provision on-premises infrastructure or a dedicated model-serving layer. The full system is available at https://gna-assistant-ai.streamlit.app/.

However, long-running RAG pipelines are not trivial to host within such constrained environments. Indeed, early deployments surfaced memory leaks. Object references from retrieval and generation modules, especially cached embeddings and session state artefacts, were not consistently released, leading to a progressive increase in memory usage during extended interactions. Identifying these leaking objects required targeted profiling, after which caching decorators and object reinitialisation strategies were introduced to reduce overhead. Although these optimisations improved stability, the underlying workload still proved too heavy for the default configuration availabilities of Streamlit Cloud.

To mitigate this, we contacted the Streamlit team and requested additional resources. As a result, the hosting environment was scaled up to 8 gibibytes (GiB), a critical increase that allowed the application to sustain concurrent sessions and longer conversational histories without interruption.

This experience underlines a broader methodological point: deploying RAG in real-world systems is not merely a matter of algorithmic optimisation but also of infrastructure alignment, where careful monitoring, memory management, and strategic resource planning or negotiation are essential for sustaining performance under operational conditions.

5.9.1 Memory Management

As discussed above, deploying a RAG workflow with multiple NLP resources integrated into a single pipeline poses considerable challenges for environments lacking access to GPUs or large memory allocations, such as free hosting platforms like Streamlit Community Cloud. To keep the system sustainable under these constraints, several optimisation strategies were adopted in terms of memory management:

- **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 persistently. At application startup, only the metadata is loaded into memory, and the Faiss index is accessed through

memory-mapping to reduce RAM footprint;

- **Asynchronous processing:** Streamlit's async capabilities are leveraged to keep the UI responsive while background tasks run in parallel, preventing memory spikes during long-running operations;

- **Cache clearing policies:** `st.session_state` objects are pruned after each session or upon manual reset by the user to prevent memory bloating during prolonged use.

5.9.2 Computational Constraints Mitigation

To counterbalance the limitations of operating under strict computational constraints and preserve system responsiveness, the following strategies were introduced:

- **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 the external Mistral API prevents the local system from being overloaded and allows scaling independently of front-end performance;

- **Timeout and fallback handlers:** if the generation request exceeds 10 seconds or fails (e.g., due to API rate limits), the interface returns a graceful fallback response, allowing users to retry 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 has been developed with sustained attention to ethical considerations and data governance, particularly in the context of cultural heritage and public information. Transparency is pursued through open release of the source code and documentation of the pipeline – covering models, prompts, parameters –, together with versioned configurations, and machine-readable evaluation artefacts; per-run JSON logs record dataset IDs, model identifiers, and retrieval settings, all of which are made accessible through the project's GitHub repository. Both the prototype and the re-engineered system are distributed under the MIT Licence, a permissive software licence that explicitly allows reuse, modification, and redistribution, provided that attribution is maintained. This choice reflects a deliberate commitment to openness and collaboration, ensuring that the system can be freely adopted, adapted, or extended by other researchers, institutions or organisations.

Privacy is safeguarded by design, as the system processes no personally identifiable information and relies exclusively on publicly available or openly licensed materials. Licensing constraints are respected, as all resources – from the knowledge base to the language models and software libraries – are drawn from projects distributed under permissive terms that guarantee lawful reuse in research and educational contexts. Auditability is ensured through persistent logging of system performance and user feedback on Streamlit Community Cloud.

On the whole, these practices turn abstract commitments – provenance, privacy, licensing, and auditability – into concrete safeguards, allowing the GNA QA system to remain technically reliable and also a responsible instrument for public engagement.

Chapter 6

Results

This chapter presents the results with regard to the evaluation of the RAG QA system across its two stages of development: the prototype and the full-scale implementation.

Each followed a distinct evaluation protocol. For the prototype, responses were assessed through a dual strategy combining human annotation on a 5-point Likert scale – measuring consistency, fluency, completeness, and relevance – with automatic scoring from an external LLM (GPT-3.5), prompted as an evaluator. By contrast, the re-engineered system was tested on a synthetic collection of 508 single-hop and 908 combined queries automatically generated from the GNA operative manual. The assessment included intrinsic IR metrics – Recall@, MRR, nDCG@, AP@, Latency – to quantify retrieval performance, alongside qualitative human feedback, to capture user-facing quality. Feedback was collected systematically through in-interface ratings tasks: each response was labelled on a 3-point Likert scale – 1-star equals "poor", 2-stars equals "fair", 3-stars equals "good" –, offering direct insight into the perceived relevance, completeness, and fluency of the generated answers.

6.1 Prototyping Phase

The prototype offered an initial proof of concept, validating that a RAG architecture could be applied to the GNA corpus. Although limited in scale, it provided important insights into feasibility.

The evaluation outcomes (Fig. 17) show a clear profile: consistency was rated at the highest level (≈ 5), indicating that generated answers rarely contradicted themselves. Relevance (≈ 4.7) and fluency (≈ 4.9) also scored strongly, confirming the system's ability to generate both accurate and linguistically natural responses. Completeness, however, achieved slightly lower values (≈ 4.6), pointing to occasional gaps in coverage where answers did not fully integrate all available evidence.

Performance testing further revealed practical constraints. Embedding generation averaged 31.2 seconds per batch, while query responses were returned in ≈ 1.26 seconds on average. Although response time was acceptable for interactive use, overall efficiency lagged behind production-ready deployments, especially under heavy load.

In sum, the prototype confirmed the viability of a RAG-based QA approach, but also highlighted the need for more scalable retrieval infrastructure.

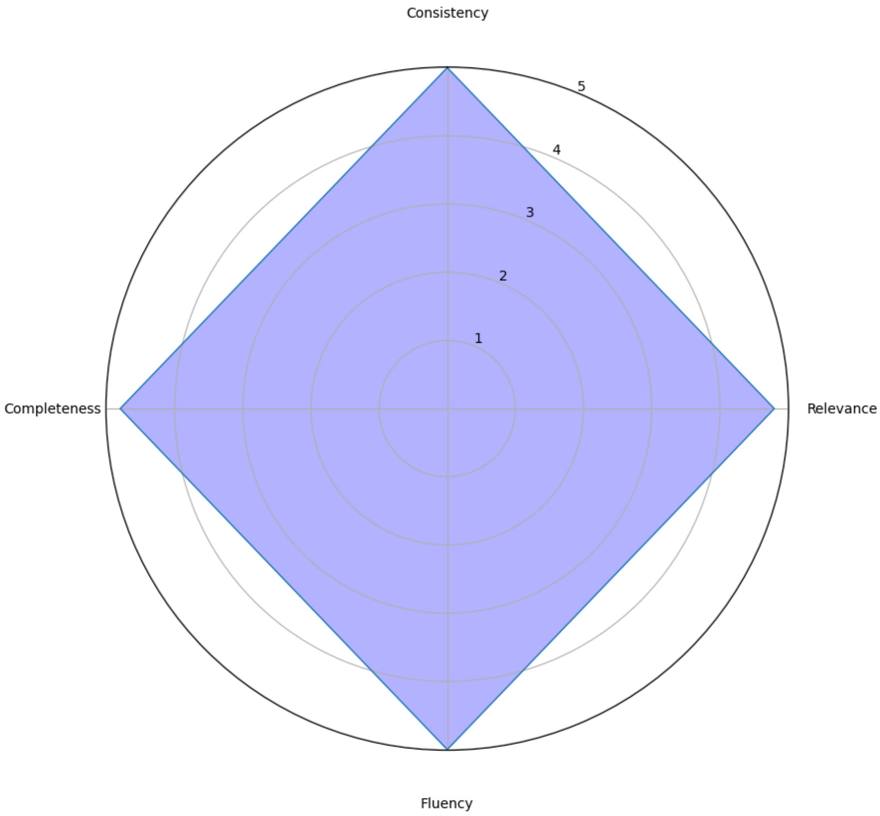


Figure 17: Evaluation results for prototype version of the GNA QA system.

6.2 Retrieval Performance and Ablation Studies

Retrieval performance pertinent to the full-scale system implementation was benchmarked in the experimental setup across different strategies: Dense, BM25, and Hybrid variants, with optional query rewriting and cross-encoder reranking. All runs were evaluated with @5 cut-offs – R@5, MRR, nDCG@5, AP@5 –, complemented by latency measurements (seconds per query). This setup enables a consistent comparison of coverage, ranking quality, and efficiency across methods. The benchmarked retrieval scores and the different configurations tested are summarised in Tab. 6.

The findings highlight distinct trade-offs across retrieval methods, presented hereafter.

Single-hop dataset.

The best overall performance was achieved by the **Hybrid + Score-blend** method without query rewriting and reranking (R@5 = 70.27, MRR = 48.59, nDCG@5 = 54.02, AP@5 = 48.59), with latency around 0.55s. This configuration consistently outperformed **Dense** retriever – the system's initial baseline – and **BM25** alone, showing the benefits of combining lexical and semantic signals. A close second was **Hybrid + Weighted RRF** (R@5 = 69.68, nDCG@5 = 52.49) at slightly lower latency (0.33s). **BM25** without extras proved a strong efficiency (R@5 = 65.15) and remained by far the fastest method ($\approx$ 0.001s), though it lagged behind hybrids in ranking quality. **Dense** retrieval showed competitive early-rank placement (MRR = 47.04) but lower recall and nDCG, confirming its limits when used in isolation.

Combined (single-hop + multi-hop).

Here, absolute scores dropped due to the increased complexity of multi-hop queries and the @5 cutoff, but the relative ordering of methods remained stable. **Hybrid + Score-blend** again provided the best balance, with R@5 = 55.35 and AP@5 = 36.69 at 0.45s. **BM25** retained an edge in precision at early ranks (MRR = 51.23, nDCG@5 = 57.13), again with negligible latency ($\approx$ 0.001s). **Hybrid + Weighted RRF** was also competitive (R@5 = 53.98, nDCG@5 = 56.92). **Dense** retrieval remained faster than hybrids (0.19–0.82s) but less effective overall.

Effect of rewriting and reranking.

Across all methods, adding query rewriting or reranking consistently reduced effectiveness and inflated latency. Reranking with the MS-MARCO English cross-encoder reordered Italian, domain-specific candidates suboptimally, while rewriting perturbed terminology that BM25 depends on. In practice, this meant added latency (up to 10 s in BM25 with rerank) and systematically lower recall and precision.

Latency and quality trade-offs.

The experiments exposed striking interplays between latency and retrieval quality across the methods evaluated. The fastest approach was **BM25** without any additional components, achieving near-instantaneous response times of approximately 0.001 seconds.[42] However, this

[42]While BM25 was also tested in isolation within these experiments, it is important to note that this method

speed came at the expense of lower recall and ranking effectiveness compared to more advanced methods. A more balanced option was provided by the **Hybrid + Score-blend** method without extras, which delivered a strong combination of recall and ranking quality while maintaining subsecond latency in the range of 0.45–0.55 seconds. This makes it particularly suitable for interactive use cases where both efficiency and retrieval accuracy are critical. In contrast, the use of **query rewriting** and **reranking** introduced substantial computational overhead, adding several seconds of latency. Crucially, these integrations did not yield meaningful improvements in retrieval quality, making them the least cost-effective option.

Overall, the findings suggest that hybrid methods offer the best compromise between quality and efficiency, whereas query rewriting and reranking may not justify their latency cost in practical deployments.

			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	<u>47.04</u>	52.18	<u>47.04</u>	<u>0.29</u>	50.78	48.21	53.70	34.98	<u>0.19</u>
	✓	✗	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	✗	✗	<u>69.68</u>	46.72	<u>52.49</u>	46.72	0.33	**53.98**	<u>50.93</u>	<u>56.92</u>	<u>36.15</u>	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	<u>53.35</u>	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 the test datasets. Best per column is bold and the second-best is underlined. Latency is measured in seconds per query. Reranking is performed using the *cross-encoder/ms-marco-MiniLM-L-6-v2* model.

is not intended to serve as a standalone retrieval strategy in a RAG pipeline. BM25 is a purely lexical approach that operates over inverted indexes and does not support vector search in a vector database. Its role in modern RAG chains is typically complementary, improving precision through the combination of keyword-based and embedding-based retrieval methods in hybrid search. Thus, its inclusion here serves primarily as an IR benchmark rather than as a practical candidate for deployment on its own.

6.3 Qualitative Analysis

The re-engineered system was evaluated not only through intrinsic metrics but also via direct user feedback, providing a complementary perspective on performance. Based on a set of 50 user ratings (see Fig. 18), 64% of answers received 3 stars ("good"), 26% received 2 stars ("fair"), and 10% received 1 star ("poor").

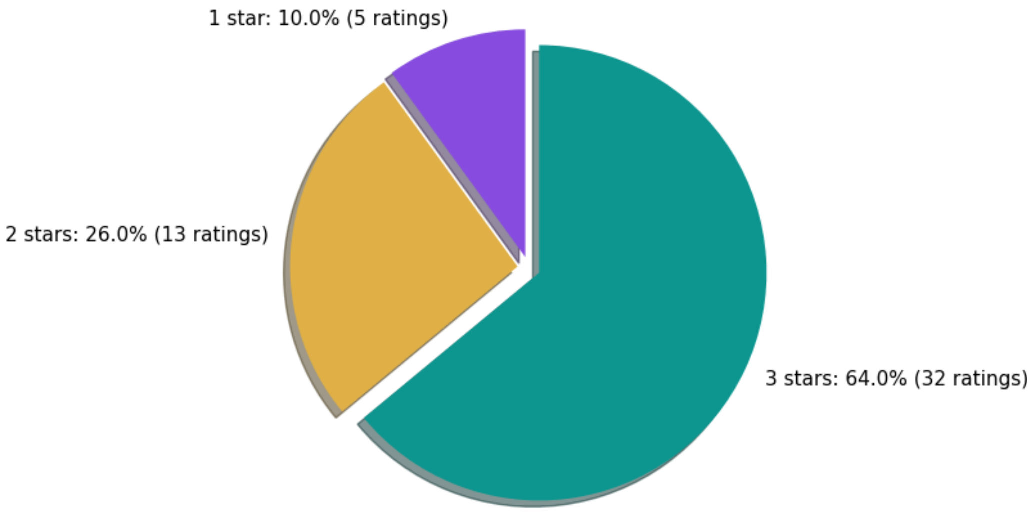


Figure 18: Distribution of user feedback ratings, showing the proportion of responses evaluated as poor (1 star), fair (2 stars), and good (3 stars).

Instances of lower relevance were generally linked to ambiguous queries or to cases where vector sparsity limited the number of retrieved chunks. Despite these shortcomings, users highlighted several strengths: the system's multilingual capabilities, the user-friendly interface, and the attempt to provide traceability through citations – though, as discussed in Chap. 7, the citation mechanism requires further refinement.

Edge-case testing yielded additional insights. For out-of-domain queries, the system largely succeeded in avoiding hallucinations by producing cautious fallback responses. For example, (Fig. 19), illustrates a query about contemporary art exhibitions in Rome: although the reply is incomplete and does not provide relevant content to the query, it faithfully reflects the system prompt's instruction to refrain from answering questions outside the GNA scope, instead redirecting the user to the knowledge base.

Equally instructive were the tests on HTML snippets retrieval – a particular feature of the GNA user manual where HTML tags are embedded as procedural instructions for the formatting of QGIS layouts. In these cases, the assistant correctly surfaced and rendered the relevant "code" snippet as a dedicated canvas (see Fig. 20), preserving both syntax and proper context. This demonstrated the system's capacity to handle heterogeneous content types aside from plain text, thus addressing one of the distinctive characteristics of the GNA documentation.

Overall, the feedback evaluation confirmed the capability and added value of a RAG pipeline tailored to the GNA corpus. The prototype established proof of concept, while the full-scale implementation improved retrieval robustness and scalability. Although some challenges persist – particularly with completeness, citation reliability, and response latency under query rewriting and reranking – the system consistently produced relevant, fluent, and context-sensitive answers, confirming its potential as a practical assistant for navigating the GNA documentation.

Figure 19: Edge case test of the GNA AI Assistant, showing an out-of-domain query about contemporary art exhibitions in Rome and the system's fallback response with redirection to the GNA knowledge base.

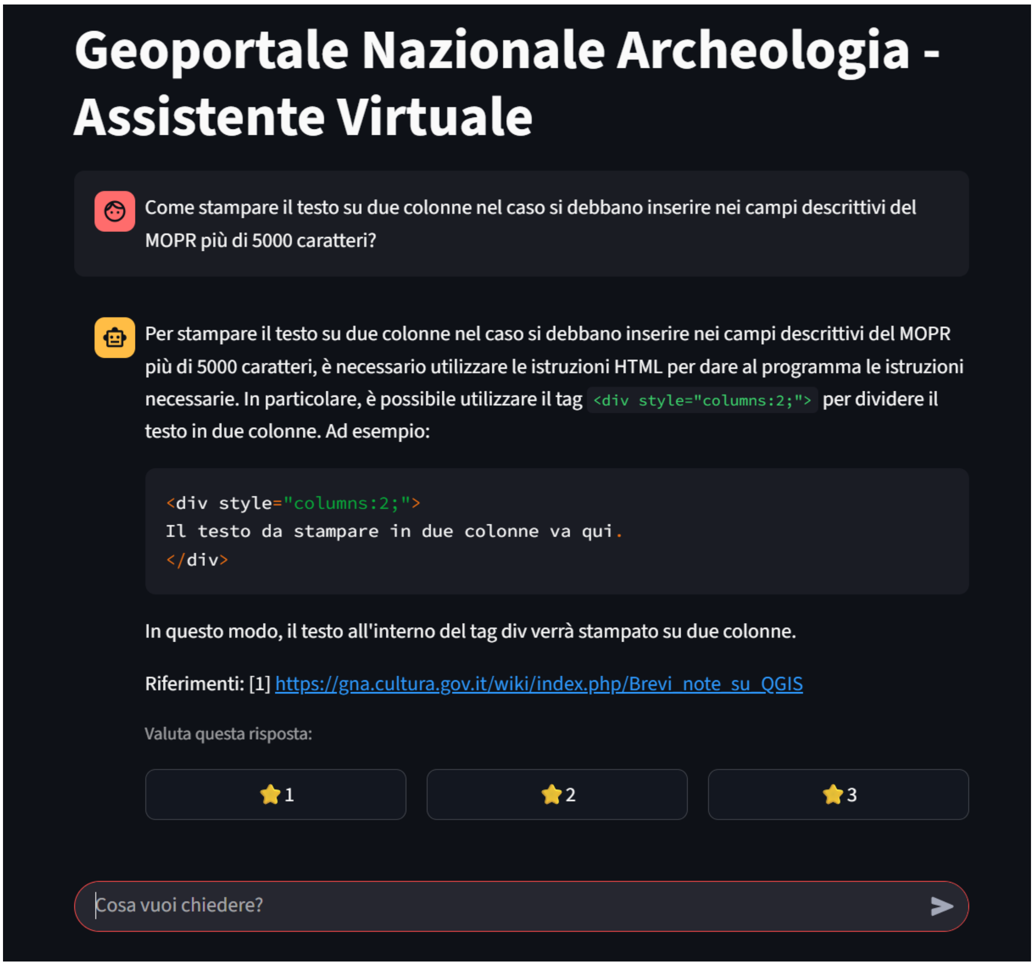


Figure 20: Example of the GNA QA system retrieving and presenting HTML code snippets from the knowledge base.

Chapter 7

Discussion

The development and evaluation of the GNA AI assistant offer a valuable vantage point for reflecting on both the methodological intricacies of retrieval-augmented generation and its broader implications within cultural heritage contexts. What began as an attempt to address a specific practical challenge – easing access to dense and highly specialised documentation – gradually unfolded into a broader inquiry into how contemporary AI methods align, or fail to align, with the realities of scholarly and professional practice in archaeology and the digital humanities. The discussion that follows is not intended as a conclusive statement to frame the project as a success story or a cautionary failure; rather, it situates the results within their larger methodological, technical, and disciplinary context.

7.1 Methodological Reflections

The project's two development phases, from prototype to full-scale implementation, trace an evolving methodological path. The prototype established the viability of pairing dense retrieval with generative AI to handle Italian-language queries, and demonstrated that, in the absence of authoritative gold standards, a custom evaluation strategy – combining human annotation with LLM-based evaluation – offered a workable, if imperfect, assessment strategy. At the same time, the prototype revealed critical bottlenecks. The dependence on LangChain undoubtedly accelerated prototyping stage, providing a convenient framework that unified stateful conversation, retrieval, and integration with Mistral and Streamlit. In parallel, its layers of abstraction came at a cost, introducing inefficiencies and concealing low-level processes such as API rate-limit handling and metadata routing. In this sense, the prototype fulfilled an essential role not simply by establishing technical feasibility, but also by acting as a diagnostic instrument, revealing the constraints of facilitated development frameworks and informing the methodological stance that guided the more mature implementation.

The re-engineered system refined these lessons into a more deliberate architectural framework. LangChain was replaced with custom modules and clearer component boundaries, together with an evaluation protocol that was more amenable to profile, ablate, and optimise. Here, the results foreground a series of needful compromises that are not merely technical but epistemic. Precision and recall in heritage QA translate into tensions between coverage and depth. Some users favour concise answers for quick consultation, while others expect fuller contextualisation; the system must balance these competing needs. In this context, results from ablation studies on retrieval reveal the uneven terrain of performance across different strategies. Hybrid approaches consistently proved the most reliable recipe, outperforming the baseline standalone dense retrieval. The score-blend variant, in particular, struck a productive balance between coverage and ranking quality, while retaining subsecond latency. These findings underline the pragmatic value of combining lexical and semantic signals in cultural heritage domains, where documents are rich in technical vocabulary but also require semantic generalisation. Likewise, they caution against uncritical reliance on techniques such as query rewriting and cross-encoder reranking, which in this setting introduced drawbacks and even depressed retrieval quality, partly due to the linguistic and domain mismatches with available cross-encoders.

User feedback painted a complementary perspective. A majority of responses were rated positively, with fluency and topical relevance consistently noted as strengths. These reactions suggest that, when sufficient contextual material was retrieved, the generative model was capable of producing responses that aligned well with user expectations in both style and substance. Still, a non-negligible portion of answers was judged incomplete or only partially satisfactory. This pattern points to retrieval obstructions – rather than deficiencies in generative quality – as the primary factor limiting user satisfaction. In other words, the weakest link in the pipeline is not the expressive power of the language model, which proved more than adequate in rendering information, but the adequacy of the retrieval step in capturing all the fragments required to compose a comprehensive and contextually grounded answer. The feedback thus reinforces a central lesson: in domain-specific QA, generation succeeds only insofar as retrieval delivers complete and precise evidence, with success depending less on linguistic fluency than on the adequacy of the retrieved material.

7.2 Technical, Infrastructural and Ethical Considerations

Deploying the system in a constrained hosting environment brought to light another layer of challenges that often remain implicit in AI research: infrastructure. Running a RAG-based QA system with multiple NLP resources on Streamlit Community Cloud, with its tight memory caps and CPU-only limitations, demanded aggressive optimisation. Lazy loading of embeddings, memory-mapped Faiss index, batch processing, cache pruning, and garbage collection routines were not optional refinements but survival strategies. Even then, memory leaks emerged, requiring close inspection of leaking objects and eventually negotiations with the platform to secure extended resources. This experience brings to the surface a broader consideration, highlighting a critical but often neglected truth about the practical life of RAG systems, which depends as much on engineering choices and infrastructural adaptability as on algorithmic innovation.

The design of the GNA AI assistant was governed both by technical feasibility and by explicit commitments to transparency, licensing compliance, and auditability. Logs of evaluation runs, public release of code and configurations, and attention to the licensing status of all components collectively ensure that the system does not devolve into a “black box” artefact, opaque to its future users and maintainers. Privacy, too, is safeguarded by design. The system processes no personal data and relies exclusively on publicly accessible or openly licensed materials. These principles are not afterthoughts but foundational elements of trustworthiness, especially in the cultural heritage sector where provenance and accountability are intrinsic values.

Nevertheless, provenance remains an unfinished frontier. Although the system is designed to provide in-answer citations pointing to the retrieved chunks or source URLs, in practice this mechanism is fragile and at times inconsistent. This limitation marks an important avenue for future refinement, for without reliable citation grounding, the epistemic authority of the system remains limited. Indeed, a robust provenance layer is more than a cosmetic feature. In domains such as archaeology, where the authority of knowledge rests on traceability, it constitutes a meaningful requirement. Fluent responses alone do not suffice; the provenance of those responses is often as important as the content itself. Strengthening this mechanism is therefore a central priority in the developmental roadmap.

7.3 Evaluating Without a Net

Since no authoritative gold standard exists for queries specific to the *Geoportale Nazionale per l'Archeologia*, nor does a legacy system provide a baseline for comparison, evaluation relied on synthetic test sets and user ratings. Despite careful design, the evaluation protocol carries its own caveats. Generating evaluation queries with the help of an LLM introduced an inevitable degree of lexical leakage. This likely biased the task toward dense retrievers, which benefit from semantic similarity, while disadvantaging lexical strategies that rely less on token overlap. Moreover, real users rarely phrase queries in such clean, corpus-mirroring forms; they abbreviate, mistype, or ask obliquely. Thus, the synthetic datasets surely provided reproducibility and diagnostic clarity, but they cannot fully substitute for faithful simulations of practice and live deployment data.

The methods adopted highlighted general tendencies but fell short of capturing the nuanced requirements of archaeologists and heritage professionals in daily workflows. It is worth acknowledging the irony: in a research landscape overflowing with QA benchmarks – ranging from early large-scale resources such as SQuAD, which defined the paradigm of span-based reading comprehension, to more compound datasets like HotpotQA, designed to test multi-hop reasoning across documents – none align with the task at hand. These resources have unquestionably driven progress in NLU, establishing common baselines and enabling systematic comparison across models. Yet their scope remains fundamentally oriented toward general-purpose QA in English, often privileging cleanly segmented passages, unambiguous answers, and simplified contexts. In opposition, the richness and complexity of the GNA user manual – with its blend of procedural guidelines, tables, and domain-specific jargon – resist the assumptions embedded in these benchmarks. What emerges is a striking mismatch; while existing datasets are indispensable for model development, they fall short as instruments for evaluating systems designed to operate within the highly contextual workflows of archaeology, cultural heritage and GLAM, which in this case are conducted in Italian and guided by practical application.

7.4 Relation to Previous Work

Positioning the GNA QA system within the broader research landscape of retrieval-augmented generation – across question answering and adjacent domains – brings into relief both continuities with existing approaches and points of departure. Recent literature documents a proliferation of RAG deployments across biomedical, legal, and cultural domains – spanning enterprise settings and academic research alike – each grappling with common challenges of retrieval quality, hallucination control, and evaluation (Agarwal et al., 2025; Bevara et al., 2025; Soman and Roychowdhury, 2024). In this context, the GNA project aligns with a growing movement to adapt RAG pipelines to corpora rooted in specialised domains, while also introducing constraints distinctive to cultural heritage infrastructures.

Methodologically, the system resonates with technical studies such as Soman and Roychowdhury (2024) work, which demonstrated the importance of chunking strategies and domain-specific terminology in retrieval accuracy. The RAG pipeline for the GNA similarly emphasised semantic preprocessing and careful segmentation of the operative manual, showing how small design choices directly affect system robustness. Like Bevara et al. (2025) in the library sciences, the GNA project foregrounded ethical and licensing considerations, where transparency and ethical accountability are understood as inseparable from technical performance.

Evaluation practices likewise reveal both alignment and shortcomings. Synthetic query sets generation, adopted here to compensate for the absence of gold standards, echoes Bor-Woei (2024) argument for synthetic corpora in low resource domains. Yet it also introduced lexical leakage, potentially advantaging dense retrieval and diminishing the ecological validity of results – an impediment less visible in biomedical and legal RAG applications, where annotated benchmarks (e.g., BioASQ) are available. The same limitations are noted in contemporary surveys (Yue, 2025), reminding that evaluation methods must be interpreted with caution, and results contextualised against the artificiality of test conditions. By the way, the practical combination of automatic metrics, synthetic datasets, and user-centred ratings situates this project within a wider methodological turn toward hybrid evaluation protocols (Abeysinghe and Circi, 2024; Gupta et al., 2024).

Concurrently, the system exposes gaps and unresolved tensions that distinguish it from many existing RAG deployments. Whereas large-scale initiatives often aim for general-purpose scalability (Ramos-Varela et al., 2025; Lála et al., 2023), this project was deliberately scoped to a single, highly specialised corpus: the operative manual of *Geoportale Nazionale per l'Archeologia*. This narrow but dense domain called for balancing lexical and semantic retrieval methods, resulting in the demonstrated superiority of hybrid retrieval approaches. In this respect, the findings echo Davis (2025), whose prototype for web archives similarly showed the advantages for retrieval precision of dense embeddings coupled with noise-reduction preprocessing, but they contrast with Ramos-Varela et al. (2025), who observed that large-context models could outperform RAG in querying multimodal museum collections. The lesson here is that performance is highly sensitive to corpus structure, query style, and domain vocabulary, framing RAG not as a one-size-fits-all solution, but as a family of adaptable techniques.

In the digital humanities specifically, the GNA AI assistant shares affinities with projects such as iREAL (Callaghan and Vieira, 2025) and CAT-IA (Barbato, 2025), which likewise confront with the epistemic stakes of provenance and cultural sensitivity. Like these initiatives, it foregrounded the necessity of provenance-tracking and interpretability, though its own citation mechanism remains a provisional feature. This limitation underscores a broader challenge also identified in Pollin et al. (2024), namely that without reliable grounding in source material, the danger for RAG is that of being perceived as obscure in its underlying processes, a risk particularly acute in cultural heritage and GLAM contexts where authority is inseparable from traceability.

Looked at in aggregate, these points of alignment and divergence position the GNA AI assistant as both participant in and contributor to ongoing debates. Its findings reinforce established lessons – such as the value of hybrid retrieval and the indispensability of provenance – while its domain, specific to the scope of technical documentation, highlights unresolved tensions around evaluation, scalability, and infrastructural constraints. Within the DH discourse, the system offers a modest but tangible case study, a reminder that applying RAG to cultural heritage entails more than merely porting established techniques into a new domain. It demands reworking them in light of epistemic values, infrastructural challenges, and humanistic priorities.

7.5 Toward a Humanistic AI

Beyond the technicalities, this project gestures toward wider implications for the digital humanities. The integration of AI systems into cultural heritage infrastructures is not merely an exercise in automation; it represents a deeper shift in how knowledge is organised, accessed, and mediated. In archaeology, where documentation is often layered, fragmented, and deeply contextual, the promise of retrieval-augmented generation lies in its capacity to interlace disparate traces of information into coherent forms of support, articulated through the lens of user queries. However, the risks of oversimplification and hallucination are ever present, demanding not blind adoption but careful stewardship.

This thesis does not claim that the GNA QA system possesses an understanding of archaeology, nor that it replaces the interpretive labour of scholars. Instead, it demonstrates how RAG techniques can operate as mediating tools: structuring access, enhancing discoverability, and supporting human decision-making. The value of such systems is not measured by abstract notions of comprehension but by their utility in enabling more efficient, transparent, and contextually reliable engagement with cultural heritage data.

The trajectory traced here is not an end but an opening. The system as implemented constitutes a functioning end-to-end assistant, capable of live deployment in relation to the GNA ecosystem. Its contribution consists in demonstrating how methodological choices, infrastructural constraints, and ethical commitments intersect in the practical design of AI for the humanities. Beyond this, the project shows potential to restructure access to fragmented and technically dense knowledge, turning an intricate manual into answers that are more immediate and digestible. Its fragility, however, rests on careful parameter tuning, infrastructural support, and evaluation strategies sensitive to specific needs of the domain.

Future development may broaden the scope beyond the operative manual to archaeological datasets more generally, strengthen provenance mechanisms through reliable citation grounding, or explore multimodal integration by weaving text, images, and geospatial data into unified retrieval workflows.

Ultimately, the discussion circles back to the tension raised at the outset: the gap between technical ingenuity and the subtleties of human knowledge. Bridging this divide is less a matter of larger models or more data, than of cultivating a sustained dialogue between AI research and humanistic inquiry. The GNA AI assistant is not to be seen as a replacement for scholarly interpretation, but as a mediator between users and the knowledge base, capable of efficiently surfacing relevant passages, shaping information flows, and enabling quicker orientation within complex documentation. In this sense, it offers a tangible contribution to the ongoing dialogue between AI and the humanities, pointing toward a future in which such systems act less as stand-ins for understanding and more as partners in the interpretation and tutelage of cultural heritage.

Chapter 8

Conclusion

This thesis has presented the design, implementation, and evaluation of an end-to-end question-answering system for the *Geoportale Nazionale Archeologia (GNA)*, showing how retrieval-augmented generation (RAG) can be applied to improve access to highly specialised cultural heritage documentation. While developed for a specific institutional use case, the GNA AI assistant offers insights of broader relevance, both technical and scholarly, into the evolving role of AI within the digital humanities.

The contributions are threefold. First, the system itself: a functioning assistant powered by AI, that systematically integrates crawling, chunking, semantic embedding, retrieval, generation, and feedback into a coherent architecture, tested across synthetic test datasets and live user interactions. Second, the evaluation framework: a dual approach that combined intrinsic retrieval metrics with human-centred assessments, making visible both capability at system level and performance as experienced by end users. Third, the methodological reflections: an exploration of the trade-offs – between precision and recall, fluency and completeness, speed and accuracy – that define the practical life of RAG systems in real-world contexts.

The results confirm that RAG can enhance access to dense archaeological resources, producing responses with a swifter and more accessible character than the laborious manual consultation of numerous records of technical documentation. At the same time, they underscore the fragility of such systems. Performance is highly sensitive to corpus structure, domain vocabulary, and infrastructural constraints; remain crucial yet challenging to sustain; and evaluation practices, when based on synthetic data, risk diverging from the texture of real user needs. The GNA QA system should therefore not be mistaken for a surrogate of scholarly interpretation. Its contribution is more modest but also more practical: it acts as an intermediary agent between users and documents, disclosing relevant passages, clarifying information flows, and enabling quicker orientation within complex materials.

Looking ahead, several fronts stand out for further development:

- Technical refinements will be required. In particular, retrieval recall, especially for multi-hop queries, must be improved through higher candidate thresholds or repacking strategies. Provenance mechanisms need stabilisation, so that citations become reliable and trust is reinforced.
- Long-term adaptability must be addressed, with strategies for updating embeddings as the knowledge base evolves and for handling security and privacy more explicitly.
- Evaluation must move toward more consistent collaboration with archaeologists and cultural heritage professionals, ensuring that system performance is judged not only by technical metrics but also by disciplinary relevance.

These improvements will be essential if RAG systems are to move from prototypes into robust infrastructures within the cultural heritage sector. However, what matters most here reaches beyond the realm of technical optimisation, as the contribution of this project aims at heading further than incremental adjustments.

This work shows how the design of AI systems, whether in cultural heritage or other domains, is shaped by the intersection of methodological choices, infrastructural realities, and ethical considerations. In this light, the integration of RAG into heritage infrastructures is not merely a technical upgrade, but a reconfiguration of how knowledge is accessed, contextualised, and mediated. Archaeology, with its dispersed and stratified documentation, offers a particularly vivid test case: here, RAG has the capacity to weave fragmented records into coherent accounts oriented around user queries. Yet such promise is inseparable from risk – the risk of oversimplification, of introducing artificial coherence, of hallucination. For this reason, critical human oversight remains indispensable, ensuring that these tools serve as mediators rather than substitutes in the interpretation of cultural knowledge.

Future research should extend this orbit, setting new vectors toward the multitude of possible directions, each illuminating a different constellation of inquiry. One promising outlet lies in multimodal retrieval, where text, images, and geospatial data are brought together within unified pipelines. Another is the development of models adapted to specific domains, multilingual in scope and capable of reflecting the linguistic and cultural diversity of heritage data. Equally important are collaborative evaluation frameworks that integrate field's experts into the assessment process, producing annotated datasets and conducting systematic user studies that better capture real information needs.

The scope of applications, too, could broaden considerably. From integration within GLAM realities and DSEs, to comparative studies with traditional cataloguing practices and educational settings, exploring such scenarios would provide a fuller sense of the scalability and adaptability of RAG systems across heterogeneous cultural contexts. In the specific case of digital scholarly editions, RAG might open new avenues for engaging with complex editorial data: users could, for instance, query textual variants, access synthesized accounts of critical notes, or navigate apparatus structures with greater efficiency and contextual awareness. When applied to TEI-encoded corpora, these systems might operate at a fine level of granularity – sections, annotations, or witness readings – while still producing responses that are coherent and transparently cited. Seen from this perspective, RAG may gradually shift from a generic AI technique toward a specialised scholarly instrument, capable of enhancing access while remaining attentive to the rigor and interpretive care central to editorial practices.

At a more conceptual level, the work opens up pressing questions. Can provenance be safeguarded without undermining usability? How can retrieval-generation pipelines scale without collapsing under their own complexity? A further issue lies in evaluation: in what ways might synthetic protocols be refined to better approximate the often messy and unpredictable demands of real-world information needs? And, perhaps most crucially, how might AI systems be designed not to replace but to complement, extend and reinforce the interpretive practices that remain at the very heart of humanistic inquiry?

The GNA AI assistant does not resolve these questions, but it makes them tangible. It shows that generative AI, when grounded in contextual retrieval, can serve as a pragmatic partner in the stewardship of cultural heritage, expanding access while respecting the epistemic specificities of the domain. More broadly, it illustrates a path forward in which AI functions less as a substitute for human understanding and more as a companion – one that structures access, enhances discoverability, and supports informed engagement with complex resources.

In this sense, the project is less an endpoint than an invitation – a doorway toward new questions, opportunities, and directions for future research. It offers a modest yet concrete contribution to the dialogue between generative AI and the humanities, a small brick laid in what may become a much larger hall of digital scholarship, portending toward a future in which AI technologies are not peripheral tools but structural beams within scholarly and heritage settings. The challenge now is to shape this integration with care, so that the ingenuity of technical design remains attuned to the subtleties of human knowledge. That challenge, and its possibilities, are both a weight and a promise. These will chart the course of the next explorations, where the paths of machine computation and human interpretation continue to converge, diverge, and intertwine.

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 *single-hop* dataset, comprising 508 queries, and on the *combined* dataset, comprising 908 queries in total (single-hop 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, spaCy, KeyBERT, NLTK, faiss-cpu, 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.
IR	Information retrieval	The field of computer science that focuses on finding relevant information in large collections of data, typically unstructured text (like documents, web pages, or articles).
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.
QA	Question answering	A task in NLP and IR that focuses on building systems capable of automatically answering questions from a user.
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 type of neural network trained on massive text corpora to understand and generate human language.
API	Application programming interface	A set of protocols and tools that allow different software applications to communicate and interact with each other.
GNA	Geoportale Nazionale Archeologia	Italy's institutional repository for archaeological data, hosting extensive documentation and resources related to the country's cultural heritage.

Continued on next page

Term	Full form	Glossary definition
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.
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, analyse, manage, and present spatial or geographic data, often used in archaeology for mapping and spatial analysis.
QGIS	Quantum GIS	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.
ML	Machine learning	A subset of AI that involves training algorithms to recognise 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.

Continued on next page

Term	Full form	Glossary definition
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.
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	SPARQL Protocol and RDF Query Language	A query language and protocol used to retrieve and manipulate data stored in 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.
CSV	Comma-Separated Values	A text file format used to store tabular data (numbers and text) where each row represents a record, and each column (field) is separated by a comma.
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 labelling.
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.

Continued on next page

Term	Full form	Glossary definition
OpenAI		An artificial intelligence research and deployment company based in San Francisco (USA).
GPT	Generative Pre-Trained Transformer	A family of LLMs developed by OpenAI.
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.
KeyBERT		A keyword extraction technique using BERT embeddings to generate the keywords and keyphrases most similar to a document.
BAAI	Beijing Academy of Artificial Intelligence	A Chinese research institute that develops and releases cutting-edge AI models. BAAI is the organization behind BGE, and also known for other large-scale AI projects.
BGE	BAAI General Embedding	A family of embedding models designed for dense retrieval and semantic search.
Intfloat	Intelligent Floating Point	A Hugging Face repository that develops and makes available open-source AI models.
E5	Embedding from Explicitly-Explained Supervision	A family of text embedding models developed by the research group Intfloat.
Intfloat/e5		A family of models available on Hugging Face and based on the implementation of *E5*.
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 specialised 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.

Continued on next page

Term	**Full form**	**Glossary definition**
TEI	Text Encoding Initiative	A set of guidelines for encoding literary and linguistic texts in XML, providing a standardised 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 scalable similarity search, supporting distributed deployment, diverse indexing methods, and efficient handling of high-dimensional vector data.
Faiss	Facebook AI Similarity Search	A library for efficient similarity search and clustering of dense vectors, optimised for performance on both CPUs and GPUs.
Qdrant	Quadrant Vector Database	An open-source vector database designed for high-performance similarity search. It supports filtering, payload (metadata) management, and scalable deployment, making it suitable for production-ready settings.
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 optimise 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.

Continued on next page

Term	Full form	Glossary definition
RAGAS	Retrieval-Augmented Generation Assessment System	An open-source evaluation framework for RAG systems. It provides metrics that assess both the retrieval and generation stages, focusing on aspects such as context relevance, answer faithfulness to retrieved documents, and overall response quality. Unlike traditional text similarity metrics (e.g., BLEU, ROUGE), RAGAS is designed to capture factual accuracy and contextual appropriateness, making it better suited for evaluating RAG-based applications like question answering and conversational agents.
AHE	Adaptive histogram equalization	A computer image processing technique designed to enhance contrast in pictures. Unlike standard histogram equalization, the adaptive approach divides the image into multiple regions, generates a separate histogram for each, and then redistributes the lightness values based on these localized histograms.
CLAHE	Contrast limited AHE	A variant of adaptive histogram equalization in which the contrast amplification is limited, so as to reduce the problem of noise amplification.
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

[MistralAI] Improve MistralAIEmbeddings by ZhangShenao · Pull Request #29242 · langchain-ai/langchain · GitHub. 2025, August. Accessed 23 August 2025. https://web.archive.org/web/20250823161804/https://github.com/langchain-ai/langchain/pull/29242.

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 3 August 2025. DOI: 10.48550/arXiv.2406.03339. http://arxiv.org/abs/2406.03339.

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 24 July 2025. https://doi.org/10.60974/GNA_03.

Ada Gabucci. 2025, July. Accessed 24 July 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 23 July 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 23 July 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 23 July 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 26 July 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 21 July 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 24 July 2025. https://aws.amazon.com/de/what-is/retrieval-augmented-generation/.

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 21 July 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 26 July 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 19 June 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 27 July 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 29 July 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 21 July 2025. DOI: 10.48550/arXiv.2412.15404. http://arxiv.org/abs/2412.15404.

Bajaj, Payal, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder et al. 2018. *MS MARCO: A Human Generated MAchine Reading COmprehension Dataset.* ArXiv:1611.09268, October. Accessed 27 August 2025. DOI: 10.48550/arXiv.1611.09268. http://arxiv.org/abs/1611.09268.

Barbato, Fabrizio. 2025. *Nasce Cat-IA, l'agente conversazionale che semplifica la consultazione del Catalogo generale dei beni culturali* [in it-IT], April. Accessed 21 July 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 19 July 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 24 July 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 23 July 2025. https://openreview.net/forum?id=DbZDbg2z9q.

Bronsdon, Conor. 2025. *Understanding Human Evaluation Metrics in AI: What They Are and How They Work* [in en], March. Accessed 2 September 2025. https://web.archive.org/web/20250902081409/https://galileo.ai/blog/human-evaluation-metrics-ai.

BUP Solutions. *BUP Solutions.* Accessed 25 July 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 26 July 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 24 July 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 19 July 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 21 July 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 23 July 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 19 July 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 23 July 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 3 August 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 26 July 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 28 July 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 26 July 2025. DOI: 10.1007/s10115-017-1100-y. http://link.springer.com/10.1007/s10115-017-1100-y.

Dinan, Emily, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli and Jason Weston. 2019. *Wizard of Wikipedia: Knowledge-Powered Conversational agents.* ArXiv:1811.01241, February. Accessed 26 August 2025. DOI: 10.48550/arXiv.1811.01241. http://arxiv.org/abs/1811.01241.

Douze, Matthijs, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini and Hervé Jégou. 2024. "The Faiss library", arXiv: 2401.08281 `[cs.LG]`.

Es, Shahul, Jithin James, Luis Espinosa-Anke and Steven Schockaert. 2023. *Ragas: Automated Evaluation of Retrieval Augmented Generation.* Accessed 19 June 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 3 August 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 19 July 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 26 July 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 27 July 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 21 July 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 26 July 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 24 July 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 3 August 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 10 July 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 24 July 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 27 July 2025. DOI: 10.1145/1460690.1460714. http://portal.acm.org/citation.cfm?doid=1460690.1460714.

GROBID. 2008–2025. https://github.com/kermitt2/grobid.

Grootendorst, Maarten. 2025. *MaartenGr/KeyBERT.* Original-date: 2020-10-22T12:21:22Z, September. Accessed 10 September 2025. https://github.com/MaartenGr/KeyBERT.

Guo, Rentong, Xiaofan Luan, Long Xiang, Xiao Yan, Xiaomeng Yi, Jigao Luo, Qianya Cheng, Weizhi Xu, Jiarui Luo, Frank Liu et al. 2022. "Manu: a cloud native vector database management system". *Proceedings of the VLDB Endowment* 15 (12): 3548–3561.

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 19 June 2025. DOI: 10.48550/arXiv.2410.12837. http://arxiv.org/abs/2410.12837.

Hajba, Gábor László. 2018. *Website Scraping with Python: Using BeautifulSoup and Scrapy.* 1st ed. 2018. Berkeley, CA: Apress : Imprint: Apress. ISBN: 9781484239254.

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 27 July 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 19 June 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 26 July 2025. DOI: 10.1017/S1351324901002807. https://www.cambridge.org/core/product/identifier/S1351324901002807/type/journal_article.

Ho, Xanh, Anh-Khoa Duong Nguyen, Saku Sugawara and Akiko Aizawa. 2020. *Constructing A Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps.* ArXiv:2011.01060, November. Accessed 26 August 2025. DOI: 10.48550/arXiv.2011.01060. http://arxiv.org/abs/2011.01060.

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

Open Source vs. Proprietary LLMs: A Comprehensive Comparison. 2025, January. Accessed 19 June 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 23 July 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 28 July 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 26 July 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 29 July 2025. DOI: 10.48550/arXiv.2001.08361. http://arxiv.org/abs/2001.08361.

Kotonya, Neema and Francesca Toni. 2020. *Explainable Automated Fact-Checking for Public Health Claims.* ArXiv:2010.09926, October. Accessed 26 August 2025. DOI: 10.48550/arXiv.2010.09926. http://arxiv.org/abs/2010.09926.

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 28 July 2025. https://www.hsozkult.de/event/id/event-142284.

Kwiatkowski, Tom, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein et al. 2019. "Natural Questions: A Benchmark for Question Answering Research" [in en]. *Transactions of the Association for Computational Linguistics* 7 (November): 453–466. ISSN: 2307-387X, accessed 26 August 2025. DOI: 10.1162/tacl_a_00276. https://direct.mit.edu/tacl/article/43518.

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 25 July 2025. DOI: 10.48550/arXiv.2312.07559. http://arxiv.org/abs/2312.07559.

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

Lee, Chris van der, Albert Gatt, Emiel van Miltenburg and Emiel Krahmer. 2021. "Human evaluation of automatically generated text: Current trends and best practice guidelines". *Computer Speech & Language* 67 (May): 101151. ISSN: 0885-2308, accessed 23 August 2025. DOI: 10.1016/j.csl.2020.101151. https://www.sciencedirect.com/science/article/pii/S0885230820300084X.

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 19 June 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 28 July 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 9 August 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 3 August 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 26 July 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 23 July 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 19 July 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 26 July 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 25 July 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 24 July 2025. https://gna.cultura.gov.it.

Develop a RAG Solution - Chunking Phase - Azure Architecture Center | Microsoft Learn. 2025, January. Accessed 25 August 2025. https://web.archive.org/web/20250825093743/https://learn.microsoft.com/en-us/azure/architecture/ai-ml/guide/rag/rag-chunking-phase.

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

Miller, George A. 1992. "WordNet: A Lexical Database for English". In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992.* Accessed 7 September 2025. https://aclanthology.org/H92-1116/.

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

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

Navigli, Roberto, Michele Bevilacqua, Simone Conia, Dario Montagnini and Francesco Cecconi. 2021. "Ten Years of BabelNet: A Survey" [in en]. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence,* 4559–4567. Montreal, Canada: International Joint Conferences on Artificial Intelligence Organization, August. ISBN: 9780999241196, accessed 7 September 2025. DOI: 10.24963/ijcai.2021/620. https://www.ijcai.org/proceedings/2021/620.

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 21 July 2025. https://amslaurea.unibo.it/id/eprint/34197/.

Normativa - FOIA [in it-it]. 2016. Accessed 3 August 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 19 July 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 23 July 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 23 July 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 29 July 2025. DOI: 10.3115/v1/D14-1162. https://aclanthology.org/D14-1162/.

Pograri, Lucrezia. 2025a. *Prototype of the AI Assistant for Geoportale Nazionale Archeologia (GNA),* July. Accessed 1 September 2025. DOI: 10.5281/zenodo.17018618. https://zenodo.org/records/17018618.

———. 2025b. *Question-Answering AI Assistant for Geoportale Nazionale Archeologia (GNA),* September. Accessed 1 September 2025. DOI: 10.5281/ZENODO.17018599. https://zenodo.org/doi/10.5281/zenodo.17018599.

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 28 July 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 28 July 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 28 July 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 27 July 2025. DOI: 10.1145/290941.291008. https://dl.acm.org/doi/10.1145/290941.291008.

Qdrant. *GitHub - qdrant/qdrant: Qdrant - High-performance, massive-scale Vector Database and Vector Search Engine for the next generation of AI. Also available in the cloud https://cloud.qdrant.io/* [in en]. Accessed 10 September 2025. https://github.com/qdrant/qdrant.

RAG vs Traditional QA - GeeksforGeeks. 2025, July. Accessed 28 July 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 28 July 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 19 July 2025. https://aclanthology.org/2025.iwsds-1.10/.

Release explosion/spacy-models it_core_news_sm-3.8.0. 2024, September. Accessed 10 September 2025. https://github.com/explosion/spacy-models/releases/tag/it_core_news_sm-3.8.0.

Riedl, Mark. 2023. *A Very Gentle Introduction to Large Language Models without the Hype* [in en], May. Accessed 26 July 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 21 July 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 23 July 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 19 July 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 23 July 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 19 June 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 19 July 2025. DOI: 10.48550/arXiv.2404.00657. http://arxiv.org/abs/2404.00657.

Streamlit Documentation v1.47.0. 2025, July. Accessed 19 June 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 3 August 2025. DOI: 10.48550/arXiv.2304.05253. http://arxiv.org/abs/2304.05253.

Team, The SQLite Development. 2025. *sqlite/sqlite.* Original-date: 2019-03-18T12:21:01Z, September. Accessed 10 September 2025. https://github.com/sqlite/sqlite.

Tesseract OCR. 2025. Original-date: 2014-08-12T18:04:59Z, September. Accessed 10 September 2025. https://github.com/tesseract-ocr/tesseract.

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 28 July 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 23 July 2025. DOI: 10.48550/ARXIV.2506.00664. https://arxiv.org/abs/2506.00664.

Topsakal, Oguzhan and Tahir Cetin Akinci. 2023. "Creating Large Language Model Applications Utilizing LangChain: A Primer on Developing LLM Apps Fast" [in en]. *International Conference on Applied Engineering and Natural Sciences* 1, no. 1 (July): 1050–1056. ISSN: 2980-3209, accessed 24 August 2025. DOI: 10.59287/icaens.1127. https://as-proceeding.com/index.php/icaens/article/view/1127.

Text REtrieval Conference (TREC) QA Data. Accessed 26 July 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 27 July 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 23 July 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 19 June 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, Jianguo, Xiaomeng Yi, Rentong Guo, Hai Jin, Peng Xu, Shengjun Li, Xiangyu Wang, Xiangzhou Guo, Chengming Li, Xiaohai Xu et al. 2021. "Milvus: A Purpose-Built Vector Data Management System". In *Proceedings of the 2021 International Conference on Management of Data,* 2614–2627.

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 25 July 2025. DOI: 10.48550/arXiv.2212.03533. http://arxiv.org/abs/2212.03533.

Wang, Liang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder and Furu Wei. 2024. *Multilingual E5 Text Embeddings: A Technical Report.* ArXiv:2402.05672, February. Accessed 10 September 2025. DOI: 10.48550/arXiv.2402.05672. http://arxiv.org/abs/2402.05672.

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 18 July 2025. DOI: 10.48550/arXiv.2407.01219. http://arxiv.org/abs/2407.01219.

Wiki GNA. 2025, August. Accessed 3 August 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 26 July 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 23 July 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 26 July 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 23 July 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 26 July 2025. DOI: 10.48550/arXiv.1906.08237. http://arxiv.org/abs/1906.08237.

Yang, Zhilin, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov and Christopher D. Manning. 2018. "HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering" [in en]. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing,* 2369–2380. Brussels, Belgium: Association for Computational Linguistics. Accessed 26 August 2025. DOI: 10.18653/v1/D18-1259. http://aclweb.org/anthology/D18-1259.

Yoon, Wonjin, Jinhyuk Lee, Donghyeon Kim, Minbyul Jeong and Jaewoo Kang. 2019. *Pre-trained Language Model for Biomedical Question Answering.* Accessed 26 July 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 26 July 2025. DOI: 10.48550/arXiv.2503.19213. http://arxiv.org/abs/2503.19213.

Zaib, Munazza, Wei Emma Zhang, Quan Z. Sheng, Adnan Mahmood and Yang Zhang. 2022. "Conversational question answering: a survey" [in en]. *Knowledge and Information Systems* 64, no. 12 (December): 3151–3195. ISSN: 0219-1377, 0219-3116, accessed 26 July 2025. DOI: 10.1007/s10115-022-01744-y. https://link.springer.com/10.1007/s10115-022-01744-y.

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 23 July 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 27 July 2025. DOI: 10.1145/3404835.3462922. https://dl.acm.org/doi/10.1145/3404835.3462922.