Universitätsmedizin Essen
Institut für KI in der Medizin (IKIM)

Investigating Retrieval-Augmented Generation for LLM Applications in the Medical Domain

Master's Thesis in Data Science

Submitted by
Fahad Deshmukh

At the
Professorship for Complex Multimedia Application Architectures
Institute of Computer Science and Computational Science
University of Potsdam

In cooperation with
IKIM – Institute for Artificial Intelligence in Medicine
Essen, Germany

Supervisors:
Dr. Stephen Tobin
Prof. Dr. med. Dr. rer. nat. Jens Kleesiek

Submission Date: May 09, 2025

Abstract

Clinicians are increasingly inundated with vast amounts of research articles, guidelines, and standard operating procedures, yet require concise and trustworthy answers directly at the point of care [KWT15]. While traditional Document Management Systems (DMS) [Bök15], used by institutions like Universitätsklinikum Essen (UK Essen) [Uni25], provide access to this information, they often fall short due to limitations in semantic understanding (relying on keyword search) and an inability to synthesize information to directly answer complex queries.

This thesis investigates how different configurations of Retrieval-Augmented Generation (RAG) [Lew+21] systems perform within the demanding context of the German medical domain, aiming to address the aforementioned limitations. We systematically implemented and compared four distinct, on-premise, open-source RAG pipelines to assess the impact of various component choices. These pipelines explored combinations of different Large Language Models [Sho+25], multilingual versus German-specialized embedding models, and both pure vector search and hybrid (dense vector + BM25) retrieval strategies.

The performance of these configurations was rigorously evaluated using a domain-specific German question-answering dataset. The evaluation employed a comprehensive suite of automated metrics (including faithfulness and answer accuracy), supplemented by an innovative "Multi-LLM based jury" [Ver+24] for qualitative assessment. Additionally, a small qualitative study benchmarked the best-performing pipeline against the incumbent system, roXtra [Sch16].

Key findings reveal how component selection influences performance in this specific domain: the multilingual embedding model showed a slight advantage over its German-specialized counterpart for document retrieval, while hybrid retrieval strategies markedly improved recall and answer groundedness. The Llama-4-Scout MoE model consistently delivered the highest overall answer quality.

Main contributions of this investigation include:

1. Empirical evidence demonstrating how RAG component choices—LLM, embedding model, and retrieval method—affect performance when applied to German documents.

2. A robust and reproducible evaluation methodology that effectively blends

automated metrics with LLM-based expert-style judgment for nuanced performance assessment within this domain.

The results demonstrate that carefully configured, completely self hosted RAG systems can effectively operate within the German medical domain, offering healthcare professionals and knowledge workers faster and more reliable access to critical knowledge, thereby mitigating the challenges of information overload.

Zusammenfassung

Klinikerinnen und Kliniker sehen sich zunehmend mit einer riesigen Menge an Forschungsartikeln, Leitlinien und Standardarbeitsanweisungen konfrontiert, benötigen jedoch prägnante und vertrauenswürdige Antworten direkt am Behandlungsort [KWT15]. Während traditionelle Dokumentenmanagementsysteme (DMS) [Bök15], wie sie von Institutionen wie dem Universitätsklinikum Essen (UK Essen) [Uni25] genutzt werden, Zugriff auf diese Informationen ermöglichen, stoßen sie aufgrund von Einschränkungen im semantischen Verständnis (basierend auf Stichwortsuche) und der Unfähigkeit, Informationen zu synthetisieren, um komplexe Anfragen direkt zu beantworten, oft an ihre Grenzen.

Diese Abschlussarbeit untersucht, wie unterschiedliche Konfigurationen von Retrieval-Augmented Generation (RAG) [Lew+21] Systemen im anspruchsvollen Kontext des deutschen medizinischen Bereichs abschneiden, mit dem Ziel, die oben genannten Einschränkungen zu adressieren. Wir implementierten und verglichen systematisch vier verschiedene, lokal betriebene Open-Source-RAG-Pipelines, um die Auswirkungen verschiedener Komponentenentscheidungen zu bewerten. Diese Pipelines untersuchten Kombinationen verschiedener Großer Sprachmodelle (Large Language Models, LLMs) [Sho+25], mehrsprachiger versus auf Deutsch spezialisierter Einbettungsmodelle sowie reiner Vektorsuche und hybrider (dichte Vektor- + BM25) Abrufstrategien.

Die Leistung dieser Konfigurationen wurde anhand eines domänenspezifischen deutschen Frage-Antwort-Datensatzes rigoros bewertet. Die Bewertung umfasste eine umfassende Reihe automatisierter Metriken (einschließlich Faktentreue und Antwortgenauigkeit), ergänzt durch eine innovative “Multi-LLM-basierte Jury” [Ver+24] zur qualitativen Bewertung. Zusätzlich wurde in einer kleinen qualitativen Studie die leistungsstärkste Pipeline mit dem bestehenden System roXtra [Sch16] verglichen.

Wichtige Erkenntnisse zeigen, wie die Auswahl der Komponenten die Leistung in diesem spezifischen Bereich beeinflusst: Das mehrsprachige Einbettungsmodell zeigte einen leichten Vorteil gegenüber seinem auf Deutsch spezialisierten Gegenstück bei der Dokumentensuche, während hybride Abrufstrategien die Trefferquote (Recall) und die Fundiertheit der Antworten deutlich verbesserten. Das Llama-4-Scout MoE-Modell lieferte durchweg die höchste Gesamtqualität der Antworten.

Hauptbeiträge dieser Untersuchung sind:

1. Empirische Belege dafür, wie sich die Wahl der RAG-Komponenten – LLM, Einbettungsmodell und Abrufmethode – auf die Leistung bei der Anwendung auf deutsche Dokumente auswirkt.

2. Eine robuste und reproduzierbare Bewertungsmethodik, die automatisierte Metriken effektiv mit LLM-basierter, expertenähnlicher Beurteilung für eine nuancierte Leistungsbewertung in diesem Bereich verbindet.

Die Ergebnisse zeigen, dass sorgfältig konfigurierte, vollständig selbst gehostete RAG-Systeme effektiv im deutschen medizinischen Bereich eingesetzt werden können. Sie bieten medizinischem Fachpersonal und Wissensarbeitern einen schnelleren und zuverlässigeren Zugriff auf kritisches Wissen und mildern so die Herausforderungen der Informationsüberflutung.

Acknowledgments

I would like to express my deepest gratitude to my supervisors, **Dr. Stephen Tobin** and **M. Sc Sameh Khattab**, whose insightful guidance, constructive feedback, and unwavering support were indispensable to this thesis.

My sincere thanks also go to **M. Sc Fin Hendrik Bhansen** and **Prof. Dr. Jens Kleesiek** from IKIM for their valuable input and support, which contributed significantly to this research environment.

My appreciation extends to the *Professur Komplexe Multimediale Anwendungsarchitekturen* at the *University of Potsdam* for providing essential resources and a stimulating research environment.

Special words of appreciation go to my dear brother and friend, **Kashif**, for his constant support. encouragement throughout this journey.

Contents

Abstract iii

Zusammenfassung v

Acknowledgments vii

Contents ix

1 Introduction 1
- 1.1 Background and Motivation . . . 1
- 1.2 Problem Statement . . . 2
- 1.3 Research Questions . . . 3
- 1.4 Thesis Structure . . . 5

2 Background and Related Work 7
- 2.1 The Challenge of Information Retrieval . . . 7
- 2.2 Evolution of Language Models and Conversational AI . . . 9
- 2.3 Domain-Specific Applications of Large Language Models . . . 12
  - 2.3.1 Hallucination and Factual Reliability . . . 12
  - 2.3.2 Knowledge Currency and Technical Constraints . . . 13
  - 2.3.3 Operational and Security Concerns . . . 13
- 2.4 Limitations of roXtra for Information Retrieval . . . 14
  - 2.4.1 roXtra's Emphasis on Document Control over Deep Information Access . . . 15
  - 2.4.2 Keyword-Based Search and the Quantifiable Limitations of Exact Matching . . . 15
  - 2.4.3 Motivating the Transition to Retrieval-Augmented Generation (RAG) . . . 16
- 2.5 Retrieval-Augmented Generation (RAG) as a Solution . . . 16
- 2.6 Evaluation Methodologies for RAG Systems . . . 17
- 2.7 Synthesis and Knowledge Gaps . . . 18

3 RAG: Architecture And Components **23**
3.1 Enhancing Large Language Models with External Knowledge . . 23
3.1.1 Addressing Inherent LLM Limitations through RAG . . . 23
3.1.2 Key Advantages of the RAG Approach for Knowledge-Intensive Tasks . . . 24
3.1.3 The RAG Workflow: From Data Ingestion to Response Generation . . . 25
3.1.4 Comparing Knowledge Enhancement Strategies: RAG vs. Fine-Tuning . . . 26
3.2 Data Preparation: Building the Foundation for Retrieval . . . 27
3.2.1 Ingestion: Acquiring and Consolidating Knowledge . . . 28
3.2.2 Parsing: Extracting and Structuring Content . . . 28
3.2.3 Chunking: Segmenting Data for Effective Processing and Retrieval . . . 29
3.3 Indexing: Structuring Knowledge for Efficient Semantic Access . . 31
3.3.1 Embedding Generation: Translating Text into Semantic Vectors . . . 31
3.3.2 Vector Stores: Databases adapted for Embedding Management and Search . . . 32
3.3.3 Indexing Algorithms: Enabling Efficient Similarity Search 33
3.4 Retrieval: Locating Relevant Information for the Query . . . 34
3.4.1 The Core Retrieval Mechanism: Semantic Similarity Search 35
3.4.2 Exploring Diverse Retrieval Strategies: Beyond Basic Semantic Search . . . 36
3.5 Generation: Synthesizing Knowledge into Coherent Responses . . 37
3.5.1 The Generator LLM: Engine for Synthesis and Formulation 37
3.5.2 Crafting Effective Prompts for Grounded Generation . . . 38
3.5.3 Generating the Final, Grounded Response . . . 39
3.6 Summary: The Synergy of RAG Components for Domain Information Analysis . . . 40

4 Evaluation Framework and Methodology **43**
4.1 Introduction to RAG Evaluation . . . 43
4.2 Evaluation Metrics . . . 43
4.2.1 Retrieval-Specific Metrics . . . 44
4.2.2 Generation-Specific Metrics . . . 45
4.2.3 End-to-End Metrics . . . 46
4.2.4 Metric Implementation and Rationale . . . 46
4.2.5 Panel of LLMs as Automated Judges . . . 46

4.3 Evaluation Dataset . . . 49
4.3.1 Dataset Construction . . . 49
4.3.2 Question Categories and Rationale . . . 50
4.3.3 Corpus and Testset Characteristics . . . 51
4.4 Qualitative Evaluation Methodology . . . 52
4.5 Experimental Design . . . 52
4.6 Summary . . . 52

5 System Design and Implementation 55
5.1 Introduction . . . 55
5.2 System Architecture Overview . . . 55
5.3 Compute Layer Implementation . . . 57
5.3.1 Text Embedding Implementation . . . 57
5.3.2 LLM Inference Implementation . . . 58
5.4 Document Cloud Implementation . . . 58
5.5 Ingestion Container Implementation . . . 58
5.5.1 Document Processing Pipeline . . . 59
5.5.2 Vector Database Implementation . . . 59
5.5.3 Citation Manager and Answer Grounding . . . 59
5.5.4 Query Engine Implementation . . . 62
5.6 Front-end Implementation . . . 63
5.7 Experimental RAG Configurations . . . 63
5.7.1 Configuration Analysis . . . 65
5.7.2 Generator Model Comparison . . . 65
5.8 Summary . . . 66

6 Results 69
6.1 Overview of RAG Architecture Performance . . . 69
6.2 Research Question 1: Impact of Embedding Models on domain specific documents in German . . . 72
6.3 Research Question 2: Effectiveness of Hybrid Retrieval Strategies 73
6.4 Research Question 3: MoE vs. Dense Generator Models . . . 74
6.5 Research Question 4: Optimal RAG Configuration for German Hospital Documentation . . . 75
6.6 Research Question 5: Qualitative Comparison with Keyword-Based Search . . . 76
6.6.1 Complex Medical Policy Questions . . . 76
6.6.2 Clinical Questions Requiring Semantic Understanding . . . 77
6.6.3 Factual Knowledge Queries . . . 78

6.7 Summary of Key Findings 79

7 Conclusion and Future Work **81**
7.1 Summary of Research and Key Findings 81
7.2 Limitations of the Study 82
7.3 Discussion and Implications 82
7.3.1 Implications for Practice 82
7.3.2 Future Research Directions 83
7.4 Concluding Remarks 84

Bibliography **i**

Declaration of Authorship **xxv**

List of Figures

3.1 Typical RAG Workflow, showcasing the interaction between retriever and generator components. [FAT24] 24

5.1 On-premises system architecture illustrating the core containerized modules, the data-ingestion pipeline, and the user-interaction flow 56
5.2 Four RAG configurations implemented and evaluated in this research. 64

6.1 Metrics Comparison Across RAG Models (Radar Chart) 70
6.2 Average Jury Score by RAG Configuration 71
6.3 Jury Quality Assessment Distribution 71
6.4 Example Keyword Search Results from Roxtra 77
6.5 Example RAG Chatbot Response Grounded in Knowledge base . . 77

List of Tables

2.1 Overview of RAG Configurations Investigated in this Thesis . . . 21

3.1 Comparison of RAG and Fine-Tuning for Knowledge Enhancement [Bal+24; Par+24; SKH24] . . . 27

5.1 Component specifications for each configuration. . . . 63
5.2 Key characteristics of dense and MoE generator models used. . . . 66

6.1 Detailed Metric Comparison Across Architectures (%) . . . 70

1 Introduction

1.1 Background and Motivation

Knowledge-intensive professions, particularly medicine, face unprecedented challenges in information management[KWT15]. The exponential growth of biomedical literature and clinical data[Din16] has precipitated what researchers term "filter failure," where conventional search and review methods struggle to surface relevant evidence efficiently. Healthcare professionals increasingly expend valuable time querying databases and navigating document repositories[Pin+21], yet critical information often remains inaccessible or difficult to locate[JLL24]. Research highlights a pronounced preference among clinicians for curated, synthesized resources over raw literature searches for point-of-care queries[Jad+22], underscoring the pressing need for more intelligent, context-aware retrieval tools integrated into daily medical practice.

Within German(language) hospital environments specifically, this challenge is compounded by the linguistic complexity of medical documentation. German medical language frequently employs complex compound words (Komposita)[Els09], exhibits significant terminological variation, and possesses grammatical structures that pose considerable challenges for standard information retrieval techniques[Lit+25]. Existing keyword-based document management systems, exemplified by platforms like roXtra[Sch16], primarily operate on lexical matching principles, struggling to capture semantic relationships or interpret complex queries that rely on contextual understanding, synonyms, or domain-specific terminology [KTS16].

Concurrently, the field of artificial intelligence has witnessed remarkable advancements in natural language processing[Wen+19]. Large Language Models (LLMs)[Cha+24] have demonstrated unprecedented capabilities in understanding and generating human language[GMT24] (see Section 2.2 for discussion on the use of cognitive terms in describing LLM capabilities), including within specialized domains like medicine. Models like GPT-4[San23] and Med-PaLM 2 [Par+] have achieved impressive performance on standardized medical licensing examinations. Med-PaLM 2 achieved an 86.5% accuracy on the MedQA benchmark, surpassing the performance of its predecessor and reaching expert-level accuracy on USMLE-style questions[Sin+25]. Similarly, GPT-4 demonstrated significant improvements over

its predecessors on the USMLE, achieving scores well above the passing threshold[Nor+23]. These developments suggest significant potential for LLM-powered conversational systems to alleviate cognitive load by synthesizing information in response to natural language queries[SBS24].

However, deploying these powerful models in information-critical fields like healthcare necessitates addressing paramount concerns regarding factual accuracy. LLMs trained on broad internet corpora frequently generate plausible yet incorrect statements ("hallucinations") and may fabricate sources [Hua+25; Mug+23], posing significant risks in high-stakes medical scenarios.

Retrieval-Augmented Generation (RAG) has emerged as a compelling architectural paradigm [Lew+21] that addresses these challenges by synergizing an LLM's generative capabilities with a dedicated retrieval module that fetches relevant context from a controlled external knowledge base. Originally proposed by Lewis et al. (2020) [Lew+21], RAG systems ground LLM outputs in verifiable evidence, enhancing factual accuracy [LYZ24]. This approach shows particular promise for specialized domains like healthcare, where information reliability is paramount [Sir+23].

The confluence of these factors—information overload in medicine, the linguistic complexity of German medical documentation, the advanced capabilities of modern LLMs, and the architectural advantages of RAG—motivates this research into optimizing information retrieval for German hospital environments. By systematically investigating the impact of various RAG components and configurations, the aim of this thesis is to develop a framework that enhances information access within these specific operational contexts.

1.2 Problem Statement

The management of medical information in German hospital environments presents a multifaceted challenge characterized by three interconnected dimensions: information volume, linguistic complexity, and retrieval limitations [PBC12].

The volume of medical information has expanded at an unprecedented rate, with PubMed alone indexing over 34 million biomedical citations and an estimated 4,000-6,000 new articles added daily [Din16]. Within hospital settings , this translates to massive collections of clinical guidelines, protocols, regulatory documents, and institutional knowledge bases. The sheer scale of this information landscape renders traditional manual search approaches increasingly impractical [Kor22], as healthcare professionals cannot efficiently navigate this vast corpus to locate specific information needed for clinical decision-making.

Existing document management systems deployed in such institutions, exemplified by roXtra [Sch16], fundamentally operate on keyword-based principles. As documented by empirical research [KTS16; LS03], these systems exhibit quantifiable limitations: lower recall and precision compared to semantic approaches, context insensitivity that fails to recognize related concepts, inadequate handling of German compound words and grammatical variations, and scalability constraints that become increasingly problematic as document collections grow. Consequently, medical professionals often must review numerous irrelevant documents before locating needed information [Slo+24], resulting in significant inefficiencies and potential information gaps [TSB09].

The fundamental problem addressed by this thesis is thus: How can we develop a retrieval system that significantly improves information access for medical knowledge workers in such crucial environments, enhancing semantic understanding and response generation for effective deployment within hospital infrastructure? More specifically, which combination of embedding models, retrieval strategies, and generation models yields optimal performance for this specific domain?

While Retrieval-Augmented Generation offers a promising architectural approach, the optimal configuration for this specific application—retrieving information from German medical documents—remains an open question [Wu+24d]. Performance is contingent upon the careful selection and integration of core components: the embedding model for semantic understanding, the retrieval strategy for identifying relevant context, and the LLM for synthesizing responses [Wan+24c]. Each component presents multiple design choices, the impact of which has not been systematically evaluated within this specialized domain and deployment context.

Addressing this problem has significant implications for clinical practice, potentially reducing information retrieval time, improving decision support quality, and enhancing operational efficiency. It also contributes to the broader scientific understanding of domain-specialized RAG systems and their component-level optimization [Sir+23].

1.3 Research Questions

To guide this investigation into optimizing Retrieval-Augmented Generation for German medical document retrieval, five specific research questions have been formulated:

RQ1 (Embedding Model Impact): How significantly does the choice between a general multilingual embedding model and a German-language optimized bilin-

gual model influence retrieval effectiveness and generation quality for hospital documents in German?

> This question examines whether language-specialized embedding models offer measurable advantages over general multilingual models when processing German medical text. It addresses the fundamental trade-off between broad language understanding and domain-specific optimization in semantic representation.

RQ2 (Retrieval Strategy Impact): To what extent does incorporating hybrid search (dense vectors + BM25 sparse vectors[WC11]) improve retrieval effectiveness and enhance the reliability of generated outputs compared to standard vector search alone?

> This question investigates the potential complementarity between neural semantic search and traditional lexical matching approaches. It explores whether combining these methods provides more robust retrieval performance across diverse query types encountered in medical information seeking.

RQ3 (MoE vs. Dense): Does employing an LLM that utilizes a Mixture-of-Experts (MoE)[Art+21] architecture yield any improvements in key generation metrics relative to employing a general instruction-following dense LLM?

> This question examines architectural choices for the generation component of the RAG pipeline, comparing traditional dense models with the emerging Mixture-of-Experts approach. It addresses whether the specialized pathways in MoE models offer advantages for synthesizing responses to medical queries.

RQ4 (Overall Performance Trade-offs): Considering retrieval effectiveness, generation quality, and operational efficiency, which RAG configuration demonstrates the most advantageous overall performance profile for retrieving information from German hospital documentation?

> This question synthesizes findings from RQ1-RQ3 to identify the optimal architecture for practical deployment, acknowledging that real-world systems must balance multiple performance dimensions rather than optimizing for a single metric.

RQ5 (Comparison with Existing Systems): In what specific functional aspects and for which types of queries does the optimal RAG configuration exhibit qualitative superiority over the incumbent keyword-based search system (`roXtra`)?

> This question provides practical context by benchmarking the optimized RAG approach against the current industry standard. It identifies specific use cases and query types where RAG offers the most significant improvements, providing guidance for potential integration or replacement strategies.

Together, these research questions form a comprehensive framework for evaluating and optimizing RAG systems specifically for German medical documentation. They progress logically from component-level analysis (RQ1-RQ3) to system-level synthesis (RQ4) and practical application (RQ5), providing both theoretical insights and applied knowledge.

1.4 Thesis Structure

This thesis is organized into seven chapters that progressively develop the investigation of Retrieval-Augmented Generation for German medical information retrieval:

Chapter 2: Background and Related Work establishes the theoretical and empirical foundations for this research. It examines the evolution of information retrieval, the rise of Large Language Models and their inherent limitations, the emergence of Retrieval-Augmented Generation as an architectural solution, and the components that influence RAG performance. The chapter sets the theoretical background required for the development of this idea and the project.

Chapter 3: RAG: Architecture And Components presents a comprehensive examination of Retrieval-Augmented Generation as a solution for enhancing LLMs with external knowledge. It systematically explores the four core components of RAG systems: data preparation (ingestion, parsing, chunking), indexing (embedding generation, vector stores), retrieval (semantic search, hybrid strategies), and generation (synthesizing coherent responses). The chapter also contrasts RAG with fine-tuning approaches and highlights the specific advantages of RAG for knowledge-intensive domains like healthcare information retrieval, demonstrating why this architecture is particularly suitable for processing German medical documents.

Chapter 4: Evaluation Framework and Methodology presents the comprehensive evaluation approach designed to assess the performance of different RAG configurations. It details the metrics used to evaluate both retrieval quality (Context Relevance, Context Recall, Context Precision) and generation quality (Faithfulness, Response Groundedness, Answer Accuracy). The chapter describes the construction of a specialized evaluation dataset comprising 90 question-answer pairs across

four question categories, the implementation of an LLM Panel/Jury methodology, and the approach to qualitative comparison with the incumbent roXtra system.

Chapter 5: System Design and Implementation details the architecture and components of the RAG system developed for this research. It describes the four primary containers that form the comprehensive architecture: the Compute Layer providing GPU-accelerated inference, the Document Cloud for document storage, the Ingestion Container housing the document processing pipeline and vector storage, and the Front-end Container delivering the user interface. The chapter explains the rationale for selecting specific components and discusses alternative architectures, particularly focusing on the Mixture-of-Experts approach.

Chapter 6: Results and Analysis presents the empirical findings from the evaluation of four distinct RAG architectures. It systematically addresses each research question, analyzing the impact of embedding models (RQ1), retrieval strategies (RQ2), and LLM architectures (RQ3). The chapter identifies the optimal RAG configuration based on comprehensive performance assessment (RQ4) and provides a qualitative comparison with the existing keyword-based search system (RQ5). The analysis examines both quantitative metrics and qualitative aspects of system performance across different question types.

Chapter 7: Conclusion and Future Work synthesizes the research findings, discusses their implications for information retrieval, acknowledges limitations of the current study, and outlines promising directions for future research. It connects the empirical results to broader themes in AI-assisted information access and system deployment within hospital environments, highlighting both theoretical contributions and practical applications.

Throughout these chapters, the thesis maintains a focus on the requirement of enhanced information access within the medical domain. The structure progresses logically from problem definition through theoretical foundations, system development, evaluation methodology, empirical results, and finally to synthesis and future directions.

2 Background and Related Work

This chapter establishes the context for the thesis by reviewing the evolution of information retrieval with time, the rise of Large Language Models (LLMs) and their inherent limitations, the emergence of Retrieval-Augmented Generation (RAG) as a promising architectural solution, the critical components influencing RAG performance, and the methodologies for evaluating such systems. The review aims to demonstrate the necessity and novelty of the research undertaken, highlighting the specific knowledge gaps addressed by this work.

2.1 The Challenge of Information Retrieval

Knowledge-intensive fields are increasingly characterized by an overwhelming volume of information [Din16]. The exponential growth in published literature, research data, domain-specific records, and practice guidelines presents a significant challenge for professionals seeking timely, relevant evidence [KWT15]. This phenomenon, often termed "filter failure," signifies the breakdown of conventional search and review methods in effectively surfacing critical information from the deluge of available data [Dae+20]. Quantitative analyses have demonstrated that healthcare professionals spend an average of around 1 hour per question searching for information, yet approximately 30% of information needs remain unaddressed due to difficulties in efficiently retrieving relevant data [Pin+21; Veg+19]. This inefficiency potentially impacts decision-making accuracy and outcomes, with studies indicating information retrieval gaps contribute to a considerable percentage of errors and suboptimal decisions [Ste+22]. The observed preference among clinicians for curated, synthesized resources over direct searches of raw literature for point-of-care queries underscores this challenge, pointing towards a pressing need for more intelligent, context-aware retrieval tools integrated into daily medical workflows [SRL24]. This preference suggests that the requirement extends beyond merely locating potentially relevant documents; it encompasses the need for reliable synthesis and interpretation, features often associated with trusted, pre-processed information sources. The reliance on such curated materials implies a demand for systems that can not only find but also present information in a digestible

and trustworthy manner, directly motivating the exploration of generative models capable of synthesis, provided their outputs can be rigorously verified.

Within specific healthcare environments, such as public hospitals, the limitations of existing information management systems are particularly apparent. Traditional keyword-based document management systems, exemplified by platforms like roXtra, often form the backbone of institutional knowledge access [Kat14]. However, these systems primarily operate on lexical matching, struggling to capture the semantic relationships between concepts or to interpret complex queries that rely on contextual understanding, synonyms, or nuanced domain-specific terminology [KTS16]. This inability to grasp meaning beyond literal keyword occurrences significantly hinders effective information retrieval. The persistence of such systems, despite their acknowledged limitations, points towards considerable practical barriers to adopting newer technologies in clinical settings, which may include factors like integration complexity, cost, the need for extensive validation, or simply the lack of demonstrably superior and trustworthy alternatives tailored to the specific institutional context. This thesis directly confronts this last barrier by providing empirical evidence for a RAG-based alternative.

The difficulties associated with information retrieval are further compounded in the German medical context due to specific linguistic characteristics. German medical language frequently employs complex compound words (Komposita)[Els09], exhibits significant terminological variation, and possesses grammatical structures that pose challenges for standard natural language processing (NLP) techniques [Név+18]. Domain-specific jargon and abbreviations add another layer of complexity [FDG17]. Consequently, keyword search often fails to retrieve relevant documents if the query uses synonyms or different phrasings than the text, while even general-purpose semantic search models developed primarily for English may underperform without specific adaptation [Pec+14]. This linguistic specificity necessitates careful consideration of language-optimized components within any advanced retrieval system designed for this environment, suggesting that solutions effective in English may require substantial modification or component substitution to achieve comparable performance in German medical settings. This directly motivates the investigation into language-specific embedding models (RQ1).

Synthesizing these observations—the overwhelming data volume, the inefficiency of traditional search methods, the limitations of keyword-based systems in handling semantic complexity, and the added linguistic challenges in German medical documentation—underscores the urgent need for advanced information retrieval solutions. These solutions must be intelligent, capable of understanding context and semantics; efficient, reducing the time burden on clinicians; and context-aware, leveraging domain-specific knowledge. This confluence of requirements sets the

stage for exploring how recent advancements in large language models and architectures like RAG can be harnessed to meet these multifaceted needs.

2.2 Evolution of Language Models and Conversational AI

The field of medical question-answering and conversational AI has undergone a profound transformation, moving from rudimentary systems to highly sophisticated neural architectures. Early iterations of medical chatbots were largely constrained by their reliance on predefined rules and keyword-matching algorithms [Zem19]. These systems lacked the flexibility and capacity for nuanced contextual processing necessary to handle the complexity inherent in medical discourse, limiting their practical utility. The contemporary landscape has been reshaped by foundational breakthroughs in NLP, particularly the development of neural networks, such as the transformer architecture, capable of learning intricate patterns and representations from vast amounts of text data [Vas+17].

As we explore these powerful architectures and the Large Language Models (LLMs) built upon them, it is crucial to address the language frequently used to describe their abilities – terms like "understanding," "knowledge," and "meaning." While LLMs demonstrate impressive performance by generating coherent text and answering complex questions, cognitive science emphasizes a fundamental distinction between their operational mechanisms and human cognition [Lam24]. An LLM's proficiency stems from learning intricate statistical correlations within massive text datasets, typically optimizing for tasks like next-word prediction [BS23]. Consequently, when an LLM appears to "understand," it is effectively leveraging distributional semantics – patterns of word co-occurrence – to generate statistically probable outputs relevant to the input [Kos+22]. This sophisticated pattern-matching allows them to mimic comprehension but lacks the grounding in real-world sensory, motor, and social experience that underpins human understanding [Lam24]. Human cognition involves robust causal reasoning [Che+24b], abstraction from limited data [FH24], common sense, and rich, multimodal conceptual representations [Wu+24c] – capabilities where current LLMs show significant limitations, as evidenced by empirical studies highlighting weaknesses in areas like robust reasoning [Als+24], handling linguistic novelty (e.g., 'leetspeak') [FP24], and metacognitive calibration.

Therefore, throughout this thesis, the use of cognitive terminology when discussing LLMs must be interpreted strictly in an operational sense. Terms like "understanding context," "semantic understanding," or "grasping meaning" refer to

the model's functional capacity to process linguistic patterns effectively, manipulate learned representations, and generate outputs appropriate to specific tasks based on its training data. This terminology denotes observed functional competence on language tasks, not an assertion of sentience, subjective experience, or genuine comprehension analogous to humans [LS21]. Maintaining this clear distinction is essential for accurately assessing the true capabilities and inherent limitations of LLMs [WKR25] and for guiding their responsible application, particularly in high-stakes environments such as medicine.

A pivotal innovation underpinning modern language models is the attention mechanism, first introduced by [BCB14] for neural machine translation. Originally designed to overcome the information bottleneck in recurrent-neural-network (RNN) encoders, attention lets the decoder dynamically focus on different parts of the input sequence, selectively weighting their relevance at each output step. This relieved the encoder from compressing all information into a single vector and substantially improved translation quality.

The next major leap came with the Transformer architecture, proposed in the seminal paper "Attention Is All You Need" [VSP17]. By discarding RNNs and relying solely on multi-head self-attention, the Transformer can process all tokens in parallel, dramatically improving computational efficiency on modern hardware. Its ability to model long-range dependencies, together with hardware advances [SK10], laid the foundation for today's large-scale language models.

Conceptually, the attention mechanism functions akin to an information retrieval system embedded within the neural network. It uses three key components—Query (Q), Key (K), and Value (V)—to determine which parts of the input are most relevant to the current processing step. The Query represents the current focus or information need, the Keys act as labels or identifiers for different elements in the input sequence, and the Values contain the actual information content associated with those elements. The mechanism computes similarity scores between the Query and all Keys, transforms these scores into attention weights (typically via a softmax function), and then calculates a weighted sum of the Value vectors based on these weights. This produces an output vector that incorporates contextually relevant information from the input sequence. The scaled dot-product attention, a common implementation, is mathematically defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where Q, K, and V represent matrices of queries, keys, and values, respectively. d_k is the dimension of the key vectors, and the scaling factor $1/\sqrt{d_k}$ is applied

to prevent excessively large dot products, which can lead to vanishing gradients during training, particularly with high-dimensional vectors. This mechanism allows models to selectively focus on relevant information, mirroring aspects of human cognitive attention [VSP17].

Building upon the Transformer foundation, research rapidly progressed towards developing larger and more capable pretrained language models (LLMs). Models like BERT (Bidirectional Encoder Representations from Transformers) introduced bidirectional context understanding, significantly improving performance on various NLP tasks by considering both left and right context simultaneously [Cla+19]. Subsequent models, such as the Generative Pre-trained Transformer (GPT) series, demonstrated remarkable few-shot learning capabilities, generating coherent and contextually relevant text across diverse domains, including medicine, often with minimal task-specific examples [FC20].

The drive towards larger models was fueled by empirical findings establishing scaling laws. Research [Kap+20] demonstrated that model performance on language tasks improves predictably with systematic increases in model size (number of parameters), dataset size, and computational resources used for training. Further studies [Hof+22] refined these laws, identifying potentially optimal ratios between model parameters and training data volume to maximize performance for a given computational budget. These insights guided the development of models with tens or hundreds of billions of parameters throughout 2022-2023. Such large-scale models began to show "emergent abilities", which can be defined as capabilities not explicitly trained for and not observed in smaller models: in complex tasks involving reasoning, arithmetic, and nuanced instruction following [Wei+22]. Specifically, models that exceed approximately 50 billion parameters demonstrated step-change improvements in reasoning tasks, with error rates decreasing by 30-40% compared to their smaller counterparts [Kap+20].

This phenomenon of emergent abilities directly informs RQ3, which compares dense architecture models like Llama-3.3-70B-Instruct with Mixture-of-Experts (MoE) architectures like Llama-4-Scout. While both approaches leverage scale, they do so differently: dense models scale through sheer parameter count, whereas MoE architectures distribute computation across specialized "expert" neural networks that are selectively activated. Each approach represents a different solution to the scaling challenge, potentially affecting how well the model can synthesize information from retrieved medical contexts. Additionally, these architectural considerations are crucial to RQ4's goal of identifying optimal RAG configurations, as the emergent reasoning capabilities of different model architectures must be balanced against other factors like retrieval quality and operational efficiency when deploying in real-world hospital environments.

Compared to earlier RNN architectures, Transformers offer significant advantages. RNNs process sequences token by token, inherently limiting parallelization [Kar+19]. This sequential nature also makes them susceptible to vanishing and exploding gradient problems, hindering their ability to capture long-range dependencies effectively [She20]. Transformers, by processing all tokens in parallel and using self-attention to directly model relationships between any two tokens in the sequence regardless of distance, overcome these limitations, enabling more efficient training and superior modeling of long-term context [Kar+19].

These technological advancements provide the foundation for addressing the information retrieval challenges outlined in Section 2.1, particularly through their application to the medical domain, which we explore next.

2.3 Domain-Specific Applications of Large Language Models

The demonstrated capabilities of LLMs have generated substantial interest within the medical community [Nor+23]. Their potential applications span a wide range, including summarizing clinical notes, answering medical questions, assisting with differential diagnoses, and potentially augmenting clinical decision support systems [Sho+25]. Advanced models like GPT-4 and the medically-tuned Med-PaLM 2[Par+] have shown impressive performance on standardized medical licensing examinations [Kun+23], achieving scores comparable to or even exceeding those of human physicians in certain assessments [Kun+23]. Med-PaLM 2, for instance, reportedly surpassed an 80% accuracy threshold on questions styled after the US Medical Licensing Exam (USMLE), marking a significant milestone.

However, despite these promising benchmarks, the direct application of general-purpose LLMs in clinical practice faces significant hurdles due to several inherent limitations. The most critical concerns can be categorized as follows:

2.3.1 Hallucination and Factual Reliability

A primary concern is the phenomenon of "hallucination," where LLMs generate outputs that are fluent, coherent, and seemingly authoritative, yet contain factual inaccuracies or fabricate information entirely [Ji+22]. This tendency arises partly from the models' training objective, which typically involves predicting the next most probable token in a sequence rather than explicitly verifying factual correctness [Sab23]. Quantitative analyses have documented hallucination rates ranging from 12% to 37% in medical question-answering tasks, with higher rates occurring

in specialized subdomains [Raw+23]. This propensity for generating misinformation creates significant barriers to clinical adoption, where even occasional errors could lead to harmful outcomes [Has25].

2.3.2 Knowledge Currency and Technical Constraints

Beyond factual reliability, LLMs face additional technical limitations that particularly affect their suitability for medical applications:

1. **Static Knowledge**: Most LLMs are trained on datasets up to a specific cutoff date and lack inherent mechanisms to access real-time information, recent research findings, or updated clinical guidelines post-training [Che+24c]. In medicine, where standards of care evolve rapidly, this static knowledge representation can lead to outdated or suboptimal recommendations [Sin+22].

2. **Bias and Fairness**: LLMs can inherit and potentially amplify biases present in their training data [Yeh+23]. Research has identified instances of demographic biases in medical language models, leading to disparities in diagnostic suggestions or treatment recommendations across different patient groups, with error-rate differences exceeding 20% between demographic groups in some scenarios.

3. **Lack of Traceability**: Standard LLM outputs typically lack clear attribution or explanation of their reasoning process, making verification difficult [Bar+24]. This opacity creates fundamental trust issues in clinical environments where source verification is essential for decision-making.

2.3.3 Operational and Security Concerns

A final, crucial set of limitations concerns practical deployment considerations:

1. **Data Security**: Many state-of-the-art LLMs are accessible primarily through cloud-based APIs[Ken22] provided by commercial entities like OpenAI [Ope20]. Transmitting institutional data or internal information (even if anonymized) to these external services creates significant operational risks and may conflict with institutional policies regarding data control and security [Bru24].

2. **Integration Barriers**: The implementation of cloud-based LLMs into existing clinical workflows involves substantial technical and bureaucratic challenges [Kon24], including integration with electronic health records,

compliance with regulatory frameworks, and alignment with established clinical processes [Fer+25].

These limitations collectively create a paradoxical situation: while LLMs demonstrate impressive performance on standardized medical knowledge tests, their practical utility in real-world clinical settings remains constrained without additional safeguards and architectural adaptations. This disconnect between benchmark performance and clinical trustworthiness highlights the need for approaches that can leverage LLMs' generative capabilities while mitigating their inherent limitations [Geb+24].

Researchers have explored domain adaptation as one potential mitigation strategy, fine-tuning models specifically on biomedical literature and data. Models like BioGPT[Luo+22] and PubMedBERT [HTZ21] exemplify this approach, showing improvements of 7–12% on specialized biomedical NLP tasks compared to general-purpose models [CJD24]. However, domain adaptation alone primarily addresses domain relevance rather than the fundamental issues of hallucination, static knowledge, or lack of traceability. It represents a strategy focused on internalizing domain knowledge within the model's parameters during training, which differs fundamentally from Retrieval-Augmented Generation (RAG), which externalizes knowledge access at inference time.

The limitations identified in this section directly motivate our investigation into RAG as a potential solution. However, before exploring RAG, we must first understand the specific limitations of current information-retrieval systems used in German medical settings, particularly roXtra, to establish a clear baseline for comparison.

2.4 Limitations of roXtra for Information Retrieval

roXtra, developed by Rossmanith GmbH [Rox25], is primarily a document control and workflow management software intended to streamline document handling in certification-driven environments [Sch16]. It provides a suite of features such as customizable workflows, revision management, archiving, and various search functionalities that collectively aim to simplify the preparation and maintenance of quality-related documentation. While these capabilities are well-suited for procedural and compliance-focused tasks, roXtra's document-retrieval mechanism reveals significant shortcomings when applied to large-scale, semantically rich domains like German medical literature.

2.4.1 roXtra's Emphasis on Document Control over Deep Information Access

The core design of roXtra is centered on tracking, auditing, and managing the lifecycle of documents rather than performing advanced information retrieval. Its workflow tools (e.g., assigning reviewers, logging revisions, issuing read-confirmation tasks) efficiently handle version control and compliance requirements but do not inherently address the complexity of extracting granular knowledge from voluminous professional texts. Consequently, while roXtra users can locate documents related to a specific process or standard, they often still need to sift through lengthy files to find the precise information they seek [Slo+24]. This manual step becomes increasingly problematic for any knowledge workers who must access timely, accurate details in a field where minor oversights can have significant repercussions.

2.4.2 Keyword-Based Search and the Quantifiable Limitations of Exact Matching

roXtra's search capabilities rely heavily on full-text and metadata-based approaches, which fundamentally operate on keyword-matching principles. Although this can be effective for smaller corpora or straightforward queries, empirical research [LS03] demonstrates several quantifiable limitations of keyword-based retrieval in complex domains:

1. **Lower Recall Performance**: Comparative analyses demonstrate that keyword-based systems retrieve approximately 25–40% fewer relevant documents than semantic search approaches in specialized medical literature [KTS16]. This measurement gap becomes particularly pronounced with complex medical queries, where recall rates can fall significantly.

2. **Semantic Blindness**: These systems operate by matching exact keywords without grasping contextual relationships or conceptual equivalence. Experimental studies [TSB09]show that keyword systems miss up to 45% of relevant documents when queries use synonyms or alternative phrasings for medical concepts, compared with 12–18% for semantic approaches [Des+21].

3. **Inefficiency at Scale**: While keyword search may rely on simpler indexing methods, its lower precision can lead to cognitive inefficiencies. Users often spend additional time reviewing less relevant results compared to those returned by semantically ranked systems [Hri+23].

These quantified limitations directly connect to our research questions, particularly RQ5, which seeks to identify specific functional aspects and query types where an optimized RAG configuration might demonstrate qualitative superiority over roXtra's keyword-based approach.

2.4.3 Motivating the Transition to Retrieval-Augmented Generation (RAG)

The documented limitations of roXtra-like keyword-based systems establish a clear need for more advanced approaches to medical information retrieval. The next section introduces Retrieval-Augmented Generation (RAG) as a promising solution that addresses these limitations through the integration of semantic understanding and generative capabilities.

2.5 Retrieval-Augmented Generation (RAG) as a Solution

Retrieval-Augmented Generation (RAG) presents a promising architectural solution to address the inherent limitations of standalone Large Language Models (LLMs) [Lew+21], particularly in high-stakes domains like medicine [PUS23]. By integrating an external information retrieval component with a generative LLM, RAG aims to significantly enhance factual accuracy and mitigate hallucinations by grounding generated responses in specific, retrieved evidence from a controlled knowledge corpus [LYZ24]. This architecture directly tackles key LLM weaknesses: it improves traceability by linking answers to sources, overcomes the static knowledge problem by accessing up-to-date external data dynamically, facilitates specialization by providing domain-specific context on demand, and supports secure, on-premises deployment crucial for maintaining data control in settings like hospitals [LYZ24]. RAG effectively externalizes knowledge management, allowing the LLM to focus on synthesis based on reliable, current information. For a detailed technical discussion of RAG architecture and its implementation, please refer to Chapter 3.

The core insight behind RAG is the separation of information retrieval from information synthesis. This architectural division reflects the idea that while LLMs are strong at understanding and generating natural language, they may not be best suited for recalling specific facts or detailed domain knowledge stored within their parameters [LYZ24]. Retrieval-augmented approaches aim to address this by incorporating external sources of information to enhance the relevance and factual

grounding of responses [Yan+25]. This can be especially valuable in specialized domains like medicine, where precision and accuracy are critical .

Current RAG implementations extend beyond simple document retrieval to incorporate sophisticated techniques like hybrid search (combining dense and sparse retrieval methods), multi-stage retrieval, query rewriting, and context distillation [Jeo23]. These advancements aim to maximize the quality and relevance of the context provided to the generator LLM, thereby optimizing the final response quality. However, achieving optimal RAG performance requires careful consideration and tuning of multiple components, from embedding models and retrieval strategies to generator model selection and prompt engineering [Gao+24].

The fundamental design choices within a RAG system directly connect to our research questions: Which embedding models best capture the semantic nuances of German medical text (RQ1)? How do different retrieval strategies influence the quality and relevance of retrieved context (RQ2)? Does the architecture of the generator LLM significantly impact its ability to synthesize accurate responses from retrieved evidence (RQ3)? And ultimately, what combination of these components yields the most effective RAG system for German medical information retrieval (RQ4)?

To systematically investigate these questions and comprehensively evaluate different RAG configurations, we require rigorous evaluation methodologies that assess both retrieval quality and generation quality. The following section outlines our approach to this crucial aspect of the research.

2.6 Evaluation Methodologies for RAG Systems

Effective evaluation of RAG systems requires a multi-faceted approach that assesses both retrieval quality and generation quality. While Chapter 4 presents our detailed experimental methodology and comprehensive evaluation framework, this section briefly introduces the key metrics and their alignment with our research questions [Fin+24].

The evaluation of RAG systems presents unique challenges compared to traditional information retrieval or text generation tasks. Unlike conventional search systems, which can be evaluated solely on their ability to retrieve relevant documents, or standard language generation, which focuses on text quality and coherence, RAG combines both retrieval and generation in an interconnected pipeline. This necessitates evaluation methods that can capture the effectiveness of each component individually as well as their synergistic performance [Wan+24c].

Our evaluation approach encompasses two primary dimensions:

1. **Retrieval Quality Metrics** assess the system's ability to surface relevant information:
    - *Context Relevance, Recall, and Precision* measure how effectively different embedding models and retrieval strategies identify pertinent information, directly informing RQ1 and RQ2.
2. **Generation Quality Metrics** evaluate the LLM's ability to synthesize accurate responses:
    - *Answer Accuracy, Faithfulness, and Response Groundedness* assess how well generator LLMs transform retrieved context into reliable answers, addressing RQ3 and RQ4.

To compare with the existing roXtra system (RQ5), we employ the RAGAS evaluation suite [Es+23] alongside query type analysis to identify specific scenarios where RAG demonstrates superior capabilities. RAGAS provides a standardized framework for RAG evaluation, incorporating metrics that assess context relevance, answer faithfulness to the retrieved context, and answer correctness relative to reference responses. This comprehensive evaluation approach enables us to identify not only which RAG configuration performs best overall but also which types of queries or information needs are particularly well-served by the RAG approach compared to traditional keyword-based systems.

The multi-faceted evaluation methodology adopted in this research ensures that our findings are robust, nuanced, and directly applicable to the real-world information retrieval challenges outlined in earlier sections. Full details of the evaluation methodology, including metric definitions and implementation specifics, are provided in Chapter 4.

2.7 Synthesis and Knowledge Gaps

The preceding review highlights significant advancements in NLP, particularly the development of transformer-based LLMs and the RAG architecture, which offer promising avenues for addressing long-standing challenges in medical information retrieval. LLMs demonstrate remarkable language understanding and generation capabilities but suffer from critical limitations regarding factual accuracy (hallucinations), static knowledge, potential biases, lack of traceability, and challenges related to using external cloud services, especially with institutional data. RAG emerges as a powerful strategy to mitigate these issues by grounding LLM generation in externally retrieved, verifiable information from curated knowledge bases,

enabling dynamic updates and facilitating secure, on-premises deployments crucial for healthcare settings.

The performance of RAG systems, however, is highly dependent on the synergistic interplay of its core components: the embedding model determining semantic understanding, the retrieval strategy balancing semantic search with keyword precision, and the generator LLM synthesizing the final response. State-of-the-art approaches involve specialized embeddings, hybrid retrieval mechanisms combining dense and sparse methods, and large, capable LLMs [Fin+24]. Comprehensive evaluation requires multi-faceted metrics assessing both retrieval quality (precision, recall, relevance) and generation quality (faithfulness, groundedness, accuracy, relevance) [Es+23].

Despite this progress, several knowledge gaps remain, particularly concerning the optimal configuration and deployment of RAG systems within the specific context of German hospital environments. This thesis aims to address the following gaps:

- **Lack of Systematic Comparison in German Medical Domain:** While RAG components are studied individually, there is a paucity of systematic, comparative research evaluating the combined impact of choices for embedding models, retrieval strategies, and generator LLMs specifically tailored to and tested on German medical documents within a framework suitable for local deployment.
- **Language-Specific vs. Multilingual Embeddings:** Limited empirical data exists directly comparing the effectiveness of state-of-the-art German-specific embedding models against leading multilingual models for RAG tasks involving complex German medical text.
- **Hybrid Retrieval Performance Quantification:** While hybrid retrieval is conceptually advantageous, its concrete performance benefits over optimized vector-only search using contemporary embedding models need rigorous quantification within this specific domain and language context.
- **Impact of LLM Architecture in RAG:** The extent to which LLMs with Mixture-of-Experts architectures offer tangible benefits over standard dense models specifically for the synthesis task within a medical RAG pipeline remains underexplored.
- **Benchmarking Against Incumbent Systems:** Most RAG research compares different RAG variants or LLMs. There is a need for benchmarks

comparing optimized RAG systems against the non-LLM, keyword-based search systems (like roXtra) actually deployed and used in clinical practice, to demonstrate practical advantages.

- **Focus on On-Premise Deployable Architectures:** Many studies utilize cloud-based APIs, whereas this work focuses explicitly on architectures using on-premise deployable, open-source components, addressing critical requirements like data control and integration for real-world enterprise adoption often overlooked in purely performance-centric research.

- **Rigorous Application of Multi-faceted Evaluation:** Applying comprehensive evaluation frameworks combining retrieval and generation metrics (like RAGAS plus custom measures) systematically across different RAG configurations in the German medical context provides needed empirical grounding.

This thesis directly addresses these gaps through its research questions. The lack of comparative data on embedding models for German medical text motivates RQ1. The need to quantify the benefits of hybrid search informs RQ2. The underexplored impact of LLM mixture of expert's abilities within RAG drives RQ3. Synthesizing these component impacts to identify an optimal configuration (RQ4) and benchmarking it against the incumbent system (RQ5) provides practical insights currently lacking. This investigation is positioned at the intersection of LLM advancements, RAG optimization, medical domain specialization, German language processing, and practical deployment constraints (like on-premise hosting and data control). Its contribution lies in the synergistic investigation of these interconnected factors within a single, rigorous empirical study, aiming to bridge the gap between academic research and the practical needs of clinical information access by demonstrating the value of optimized RAG over existing workflows.

The specific RAG configurations designed to systematically investigate these questions are summarized in Table 2.1, providing a structured framework for the experimental evaluation that follows in subsequent chapters. Each configuration represents a specific combination of embedding model, retrieval strategy, and generator LLM chosen to isolate and assess the impact of individual components while building towards an optimized system that addresses the limitations identified in current approaches. These configurations form the foundation for our experimental evaluation, enabling systematic investigation of how different component choices affect overall RAG performance in the German medical domain. The results of this evaluation will provide empirical evidence to guide the development of optimized

information retrieval systems that address the challenges identified in this literature review, potentially transforming how medical professionals access and utilize critical information in their daily practice.

Table 2.1: Overview of RAG Configurations Investigated in this Thesis

Configuration Name	Embedding Model	Retrieval Strategy	Generator LLM	Key Hypotheses Targeted
Multilingual RAG	intfloat/multilingual-e5-large (Multilingual)	Vector Search (Dense)	Llama-3.3-70B-Instruct	Baseline
RAG with German specialized Embedding model	mixedbread-ai/deepset-mxbai-embed-de-large-v1 (German)	Vector Search (Dense)	Llama-3.3-70B-Instruct	RQ1
Hybrid RAG	mixedbread-ai/deepset-mxbai-embed-de-large-v1 (German)	Hybrid Search (Dense+BM25)	Llama-3.3-70B-Instruct	RQ1, RQ2
RAG with Mixture-of-experts LLM as Generator	mixedbread-ai/deepset-mxbai-embed-de-large-v1 (German)	Hybrid Search (Dense+BM25)	Llama-4-Scout-17B-16E-Instruct	RQ1, RQ2, RQ3

3 RAG: Architecture And Components

3.1 Enhancing Large Language Models with External Knowledge

Retrieval-Augmented Generation (RAG) represents a pivotal architectural approach that addresses fundamental limitations of Large Language Models (LLMs) by dynamically integrating external knowledge into the text generation process [Kar+20; Lew+21]. Rather than relying solely on static training data, RAG systems retrieve relevant information segments from an external corpus before generating responses, enabling outputs that are more accurate, relevant, and grounded in verifiable facts [Che+24c; Fin+24].

At its core, RAG combines two powerful technologies: an information retrieval [Tha+21] system and a generative language model [Sho+25]. As illustrated in Figure 3.1, user input flows through a retriever component that selects relevant data, followed by a generator component that formulates the final output. This architecture transforms LLMs from closed systems limited by their training cutoff [Che+24c] into frameworks capable of accessing contextually relevant, up-to-date information during inference. This dynamic grounding makes RAG particularly suitable for knowledge-intensive domains like medicine [Sir+23], where access to current and specific information is paramount.

3.1.1 Addressing Inherent LLM Limitations through RAG

Standard LLMs face several critical limitations that RAG directly addresses [WKR25]. First, their knowledge is frozen at training time, rendering them unaware of subsequent developments—a significant deficiency in rapidly evolving fields like medicine [Che+24c; Kor22]. Second, LLMs trained on general web data often lack the deep, domain-specific knowledge required for specialized tasks [She+24]. Third, they are prone to "hallucinations"—factually incorrect or nonsensical statements that arise from identifying statistical patterns in language without true comprehension [Ji+22; Sab23]. Finally, their decision-making process remains opaque, making it difficult to trace the sources underlying generated output [FBS24].

By explicitly retrieving and conditioning the generation process on specific,

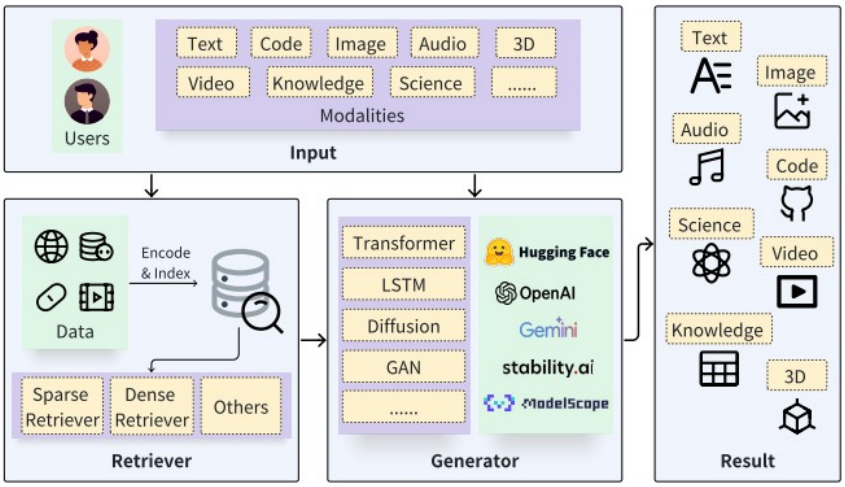


Figure 3.1: Typical RAG Workflow, showcasing the interaction between retriever and generator components. [FAT24]

verifiable information from designated external sources, RAG provides a pathway to traceable, factually grounded responses—indispensable when handling sensitive medical information where accuracy and accountability are non-negotiable [Fin+24; Mag+24; Zho+24b].

3.1.2 Key Advantages of the RAG Approach for Knowledge-Intensive Tasks

RAG implementation offers compelling benefits for applications demanding high factual accuracy, access to current information, and domain specificity [Li+24b]. By providing LLMs with relevant, externally sourced facts during generation, RAG substantially mitigates hallucinations and improves factual correctness—effectively giving the model access to reference materials for an "open-book exam" [Gao+24; Mag+24; Wan+24c].

This approach enables incorporation of information more recent than the LLM's training cutoff by connecting to dynamically updated knowledge bases [Fan+24; Sir+23]. For specialized domains like medical document analysis, RAG facilitates effective domain adaptation without the prohibitive cost of retraining the base LLM [Alg+24]. Organizations can configure the system to retrieve information exclusively from proprietary datasets or curated medical literature, allowing a general-purpose LLM to effectively "specialize" through context provided by domain-specific sources [Sir+23].

Moreover, reliance on retrieved documents enhances verifiability and transparency, as well-designed RAG systems can provide citations pointing back to specific source documents [Wu+24b; Zho+24b]. This traceability fosters user trust and enables independent verification—essential for enterprise utility or research integrity [Xia+93].

From a resource perspective, adapting an LLM's knowledge base using RAG is typically more cost-effective and agile than retraining or fine-tuning [Liu+22]. It allows for incremental knowledge base updates without altering the underlying model while granting developers explicit control over information sources, enabling curation and quality control that further enhances reliability [Gao+24].

3.1.3 The RAG Workflow: From Data Ingestion to Response Generation

A typical RAG system operates through an orchestrated pipeline, processing information across distinct stages to achieve grounded text generation [FM24]. While specific implementations vary, the fundamental workflow involves transforming external knowledge into a searchable format, identifying relevant knowledge based on a user query, and synthesizing this retrieved information into a coherent response.

The process begins with **Data Preparation**, a foundational phase focused on converting raw external knowledge – often originating from diverse, unstructured sources common in healthcare (like clinical notes, research papers, or information leaflets) [Aya+21] – into a clean, structured format suitable for processing. This involves acquiring the data (*Ingestion*), cleaning it and extracting meaningful content (*Parsing*), and segmenting this content into smaller, manageable units (*Chunking*) [HSM24]. The thoroughness and quality of data preparation profoundly influence the effectiveness of all subsequent steps [Wan+24c].

Next is the **Indexing** stage. Here, the prepared data chunks are transformed into numerical representations, known as embeddings [RS21], which capture their semantic meaning. These embeddings, along with metadata linking them back to their original source (e.g., document name, page number, section), are loaded into specialized vector databases [PWL24], or vector stores. These databases employ efficient algorithms to create a searchable index, enabling the system to find information based on conceptual similarity rather than just keyword matching [GMM03].

When a user submits a query [YM98], the **Retrieval** stage is initiated. The system first converts the user's query into an embedding using the same model employed during indexing [LZM22]. This query embedding is then used to search

the index within the vector store, identifying and retrieving the data chunks whose embeddings are semantically closest to the query embedding [Fin+24]. This step effectively pinpoints the most relevant pieces of information within the knowledge base related to the user's request.

Finally, the process culminates in the **Generation** stage. The retrieved data chunks, representing the most relevant context, are combined with the original user query to form an augmented prompt [Fin+24]. This prompt, containing both the question and its supporting evidence, is then fed into the generator LLM. The LLM's task is to synthesize the information from the retrieved context, guided by the query, to produce a coherent, contextually grounded, and informative final response [Lew+21].

This pipeline mirrors the research process: Data preparation is similar to acquiring, organizing, and cleaning research materials. Indexing resembles creating a detailed catalog or database for efficient searching [Boo+21]. Retrieval corresponds to using the catalog to find relevant books or articles for a specific research question. Generation parallels the researcher reading the selected materials and synthesizing the gathered information into a comprehensive report or answer [CC17].

The success of any RAG system hinges on the quality and relevance of its underlying knowledge sources and the precision of its data preparation, indexing, and retrieval mechanisms [KR18]. The following sections will elaborate on each stage, exploring critical components, techniques, and design considerations relevant to building an effective RAG system, particularly one tailored for the unique challenges of processing German medical documents as undertaken in this thesis.

3.1.4 Comparing Knowledge Enhancement Strategies: RAG vs. Fine-Tuning

Although RAG provides a powerful method to integrate LLMs with external knowledge, it coexists with another primary technique for model adaptation: fine-tuning [Par+24]. Understanding their distinctions clarifies the rationale behind choosing RAG for specific applications, including the system developed in this thesis.

Fine-tuning adapts a pre-trained LLM by continuing its training process on a smaller, domain-specific dataset [Par+24]. This process modifies the LLM's internal parameters (weights) to specialize its behavior and knowledge for particular tasks or domains, leveraging the principle of transfer learning [Nic+17]. Unlike RAG, which provides external context at inference time, fine-tuning encodes specialized knowledge directly into the model parameters.

The key differences influence their suitability for different scenarios [Alg+24; Bal+24; SKH24] :

Table 3.1: Comparison of RAG and Fine-Tuning for Knowledge Enhancement [Bal+24; Par+24; SKH24]

Aspect	RAG	Fine-Tuning
Knowledge Access	Dynamic, via external retrieval	Static, encoded in model parameters
Updating Knowledge	Relatively easy (update knowledge base)	Requires retraining the model
Computational Cost	Lower upfront, higher per-query latency	High upfront (training), faster inference
Adaptability to New Info	High	Lower without retraining
Traceability/ Verifiability	High (can point to retrieved sources)	Limited (internal reasoning opaque)
Hallucination Risk	Lower (when grounded in reliable sources)	Can persist, especially outside training data

RAG is often favored when continuous access to the latest information is critical (as in medicine), when dealing with rapidly evolving knowledge domains, or when the ability to verify the source of information is paramount [SKH24]. Fine-tuning might be preferred when the goal is to deeply adapt the model's style, tone, or inherent understanding of specific concepts that are relatively stable, or when inference speed is the absolute priority after an initial training investment [Par+24]. For this thesis, the ability of RAG to leverage specific, updatable corpora of German medical documents and provide traceable answers makes it the more suitable architecture.

3.2 Data Preparation: Building the Foundation for Retrieval

The data preparation phase is the cornerstone of an effective RAG system [HSM24]. Its objective is to transform raw, often heterogeneous information sources into a clean, consistently formatted, and segmented dataset optimized for semantic search and subsequent LLM processing. The quality and appropriateness of the data

prepared here directly determine the potential accuracy of retrieval and the ultimate reliability of the generated responses [Fin+24]. This phase typically involves three sequential processes: ingestion, parsing, and chunking.

3.2.1 Ingestion: Acquiring and Consolidating Knowledge

Ingestion marks the beginning of the RAG system's data pipeline, encompassing the systematic collection and consolidation of data from the various repositories or files intended to form its external knowledge base [Wan+24c]. This foundational step significantly influences the scope, timeliness, and potential bias of the information the system can ultimately access and retrieve. RAG systems are adaptable and can potentially ingest data from a wide array of sources [Fin+24]. While structured data from databases (SQL/NoSQL) or dynamic data streams via Application Programming Interfaces (APIs) can be utilized [Per16], many RAG applications, especially in enterprise settings, focus heavily on processing unstructured and semi-structured content [Yep+24]. Common inputs often include documents residing in file systems, such as Portable Document Format (PDFs), Microsoft Word documents (DOCX), web pages (HTML), or plain text/Markdown files [Bav+21].

Several practical considerations are paramount during the ingestion phase, particularly when dealing with sensitive or rapidly changing information characteristic of the healthcare field. Data quality - which encompasses accuracy, completeness and currency - is not negotiable. Ingesting inaccurate or outdated medical information would critically undermine the reliability of the RAG system and could potentially lead to harmful output [PV24]. Consequently, rigorous source vetting and quality control measures are essential prerequisites, which may require manual effort. Furthermore, because medical knowledge is subject to constant evolution (due to new research, updated guidelines, revised protocols) [JLL24], establishing robust update mechanisms is vital [AGK95]. The ingestion process must incorporate effective strategies for regularly incorporating new information and potentially retiring outdated documents [Boo+21]. This ensures the RAG system's knowledge base remains current, accurate, and trustworthy over the long term.

3.2.2 Parsing: Extracting and Structuring Content

Once data is ingested, it requires parsing to convert it from its raw format into a clean, usable structure [Yan+24]. The primary objectives are to accurately extract the relevant textual content while discarding irrelevant elements (e.g., complex formatting, navigation menus, advertisements) and to standardize this content for reliable downstream processing.

Different data formats necessitate different parsing techniques. For PDF documents, which are prevalent in medical literature and clinical documentation, libraries like PyMuPDF and docling offer robust capabilities for extracting text content [Art25]. Such tools can often handle various PDF structures, aiming to preserve textual flow even across columns or around figures, which is important for maintaining the coherence of professional texts [Lla25; PV24]. While no parser is perfect, especially with highly complex layouts, tools like PyMuPDF provide a practical balance of efficiency and accuracy for many common PDF structures encountered in medical documents. Similar specialized parsers exist for HTML, DOCX, and other formats, often focusing on isolating the core textual content.

A crucial aspect of parsing is the extraction and preservation of metadata [Yan+24]. Information such as the document title, authors, publication date, source URL, section headings within the document, or relevant keywords provides invaluable context. This metadata, stored alongside the extracted text, can significantly enhance retrieval relevance by enabling filtering (e.g., retrieving information only from guidelines published after a certain date) and is essential for generating accurate citations in the final response, thereby improving the system's transparency, traceability and trustworthiness [Fin+24].

3.2.3 Chunking: Segmenting Data for Effective Processing and Retrieval

Following parsing, the cleaned text content, which can often be lengthy (e.g., entire research articles or book chapters), is typically subjected to chunking [Zho+24a]. This process involves dividing the text into smaller, more manageable segments or "chunks." Chunking is a critical optimization driven by several technical and performance considerations [QTB24].

Firstly, chunking ensures that the data segments conform to the input length limitations (context windows) of both the embedding models used for indexing and the generator LLMs used for synthesizing the final response [Fin+24]. Processing text segments that exceed these limits can lead to errors or silent truncation, potentially losing valuable information.

Secondly, chunking aims to enhance retrieval precision [QTB24]. Smaller, more focused chunks generally allow the retrieval system to pinpoint information directly relevant to a specific query more effectively. If chunks are too large, a relevant sentence might be retrieved, but it could be buried within a large amount of surrounding irrelevant text, potentially diluting the semantic signal for the generator LLM and leading to less precise or overly broad answers [Zho+24a].

Lastly, processing smaller chunks is generally more computationally efficient during embedding, indexing, and retrieval compared to handling entire large documents at once [Wan+24c].

Various strategies exist for chunking text, each presenting trade-offs between implementation simplicity, computational cost, and the degree to which semantic coherence is preserved [Wan+24c]. A straightforward approach is fixed-size chunking [Yep+24]. This method splits the text into segments of a predetermined length (e.g., measured in tokens or characters), often with some overlap between consecutive chunks to maintain context across boundaries [PV24]. While simple and fast, fixed-size chunking risks arbitrarily splitting sentences, paragraphs, or logical units of thought, potentially disrupting the semantic meaning captured by the embedding model and hindering the generator LLM's comprehension . This can be particularly problematic for complex medical texts where sentence structure and paragraph breaks often delineate important steps in reasoning or lists.

To mitigate semantic disruption, semantic or content-aware segmentation methods attempt to divide the text along natural linguistic or structural boundaries [QTB24]. These strategies might leverage sentence-ending punctuation, paragraph breaks (e.g., identified by double newlines), or structural elements identified during parsing (like section headings). By aligning chunks with the inherent structure of the text, these approaches generally produce more semantically coherent segments [MS25]. This often leads to more meaningful embeddings and improves the relevance of retrieval results, as each chunk is more likely to represent a self-contained idea or piece of information.

A popular technique that balances simplicity and semantic awareness is recursive chunking [Wan+24c]. This strategy employs a hierarchical list of separators (e.g., trying paragraph breaks first, then sentence breaks, then perhaps specific punctuation like commas, and finally character-level splitting as a last resort). It recursively applies these separators to split larger pieces into smaller ones until all resulting chunks satisfy a specified size constraint. This provides a structured way to prioritize more significant semantic breaks over less meaningful ones.

The optimal chunking strategy depends heavily on the characteristics of the source documents (e.g., structure, density, language complexity of German medical texts), the specifications of the chosen embedding model (its context window, sensitivity to semantic breaks)[NNN24], and the specific requirements of the RAG application (e.g., the desired granularity for retrieval and citation). Careful consideration and potentially empirical testing are often needed to select the most effective chunking approach for a given project [Wan+24c].

3.3 Indexing: Structuring Knowledge for Efficient Semantic Access

With the source documents prepared and segmented into chunks, the indexing phase transforms this processed text into a format optimized for rapid semantic search. This involves two primary steps: generating numerical vector representations (embeddings) that capture the meaning of each chunk, and storing these embeddings, along with associated metadata, in specialized database systems – vector stores – designed for efficient similarity-based retrieval [SMS24a].

3.3.1 Embedding Generation: Translating Text into Semantic Vectors

Embedding generation is the cornerstone of enabling semantic understanding within a RAG system [RS21]. This process utilizes specialized machine learning models [Che+24a], known as embedding models [Wan+19], to convert textual data (the prepared chunks) into numerical vectors. These models are trained to map text passages with similar meanings to points that are geometrically close in a high-dimensional vector space, while mapping dissimilar passages to points that are far apart [TP10]. This geometric representation allows the system to search for information based on conceptual relatedness, moving beyond simple keyword matching to understand nuances of meaning, synonyms, and paraphrasing [HS03].

During the embedding process, each text chunk is fed into the chosen embedding model, which then outputs a dense vector – typically an array of hundreds or thousands of floating-point numbers [Mue+22]. This vector encapsulates the chunk's semantic meaning within the model's learned representational space. The quality of these embeddings is paramount, as it directly determines the upper bound on the effectiveness of the subsequent retrieval step [GR09]. Poor embeddings that fail to capture semantic nuances will lead to irrelevant retrieval results, regardless of how sophisticated the search mechanism is. This is also a motivation behind our research to compare multiple embedding models for optimal RAG performance.

Many state-of-the-art embedding models employed for dense retrieval within Retrieval-Augmented Generation (RAG) systems are based on the Transformer architecture [VSP17]. Prominent examples include BERT (Bidirectional Encoder Representations from Transformers) [Cla+19] and its derivatives, such as Sentence-BERT [RG19]. These models generate dense vector representations, characterized by predominantly non-zero elements, which allows them to effectively encode nuanced semantic distinctions [Wan+19]. For this thesis, which focuses on German medical

documents, it is crucial to select an embedding model with demonstrated strong performance in German text, or a state-of-the-art multilingual model [Des+21]. This choice is essential to accurately capture the specific meaning of specialized German terminology and complex scientific concepts. The Massive Text Embedding Benchmark (MTEB) [Mue+22] provides a valuable resource for this selection process, offering quantitative evaluations and leaderboards to compare the suitability of different models [Mue+22].

3.3.2 Vector Stores: Databases adapted for Embedding Management and Search

Once text chunks are transformed into embedding vectors, they must be stored in a way that allows for efficient searching. This is a function of vector stores (also known as vector databases) [PWL24]. These are database systems specifically engineered to handle the storage, management, and querying of large collections of high-dimensional vectors, such as those generated from text [Wan+21]. They provide the essential infrastructure for performing fast semantic similarity searches at scale, acting as the persistent, indexed repository for the knowledge of the RAG system [HLW23].

Vector stores offer several key capabilities crucial for RAG systems [Fil+24]. They provide efficient storage mechanisms optimized for high-dimensional vectors, often incorporating techniques like quantization or compression to manage storage requirements while attempting to minimize the impact on retrieval accuracy [PWL24]. Critically, they store associated metadata alongside each vector, maintaining the link back to the original text chunk and its source document attributes (e.g., document ID, page number, publication date), which is crucial to maintain traceability of the generated responses.

A core feature is their implementation of algorithms for Approximate Nearest Neighbor (ANN) search [IM99]. Performing an exact similarity calculation between a query vector and every vector in a large database (brute-force search) is computationally infeasible for real-time applications [Sch+93]. ANN algorithms build specialized index structures over the vectors (discussed in 3.3.3) that allow the system to find vectors highly likely to be among the true closest neighbors much faster, accepting a small, often negligible, trade-off in recall (potentially missing the absolute closest neighbor occasionally) for significant gains in speed [WJ96].

Vector stores provide querying interfaces (e.g., APIs, query language extensions) allowing applications to submit a query vector and retrieve the 'k' most similar vectors based on a chosen similarity metric (e.g., cosine similarity, Euclidean distance)

[Wan+24c]. Crucially for sophisticated applications, they support metadata filtering [Chr23]. This allows combining semantic search with traditional attribute-based filtering. For instance, in a medical context, a query could search for semantically relevant chunks only within documents classified as "clinical trial results," published after "2023," significantly enhancing retrieval precision and relevance by narrowing the search space based on known criteria [Shi+24].

The landscape of vector stores offers diverse options. Managed cloud services (e.g., Pinecone [Pin24], managed versions of open-source databases) provide scalability and reduce operational burden. Self-hostable open-source databases (e.g., Milvus [Wan+21], Weaviate [Eti], Qdrant [Qdr23], Chroma [Chr23]) offer greater control, customization, and potential cost savings but require more management effort. Additionally, established database systems like PostgreSQL (with extensions like pgvector [pgv]) and Elasticsearch/OpenSearch [GT15] have integrated vector search capabilities, allowing organizations to leverage existing infrastructure. The choice depends on factors like scale, budget, required features (e.g., specific filtering capabilities), and operational capacity [Wan+24c].

3.3.3 Indexing Algorithms: Enabling Efficient Similarity Search

Within the vector store, indexing algorithms are the specific techniques used to organize the embedding vectors to enable rapid Approximate Nearest Neighbor (ANN) search, avoiding the prohibitive cost of brute-force comparison [WJ96]. The selection of an indexing algorithm involves navigating fundamental trade-offs between search speed (latency), search accuracy (recall), memory (RAM) consumption, index build time, and the ease of updating the index with new data [BC05].

The simplest baseline is a Flat Index, which essentially performs brute-force search [Vim+24]. It stores vectors without any special organizational structure. Finding nearest neighbors requires comparing the query vector to every stored vector. While guaranteeing perfect recall (finding the exact nearest neighbors), its linear scaling makes it impractically slow for typical RAG datasets [JDJ17].

To achieve practical performance, various ANN indexing methods are employed, often implemented using libraries such as FAISS [Dou+24] or integrated within vector database systems . One common family relies on space partitioning, exemplified by the Inverted File Index (IVF) approach [MB11]. IVF first divides the vector space into clusters (using an algorithm like k-means [ASI20]) and assigns each vector to its nearest cluster centroid. During search, the system identifies the cluster(s) closest to the query vector and restricts the search to only the vectors

within those selected clusters (cells). This drastically reduces the number of comparisons needed. However, IVF requires an initial training step for clustering and tuning parameters like the number of clusters and the number of probed clusters during search [Baz23].

Another widely used and often high-performing category is based on graph structures, notably Hierarchical Navigable Small World (HNSW) [WFG24]. HNSW constructs a multi-layered graph where nodes are vectors and edges connect vectors to their neighbors at different proximity scales. Searching involves navigating this graph greedily, starting from an entry point and moving progressively closer to the query vector across layers until the likely nearest neighbors are found . HNSW often provides an excellent balance of high speed and high recall and supports dynamic data addition reasonably well [Lin24a]. However, building HNSW indexes can be computationally intensive and memory-hungry, and its performance is sensitive to several configuration parameters set during index construction and querying [MY16].

The choice between algorithms like IVF, HNSW, or others depends on the specific application requirements, such as the size of the dataset, the required query latency, the acceptable accuracy trade-off, available memory resources, and how frequently the underlying data needs to be updated [Lin24b]. For instance, a system requiring very low latency might prioritize speed over perfect recall, while a system where missing relevant information is critical (as potentially in medicine) might favor algorithms offering higher recall, even at the cost of slightly higher latency or memory usage.

3.4 Retrieval: Locating Relevant Information for the Query

The retrieval stage is where the RAG system actively searches its indexed knowledge base to identify the information most relevant to the user's current query [Bar24]. This core process involves translating the user's query into a comparable vector representation and then utilizing search strategies within the vector store to retrieve the content chunks that will best inform the LLM's subsequent response generation.

3.4.1 The Core Retrieval Mechanism: Semantic Similarity Search

The foundation of most modern RAG retrieval systems is semantic similarity search conducted within the high-dimensional vector space established during indexing [Jad+22]. The process typically proceeds as follows:

First, the user's input query (which could range from a simple keyword to a complex natural language question) is transformed into a *Query Embedding* [Fin+24]. Critically, this transformation must use the exact same embedding model employed during the indexing phase to embed the document chunks [Yep+24]. This consistency ensures that the query vector and the document chunk vectors reside within the same semantic space, making their relative positions and distances meaningful for comparison.

Next, this query vector is sent to the vector store to perform the *Similarity Search* [Fin+24]. Leveraging its pre-built ANN index (e.g., HNSW or IVF), the vector store efficiently navigates the potentially vast collection of stored document chunk vectors. Instead of the infeasible brute-force comparison, the ANN index rapidly identifies a set of candidate vectors highly likely to be among the closest (most similar) to the query vector, according to a predefined mathematical measure of similarity or distance [IM99].

The concept of "similarity" between the query vector ($\mathbf{q}$) and document chunk vectors ($\mathbf{d}$) is quantified using specific mathematical metrics [UH05]. The selection of an appropriate metric significantly impacts retrieval results and should align with the properties of the chosen embedding model [SAW15]. Three metrics commonly employed in RAG systems are:

1. **Cosine Similarity**: Measures the cosine of the angle between vectors ($\text{sim}(\mathbf{q}, \mathbf{d}) = \frac{\mathbf{q} \cdot \mathbf{d}}{\|\mathbf{q}\| \|\mathbf{d}\|}$), ranging from -1 to 1, with 1 indicating perfect similarity. This metric is widely preferred for text embeddings as it focuses on semantic orientation rather than magnitude, which can vary with chunk length [GSB18].

2. **Euclidean Distance**: Calculates the straight-line distance between vector endpoints ($d(\mathbf{q}, \mathbf{d}) = \|\mathbf{q} - \mathbf{d}\|_2 = \sqrt{\sum_{i=1}^{n} (q_i - d_i)^2}$), where smaller values indicate greater similarity. Unlike cosine similarity, Euclidean distance is sensitive to vector magnitude [Gow85].

3. **Dot Product**: Computed as $\mathbf{q} \cdot \mathbf{d} = \sum_{i=1}^{n} q_i d_i$, this metric considers both vector alignment and magnitude. When vectors are normalized to unit length (a standard preprocessing step [Car05]), dot product becomes mathematically equivalent to cosine similarity.

These metrics offer different trade-offs between computational efficiency and

semantic sensitivity, with cosine similarity typically providing the most balanced performance for text retrieval applications [WJ96].

Finally, the retrieval step *Outputs* a ranked list, typically containing the top 'k' document chunks whose vectors were found to be most similar to the query vector (where 'k' is a configurable parameter, e.g., retrieve the 5 most similar chunks) [Wan+24c]. This output usually includes the similarity score, the original text content of the chunk, and associated metadata (like source document identifiers, page numbers), which are crucial for the generation stage and for potential citation rendering [Fin+24].

3.4.2 Exploring Diverse Retrieval Strategies: Beyond Basic Semantic Search

While dense semantic retrieval using neural embeddings forms the backbone of many RAG systems, understanding its strengths and limitations in comparison to other paradigms is beneficial [Wan+24c]. Often, the most robust retrieval systems, particularly for complex domains like medicine, employ *Hybrid Retrieval* strategies that intelligently combine different approaches to leverage their complementary strengths [SMS24b].

Dense Retrieval, as detailed above, relies on comparing dense vector embeddings generated by neural models [Kar+20]. Its primary strength is its ability to capture semantic meaning, context, and user intent, enabling it to find relevant information even when keywords don't match exactly . It excels at handling synonyms (e.g., finding "cardiac insufficiency" for a query about "heart failure"), paraphrasing, and conceptual similarity. However, dense retrieval can sometimes struggle with queries demanding precise literal matches (e.g., specific medical codes, drug names with unique spellings) [Lin24a] and is generally more computationally intensive and requires more storage than traditional methods [Sul+24].

Sparse Retrieval (Lexical Search) represents the traditional information retrieval approach, focusing on keyword matching [Lua+21]. Algorithms like Okapi BM25 are prominent examples [Rob10]. BM25 ranks documents based on the statistical occurrence of query keywords, considering term frequency (TF - how often a word appears in a document) and inverse document frequency (IDF - how rare a word is across the collection) [SB09]. Sparse retrieval excels where dense retrieval might falter: finding documents containing exact keywords, acronyms, specific codes, or unique identifiers [Lin24a]. It is computationally efficient and highly interpretable [Lua+21]. Its main limitation is the lack of true semantic understanding; it cannot

grasp synonyms or conceptual similarity unless explicitly handled (e.g., through query expansion) [WC11].

Recognizing these complementary characteristics, *Hybrid Retrieval* seeks to combine both dense and sparse methods to achieve superior overall performance [BGI24]. This approach aims to harness the deep semantic understanding of dense retrieval while retaining the keyword precision of sparse retrieval [BGI24]. A common technique involves running both sparse and dense searches in parallel and then intelligently merging the results using a re-ranking algorithm [SMS24b].

For processing German medical documents, a hybrid approach appears particularly advantageous. It allows the system to capture the semantic meaning of complex clinical descriptions, symptoms, and conditions expressed in natural language (dense retrieval's strength) while simultaneously ensuring precise matching of specific medical terminology, drug names (e.g., "Acetylsalicylsäure"), diagnostic codes (e.g., ICD-10 codes), and anatomical references where exactitude is crucial (sparse retrieval's strength). This combination enhances robustness across diverse query types, from conceptual questions to requests for information containing specific entities.

3.5 Generation: Synthesizing Knowledge into Coherent Responses

The final stage in the RAG pipeline is **Generation**. This phase harnesses the capabilities of a Large Language Model (LLM), designated as the "generator," to synthesize the retrieved information into a useful response [Fin+24]. The generator LLM receives the user's original query along with the set of relevant contextual passages retrieved from the knowledge base. Its critical function is to integrate this evidence and formulate a coherent, informative, and contextually grounded answer that directly addresses the user's information need [Lew+21].

3.5.1 The Generator LLM: Engine for Synthesis and Formulation

Within the RAG architecture, the generator LLM acts as the engine for synthesis, reasoning (within the bounds of the provided context), and fluent natural language expression [Yu+24]. It processes an augmented prompt containing both the user's query and the supporting text chunks retrieved from the external knowledge source. Its role transcends simple extraction or summarization; it is expected to

intelligently weave together information from potentially multiple retrieved chunks, using them as factual grounding to construct a comprehensive answer to the query [FBS24]. A primary goal is to ensure the generated response remains faithful to the provided contextual evidence, minimizing reliance on the LLM's internal parametric knowledge, which might be outdated, inaccurate, or overly general for the specific task [Che+24c].

A variety of powerful LLMs can serve as the generator component [Nav+23]. Widely used examples include models from OpenAI's GPT series (e.g., GPT-3.5-turbo, GPT-4 variants) [San23], known for strong instruction following and generative fluency. Anthropic's Claude models [Ant25] are recognized for robust performance on complex reasoning and handling very long contexts. Meta's Llama family [AI22] offers high-performance open-source alternatives, providing flexibility for customization and local deployment. Numerous other models, both commercial and open-source, are also viable options [SWK24].

Selecting an appropriate generator LLM for the specific application of analyzing German medical documents requires careful consideration of certain characteristics [Zha+24]. Foremost is multilingual capability, specifically high proficiency in German, to accurately interpret both the German source documents and potentially German user queries containing technical medical terms [AI22]. Furthermore, a model possessing some inherent understanding of, or having been fine-tuned on, biomedical terminology and concepts is likely to produce more nuanced, accurate, and contextually appropriate responses within the medical domain [Par+]. The model's context window size is also a practical constraint, determining how much retrieved information can be processed simultaneously [An+24].

3.5.2 Crafting Effective Prompts for Grounded Generation

The way in which the user's query and the retrieved context are presented to the generator LLM – the structure and content of the augmented prompt – significantly influences the quality, faithfulness, and overall success of the final output [Gir23]. Prompt engineering, the practice of carefully designing these prompts, is crucial in RAG systems [Whi+23]. Effective prompts guide the LLM to prioritize the provided context, maintain factual grounding in the retrieved evidence, and resist the tendency to hallucinate or rely solely on its potentially less relevant internal knowledge [Mes23].

Having established the functional roles of the retriever and generator within the RAG architecture, we now turn our attention to a critical aspect of the system's practical implementation and configuration: the design of the prompt itself. This element acts as the crucial interface conveying retrieved information and task

instructions to the generator. Crafting effective prompts, therefore, involves several key techniques aimed at ensuring the generator utilizes the provided context appropriately to produce grounded outputs [Gir23].

Foremost among these techniques are *explicit instructions*, which entail clearly stating the LLM's task and constraints, such as directing it to answer based *only* on the provided context and to explicitly state if the answer cannot be found, thereby mitigating hallucinations [Wan+23]. Specifying a *role* (e.g., "You are a helpful assistant analyzing medical texts") can also guide the response style [Whi+23]. Equally important is providing *clear structure and demarcation*, using consistent delimiters like XML-like tags (`<context>`, `<query>`) or Markdown formatting (e.g., headers like `### Context Documents`, `### User Query`, or enclosing context in triple backticks) to help the LLM distinguish between system instructions, the retrieved evidence passages, and the user's query [Whi+23]. Practical *context handling strategies* are also needed for when the combined length of retrieved text exceeds the LLM's input limits [Liu+23]; common approaches include simple truncation (keeping only top-ranked chunks), though this risks information loss, or more sophisticated methods like re-ranking chunks or employing summarization techniques (potentially with another LLM call) to condense the context [Mes23]. Finally, considering *context placement considerations* may be beneficial, as some research suggests strategically placing critical information or the query at the start or end of very long prompts might help counteract potential 'lost-in-the-middle' effects where the model underutilizes information from the middle of the context [Liu+23]. The augmented prompt acts as the crucial interface for ensuring the "Augmented" aspect of RAG is realized. It channels the LLM's powerful generative capabilities, constraining them with relevant external knowledge, thereby enhancing the accuracy, relevance, and trustworthiness of the final response – qualities essential in high-stakes domains like medicine.

3.5.3 Generating the Final, Grounded Response

Guided by the meticulously crafted augmented prompt, the generator LLM produces a natural language output. For the RAG system to be deemed successful, this final response should ideally embody several key characteristics, especially critical in the medical context [Wan+24c]. First, it must demonstrate *accuracy*, meaning it is factually correct and strictly aligned with the information presented in the retrieved context documents, avoiding external facts or contradictions [Es+23]. The response must also be *relevant*, directly addressing the user's specific query and satisfying their underlying information need without tangential diversions [Zhu+24]. Furthermore, *coherence* is essential; the response needs to be well-

structured, grammatically correct, use clear and unambiguous language appropriate for the target audience (e.g., clinicians or patients), and exhibit logical flow [Es+23]. Crucially, it must be *grounded*, demonstrably derived from the provided context, with speculation, excessive inference, or hallucination of information not present in the sources minimized [Es+23]. Finally, the response should possess *utility*, presenting the information in a format, style, and level of detail that is genuinely helpful to the user for their specific task, such as summarizing findings or answering a clinical question [Wu+24a].

Depending on the application's requirements, post-processing steps may be applied to the LLM's raw output. A particularly valuable step for enhancing trustworthiness in RAG is *citation generation* [Zho+24b]. This involves identifying which parts of the response were derived from which specific retrieved chunks and embedding references or links back to those sources within the final output [FBS24]. This allows users to easily verify the information and consult the original source material, fostering trust and facilitating deeper investigation – a critical feature for medical and scientific applications.

3.6 Summary: The Synergy of RAG Components for Domain Information Analysis

Retrieval-Augmented Generation (RAG) offers a compelling architectural solution for overcoming inherent limitations of Large Language Models, particularly in domains demanding access to current, specialized, and verifiable information [LYZ24]. By dynamically integrating external knowledge sources into the generation process, RAG significantly enhances the factual grounding, relevance, and trustworthiness of LLM outputs, making it exceptionally well-suited for applications like the analysis of German medical documents explored in this thesis.

The effectiveness of a RAG system relies on the seamless integration and optimization of its core components [Pal+24]. The journey begins with meticulous **Data Preparation**, transforming raw medical documents into clean, structured, and appropriately segmented chunks ready for processing. This is followed by **Indexing**, where these chunks are converted into semantic embeddings and organized within efficient vector stores, enabling rapid retrieval based on meaning. The **Retrieval** stage then leverages these indexed embeddings to identify the most relevant pieces of information in response to a user's query, often employing hybrid strategies to balance semantic understanding with keyword precision crucial for medical terminology. Finally, the **Generation** stage utilizes a capable LLM, guided by a carefully constructed prompt containing the query and retrieved context, to

synthesize an accurate, coherent, and grounded response, ideally with traceable citations back to the source documents.

Compared to alternatives like fine-tuning, RAG provides distinct advantages in terms of knowledge updateability, transparency through source attribution, and inherent mechanisms for reducing factual inaccuracies by grounding responses in retrieved evidence [Bal+24]. These characteristics directly address the critical requirements of working with sensitive and dynamic medical information. The specific implementation choices across the RAG pipeline – from data ingestion and chunking strategies to the selection of embedding models, vector stores, retrieval methods, and generator LLMs – must be carefully considered and tailored to the unique challenges and demands of the target application, such as navigating the complexities of the German language and the specific nature of medical documentation, to build a truly effective and reliable system [Jeo23].

4 Evaluation Framework and Methodology

4.1 Introduction to RAG Evaluation

Evaluating Retrieval-Augmented Generation (RAG) systems necessitates careful consideration due to their inherent hybrid nature, integrating distinct retrieval and generation components whose performances are intrinsically linked [Zhu+24]. The foundational RAG framework, as introduced by Lewis et al. [Lew+21], aims to leverage retrieved documents to enhance the factual accuracy of Large Language Model (LLM) outputs [KB24] and mitigate the occurrence of hallucinations.

The challenges associated with RAG evaluation are particularly pronounced within the medical domain, arising from multiple factors [Xio+24]. Medical information retrieval requires exceptionally high precision and recall, as inaccuracies or omissions could significantly impact evidence review processes [Kir+16]. The German medical corpus introduces additional complexity through domain-specific terminology, prevalent compound words, and localized clinical practices [Kam+23]. Furthermore, stringent healthcare privacy regulations necessitate on-premises system deployment, imposing constraints on model selection and implementation approaches [Sch20].

To navigate these complexities, we have developed a comprehensive evaluation framework that synthesizes quantitative metrics with qualitative assessments. This framework provides a structured approach to evaluate four distinct RAG architectures, facilitating a comparative analysis of their performance against one another and relative to the incumbent roXtra keyword-based search system currently in use [Sch16].

4.2 Evaluation Metrics

The rigorous assessment of RAG systems hinges on the selection of appropriate evaluation metrics [Zhu+24]. Drawing upon established RAG evaluation frameworks like RAGAS [Es+23] and TruLens [Tru], we identified six core metrics designed to provide a holistic performance profile. These metrics are logically categorized according to the specific component of the RAG pipeline (retrieval or generation)

they primarily evaluate, complemented by metrics assessing the end-to-end system performance.

4.2.1 Retrieval-Specific Metrics

These metrics focus on the quality and sufficiency of the context retrieved prior to the generation phase:

Context Relevance

This metric assesses the pertinence of the retrieved context passages (chunks) to the user's query. Its calculation involves two independent "LLM-as-a-judge" evaluations [Zhe+23], each rating relevance on a three-point scale: '0' indicating no relevance, '1' signifying partial relevance, and '2' representing complete relevance. These numerical ratings are subsequently normalized to a $[0, 1]$ scale and averaged to yield the final score [Es+23]. Higher Context Relevance scores suggest the retrieval mechanism successfully identifies information closely aligned with the user's query, a prerequisite for accurate response generation. Conversely, low scores may indicate the inclusion of irrelevant or tangential information.

Context Recall

This metric evaluates whether the retrieved context encompasses all necessary information to answer the query comprehensively. It operates by determining if claims made in a reference answer can be substantiated by the retrieved context [Es+23]. Context Recall is quantified as the proportion of claims in the reference answer (S_{ref}) that are supported by the retrieved context ($S_{context}$), relative to the total number of claims in the reference answer (S_{ref}):

$$ContextRecall = \frac{|S_{context} \cap S_{ref}|}{|S_{ref}|}$$

This LLM-based metric requires the user input, reference answer, and retrieved contexts for computation. Utilizing the reference answer as a proxy for ideal context circumvents the labor-intensive process of annotating reference contexts directly [Oro+24]. Ideally, all claims in the reference answer should be traceable back to the retrieved context.

Context Precision with Reference

Employed when both retrieved contexts and a reference answer are available, this metric gauges the relevance of retrieved passages by comparing each chunk against the reference answer using an LLM. Higher scores reflect the retrieval of highly pertinent information essential for accurately addressing the query [Saa+24].

4.2.2 Generation-Specific Metrics

These metrics scrutinize the LLM's effectiveness in utilizing the provided retrieved context:

Faithfulness

Measuring the factual consistency between the generated response and the retrieved context, Faithfulness scores range from 0 to 1. A higher score indicates stronger alignment, meaning all claims within the response are supported by the context. The process involves identifying all claims made in the response, verifying each claim against the retrieved context, and calculating the score as the ratio of supported claims ($V_{supported}$) to the total number of claims (V_{total}):

$$Faithfulness = \frac{|V_{supported}|}{|V_{total}|}$$

Faithfulness is paramount in the medical domain, directly influencing the system's trustworthiness and utility [Es+23; Saa+24].

Response Groundedness

This metric assesses the extent to which a response is supported or "grounded" by the retrieved contexts, checking if each claim in the response can be located, wholly or partially, within the provided passages. Ratings are assigned on a scale where '0' denotes no grounding, '1' indicates partial grounding, and '2' signifies full grounding (every statement verifiable from the context). These ratings are normalized to a $[0, 1]$ scale for the final score. While conceptually similar to Faithfulness, Response Groundedness offers a more holistic assessment, providing a complementary perspective on generation quality [Saa+24; Wan+24c].

4.2.3 End-to-End Metrics

This category evaluates the overall system performance from the user's viewpoint:

Answer Accuracy

This metric quantifies the agreement between the model's generated response and a reference ground truth answer for a given question. Calculation involves two separate "LLM-as-a-judge" prompts, each assigning a rating of '0' (inaccurate/off-topic), '2' (partially aligned), or '4' (exact alignment) [Tru]. These ratings are converted to a $[0, 1]$ scale, and the average of the two scores is taken. Answer Accuracy provides a comprehensive measure, encompassing factual correctness, relevance, completeness, and overall helpfulness in addressing the user's information need [Oro+24].

4.2.4 Metric Implementation and Rationale

We used these six metrics utilizing the RAGAS framework [Es+23], leveraging its standardized implementations for evaluating RAG systems and employing LLM-based assessments where appropriate, following recommended procedures [Sim+24].

The selection of this specific suite of metrics was guided by three primary considerations. Firstly, it enables **component-level diagnosis**, allowing us to attribute performance bottlenecks to either inadequate retrieval (indicated by, for instance, low Context Recall) or problematic generation (such as low Faithfulness). Secondly, these metrics directly address **critical requirements for medical information systems**, emphasizing factual accuracy (via Faithfulness and Answer Accuracy) and the need for comprehensive information retrieval (measured by Context Recall) [Tha+21]. Finally, the inclusion of **user-centric evaluation** through end-to-end metrics like Answer Accuracy ensures that technical improvements correspond to tangible enhancements in the system's practical utility and user experience.

4.2.5 Panel of LLMs as Automated Judges

Building upon the metrics defined above, our evaluation framework implements these assessments through a panel of LLMs serving as automated judges. This approach operationalizes the metrics while providing standardized quality judgments across all evaluated systems.

Recent research has increasingly explored the potential of large language models (LLMs) to serve as automated 'judges' for evaluating the quality of machine-

generated text [GJ23; Zhe+23]. This approach leverages the sophisticated linguistic understanding and reasoning capabilities inherent in modern LLMs to assess various quality dimensions, offering a scalable alternative or supplement to human evaluation. Our evaluation framework employs LLMs specifically to implement the metrics described in the previous sections, analyzing generated responses against reference answers and assigning quantitative and qualitative quality scores consistent with our defined metrics.

To enhance the reliability and robustness of our automated evaluation process, we utilize a panel of multiple LLMs rather than relying on a single model as the sole judge. This decision is motivated by findings that individual LLMs, much like human annotators, can exhibit specific biases or inconsistencies in their judgments [Zhe+23]. By adopting a panel-based approach, we aim to reduce variance stemming from any single judge and mitigate the potential biases inherent to any particular model architecture or training data. In our setup, each LLM judge within the panel receives the identical prompt (detailed in Section 4.2.5) and generates an independent assessment. We then aggregate these verdicts using simple majority voting for categorical labels and the arithmetic mean for numerical scores, ensuring a more stable and representative final judgment.

Our choice to employ five distinct frontier-LLMs as judges was guided by research, such as work by Cohere [Ver+24], suggesting that diversity within the judge panel is crucial for achieving comprehensive and robust assessment. By selecting models from different development labs and LLM families (see Appendix C for details), we aimed to capture a wider spectrum of evaluation perspectives and minimize the risk of family-specific biases influencing the outcome. While research indicates that panels multiple models might offer optimal evaluation stability [Ver+24], we selected five models as a practical balance between maximizing evaluation robustness and managing computational resource utilization, and maintaiing diversity.

Quality Rubric

The evaluation rubric provided to each LLM judge defines three ordinal quality classes alongside a continuous score.

- **Good**: Assigned to responses that are fully accurate, complete, and directly supported by the provided context or align closely with the reference answer.
- **Acceptable**: Assigned to responses that contain only minor omissions or slight factual inaccuracies but remain fundamentally helpful and relevant to the query.

- **NotAcceptable**: Assigned to responses exhibiting significant factual errors, hallucinations, irrelevance, or a complete failure to address the user's query.

We established a fixed mapping from the continuous score $S \in [0, 1]$ to these labels: Good corresponds to scores $S \geq 0.80$, Acceptable to scores $0.50 \leq S < 0.80$, and NotAcceptable to scores $S < 0.50$.

Judge Prompt

Our evaluation framework employs a standardized prompt structure for all LLM judges, ensuring consistency in the task definition:

LLM Judge Prompt

[System] You are an impartial, meticulous judge of answer quality.
[User] You will receive: **Question**, **ReferenceAnswer**, **CandidateAnswer**.
Evaluation Criteria: 1. **Factual correctness:** Agrees with ReferenceAnswer. 2. **Completeness & relevance:** Covers key points of ReferenceAnswer, no irrelevant/contradictory material. 3. **Reasoning clarity (if present):** Sound reasoning.
Scoring & Labeling: Assign score $S \in [0, 1]$ and map to a label:

- $S \geq 0.80$: **Good** (Perfect to Very Good)
- $0.50 \leq S < 0.80$: **Acceptable** (Partly correct but flawed)
- $S < 0.50$: **NotAcceptable** (Major errors/irrelevant)

Output Format: Respond with exactly one line: `$$$ score: <S rounded to 2 decimals> Quality: <Label>`
Inputs: Question: {{question}} **ReferenceAnswer:** {{reference_answer}} **CandidateAnswer:** {{generated_answer}}

Aggregation and Variance Control

To derive a final quality assessment for each candidate answer, we aggregate the independent judgments from the panel. For the ordinal decision (Good, Acceptable, NotAcceptable), the class selected by the majority of the LLM judges becomes the final assigned label. This majority voting mechanism helps to smooth out potential outliers and reduce the impact of any single judge's bias, leading to more consistent evaluations. For the continuous score, we compute the arithmetic mean of the

scores assigned by all judges. This provides a fine-grained performance measure that captures nuances potentially lost in the broader categorical labels.

Implementation Details

The evaluation process is executed in parallel, with the standardized prompt dispatched concurrently to all five LLM judges. This parallelization significantly reduces the overall wall-clock time required for assessment without compromising the quality of the evaluation. In the infrequent scenario of a tie in the majority vote for the ordinal label (e.g., a 2-2-1 split across the three classes with five judges), we implement a tie-breaking rule by selecting the class corresponding to the highest arithmetic mean score among the tied categories. This panel-based judging scheme delivers repeatable, high-confidence quality judgments. By leveraging a diverse set of judges (detailed in Appendix C, including models and hyperparameters) and aggregating their assessments, we effectively mitigate over-reliance on any single model and achieve a robust evaluation framework that balances thoroughness with computational feasibility.

4.3 Evaluation Dataset

4.3.1 Dataset Construction

A specialized evaluation dataset was curated, comprising 90 question-answer pairs derived from a corpus of documents housed within an internal knowledge management system (roXtra). The selected corpus consisted of 10 German-language documents with a total length of approximately 400+ pages, containing domain-specific healthcare information rather than general content. This focused selection ensured our evaluation remained targeted toward specialized professional knowledge scenarios, distinguishing it from generic evaluations. Inspired by methodologies like the Giskard RAGET approach [Tea24a], we formulated diverse question types aimed at probing various facets of RAG system performance. The construction process involved analyzing the internal document corpus, defining distinct query categories relevant to clinical information needs, generating representative questions within each category, compiling comprehensive ground truth answers, and validating a subset of question-answer pair manually.

4.3.2 Question Categories and Rationale

To ensure a thorough evaluation covering different information needs and system capabilities, questions were developed across four primary categories:

Simple Questions

These queries seek straightforward factual information typically found within a single document or section. An example is: *"Wie sollte der Spontanatemversuch bei einem endotracheal intubierten oder tracheotomierten Patienten durchgeführt werden?"* (How should the spontaneous breathing trial be conducted for an endotracheally intubated or tracheotomized patient?). Their rationale is to test fundamental retrieval effectiveness and the system's ability to provide clear answers for standard clinical protocols.

Questions with Distracting Elements

Moving beyond simple factual queries, this category includes questions containing potentially irrelevant information designed to distract from the core information need. An illustrative example is: *"Wie wird der Rapid Shallow Breathing Index (RSBI) bei Patienten mit progredienten Erkrankungen und infauster Prognose verwendet, um Entscheidungen über die Therapiereduktion oder den Therapieverzicht zu unterstützen?"* (How is the Rapid Shallow Breathing Index (RSBI) used in patients with progressive diseases and poor prognosis to support decisions about therapy reduction or withdrawal?).

The rationale for this question type is twofold. Firstly, it tests the system's ability to discern and focus on the pertinent aspects of the query (the use of RSBI for therapy decisions) while ignoring potentially distracting clinical details (like 'progredienten Erkrankungen und infauster Prognose'), a crucial skill in real-world settings where queries may contain extraneous information [Tea24a]. Secondly, this example deliberately includes the English phrase 'Rapid Shallow Breathing Index (RSBI)', highlighting the challenge of code-mixing. This reflects the linguistic reality of the medical corpora underpinning this work, which, while predominantly German, contain numerous English medical terms and acronyms frequently used untranslated in German clinical practice and documentation. Other languages, such as Arabic or Turkish, were minimally present in the core clinical texts and typically confined to specific patient information brochures. Consequently, these questions assess the RAG system's robustness not only to contextual distractors but also to the common occurrence of key medical concepts being expressed in English within German questions.

Complex Questions

Requiring synthesis of information from multiple documents or sections, these questions often involve understanding nuanced relationships between medical, ethical, or legal concepts. An example is: *"Unter welchen Bedingungen ist eine Patientenverfügung, die ein Patient im Zustand der Entscheidungsfähigkeit ausgefertigt hat, für den behandelnden Arzt verbindlich?"* (Under what conditions is an advance directive, prepared by a patient with decision-making capacity, binding for the treating physician?). These questions assess the system's capability to integrate disparate information and generate comprehensive answers to multifaceted inquiries.

Out-of-Scope Questions

These queries request information not contained within the provided knowledge base, testing the system's ability to recognize its limitations and avoid hallucination [Mag+24]. An example: *"Wie viele Jahre Erfahrung hat Dr. Müller in der Intensivmedizin, bevor er zur Therapiereduktion beiträgt?"* (How many years of experience does Dr. Müller have in intensive care medicine before contributing to therapy reduction?). Evaluating the response to such questions assesses a critical safety feature: the system's capacity to appropriately state when reliable information cannot be provided.

4.3.3 Corpus and Testset Characteristics

The evaluation corpus comprised 10 documents sourced from an internal knowledge management system (roXtra), totaling approximately 400 pages of domain-specific German-language content. This collection represented a diverse range of materials typical in German hospital environments, including medical guidelines, procedural protocols, regulatory information, and clinical reference texts. These documents were characterized by technical medical terminology specific to German healthcare, complex compound words prevalent in German medical writing formal regulatory language, and a mixture of general guidelines and specialized protocols, often requiring information synthesis across sections for complete understanding.

From this corpus, the test dataset of 90 question-answer pairs was derived. It was carefully constructed to mirror realistic information needs while ensuring balanced coverage across the four question categories (approximately 20-25 questions per category). Each question was paired with a crafted reference answer, crafted from the source. This deliberate distribution ensures our evaluation thoroughly probes RAG system performance across various query types and complexities, from basic retrieval to complex reasoning and handling of out-of-domain requests.

4.4 Qualitative Evaluation Methodology

Complementing the quantitative metrics, our qualitative evaluation examined the best-performing RAG system alongside the incumbent roXtra keyword-based search system [Sch16] within realistic hospital information retrieval scenarios. This involved executing identical queries on both systems and documenting the search results and response characteristics (such as retrieved documents, generated summaries, and presentation format). While our RAG system analysis was comprehensive, the comparison with roXtra was targeted and focused on selected representative examples rather than exhaustive. We assessed performance across dimensions like retrieval effectiveness, information synthesis capability, creating comparative analysis matrices to highlight key differences. Though limited in scope, this targeted comparison nevertheless revealed clear patterns that demonstrate practical improvements in information accessibility and usability offered by the RAG system compared to the traditional keyword search [KTS16].

4.5 Experimental Design

Our experimental setup involved the evaluation of four distinct RAG architectures, each configured with variations in key pipeline components (e.g., different retrieval strategies or generation models). For detailed specifications of these architectures and their configurations, please refer to Section 5.7, which provides comprehensive information about each implementation variant.

The performance of each architecture was systematically assessed using the described quantitative metrics (Context Relevance, Recall, Precision; Faithfulness, Groundedness; Answer Accuracy), subjected to the LLM panel assessment for quality categorization ("Good," "Acceptable," "Not Acceptable") and scoring, and further analyzed through the qualitative comparison of the optimal RAG configuration against the baseline roXtra system. This multi-pronged approach allows for a robust identification of the most effective RAG configuration for German hospital documentation while rigorously quantifying its advantages over the existing keyword-based search paradigm.

4.6 Summary

In summary, our evaluation framework provides a robust and comprehensive methodology for assessing RAG architectures tailored to German hospital documentation. It integrates quantitative metrics targeting specific pipeline components

and overall performance, employs an LLM jury panel for consistent quality judgment, and incorporates qualitative analysis for real-world system comparison. By utilizing a purpose-built, diverse German-language medical testset encompassing simple, complex, distracting, and out-of-scope questions, we ensure the evaluation addresses the full spectrum of information needs encountered in clinical settings. This multi-faceted approach—combining component-specific metrics, jury-based quality assessment, and direct system comparison—enables us to draw well-supported conclusions regarding the optimal RAG configuration for German medical information retrieval and provides tangible evidence of its practical benefits over traditional keyword search methods.

5 System Design and Implementation

5.1 Introduction

This chapter details the implementation of a Retrieval Augmented Generation (RAG) system [Lew+21] designed for the German medical domain. The system's architecture directly addresses two critical requirements: (1) ensuring data privacy through on-premises deployment, and (2) providing accurate information retrieval and generation capabilities for German medical content. The implementation employs containerized, open-source components configured to operate entirely within on-premise infrastructure, thereby ensuring consistency and scalability.

The RAG implementation serves as a technical foundation for investigating three specific research questions: the impact of language-specialized embedding models (RQ1), the effectiveness of hybrid retrieval strategies (RQ2), and the performance-efficiency tradeoffs between dense and Mixture-of-Experts LLM architectures (RQ3). All components were selected to facilitate controlled experimentation while maintaining consistent deployment parameters.

5.2 System Architecture Overview

The system architecture, depicted in Figure 5.1, consists of four containerized modules that collectively implement a complete RAG pipeline:

1. **Compute Layer (A)** provides GPU-accelerated inference for both text embedding generation and LLM response synthesis.

2. **Document Cloud (B)** securely stores the medical document corpus.

3. **Ingestion Container (C)** processes documents, generates embeddings, maintains the vector database, and handles query processing. This container also implements the Citation Manager, which ensures proper source attribution in generated responses.

4. **Front-end Container (D)** delivers the user interface.

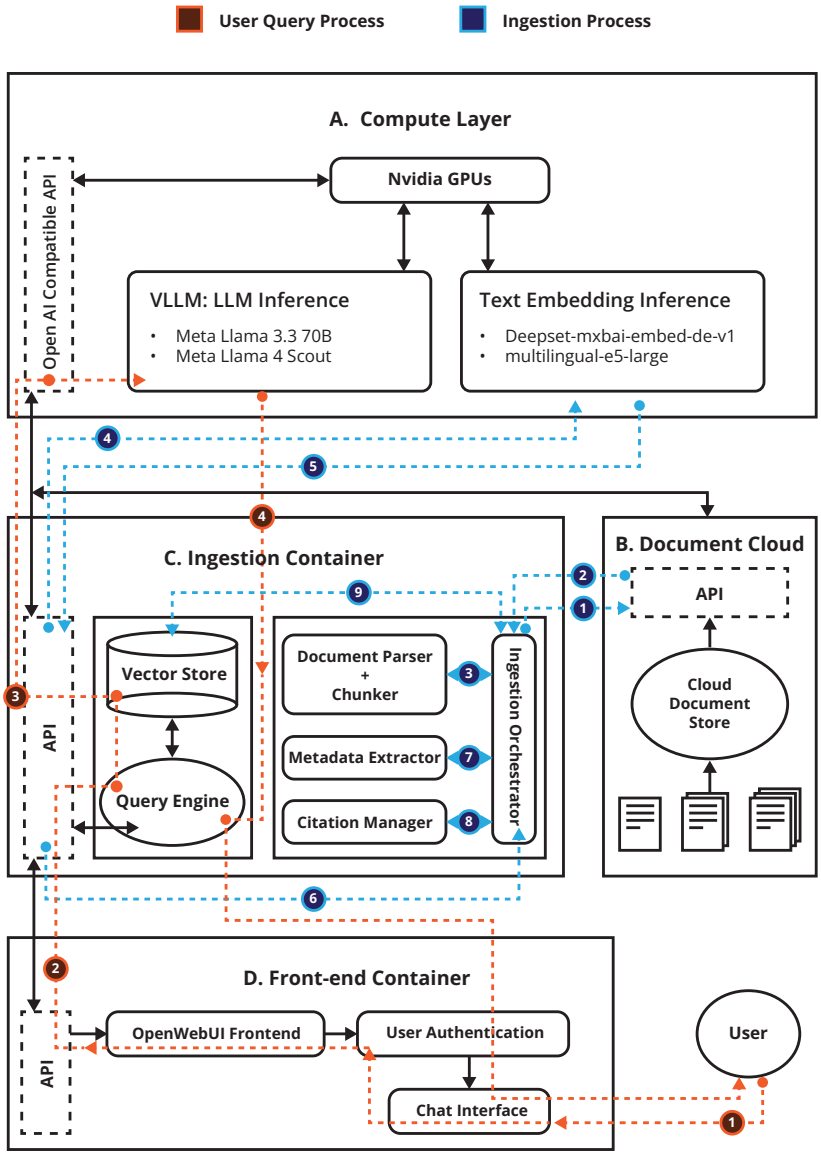


Figure 5.1: On-premises system architecture illustrating the core containerized modules, the data-ingestion pipeline, and the user-interaction flow

The information flow begins with source documents in the Document Cloud (B), which are processed by the Ingestion Container (C). This processing includes parsing, chunking, metadata extraction, and embedding generation (with computation delegated to the Compute Layer). The resulting embeddings and metadata are indexed in a vector database within the Ingestion Container. When a user submits a query via the Front-end (D), the Ingestion Container retrieves relevant document chunks and passes them, along with the query, to the Compute Layer for LLM inference. The generated response, including citations to source documents, is then presented to the user through the Front-end interface.

This architecture ensures all data processing occurs entirely within the on-premise network. No internal documents, queries, or responses are transmitted to external services, thereby maintaining data privacy and regulatory compliance.

5.3 Compute Layer Implementation

The Compute Layer centralizes all GPU-accelerated inference, providing two primary services: text embedding generation and LLM inference.

The layer utilizes NVIDIA GPUs with CUDA [SK10] and cuDNN [Che+14] optimizations for efficient neural network execution. Resources are shared between embedding and LLM processes through container-level resource allocation.

5.3.1 Text Embedding Implementation

The embedding service [Hug23] converts text chunks and queries into high-dimensional vector representations. Two distinct embedding models are implemented:

1. `deepset-mxbai-embed-de-large-v1`: A bilingual model optimized specifically for German language content [AI24], generating 1024-dimensional embeddings.

2. `intfloat/multilingual-e5-large`: A general-purpose multilingual model [Wan+24a] based on XLM-RoBERTa [Pry24], processing approximately 100 languages including German and generating 1024-dimensional embeddings.

Both models process input sequences up to 512 tokens. When using the E5-based model, the implementation adds required prefixes ("passage:" for document chunks, "query:" for user queries) according to the model's training specification [Wan+24a]. The service exposes a REST API for embedding generation requests.

5.3.2 LLM Inference Implementation

The LLM inference service generates responses based on user queries and retrieved context. Two LLMs are implemented for experimental comparison:

1. `Llama-3.3-70B-Instruct`: A dense transformer model with 70 billion parameters [Gra+24], incorporating Grouped Query Attention [FZS22] for inference optimization. This model utilizes all 70B parameters for each token processed and supports a context window of 128K tokens.

2. `Llama-4-Scout-17B-16E`: An MoE model with approximately 17B active parameters per token out of 109B total parameters [AI22]. This architecture activates only a subset of parameters during inference through sparse expert routing and supports a context window of up to 10M tokens.

Both models are deployed using vLLM [WW24], a high-throughput serving engine that implements PagedAttention for memory-efficient inference. The vLLM configuration provides an OpenAI-compatible API [Ope20] to standardize access across both models, simplifying experimental comparisons.

5.4 Document Cloud Implementation

The Document Cloud provides secure, private storage for the medical corpus. This component uses Seafile [Tei+19], an open-source document management system deployed on the local network. Seafile was selected for its self-hosting capabilities and comprehensive REST API, which enables programmatic document access and version tracking.

Document retrieval is implemented through two primary API interactions: (1) listing directory contents to discover available documents, and (2) obtaining temporary download links for specific files. Both operations are authenticated using repository-specific tokens. This API-driven approach decouples the RAG pipeline from storage implementation details, enabling automated document discovery and retrieval without exposing internal storage mechanisms.

5.5 Ingestion Container Implementation

The Ingestion Container implements the document processing pipeline, vector database, citation manager, and query engine. It coordinates the transformation of raw documents into retrievable vector representations and ensures proper source attribution in generated responses.

5.5.1 Document Processing Pipeline

The document processing workflow is managed by a Python service that orchestrates several processing stages:

Document Parsing: PDF documents are processed using PyMuPDF [Art25], which extracts text while preserving structural information such as tables and layouts. This implementation leverages PyMuPDF's C-based architecture for efficient processing.

Text Chunking: The extracted text is segmented using Recursive Character Text Splitting with a chunk size of 512 tokens and 50-token overlap. This approach splits text along natural boundaries (paragraphs, sentences) before resorting to character-level splits, helping to preserve semantic coherence [Wan+24c]. The 512-token size aligns with embedding model input limits, while the overlap helps maintain context across adjacent chunks.

Metadata Management: The implementation extracts two types of metadata: (1) citation information mapping each chunk to its source document and page number, and (2) document metadata including titles, authors, and dates. These are stored alongside embeddings to enable filtered retrieval and citation generation.

5.5.2 Vector Database Implementation

Vector storage and retrieval are implemented using ChromaDB [Chr23], an open-source vector database optimized for RAG applications. The implementation stores document chunks together with their embeddings and associated metadata in collections organized by embedding model type.

For similarity search, the implementation uses the Hierarchical Navigable Small World algorithm [WFG24] with cosine similarity as the distance metric. This provides approximate nearest-neighbor search with logarithmic complexity, enabling efficient retrieval at scale. The implementation also supports combined queries using both vector similarity and metadata filters, allowing retrieval refinement based on document attributes.

5.5.3 Citation Manager and Answer Grounding

The Citation Manager component ensures generated responses integrate verifiable references directly within the text through inline, clickable Markdown links to Seafile source documents. This approach embeds citations seamlessly into the response text, eliminating the need for separate footnotes or reference sections.

Seafile URL Generation

During document ingestion, the system leverages Seafile's REST API (v2.1) [Zho+14] to generate permanent, authenticated URLs for each source document. These URLs follow Seafile's standard structure:

"' https://[seafile-instance]/lib/[repository-id]/file/[path-to-file] "'

For documents containing distinct sections, the system generates section-specific URLs using fragment identifiers:

"' https://[seafile-instance]/lib/[repository-id]/file/[path-to-file]#[section-id] "'

To accommodate varying authentication scenarios, the implementation dynamically switches between standard repository URLs that utilize the user's existing Seafile session and temporary share links with configured expiration periods for broader access [Tei+19]. This flexibility ensures that citations remain accessible to authorized users while maintaining document security in accordance with medical information handling requirements [Sch20].

The system maintains a comprehensive mapping between document chunks and their corresponding Seafile URLs, enabling precise inline citation of specific document sections. This granular approach to citation is particularly important in the medical domain, where claiming information from a specific section of clinical guidelines or research papers requires precise attribution [Zho+24a].

Inline Citation Prompt Engineering

The Citation Manager employs specialized prompt templates that instruct the LLM to embed Seafile URLs directly within the response text as Markdown links. The primary citation prompt emphasizes inline citations:

Inline Seafile Citation Prompt

You are a precise research assistant that provides verifiable information. Answer the query based SOLELY on the provided Seafile sources.
CRITICAL INSTRUCTIONS FOR CITATIONS: - Embed citations INLINE as clickable Markdown links - Format: [descriptive text](seafile-url) - Example: "According to [medical guidelines](seafile.example.org/lib/abc123/file/guidelines.pdf), the recommendation..." - Citations must be directly embedded within your sentences - EVERY factual claim must have an embedded citation - DO NOT add separate footnotes or reference sections - If sources contradict, acknowledge this with multiple embedded citations - If no sources provide relevant information, state this clearly
Sources: {context_str}
Query: {query_str} Answer:

The refinement prompt maintains this inline citation approach when integrating additional information:

Inline Citation Refinement Prompt

You are refining an answer while maintaining all inline Seafile citations.
Existing answer: {existing_answer}
Additional Seafile sources to incorporate: {context_msg}
Refine the answer by integrating new sources with these instructions: - Preserve ALL existing inline Markdown citations - Add new information with inline citations in the format [text](seafile-url) - Maintain seamless integration of citations within sentences - DO NOT add footnotes or separate reference sections - If new sources contradict existing information, include multiple embedded citations - Preserve the overall flow and readability of the text
Query: {query_str} Refined answer:

This prompt engineering approach is based on findings from research on citation effectiveness [Mes23] and ensures that citations are embedded naturally within the text flow, enhancing readability while maintaining direct access to source documents.

Markdown URL Embedding

The system converts Seafile URLs into properly formatted Markdown links before presenting them to the LLM. This preprocessing includes URL encoding of special characters in file paths, generation of descriptive anchor text based on document metadata, and standardization of URL formats for consistent presentation.

When embedding citations, the system preferentially uses semantically meaningful anchor text derived from document metadata (e.g., document title, section heading) rather than generic citation numbers. For example:

“‘ According to [Clinical Guidelines for Hypertension Management](seafile.example.org/lib/abc123/file/guidelines.pdf#hypertension), patients should... “‘

This approach maintains natural reading flow while providing clear attribution to specific sources, which has been shown to increase user trust in generated medical information.

The inline citation approach provides a user experience where verification becomes a natural extension of reading rather than a separate task, enhancing information trustworthiness while maintaining text coherence and readability [Dey+19]. This is particularly valuable in the medical domain, where information verification is critical for clinical decision-making.

5.5.4 Query Engine Implementation

The Query Engine component processes user queries and orchestrates the retrieval-generation sequence. The implementation performs several functions:

1. Query preprocessing, including adding model-specific prefixes for embedding generation. 2. Vector similarity search against the indexed document chunks. 3. For hybrid retrieval configurations, a parallel keyword-based search using BM25 [WC11]. 4. Result fusion using Reciprocal Rank Fusion [Wan+24b] when multiple retrieval methods are employed. 5. Context assembly, collecting and ordering retrieved chunks. 6. Prompt construction, formatting the query and context according to the LLM’s expected input structure, incorporating the citation instructions detailed in Section 5.5.3.

The Query Engine communicates with the Compute Layer via REST APIs, sending embedding requests and LLM inference requests. Responses from the LLM that include inline Seafile citations are delivered to the Front-end Container for presentation to the user.

5.6 Front-end Implementation

The Front-end Container implements the user interface using OpenWebUI [Tea24b], an open-source chat interface for LLM interactions. This component is configured to operate entirely within the local network, ensuring user interactions remain private.

The implementation provides a chat interface that supports conversation history, markdown-formatted responses, and clickable inline citations. Backend communication is implemented through REST APIs to the Query Engine. Authentication is handled locally, with no external service dependencies.

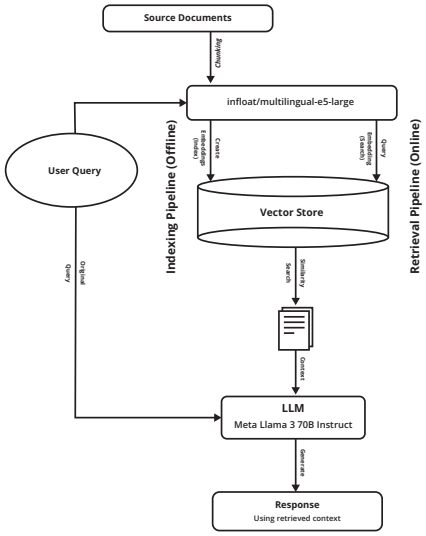
The OpenWebUI implementation was modified to properly render Markdown links as clickable elements, enabling users to directly access source documents in Seafile when verifying information. This seamless citation access is critical for eneterprise applications where response validation is essential [Dey+19].

5.7 Experimental RAG Configurations

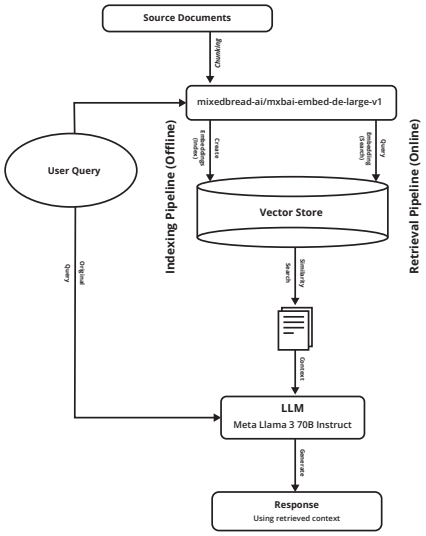
Using the implemented components, four distinct RAG configurations were created to address the research questions outlined in Chapter 2. Each configuration represents a specific combination of embedding models, retrieval strategies, and generator architectures, as illustrated in Figure 5.2.

Table 5.1: Component specifications for each configuration.

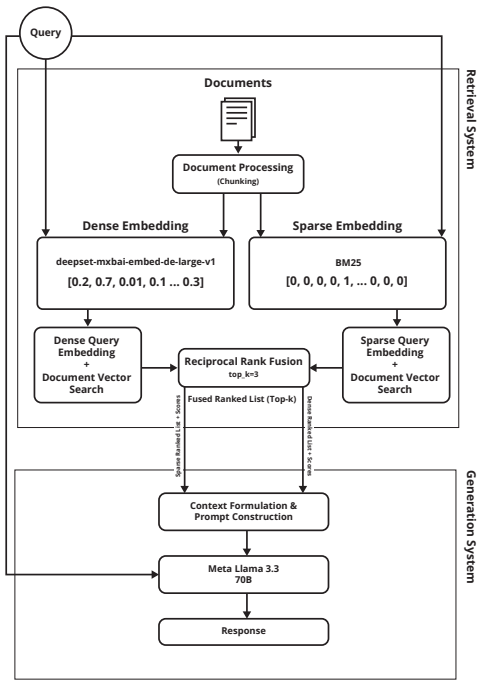
Component	Configuration 1	Configuration 2	Configuration 3	Configuration 4
Embedding	`multilingual-e5-large`	`mxbai-embed-de-large-v1`	`mxbai-embed-de-large-v1`	`mxbai-embed-de-large-v1`
Retrieval	Dense vector	Dense vector	Hybrid (Dense + BM25)	Hybrid (Dense + BM25)
Fusion	N/A	N/A	RRF	RRF
Generator	Llama-3.3-70B-Instruct	Llama-3.3-70B-Instruct	Llama-3.3-70B-Instruct	Llama-4-Scout-17B-16E
Research Focus	Baseline (RQ1)	Language specialization (RQ1)	Retrieval strategy (RQ2)	Generator architecture (RQ3)



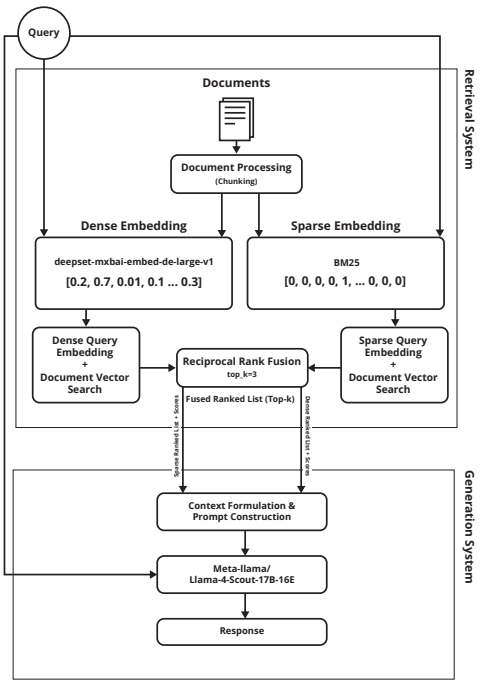
(a) Configuration 1: Multilingual Base RAG using general-purpose embeddings with dense retrieval.



(b) Configuration 2: German-Optimized RAG using specialized German embeddings with dense retrieval.



(c) Configuration 3: Hybrid Retrieval RAG combining dense and sparse retrieval methods.



(d) Configuration 4: MoE Generator RAG using mixture-of-experts architecture.

Figure 5.2: Four RAG configurations implemented and evaluated in this research.

Each configuration was designed to isolate specific variables for experimental comparison, as detailed in Table 5.1. All configurations maintain consistent citation management and user interface components.

5.7.1 Configuration Analysis

Configuration 1: Multilingual Base RAG (Figure 5.2 (a)) establishes the baseline using a general-purpose multilingual embedding model. This configuration employs the `intfloat/multilingual-e5-large` model with conventional dense vector retrieval and the Llama-3.3-70B-Instruct generator. It serves as the comparison point for evaluating language-specialized embeddings.

Configuration 2: German-Optimized RAG (Figure 5.2 (b)) introduces language specialization by replacing only the embedding model with `deepset-mxbai-embed-de-large-v1`, which is optimized for German language content. All other components remain identical to Configuration 1, enabling direct comparison of the impact of language-specialized versus multilingual embeddings (RQ1).

Configuration 3: Hybrid Retrieval RAG (Figure 5.2 (c)) builds upon Configuration 2 by introducing hybrid retrieval. It combines dense vector search with the lexical BM25 algorithm [WC11], integrating results through Reciprocal Rank Fusion [CCB09]. This configuration addresses RQ2 by evaluating whether hybrid retrieval improves upon the German-optimized dense retrieval approach.

Configuration 4: MoE Generator RAG (Figure 5.2 (d)) maintains the hybrid retrieval mechanism from Configuration 3 but replaces the dense Llama-3.3-70B-Instruct model with the MoE-based Llama-4-Scout-17B-16E. This final configuration addresses RQ3 by evaluating whether MoE architectures offer efficiency advantages without compromising response quality.

5.7.2 Generator Model Comparison

A central aspect of our experimental design is the comparison between dense and MoE generator architectures. Table 5.2 summarizes the key differences between these models.

The key efficiency trade-off between these models is that the MoE architecture activates fewer parameters per token (17B vs. 70B), potentially lowering computational requirements at inference time [FZS22]. This is particularly relevant for on-premises deployments where GPU resources may be constrained. While the MoE model has a larger total parameter count (109B), only a subset of experts is activated for each token, creating a favorable computation-to-capability ratio.

Table 5.2: Key characteristics of dense and MoE generator models used.

Feature	Llama-3.3-70B-Instruct	Llama-4-Scout-17B-16E
Architecture	Dense Transformer	Mixture of Experts Transformer
Active parameters	70 billion	17 billion
Total parameters	70 billion	109 billion (16 experts)
Context window	128K	10M tokens
Inference efficiency	Grouped Query Attention	Sparse Expert Activation
Key advantage	Higher per-token compute	Lower inference cost, larger context
Potential limitation	Higher resource requirements	Routing overhead, fewer active parameters

Though both models support large context windows, the RAG approach primarily relies on retrieved context rather than the model's internal context length. Nevertheless, the larger context capacity of the MoE architecture could enable future extensions of the system to incorporate more comprehensive document context when needed.

5.8 Summary

This chapter has detailed the implementation of an on-premises RAG system for the German medical domain. The containerized architecture ensures consistency and reproducibility while enabling systematic experimentation with different component configurations. The system integrates specialized components for each phase of the RAG pipeline, from document ingestion to response generation with verifiable citations.

A key contribution of this implementation is the Citation Manager, which embeds clickable Seafile URLs directly within generated responses, enabling seamless verification of information sources. This approach is particularly valuable in the medical domain, where traceability to authoritative sources is essential for trust and clinical utility.

Four specific RAG configurations have been implemented to investigate German-specialized embeddings (RQ1), hybrid retrieval strategies (RQ2), and MoE versus dense generator architectures (RQ3). These configurations maintain consistent citation and verification capabilities while varying specific components to address the research questions.

The implementation uses open-source components throughout, including Seafile for document storage, PyMuPDF for document parsing, ChromaDB for vector storage, vLLM for model serving, and OpenWebUI for the user interface. This approach ensures reproducibility while maintaining full control over data flow and processing. The system provides the technical foundation for the empirical evaluations presented in subsequent chapters.

6 Results

This chapter details the performance evaluation of the four Retrieval-Augmented Generation (RAG) [Lew+21] configurations investigated in this thesis. We begin with a comparative overview before delving into the specific findings related to each research question.

6.1 Overview of RAG Architecture Performance

The evaluation revealed significant variations in performance across the four RAG architectures along multiple dimensions. As illustrated in Figure 6.1, the comparative metrics assessment (Context Recall, Faithfulness, etc.) demonstrates distinct performance profiles for each architecture. The MoE and Hybrid architectures achieved the highest performance levels. Both architectures attained identical jury scores of 0.85 on a 0–1 scale as shown in Figure 6.2, likely attributable to their shared hybrid retrieval mechanism [BGI24], a concept discussed in work such as Bruch et al. [BGI24].

Following the top performers, the Multilingual architecture registered a jury score of 0.78. The Bilingual architecture scored lowest at 0.72.

To provide a more nuanced understanding of these performance differences, we also examined the distribution of quality assessments assigned by the LLM jury panel [Ver+24] to each architecture's responses. These quality categorization assessments (Figure 6.3) further substantiated the overall performance findings. The MoE architecture yielded the highest percentage of responses rated as "Good" (87.91%), marginally exceeding the Hybrid architecture (86.81%). Both significantly outperformed the Multilingual (80.22%) and Bilingual (73.33%) implementations. Notably, the Bilingual architecture produced the highest rate of "Not Acceptable" responses (23.33%), more than double the rate observed in the leading MoE architecture (9.89%).

A detailed comparison across six key metrics (precision with reference, context recall, faithfulness, answer accuracy, response groundedness, and context relevance, drawing from the RAGAS framework [Es+23]) is presented in Table 6.1 and visualized in the radar chart in Figure 6.1.

Figure 6.1 highlights the performance profiles, showing similar patterns of

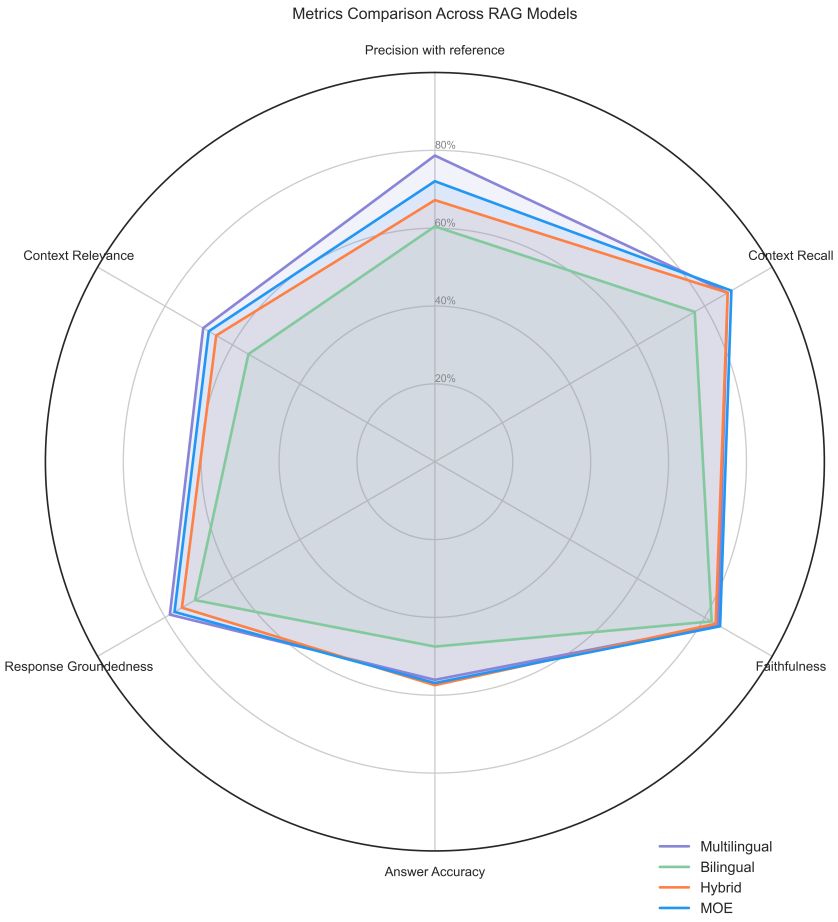


Figure 6.1: Metrics Comparison Across RAG Models (Radar Chart)

Table 6.1: Detailed Metric Comparison Across Architectures (%)

Architecture	Precision with Reference	Context Recall	Faithfulness	Answer Accuracy	Response Groundedness	Context Relevance
MoE	**77.3**	68.4	**84.5**	**72.1**	79.8	82.5
Hybrid	75.8	69.2	83.0	71.5	**81.2**	**83.6**
Multilingual	75.1	**74.7**	83.9	68.4	76.3	79.2
Bilingual	68.2	65.3	82.0	63.7	72.4	74.8

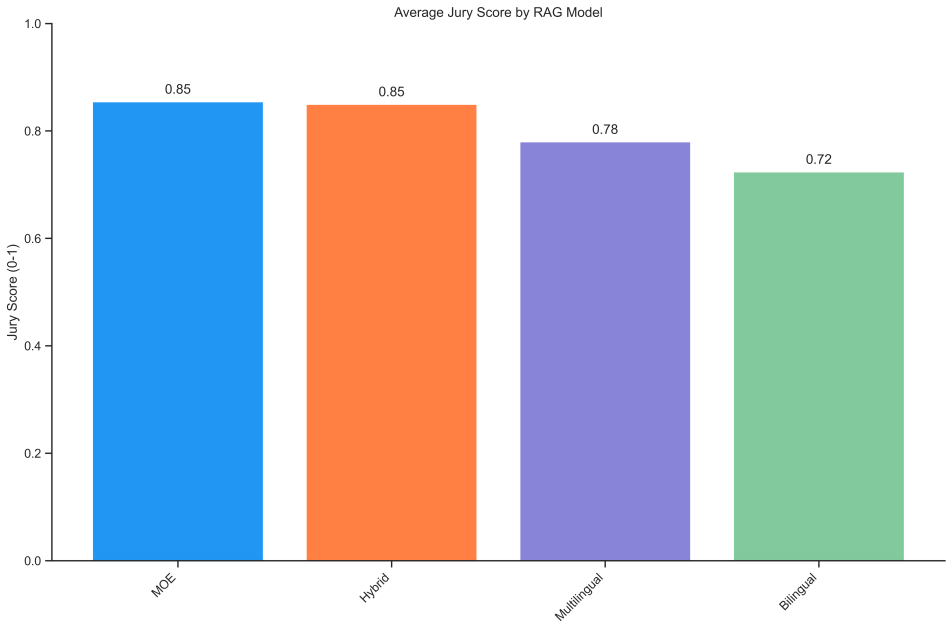


Figure 6.2: Average Jury Score by RAG Configuration

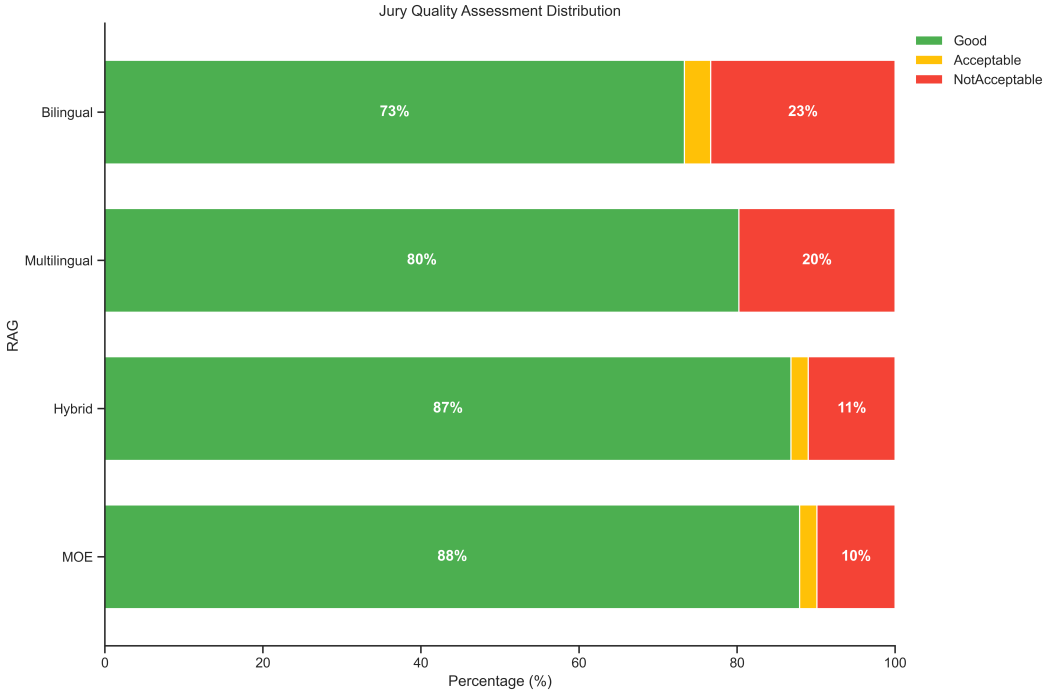


Figure 6.3: Jury Quality Assessment Distribution

strength for the MoE and Hybrid architectures, particularly in context relevance and response groundedness. In contrast, the Multilingual architecture demonstrated superior context recall compared to the other implementations.

Having established this general performance landscape, we now turn to addressing each of our research questions in detail.

6.2 Research Question 1: Impact of Embedding Models on domain specific documents in German

Our first research question investigated how the choice between a general multilingual embedding model (`intfloat/multilingual-e5-large` [Wan+24a]) and a German-language optimized bilingual model (`mixedbread-ai/deepset-mxbai-embed-de-large-v1` [mix23]) influences retrieval effectiveness and generation quality for domain specific documents in German.

Contrary to initial expectations, the results showed meaningful performance variations favoring the Multilingual architecture. The `intfloat/multilingual-e5-large` embedding model outperformed the German-optimized `mixedbread-ai/deepset-mxbai-embed-de-large-v1` model (Bilingual architecture) on most key metrics. The initial hypothesis favoring German-optimized embeddings was based on prior research suggesting language-specific embeddings should better capture nuances in specialized domains [VPJ22].

Specifically, the Multilingual architecture achieved a jury score 8.3% higher than the Bilingual architecture (0.78 vs. 0.72, see Figure 6.2), and delivered 9.4% more responses categorized as "Good" (80.22% vs. 73.33%, see Figure 6.3). This performance advantage materialized despite the hypothesis that German-optimized embeddings might yield superior results for the predominantly German-language hospital documents.

Analysis of the detailed metrics in Figure 6.1 reveals that the Multilingual architecture particularly excelled in context recall (74.7% vs. 65.3% for Bilingual) and precision with reference (75.1% vs. 68.2%). This suggests that the broader linguistic capabilities of the multilingual embedding model [TR16] may have provided more effective document representation for retrieval, even within the specialized German medical domain.

When examining performance by question type, the Multilingual architecture demonstrated notable strength with complex questions, achieving a jury score of 0.92 compared to the Bilingual architecture's 0.74—a significant 24.3% improvement.

This finding indicates that the general multilingual embedding model might be better equipped to capture the semantic relationships inherent in complex medical queries [TR16].

A contributing factor to the multilingual model's superior performance may be its enhanced ability to handle code-mixing—the inclusion of English medical terminology within predominantly German text. As noted in Section 4.3.2, the German medical corpus contains numerous untranslated English medical terms and acronyms, which the multilingual model appears better equipped to process semantically.

6.3 Research Question 2: Effectiveness of Hybrid Retrieval Strategies

Moving from embedding models to retrieval methods, our second research question examined the extent to which incorporating hybrid search (combining dense vectors with BM25 [WC11] sparse vectors) [SMS24a] improves retrieval effectiveness and enhances the reliability of generated outputs compared to standard vector search alone [Wan+24c].

The results demonstrate clear advantages for the hybrid approach [BGI24]. The Hybrid architecture, combining dense embeddings with BM25 sparse search (weighted 0.7 for vector similarity, 0.3 for BM25, substantially outperformed both the Multilingual and Bilingual architectures (which relied solely on vector search).

As shown in Figure 6.2, the Hybrid architecture's jury score of 0.85 represents improvements of 9.0% over the Multilingual (0.78) and 18.1% over the Bilingual (0.72) systems.

The quality distribution assessment (Figure 6.3) revealed even more pronounced differences in reliability. The Hybrid architecture delivered "Good" quality responses in 86.81% of cases, compared to 80.22% for Multilingual (+8.2% improvement) and 73.33% for Bilingual (+18.4% improvement). Concurrently, the rate of "Not Acceptable" responses was substantially reduced in the Hybrid architecture (10.99%) compared to Multilingual (19.78%) and Bilingual (23.33%), indicating significantly increased reliability.

Insights from the radar chart (Figure 6.1) show the Hybrid architecture's particular strengths in context relevance (83.6%), response groundedness (81.2%), and answer accuracy (71.5%). This suggests that the synergy between dense and sparse retrieval methods provides more robust context selection [Wan+24c], enabling the generator model to produce more accurate and well-supported responses.

Interestingly, while the Hybrid approach moderately improved context recall compared to the Bilingual model (69.2% vs 65.3%, see Table 6.1), it did not fully match the recall capabilities of the Multilingual architecture (74.7%). This indicates a potential trade-off: hybrid retrieval enhances precision and relevance, but the broader language coverage of multilingual embeddings might still offer advantages for maximizing the retrieval of all potentially relevant information chunks [QTB24].

Across different question types, the Hybrid architecture maintained strong and consistent performance, with jury scores never falling below 0.82 for any category, demonstrating its robustness for handling diverse medical queries.

6.4 Research Question 3: MoE vs. Dense Generator Models

Our third research question investigated whether employing a Large Language Model (LLM) utilizing a Mixture-of-Experts (MoE) architecture (`Llama 4 Scout` [AI22]) yields considerable improvements in key generation metrics compared to using a general instruction-following dense LLM (`Llama-3.3-70B-Instruct` [Gra+24]), when both are paired with the same hybrid retrieval system.

The Mixture-of-Experts architecture, an approach that routes different input types to specialized sub-networks or "experts" within the model [FZS22], has gained attention for potentially improving efficiency and task-specific performance. MoE models dynamically activate only a small subset of parameters for any given input, potentially allowing for more specialized processing pathways.

The results demonstrate certain advantages for the MoE architecture regarding output quality. While achieving an equivalent overall jury score (0.85) to the Hybrid architecture (which used the dense Llama 3.3 model, see Figure 6.2), the MoE configuration delivered several notable improvements in specific generation-focused areas.

In terms of quality categorization (Figure 6.3), the MoE model slightly outperformed the dense implementation, generating 87.91% “Good” responses versus 86.81%, coupled with a marginally lower “Not Acceptable” rate (9.89% vs. 10.99%). Although these differences are small, they suggest the MoE architecture may provide modest gains in overall response quality and reliability.

Furthermore, the MoE architecture achieved the highest average faithfulness score (84.5%) among all tested configurations (Table 6.1), compared to 83.0% for the dense model in the Hybrid setup. This suggests that the specialized expert pathways activated within the MoE model might be particularly effective at maintaining factual consistency between the generated response and the retrieved context.

Examining performance by question type revealed MoE's strong results, particularly for complex questions (jury score 0.91 vs. 0.90 for the dense model), implying that MoE's specialized pathways could benefit synthesis and complex reasoning tasks. The MoE model also exhibited less variability across question types—for example, scoring 0.82 for distracting-element questions versus 0.84 for dense, and 0.83 for simple questions versus 0.82 for dense—suggesting more consistent performance regardless of query structure.

6.5 Research Question 4: Optimal RAG Configuration for German Hospital Documentation

Considering both retrieval effectiveness and generation quality, our analysis aimed to identify which RAG configuration demonstrated the most advantageous overall performance profile for retrieving information from German hospital documentation.

Based on the comprehensive evaluation across multiple quality metrics, the **MoE architecture emerges as the optimal configuration** identified in this study. By combining hybrid retrieval (dense embeddings + BM25, weighted 0.7/0.3) with the `Llama 4 Scout` MoE generator model, this setup achieved the best overall balance of performance.

Key indicators supporting this conclusion include:

- The MoE architecture delivered the highest percentage of "Good" quality assessments at 87.91% (Figure 6.3), marginally outperforming the Hybrid setup (86.81%) and substantially exceeding the Multilingual (80.22%) and Bilingual (73.33%) configurations.
- Simultaneously, it maintained the lowest rate of "Not Acceptable" responses at 9.89%, less than half the rate observed in the Bilingual architecture (23.33%).
- The radar chart (Figure 6.1) illustrates the MoE configuration's well-rounded profile, exhibiting strong performance across all measured metrics. While the Multilingual architecture had slightly better context recall and the Hybrid architecture slightly better context relevance, the MoE system achieved the most robust combined performance.

Performance analysis by question type further reinforced the MoE architecture's suitability. Our evaluation revealed that MoE's jury scores ranged consistently

high, from 0.82 (distracting elements) to 0.91 (complex). For complex questions, it achieved a 95.45% "Good" response rate with only 4.55% deemed "Not Acceptable." Notably, even for challenging out-of-scope questions, it maintained an 86.36% "Good" rate (compared to 68.18% for Multilingual), highlighting its capability to effectively recognize information boundaries and avoid hallucination, a known strength of MoE implementations [Has25] related to approaches like DeepSpeed-MoE [Raj+22].

These empirical findings align with recent research suggesting that MoE architectures may offer particular advantages in domain-specific applications and in settings requiring high precision and factual reliability.

6.6 Research Question 5: Qualitative Comparison with Keyword-Based Search

Having identified the MoE architecture as the optimal RAG configuration in our tests, our fifth research question examined how this advanced system qualitatively outperforms the incumbent keyword-based search system (Roxtra [Sch16]) currently used for the German hospital document corpus, particularly on complex or semantically nuanced queries. We sought to understand the underlying mechanisms driving these differences.

Comparative tests were conducted using identical queries in both the optimal RAG system and Roxtra. We focused on three representative query types:

6.6.1 Complex Medical Policy Questions

Query: "Unter welchen Bedingungen ist eine Organspende in Deutschland möglich?" (Under what conditions is organ donation possible in Germany?)

Roxtra: Returned approximately 19 document titles containing the keywords "Organspende" and "Deutschland." This required users to manually open and review potentially relevant documents with minimal initial context (See example in Figure 6.4).

RAG: Provided a synthesized, coherent answer summarizing key conditions like consent requirements (e.g., explicit consent, presumed consent depending on specifics), brain death criteria , and connections to advance directives . Crucially, it included precise citations (e.g., "BZgA_Pflege.pdf (p.14)"), enabling direct verification and deeper exploration of the source material (See example RAG interface in Figure 6.5).

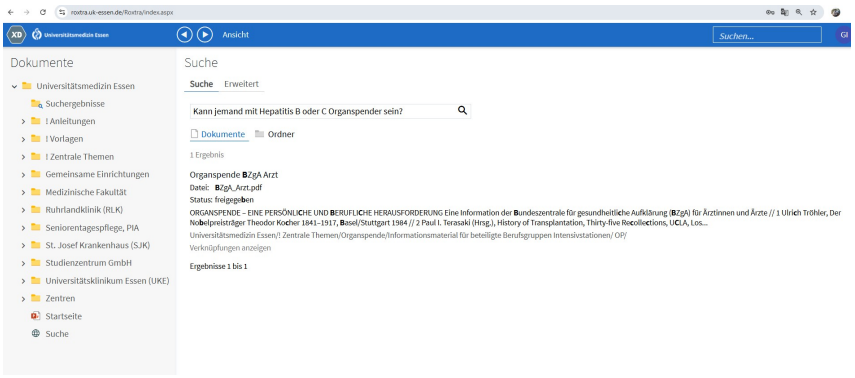


Figure 6.4: Example Keyword Search Results from Roxtra

6.6.2 Clinical Questions Requiring Semantic Understanding

Query: "Kann jemand mit Hepatitis B oder C Organspender sein?" (Can someone with Hepatitis B or C be an organ donor?)

Roxtra: Returned a general physician guide on organ donation but lacked explicit details regarding Hepatitis B/C eligibility, forcing the user into further manual searching within documents.

RAG: Delivered a direct answer explaining that individuals with Hepatitis B or C *can* potentially be donors, often for recipients who also have the same condition, outlining relevant considerations. It synthesized this information and cited specific sources like "BZgA_Pflege.pdf (p.16)" and "07_Anamnesebogen_Organspende.pdf (p.1)," demonstrating effective cross-document synthesis based on semantic understanding rather than just keyword overlap.

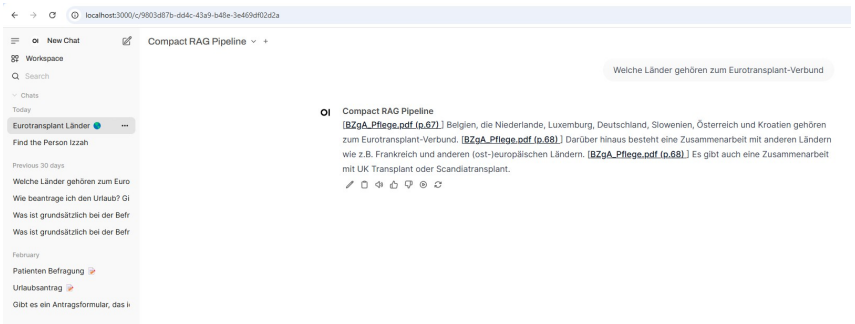


Figure 6.5: Example RAG Chatbot Response Grounded in Knowledge base

6.6.3 Factual Knowledge Queries

Query: "Welche Länder gehören zum Eurotransplant-Verbund?" (Which countries belong to the Eurotransplant network?)

Roxtra: Returned approximately 18 documents related to transplantation guidelines and processes, none of which immediately presented a list of member countries. The user would need to skim multiple documents to find this specific fact.

RAG: Directly extracted and listed the eight member countries (Germany, Netherlands, Belgium, Luxembourg, Austria, Hungary, Slovenia, Croatia), citing the source document and page number, providing an immediate, accurate answer to the factual query.

Underlying Mechanisms for RAG Superiority:

The qualitative examples highlight key mechanisms by which the RAG system overcomes the limitations of traditional keyword search [KTS16]:

1. **Semantic Understanding vs. Keyword Matching:** RAG leverages embeddings to grasp the *meaning* of the query, allowing it to retrieve relevant information even if the exact keywords are absent or phrased differently [TP10]. Roxtra relies on literal keyword presence.

2. **Information Synthesis vs. Document Lists:** RAG's generator model actively synthesizes information from multiple retrieved passages into a single, coherent answer [Ars+24], whereas keyword search typically returns a list of potentially relevant documents requiring manual synthesis.

3. **Context-Aware Presentation:** The RAG system provides answers within an explanatory context, rather than presenting isolated facts or document snippets.

4. **Precise Source Attribution:** Page-level citations are a core feature, offering transparency and enabling users to easily verify the information and explore sources in depth [Zho+24a]. Keyword systems usually lack this granularity.

5. **Cross-Document Reasoning:** RAG can integrate information scattered across multiple documents to answer nuanced queries that cannot be addressed by any single document alone [Bav+21].

These mechanisms collectively empower the RAG system to handle complex and semantically rich medical queries far more effectively than traditional keyword-based approaches [LS03; Wu+24a].

6.7 Summary of Key Findings

The comprehensive evaluation of the four RAG architectures yielded several key findings, directly addressing our initial research questions:

- **RQ1 (Embedding Impact):** The general multilingual embedding model (`intfloat/multilingual-e5-large` [Wan+24a]) unexpectedly outperformed the German-optimized bilingual model (`mixedbread-ai/deepset-mxbai-embed-de-large-v1` [mix23]). The Multilingual architecture achieved an 8.3% higher jury score and demonstrated superior context recall and precision, particularly excelling on complex questions (24.3% higher jury score than the Bilingual setup).

- **RQ2 (Retrieval Strategy):** Implementing hybrid search (dense vectors + BM25 [WC11]) provided significant benefits over vector-only search. The Hybrid architecture improved jury scores by 9.0–18.1% compared to vector-only architectures and substantially reduced "Not Acceptable" responses (by 44.4–52.9%). Its main strengths lay in enhancing context relevance and response groundedness.

- **RQ3 (MoE vs. Dense Generator):** The `Llama 4 Scout` [AI22] MoE generator showed modest advantages over the dense `Llama 3.3 70B` [Gra+24] model when paired with hybrid retrieval. It achieved slightly higher "Good" response ratings (87.91% vs. 86.81%), superior faithfulness (84.5% vs. 83.0%), and demonstrated more consistent performance across diverse question types.

- **RQ4 (Optimal Configuration):** The MoE architecture, integrating hybrid retrieval with the MoE generator, emerged as the most advantageous configuration overall. It secured the highest "Good" response rate (87.91%), the lowest "Not Acceptable" rate (9.89%), and proved robust in handling both complex (95.45% Good rate) and out-of-scope questions (86.36% Good rate), leveraging known MoE strengths [FZS22].

- **RQ5 (Keyword-Search Comparison):** Qualitative analysis confirmed the RAG system's superiority over the incumbent Roxtra [Sch16] keyword search. RAG's capabilities in semantic understanding [Jad+22], answer synthesis, context-aware presentation, precise citation, and cross-document reasoning enabled it to effectively answer complex and nuanced medical queries where keyword search faltered.

These findings offer actionable insights for deploying RAG systems within the medical domain [Wu+24a], particularly in German-language contexts. They underscore the value of employing hybrid retrieval strategies [BGI24] and exploring MoE generator architectures [Li+24a] to achieve high-quality, reliable, and contextually appropriate responses across diverse query types, ultimately validating the practical advantages of RAG over traditional keyword search in this specific hospital documentation setting [KTS16].

7 Conclusion and Future Work

7.1 Summary of Research and Key Findings

This thesis investigated the systematic optimization of Retrieval-Augmented Generation (RAG) to enhance information access for medical knowledge workers within German hospital environments, focusing on privacy-preserving on-premises deployment. Through evaluating four distinct RAG configurations against five research questions, we assessed the impact of embedding models, retrieval strategies, LLM architectures, overall performance, and comparison to keyword search. Our evaluation yielded several key findings that collectively validate RAG's potential in this setting.

Regarding the **embedding model's impact (RQ1)**, a general multilingual model (‘intfloat/multilingual-e5-large‘) unexpectedly outperformed its German-optimized counterpart (‘mixedbread-ai/deepset-mxbai-embed-de-large-v1‘), achieving an 8.3% higher jury score. This advantage was particularly pronounced in context recall, precision, and handling complex queries (+24.3% jury score), suggesting broad training may offer robust semantic understanding even in specialized domains [Wan+24a].

Turning to the **retrieval strategy (RQ2)**, implementing a hybrid search combining dense vectors (0.7 weight) with BM25 sparse vectors (0.3 weight) substantially improved performance [Wan+24c]. This approach yielded significantly higher jury scores (by 9.0–18.1%) and enhanced reliability, reducing “NotAcceptable” responses by up to 52.9% compared to vector-only methods, thus validating the combined strength of semantic and keyword-based retrieval [SMS24a].

In comparing generator architectures (**MoE vs. Dense, RQ3**), the Llama 4 Scout Mixture-of-Experts (MoE) model demonstrated marginal gains over the dense Llama-3.3-70B model in “Good” quality ratings (87.91% vs. 86.81%) and offered notably better faithfulness (84.5% vs. 83.0%), suggesting potential advantages for complex reasoning inherent in MoE architectures [FZS22].

Considering **overall performance trade-offs (RQ4)**, the configuration merging hybrid retrieval with the MoE generator emerged as optimal. This setup achieved the highest “Good” rate (87.91%), the lowest “NotAcceptable” rate (9.89%), and demonstrated robust performance across various question types.

Finally, the **comparison with keyword-based search (RQ5)** showed the opti-

mal RAG system's clear qualitative superiority over the incumbent Roxtra system. RAG's capacity for semantic understanding, information synthesis across documents, and precise source attribution marked a significant advancement over traditional document list retrieval.

7.2 Limitations of the Study

Despite these findings, several limitations frame the scope of this research. The **representativeness of the evaluation dataset** is one factor; derived from approximately 400 pages of source material yielding 90 question-answer pairs, it constitutes a limited sample of the vast medical domain. While curated for diversity, findings are specific to this subset and necessitate validation on larger, more varied corpora .

Hardware limitations also constrained the study, restricting analysis to models up to 70B parameters. This precluded investigation of larger state-of-the-art models or certain advanced training methodologies (like RL-based reasoning approaches [Guo+25]) that might yield different performance characteristics.

Furthermore, the **lack of formal user studies** means the evaluation relied on automated metrics and LLM panels, which cannot fully capture real-world applicability. Rigorous evaluation of deployment feasibility and clinical effectiveness requires engaging healthcare professionals through methods like observational usability testing (e.g., think-aloud protocols during representative tasks) or pilot deployments within clinical units, collecting feedback via surveys, interviews, and usage logs to understand practical utility and barriers.

Lastly, the **document chunking strategy** employed—a fixed-size recursive approach (512 tokens, 50 overlap)—does not leverage semantic content or document structure. Exploring alternative, potentially structure-aware methods (e.g., based on titles and paragraphs as explored by Jimeno Yepes et al. (2024) [Yep+24]) could enhance retrieval performance and mitigate context-related issues like the "Lost-in-the-Middle" effect [Liu+23], representing an area unaddressed due to the static approach used here.

7.3 Discussion and Implications

7.3.1 Implications for Practice

The empirical findings translate directly into actionable guidance for organizations deploying AI-assisted information retrieval in healthcare. Firstly, the unexpected

outperformance of the multilingual embedding model (RQ1) implies that **practitioners must empirically validate embedding choices** on their specific data, challenging assumptions about automatic benefits from language specialization and recognizing that broad training may offer robust semantic capture [Wan+24a].

Secondly, given its substantial boost to reliability and quality through improved relevance and groundedness (RQ2), **hybrid retrieval strategies combining dense and sparse methods should be prioritized** over vector-only approaches to ensure trustworthy RAG outputs in clinical settings [BGI24].

Regarding the generator, **Mixture-of-Experts (MoE) models present a compelling option**, offering slight edges in faithfulness and consistency (RQ3, RQ4). This suggests a pathway for balancing high-quality generation with potential computational efficiency, valuable in resource-aware environments [Raj+22].

Crucially, this study **confirms the feasibility of deploying effective RAG systems securely on-premises**. This assures the data sovereignty essential for healthcare institutions concerned with privacy and governance [Sch20].

Moreover, RAG's demonstrated ability to handle semantic complexity and synthesize information (RQ5) positions it as a **powerful enhancement layer over traditional keyword search**, particularly valuable for tackling complex clinical queries where older systems falter.

Finally, the optimal configuration's **robust performance across diverse query types** (RQ4) indicates its suitability as a reliable general-purpose tool baseline for varied hospital information needs, although further specialization remains possible.

7.3.2 Future Research Directions

Building upon this work, several promising research avenues emerge. One key direction involves **integrating multimodal information**, exploring architectures that incorporate vision-language models (e.g., CLIP [She+21], Med-PaLM M [Par+]) to process medical images, charts, and tables alongside text for more holistic data understanding. Complementary to handling broader data types is achieving deeper understanding within text itself; research into **structure-aware document processing**, using models that leverage sections, lists, and tables (e.g., Longformer-style attention [BPC20]), could improve context utilization and faithfulness over current chunking methods.

Further enhancements could come from **advancing retrieval effectiveness through dynamic query processing**. Techniques like query expansion using medical ontologies (e.g., SNOMED CT [Lee+14], ICD , query decomposition [Wan+24c], or adaptive retrieval methods like HyDE [Wan+24c] warrant investigation to potentially surpass the gains already achieved with static hybrid search (RQ2). As

capabilities advance, **developing clinically focused explainability** becomes paramount. This requires moving beyond source citations (RQ5) to include robust confidence scoring, transparent evidence provenance visualization, and methods to flag uncertainty or contradictions [Zho+24b], fostering essential clinical trust.

Expanding language capabilities, building on findings regarding multilingual models (RQ1), points towards **enabling true cross-lingual RAG** where users can query in one language and retrieve synthesized information from documents in another [Par+]. Architecturally, exploring alternatives like **Retrieval-Augmented Fine-Tuning (RAFT)**—integrating retrieval into the fine-tuning process [Zha+24]—offers another pathway to creating domain-specialized yet grounded models.

To ensure rigor across these diverse research streams, **standardizing evaluation for medical RAG** is crucial. Developing benchmark datasets reflecting clinical needs and robust, reproducible metrics (inspired by work like RAGAS [Es+23]) will enable meaningful comparisons. Lastly, addressing the **temporal dynamics** of medical information—handling evolving guidelines, research, and patient data—remains a vital long-term challenge requiring mechanisms for versioning awareness and time-sensitive information prioritization.

7.4 Concluding Remarks

This thesis has demonstrated the practical realization and effectiveness of privacy-preserving Retrieval-Augmented Generation systems specifically optimized for navigating complex German hospital documentation. By systematically dissecting the impact of embedding models, retrieval strategies, and generator architectures, we have not only identified RAG configurations that substantially outperform traditional keyword-based search but also derived key insights for practical deployment.

Our findings highlight the crucial role of hybrid retrieval for reliability, the potential of MoE models for balancing quality and efficiency, and the need for empirical validation over assumed specialization when selecting embedding models. Crucially, this work validates that such advanced AI capabilities can be achieved securely on-premises, providing a viable blueprint for healthcare institutions to leverage LLMs responsibly—tackling information overload while ensuring outputs are grounded in evidence and data sovereignty is maintained.

RAG represents a powerful tool for transforming medical information access from passive search to active synthesis. While this study provides a strong foundation, future progress hinges on embracing richer data modalities, deeper document understanding, and robust explainability to foster clinical trust. Realizing RAG's full potential requires continued innovation that balances cutting-edge performance

with the non-negotiable demands of privacy, reliability, and usability in real-world healthcare workflows.

Bibliography

[AGK95] Brad Adelberg, Hector Garcia-Molina, and Ben Kao. **Applying update streams in a soft real-time database system**. *ACM SIGMOD Record* 24:2 (1995), 245–256 (see page 28).

[AI22] Meta AI. *The Llama 4 herd: The beginning of a new era of natively multimodal AI innovation*. Meta.com, 2022. URL: https://ai.meta.com/blog/llama-4-multimodal-intelligence/ (see pages 38, 58, 74, 79).

[AI24] Mixedbread AI. *mixedbread-ai/mxbai-embed-large-v1 · Hugging Face*. Huggingface.co, 2024. URL: https://huggingface.co/mixedbread-ai/mxbai-embed-large-v1 (see page 57).

[Alg+24] Simone Alghisi, Massimo Rizzoli, Gabriel Roccabruna, Seyed Mahed Mousavi, and Giuseppe Riccardi. **Should we fine-tune or RAG? Evaluating different techniques to adapt LLMs for dialogue** (2024). eprint: 2406.06399 (cs.CL) (see pages 24, 27).

[Als+24] Dana Alsagheer, Rabimba Karanjai, Nour Diallo, Weidong Shi, Yang Lu, Suha Beydoun, and Qiaoning Zhang. **Comparing rationality between large language models and humans: Insights and open questions**. *arXiv preprint arXiv:2403.09798* (2024) (see page 9).

[An+24] Shengnan An, Zexiong Ma, Zeqi Lin, Nanning Zheng, Jian-Guang Lou, and Weizhu Chen. **Make your llm fully utilize the context**. *Advances in Neural Information Processing Systems* 37 (2024), 62160–62188 (see page 38).

[Ant25] Anthropic. *Claude AI*. https://claude.ai/. Large language model by Anthropic. Accessed: May 6, 2025. 2025 (see page 38).

[Ars+24] Muhammad Arslan, Hussam Ghanem, Saba Munawar, and Christophe Cruz. **A Survey on RAG with LLMs**. *Procedia Computer Science* 246 (Nov. 2024), 3781–3790. DOI: 10.1016/j.procs.2024.09.178. URL: https://www.sciencedirect.com/science/article/pii/S1877050924021860#keys0001 (see page 78).

[Art+21] Mikel Artetxe, Shruti Bhosale, Naman Goyal, Todor Mihaylov, Myle Ott, Sam Shleifer, Xi Victoria Lin, Jingfei Du, Srinivasan Iyer, Ramakanth Pasunuru, et al. **Efficient large scale language modeling with mixtures of experts**. *arXiv preprint arXiv:2112.10684* (2021) (see page 4).

[Art25] Artifex Software, Inc. *PyMuPDF: Python bindings for MuPDF's rendering library*. https://github.com/pymupdf/PyMuPDF. Software repository. 2025. (Visited on 05/08/2025) (see pages 29, 59).

[ASI20] Mohiuddin Ahmed, Raihan Seraj, and Syed Mohammed Shamsul Islam. **The k-means algorithm: A comprehensive survey and performance evaluation**. *Electronics* 9:8 (2020), 1295 (see page 33).

[Aya+21] Muhammad Ayaz, Muhammad F Pasha, Mohammed Y Alzahrani, Rahmat Budiarto, and Deris Stiawan. **The Fast Health Interoperability Resources (FHIR) standard: systematic literature review of implementations, applications, challenges and opportunities**. *JMIR medical informatics* 9:7 (2021), e21929 (see page 25).

[Bal+24] Angels Balaguer, Vinamra Benara, Renato Luiz de Freitas Cunha, Roberto de M Estevão Filho, Todd Hendry, Daniel Holstein, Jennifer Marsman, Nick Mecklenburg, Sara Malvar, Leonardo O Nunes, Rafael Padilha, Morris Sharp, Bruno Silva, Swati Sharma, Vijay Aski, and Ranveer Chandra. **RAG vs fine-tuning: Pipelines, tradeoffs, and a case study on agriculture** (2024). eprint: 2401.08406 (cs.CL) (see pages 27, 41).

[Bar+24] Oren Barkan, Yonatan Toib, Yehonatan Elisha, Jonathan Weill, and Noam Koenigstein. **LLM Explainability via Attributive Masking Learning**. In: *Findings of the Association for Computational Linguistics: EMNLP 2024*. 2024, 9522–9537 (see page 13).

[Bar24] Zuzanna Bartczak. *From RAG to Riches: Evaluating the Benefits of Retrieval-Augmented Generation in SQL Database Querying*. DIVA, 2024. URL: https://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-540215 (visited on 04/21/2025) (see page 34).

[Bav+21] Dipali Baviskar, Swati Ahirrao, Vidyasagar Potdar, and Ketan Kotecha. **Efficient automated processing of the unstructured documents using artificial intelligence: A systematic literature review and future directions**. *IEEE Access* 9 (2021), 72894–72936 (see pages 28, 78).

[Baz23] Anton Bazdyrev. **Semi-supervised inverted file index approach for approximate nearest neighbor search**. *System research and information technologies* (Dec. 2023), 69–75. DOI: 10.20535/srit.2308-8893.2023.4.05. (Visited on 04/21/2025) (see page 34).

[BC05] Stefan Büttcher and Charles LA Clarke. **Indexing time vs. query time: trade-offs in dynamic information retrieval systems**. In: *Proceedings of the 14th ACM international conference on Information and knowledge management*. 2005, 317–318 (see page 33).

[BCB14] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. **Neural machine translation by jointly learning to align and translate**. *arXiv preprint arXiv:1409.0473* (2014) (see page 10).

[BGI24] Sebastian Bruch, Siyu Gai, and Amir Ingber. **An Analysis of Fusion Functions for Hybrid Retrieval**. *ACM Transactions on Information Systems* 42 (Jan. 2024), 1–35. DOI: 10.1145/3596512. URL: https://arxiv.org/abs/2210.11934 (visited on 04/05/2024) (see pages 37, 69, 73, 80, 83).

[Bök15] Ann-Catrin Bökel. *Dokumenten-Management-Systeme zur elektronischen Verwaltung von Dokumenten des Qualitäts-Managements im Gesundheitswesen - Eine Befragung der Universitätsklinika in Deutschland*. bachelorthesis. 2015 (see pages iii, v).

[Boo+21] Andrew Booth, Marrissa Martyn-St James, Mark Clowes, and Anthea Sutton. **Systematic approaches to a successful literature review** (2021) (see pages 26, 28).

[BPC20] Iz Beltagy, Matthew E. Peters, and Arman Cohan. *Longformer: The Long-Document Transformer*. arXiv.org, Dec. 2020. DOI: 10.48550/arXiv.2004.05150. URL: https://arxiv.org/abs/2004.05150v2 (visited on 12/06/2023) (see page 83).

[Bru24] Tilmann Bruckhaus. **Rag does not work for enterprises**. *arXiv preprint arXiv:2406.04369* (2024) (see page 13).

[BS23] Marcel Binz and Eric Schulz. **Using cognitive psychology to understand GPT-3**. *Proceedings of the National Academy of Sciences* 120:6 (2023), e2218523120 (see page 9).

[Car05] Martin Carlsen. **Conceptual understanding of the dot product**. *Nordic Studies in Mathematics Education* 10:3-4 (2005), 3–28 (see page 35).

[CC17] John W Creswell and J David Creswell. **Research design: Qualitative, quantitative, and mixed methods approaches**. Sage publications, 2017 (see page 26).

[CCB09] Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. **Reciprocal rank fusion outperforms condorcet and individual rank learning methods**. *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval* (July 2009). DOI: 10.1145/1571941.1572114 (see page 65).

[Cha+24] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. **A survey on evaluation of large language models**. *ACM transactions on intelligent systems and technology* 15:3 (2024), 1–45 (see page 1).

[Che+14] Sharan Chetlur, Cliff Woolley, Philippe Vandermersch, Jonathan Cohen, John Tran, Bryan Catanzaro, and Evan Shelhamer. **cuDNN: Efficient Primitives for Deep Learning**. *arXiv:1410.0759 [cs]* (Dec. 2014). URL: https://arxiv.org/abs/1410.0759 (visited on 05/04/2022) (see page 57).

[Che+24a] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. *BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation*. arXiv.org, 2024. URL: https://arxiv.org/abs/2402.03216 (visited on 04/22/2025) (see page 31).

[Che+24b] Lu Chen, Yuxuan Huang, Yixing Li, Yaohui Jin, Shuai Zhao, Zilong Zheng, and Quanshi Zhang. **Alignment Between the Decision-Making Logic of LLMs and Human Cognition: A Case Study on Legal LLMs**. *arXiv preprint arXiv:2410.09083* (2024) (see page 9).

[Che+24c] Jeffrey Cheng, Marc Marone, Orion Weller, Dawn Lawrie, Daniel Khashabi, and Benjamin Van Durme. **Dated data: Tracing knowledge cutoffs in large language models**. *arXiv preprint arXiv:2403.12958* (2024) (see pages 13, 23, 38).

[Chr23] Chroma. *chroma-core/chroma*. GitHub, Oct. 2023. URL: https://github.com/chroma-core/chroma (see pages 33, 59).

[CJD24] Vicente Sanchez Carmona, Shanshan Jiang, and Bin Dong. **A Multilevel Analysis of PubMed-only BERT-based Biomedical Models**. In: *Proceedings of the 6th Clinical Natural Language Processing Workshop*. 2024, 105–110 (see page 14).

[Cla+19] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. **What does bert look at? an analysis of bert's attention**. *arXiv preprint arXiv:1906.04341* (2019) (see pages 11, 31).

[Dae+20] Azra Daei, Mohammad Reza Soleymani, Hasan Ashrafi-Rizi, Ali Zargham-Boroujeni, and Roya Kelishadi. **Clinical information seeking behavior of physicians: A systematic review**. *International journal of medical informatics* 139 (2020), 104144 (see page 7).

[Des+21] Danilo Dessi, Rim Helaoui, Vivek Kumar, Diego Reforgiato Recupero, and Daniele Riboni. **TF-IDF vs word embeddings for morbidity identification in clinical notes: An initial study**. *arXiv preprint arXiv:2105.09632* (2021) (see pages 15, 32).

[Dey+19] P Peter Dey, B Raj Sinha, Mohammad Amin, and Hassan Badkoobehi. **Best practices for improving user interface design**. *International Journal of Software Engineering & Applications* 10:5 (2019), 71–83 (see pages 62, 63).

[Din16] Ivo D Dinov. **Volume and value of big healthcare data**. *Journal of medical statistics and informatics* 4 (2016), 3 (see pages 1, 2, 7).

[Dou+24] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. *The Faiss library*. arXiv.org, Jan. 2024. DOI: 10.48550/arXiv.2401.08281. URL: https://arxiv.org/abs/2401.08281 (see page 33).

[Els09] Hilke Elsen. **Komplexe Komposita und Verwandtes**. *Germanistische Mitteilungen* 69 (2009), 57 (see pages 1, 8).

[Es+23] Sidorkina Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. **RAGAS: Automated Evaluation of Retrieval Augmented Generation**. *arXiv (Cornell University)* (Sept. 2023). DOI: 10.48550/arxiv.2309.15217 (see pages 18, 19, 39, 40, 43–46, 69, 84).

[Eti] Etienne Dilocker and Bob van Luijt and Byron Voorbach and Mohd Shukri Hasan and Abdel Rodriguez and Dirk Alexander Kulawiak and Marcin Antas and Parker Duckworth. *Weaviate*. URL: https://github.com/weaviate/weaviate (see page 33).

[Fan+24] Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. **A survey on rag meeting llms: Towards retrieval-augmented large language models**. In: *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2024, 6491–6501 (see page 24).

[FAT24] Ali Forootani, Danial Esmaeili Aliabadi, and Daniela Thraen. **Bio-Eng-LMM AI Assist chatbot: A Comprehensive Tool for Research and Education**. *arXiv preprint arXiv:2409.07110* (2024) (see page 24).

[FBS24] Robert Friel, Masha Belyi, and Atindriyo Sanyal. *RAGBench: Explainable Benchmark for Retrieval-Augmented Generation Systems*. arXiv.org, 2024. URL: https://arxiv.org/abs/2407.11005 (visited on 11/27/2024) (see pages 23, 38, 40).

[FC20] Luciano Floridi and Massimo Chiriatti. **GPT-3: Its nature, scope, limits, and consequences**. *Minds and Machines* 30 (2020), 681–694 (see page 11).

[FDG17] Alessio Ferrari, Beatrice Donati, and Stefania Gnesi. **Detecting domain-specific ambiguities: an NLP approach based on Wikipedia crawling and word embeddings**. In: *2017 IEEE 25th international requirements engineering conference workshops (REW)*. IEEE. 2017, 393–399 (see page 8).

[Fer+25] Georgios Feretzakis, Evangelia Vagena, Konstantinos Kalodanis, Paraskevi Peristera, Dimitris Kalles, and Athanasios Anastasiou. **GDPR and Large Language Models: Technical and Legal Obstacles**. *Future Internet* 17:4 (2025), 151 (see page 14).

[FH24] Teppo Felin and Matthias Holweg. **Theory is all you need: AI, human cognition, and decision making**. *Human Cognition, and Decision Making (February 23, 2024)* (2024) (see page 9).

[Fil+24] Elena Filipovska, Ana Mladenovska, Jovana Dobreva, Dimitar Kitanovski, Goran Mitrov, Petre Lameski, and Eftim Zdravevski. **Evaluation of Vector Databases and LLMs in RAG-Based Multi-document Question Answering**. In: *International Conference on ICT Innovations*. Springer. 2024, 3–18 (see page 32).

[Fin+24] Paulo Finardi, Leonardo Avila, Rodrigo Castaldoni, Pedro Gengo, Celio Larcher, Marcos Piau, Pablo Costa, and Vinicius Caridá. **The chronicles of rag: The retriever, the chunk and the generator**. *arXiv preprint arXiv:2401.07883* (2024) (see pages 17, 19, 23, 24, 26, 28, 29, 35–37).

[FM24] Mats Finsås and Joachim Maksim. **Optimizing RAG Systems for Technical Support with LLM-based Relevance Feedback and Multi-Agent Patterns**. MA thesis. NTNU, 2024 (see page 25).

[FP24] Jan Fillies and Adrian Paschke. **Simple LLM based approach to counter algospeak**. In: *Proceedings of the 8th Workshop on Online Abuse and Harms (WOAH 2024)*. 2024, 136–145 (see page 9).

[FZS22] William Fedus, Barret Zoph, and Noam Shazeer. *Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity*. arXiv.org, June 2022. DOI: 10.48550/arXiv.2101.03961. URL: https://arxiv.org/abs/2101.03961 (visited on 11/17/2023) (see pages 58, 65, 74, 79, 81).

[Gao+24] Yunfan Gao, Yun Xiong, Meng Wang, and Haofen Wang. **Modular rag: Transforming rag systems into lego-like reconfigurable frameworks**. *arXiv preprint arXiv:2407.21059* (2024) (see pages 17, 24, 25).

[Geb+24] Senay A Gebreab, Khaled Salah, Raja Jayaraman, Muhammad Habib ur Rehman, and Samer Ellaham. **Llm-based framework for administrative task automation in healthcare**. In: *2024 12th International Symposium on Digital Forensics and Security (ISDFS)*. IEEE. 2024, 1–7 (see page 14).

[Gir23] Louie Giray. **Prompt engineering with ChatGPT: a guide for academic writers**. *Annals of biomedical engineering* 51:12 (2023), 2629–2633 (see pages 38, 39).

[GJ23] Jiawei Gu and Xuhui Jiang. *A Survey on LLM-as-a-Judge*. Arxiv.org, 2023. URL: https://arxiv.org/html/2411.15594v1 (visited on 04/22/2025) (see page 47).

[GMM03] Ramanathan Guha, Rob McCool, and Eric Miller. **Semantic search**. In: *Proceedings of the 12th international conference on World Wide Web*. 2003, 700–709 (see page 25).

[GMT24] Oscar G. Lira, Alberto Marroquin, and Marco Antonio To. **Harnessing the advanced capabilities of llm for adaptive intrusion detection systems**. In: *International Conference on Advanced Information Networking and Applications*. Springer. 2024, 453–464 (see page 1).

[Gow85] John Clifford Gower. **Properties of Euclidean and non-Euclidean distance matrices**. *Linear algebra and its applications* 67 (1985), 81–97 (see page 35).

[GR09] Yair Goldberg and Ya'acov Ritov. **Local procrustes for manifold embedding: a measure of embedding quality and embedding algorithms**. *Machine learning* 77 (2009), 1–25 (see page 31).

[Gra+24] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. **The llama 3 herd of models**. *arXiv preprint arXiv:2407.21783* (2024) (see pages 58, 74, 79).

[GSB18] Dani Gunawan, CA Sembiring, and Mohammad Andri Budiman. **The implementation of cosine similarity to calculate text relevance between two documents**. In: *Journal of physics: conference series*. Vol. 978. IOP Publishing. 2018, 012120 (see page 35).

[GT15] Clinton Gormley and Zachary Tong. **Elasticsearch: the definitive guide: a distributed real-time search and analytics engine**. " O'Reilly Media, Inc.", 2015 (see page 33).

[Guo+25] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. **Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning**. *arXiv preprint arXiv:2501.12948* (2025) (see page 82).

[Has25] Mohammad Hassan. **Measuring the Impact of Hallucinations on Human Reliance in LLM Applications**. *Journal of Robotic Process Automation, AI Integration, and Workflow Optimization* 10:1 (2025), 10–20 (see pages 13, 76).

[HLW23] Yikun Han, Chunjiang Liu, and Pengfei Wang. **A comprehensive survey on vector database: Storage and retrieval technique, challenge**. *arXiv preprint arXiv:2310.11703* (2023) (see page 32).

[Hof+22] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. **Training compute-optimal large language models**. *arXiv preprint arXiv:2203.15556* (2022) (see page 11).

[Hri+23] Vagelis Hristidis, Nicole Ruggiano, Ellen L Brown, Sai Rithesh Reddy Ganta, and Selena Stewart. **ChatGPT vs Google for queries related to dementia and other cognitive decline: comparison of results**. *Journal of Medical Internet Research* 25 (2023), e48966 (see page 15).

[HS03] Gísli R Hjaltason and Hanan Samet. **Properties of embedding methods for similarity searching in metric spaces**. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25 (May 2003), 530–549. DOI: 10.1109/tpami.2003.1195989. (Visited on 11/27/2023) (see page 31).

[HSM24] Gerry Hosea, IT Student, and North Sumatra Medan. **Transforming Data Warehouses into Dynamic Knowledge Bases for RAG**. *Scientific Research Journal of Science, Engineering and Technology* 2:1 (2024), 5–10 (see pages 25, 27).

[HTZ21] Qing Han, Shubo Tian, and Jinfeng Zhang. **A PubMedBERT-based classifier with data augmentation strategy for detecting medication mentions in tweets**. *arXiv preprint arXiv:2112.02998* (2021) (see page 14).

[Hua+25] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. **A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions**. *ACM Transactions on Information Systems* 43:2 (2025), 1–55 (see page 2).

[Hug23] Hugging Face. *Text Embeddings Inference Documentation*. https://huggingface.co/docs/text-embeddings-inference/en/index. 2023 (see page 57).

[IM99] Piotr Indyk and Rajeev Motwani. *Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality (preliminary version)*. 1999. URL: https://graphics.stanford.edu/courses/cs468-06-fall/Papers/06%20indyk%20motwani%20-%20stoc98.pdf (visited on 04/21/2025) (see pages 32, 35).

[Jad+22] Ashutosh Jadhav, Tyler Baldwin, Joy Wu, Vandana Mukherjee, and Tanveer Syeda-Mahmood. **Semantic Expansion of Clinician Generated Data Preferences for Automatic Patient Data Summarization**. *AMIA Annual Symposium Proceedings* 2021 (Feb. 2022), 571. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC8861768/ (visited on 04/21/2025) (see pages 1, 35, 79).

[JDJ17] Jeff Johnson, Matthijs Douze, and Hervé Jégou. *Billion-scale similarity search with GPUs*. arXiv.org, Feb. 2017. DOI: 10.48550/arXiv.1702.08734. URL: https://arxiv.org/abs/1702.08734.pdf (see page 33).

[Jeo23] Cheonsu Jeong. **A study on the implementation of generative ai services using an enterprise data-based llm application architecture**. *arXiv preprint arXiv:2309.01105* (2023) (see pages 17, 41).

[Ji+22] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. **Survey of Hallucination in Natural Language Generation**. *ACM Computing Surveys* 55 (Nov. 2022). DOI: 10.1145/3571730 (see pages 12, 23).

[JLL24] Qiao Jin, Robert Leaman, and Zhiyong Lu. **PubMed and beyond: biomedical literature search in the age of artificial intelligence**. *EBioMedicine* 100 (2024) (see pages 1, 28).

[Kam+23] Niklas Kammer, Florian Borchert, Silvia Winkler, Gerard De Melo, and Matthieu-P Schapranow. **Resolving elliptical compounds in german medical text**. In: *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*. 2023, 292–305 (see page 43).

[Kap+20] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. **Scaling Laws for Neural Language Models**. *arXiv preprint arXiv:2001.08361* (2020). Published January 22, 2020. URL: https://doi.org/10.48550/arXiv.2001.08361 (see page 11).

[Kar+19] Shigeki Karita, Nanxin Chen, Tomoki Hayashi, Takaaki Hori, Hirofumi Inaguma, Ziyan Jiang, Masao Someki, Nelson Enrique Yalta Soplin, Ryuichi Yamamoto, Xiaofei Wang, et al. **A comparative study on transformer vs rnn in speech applications**. In: *2019 IEEE automatic speech recognition and understanding workshop (ASRU)*. IEEE. 2019, 449–456 (see page 12).

[Kar+20] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. **Dense Passage Retrieval for Open-Domain Question Answering**. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2020). DOI: 10.18653/v1/2020.emnlp-main.550 (see pages 23, 36).

[Kat14] Maryam Katani. **Challenges of implementing an electronic document management system in a large health care facility in Southern California**. PhD thesis. 2014 (see page 8).

[KB24] Jason Kirchenbauer and Caleb Barns. *Hallucination reduction in large language models with retrieval-augmented generation using wikipedia knowledge*. 2024 (see page 43).

[Ken22] Benjamin Kenwright. "Introduction to the webgpu api." In: *Acm siggraph 2022 courses*. 2022, 1–184 (see page 13).

[Kir+16] Roberts Kirk, DF Dina, EM Voorhees, and R Hersh William. **Overview of the TREC 2016 clinical decision support track**. In: *Proceedings of the 15th text retrieval conference*. 2016 (see page 43).

[Kon24] Konferenz der unabhängigen Datenschutzaufsichtsbehörden des Bundes und der Länder (DSK). **Orientierungshilfe: Künstliche Intelligenz und Datenschutz**. Orientierungshilfe. Version 1.0. Version 1.0. Veröffentlicht am 6. Mai 2024. Konferenz der unabhängigen Datenschutzaufsichtsbehörden des Bundes und der Länder, May 2024. URL: https://www.datenschutzkonferenz-online . de / media / oh / 20240506 _ DSK _ Orientierungshilfe _ KI _ und _ Datenschutz.pdf (visited on 05/05/2025) (see page 13).

[Kor22] Modest von Korff. **Exhaustive Indexing of PubMed Records with Medical Subject Headings**. *ACL Anthology* (June 2022), 8–15. URL: https://aclanthology.org/2022.cmlc-1.2/ (visited on 04/21/2025) (see pages 2, 23).

[Kos+22] Eliza Kosoy, David M Chan, Adrian Liu, Jasmine Collins, Bryanna Kaufmann, Sandy Han Huang, Jessica B Hamrick, John Canny, Nan Rosemary Ke, and Alison Gopnik. **Towards understanding how machines can learn causal overhypotheses**. *arXiv preprint arXiv:2206.08353* (2022) (see page 9).

[KR18] Monique F Kilkenny and Kerin M Robinson. *Data quality:"Garbage in–garbage out"*. 2018 (see page 26).

[KTS16] Ranjeet Kumar, RC Tripathi, and Vrijendra Singh. **Keyword based search and its limitations in the patent document to secure the idea from its infringement**. *Procedia Computer Science* 78 (2016), 439–446 (see pages 1, 3, 8, 15, 52, 78, 80).

[Kun+23] Tiffany H. Kung, Morgan Cheatham, Arielle Medenilla, Czarina Sillos, Lorie De Leon, Camille Elepaño, Maria Madriaga, Rimel Aggabao, Giezel Diaz-Candido, James Maningo, and Victor Tseng. **Performance of ChatGPT on USMLE: Potential for AI-assisted medical education using large language models**. *PLOS Digital Health* 2 (Feb. 2023). Ed. by Alon Dagan, e0000198. DOI: 10.1371/journal.pdig.0000198. URL: https://journals.plos.org/digitalhealth/article?id=10.1371/journal.pdig.0000198 (see page 12).

[KWT15] Irma Klerings, Alexandra S Weinhandl, and Kylie J Thaler. **Information overload in healthcare: too much of a good thing?** *Zeitschrift fur Evidenz, Fortbildung und Qualitat im Gesundheitswesen* 109 (2015), 285–90. DOI: 10.1016/j.zefq.2015.06.005. URL: https://www.ncbi.nlm.nih.gov/pubmed/26354128 (see pages iii, v, 1, 7).

[Lam24] Sotiris Lamprinidis. **LLM Cognitive Judgements Differ from Human**. In: *Frontiers of Artificial Intelligence, Ethics, and Multidisciplinary Applications*. Ed. by Mina Farmanbar, Maria Tzamtzi, Ajit Kumar Verma, and Antorweep Chakravorty. Singapore: Springer Nature Singapore, 2024, 17–23. ISBN: 978-981-99-9836-4 (see page 9).

[Lee+14] Dennis Lee, Nicolette de Keizer, Francis Lau, and Ronald Cornet. **Literature review of SNOMED CT use**. *Journal of the American Medical Informatics Association* 21:e1 (2014), e11–e19 (see page 83).

[Lew+21] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. arXiv.org, Apr. 2021. DOI: 10.48550/arXiv.2005.11401. URL: https://arxiv.org/abs/2005.11401 (see pages iii, v, 2, 16, 23, 26, 37, 43, 55, 69).

[Li+24a] Jiachen Li, Xinyao Wang, Sijie Zhu, Chia-Wen Kuo, Lu Xu, Fan Chen, Jitesh Jain, Humphrey Shi, and Longyin Wen. **CuMo: Scaling Multimodal LLM with Co-Upcycled Mixture-of-Experts**. In: *Advances in Neural Information Processing Systems 37*. Ed. by Amir Globerson, Lihong Mackey, Danielle Belgrave, Alice Fan, Urs Paquet, Jakub Tomczak, and Cheng Zhang. Vol. 37. Dec. 2024, 131224–131246 (see page 80).

[Li+24b] Moxin Li, Yong Zhao, Yang Deng, Wenxuan Zhang, Shuaiyi Li, Wenya Xie, See-Kiong Ng, and Tat-Seng Chua. **Knowledge boundary of large language models: A survey** (Dec. 2024). arXiv: 2412.12472 [cs.CL] (see page 24).

[Lin24a] Jimmy Lin. **Operational Advice for Dense and Sparse Retrievers: HNSW, Flat, or Inverted Indexes?** *arXiv (Cornell University)* (Sept. 2024). DOI: 10.48550/arxiv.2409.06464. (Visited on 04/21/2025) (see pages 34, 36).

[Lin24b] Jimmy Lin. **Operational Advice for Dense and Sparse Retrievers: HNSW, Flat, or Inverted Indexes?** *arXiv preprint arXiv:2409.06464* (2024) (see page 34).

[Lit+25] Robert Litschko, Oliver Kraus, Verena Blaschke, and Barbara Plank. **Cross-Dialect Information Retrieval: Information Access in Low-Resource and High-Variance Languages**. In: *Proceedings of the 31st International Conference on Computational Linguistics*. Ed. by Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert. Abu Dhabi, UAE: Association for Computational Linguistics, Jan. 2025, 10158–10171. URL: https://aclanthology.org/2025.coling-main.678/ (see page 1).

[Liu+22] Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A. Raffel. **Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning**. In: *Advances in Neural Information Processing Systems*. Ed. by Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato. Vol. 35. Curran Associates, Inc., 2022, 1950–1965 (see page 25).

[Liu+23] Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. **Lost in the Middle: How Language Models Use Long Contexts**. *ARXIV* (July 2023). DOI: 10.48550/arxiv.2307.03172 (see pages 39, 82).

[Lla25] LlamaIndex.ai. *LlamaParse: Transform unstructured data into LLM optimized formats*. Web Page. Product page by LlamaIndex.ai, which describes itself as a platform to "Build Knowledge Assistants over your Enterprise Data." Accessed on April 21, 2025. LlamaIndex.ai, 2025. URL: https://www.llamaindex.ai/llamaparse (visited on 04/21/2025) (see page 29).

[LS03] Ryan LaBrie and Robert St Louis. **Information retrieval from knowledge management systems: Using knowledge hierarchies to overcome keyword limitations**. *AMCIS 2003 Proceedings* (2003), 333 (see pages 3, 15, 78).

[LS21] Kevin Laland and Amanda Seed. **Understanding human cognitive uniqueness**. *Annual Review of Psychology* 72:1 (2021), 689–716 (see page 10).

[Lua+21] Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. **Sparse, dense, and attentional representations for text retrieval**. *Transactions of the Association for Computational Linguistics* 9 (2021), 329–345 (see page 36).

[Luo+22] Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. **BioGPT: generative pre-trained transformer for biomedical text generation and mining**. *Briefings in bioinformatics* 23:6 (2022), bbac409 (see page 14).

[LYZ24] Jiarui Li, Ye Yuan, and Zehua Zhang. **Enhancing llm factual accuracy with rag to counter hallucinations: A case study on domain-specific queries in private knowledge-bases**. *arXiv preprint arXiv:2403.10446* (2024) (see pages 2, 16, 40).

[LZM22] Ruiqi Li, Xiang Zhao, and Marie-Francine Moens. **A Brief Overview of Universal Sentence Representation Methods: A Linguistic View**. *ACM Computing Surveys* 55:3 (Mar. 2022). ISSN: 0360-0300. DOI: 10.1145/3482853 (see page 25).

[Mag+24] Varun Magesh, Faiz Surani, Matthew Dahl, Mirac Suzgun, Christopher D Manning, and Daniel E Ho. **Hallucination-Free? Assessing the Reliability of Leading AI Legal Research Tools**. *Journal of Empirical Legal Studies* (2024) (see pages 24, 51).

[MB11] Ajit Kumar Mahapatra and Sitanath Biswas. **Inverted indexes: Types and techniques**. *International Journal of Computer Science Issues (IJCSI)* 8:4 (2011), 384 (see page 33).

[Mes23] Bertalan Meskó. **Prompt engineering as an important emerging skill for medical professionals: tutorial**. *Journal of medical Internet research* 25 (2023), e50638 (see pages 38, 39, 61).

[mix23] mixedbread-ai. *deepset-mxbai-embed-de-large-v1*. https://huggingface.co/mixedbread-ai/deepset-mxbai-embed-de-large-v1. Accessed: 2024-05-06. 2023 (see pages 72, 79).

[MS25] Carlo Merola and Jaspinder Singh. **Reconstructing Context: Evaluating Advanced Chunking Strategies for Retrieval-Augmented Generation**. *arXiv preprint arXiv:2504.19754* (2025) (see page 30).

[Mue+22] Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. **MTEB: Massive text embedding benchmark**. *arXiv preprint arXiv:2210.07316* (2022) (see pages 31, 32).

[Mug+23] Joseph Mugaanyi, Liuying Cai, Sumei Cheng, Caide Lu, and Jing Huang. **Citations and References in Scholarly Writing: A cross-disciplinary Evaluation of Large Language Model Performance and Reliability. (Preprint)**. *JMIR. Journal of medical internet research/Journal of medical internet research* (Sept. 2023). DOI: 10.2196/52935 (see page 2).

[MY16] Yu A Malkov and D A Yashunin. **Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs** (Mar. 2016). DOI: 10.48550/arxiv.1603.09320 (see page 34).

[Nav+23] Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. **A comprehensive overview of large language models**. *arXiv preprint arXiv:2307.06435* (2023) (see page 38).

[Név+18] Aurélie Névéol, Hercules Dalianis, Sumithra Velupillai, Guergana Savova, and Pierre Zweigenbaum. **Clinical natural language processing in languages other than English: opportunities and challenges**. *Journal of biomedical semantics* 9 (2018), 1–13 (see page 8).

[Nic+17] Raffaele Nicastro, Alessandro Sardu, Nicolas Panchaud, and Claudio De Virgilio. **The architecture of the Rag GTPase signaling network**. *Biomolecules* 7:3 (2017), 48 (see page 26).

[NNN24] H. T. Nguyen, T. D. Nguyen, and V. H. Nguyen. **Enhancing Retrieval Augmented Generation with Hierarchical Text Segmentation Chunking**. In: *Information and Communication Technology – SOICT 2024: 13th International Symposium, Danang, Vietnam, December 13–15, 2024, Proceedings, Part I*. Ed. by Wray Buntine, Morten Fjeld, Truyen Tran, Minh-Triet Tran, Binh Huynh Thi Thanh, and Takumi Miyoshi. Vol. 2350. Communications in

Computer and Information Science. Singapore: Springer Nature Singapore, Dec. 2024, 209–220 (see page 30).

[Nor+23] Harsha Nori, Nicholas King, Scott Mayer McKinney, Dean Carignan, and Eric Horvitz. **Capabilities of GPT-4 on Medical Challenge Problems** (Mar. 2023). DOI: 10.48550/arxiv.2303.13375 (see pages 2, 12).

[Ope20] OpenAI. *OpenAI*. Accessed: 2025-05-06. OpenAI. 2020. URL: https://openai.com/ (see pages 13, 58, xxv).

[Oro+24] Ermelinda Oro, Francesco Maria Granata, Antonio Lanza, Amir Bachir, Luca De Grandis, and Massimo Ruffolo. **Evaluating Retrieval-Augmented Generation for Question Answering with Large Language Models** (2024) (see pages 44, 46).

[Pal+24] Somasundaram Palaniappan, Rohit Mali, Luca Cuomo, Mauro Vitale, Ali Youssef, Athira Puthanveetil Madathil, Malathi Murugesan, Alessandro Bettini, Giovanni De Magistris, and Giacomo Veneri. **Enhancing Enterprise-Wide Information Retrieval through RAG Systems Techniques, Evaluation, and Scalable Deployment**. *ADIPEC* (Nov. 2024). DOI: 10.2118/222032-ms. URL: https://onepetro.org/SPEADIP/proceedings-abstract/24ADIP/24ADIP/585627 (visited on 12/25/2024) (see page 40).

[Par+] Kendall Park, Rory Sayres, Andrew Sellergren, Tom Pollard, Fayaz Jamil, Timo Kohlberger, Charles Lau, and Atilla Kiraly. **Application of Med-PaLM 2 in the refinement of MIMIC-CXR labels** () (see pages 1, 12, 38, 83, 84).

[Par+24] Venkatesh Balavadhani Parthasarathy, Ahtsham Zafar, Aafaq Khan, and Arsalan Shahid. **The ultimate guide to fine-tuning llms from basics to breakthroughs: An exhaustive review of technologies, research, best practices, applied research challenges and opportunities**. *arXiv preprint arXiv:2408.13296* (2024) (see pages 26, 27).

[PBC12] Carol Peters, Martin Braschler, and Paul Clough. **Multilingual information retrieval: From research to practice**. Springer, 2012 (see page 2).

[Pec+14] Pavel Pecina, Ondřej Dušek, Lorraine Goeuriot, Jan Hajič, Jaroslava Hlaváčová, Gareth JF Jones, Liadh Kelly, Johannes Leveling, David Mareček, Michal Novák, et al. **Adaptation of machine translation for multilingual information retrieval in the medical domain**. *Artificial intelligence in medicine* 61:3 (2014), 165–185 (see page 8).

[Per16] Caio Ribeiro Pereira. **Building APIs with Node.js**. Berkeley, CA: Apress, 2016. ISBN: 978-1-4842-2442-7. DOI: 10.1007/978-1-4842-2442-7. URL: https://doi.org/10.1007/978-1-4842-2442-7 (see page 28).

[pgv] pgvector contributors. *pgvector: Open-source vector similarity search for Postgres* (see page 33).

[Pin+21] Yuliya Pinevich, Kathryn J Clark, Andrew M Harrison, Brian W Pickering, and Vitaly Herasevich. **Interaction time with electronic health records: a systematic review**. *Applied clinical informatics* 12:04 (2021), 788–799 (see pages 1, 7).

[Pin24] Pinecone, Inc. *Pinecone: The vector database to build knowledgeable AI*. https://www.pinecone.io/. 2024 (see page 33).

[Pry24] M Prytula. **Fine-tuning BERT, DistilBERT, XLM-RoBERTa and Ukr-RoBERTa models for sentiment analysis of Ukrainian language reviews**. *Machine learning* 3:4 (2024) (see page 57).

[PUS23] Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. *Med-HALT: Medical Domain Hallucination Test for Large Language Models*. Accepted at EMNLP 2023 (The SIGNLL Conference on Computational Natural Language Learning). 2023. DOI: 10.48550/arXiv.2307.15343. arXiv: 2307.15343 [cs.CL]. URL: https://arxiv.org/abs/2307.15343 (see page 16).

[PV24] Arnau Perez and Xavier Vizcaino. *Advanced ingestion process powered by LLM parsing for RAG system*. arXiv preprint arXiv:2412.15262 [cs.CL]. 2024. DOI: 10.48550/arXiv.2412.15262. arXiv: 2412.15262 [cs.CL]. URL: https://arxiv.org/abs/2412.15262 (see pages 28–30).

[PWL24] James Jie Pan, Jianguo Wang, and Guoliang Li. **Vector database management techniques and systems**. In: *Companion of the 2024 International Conference on Management of Data*. 2024, 597–604 (see pages 25, 32).

[Qdr23] Qdrant. *Qdrant: Neural Search Engine*. https://qdrant.tech/. 2023 (see page 33).

[QTB24] Renyi Qu, Ruixuan Tu, and Forrest Bao. **Is Semantic Chunking Worth the Computational Cost?** *arXiv preprint arXiv:2410.13070* (2024) (see pages 29, 30, 74).

[Raj+22] Samyam Rajbhandari, Conglong Li, Zhewei Yao, Minjia Zhang, Aminabadi Reza Yazdani, Ammar Ahmad Awan, Jeff Rasley, and Yuxiong He. *DeepSpeed-MoE: Advancing Mixture-of-Experts Inference and Training to Power Next-Generation AI Scale*. arXiv.org, 2022. URL: https://arxiv.org/abs/2201.05596 (visited on 04/01/2025) (see pages 76, 83).

[Raw+23] Vipula Rawte, Swagata Chakraborty, Agnibh Pathak, Anubhav Sarkar, SM Towhidul Islam Tonmoy, Aman Chadha, Amit Sheth, and Amitava Das. **The Troubling Emergence of Hallucination in Large Language Models-An Extensive Definition, Quantification, and Prescriptive Remediations**. In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 2023, 2541–2573 (see page 13).

[RG19] Nils Reimers and Iryna Gurevych. **Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks** (Aug. 2019). DOI: 10.48550/arxiv.1908.10084 (see page 31).

[Rob10] Stephen Robertson. **The Probabilistic Relevance Framework: BM25 and Beyond**. *Foundations and Trends® in Information Retrieval* 3 (2010), 333–389. DOI: 10.1561/1500000019 (see page 36).

[Rox25] Roxtra GmbH. *Über die Roxtra GmbH*. Web Page. Accessed on May 7, 2025. 2025. URL: https://www.roxtra.com/unternehmen/ (visited on 05/07/2025) (see page 14).

[RS21] Pedro Rodriguez and Arthur Spirling. **Word Embeddings: What works, what doesn't, and how to tell the difference for applied research**. *The Journal of Politics* (May 2021). DOI: 10.1086/715162 (see pages 25, 31).

[Saa+24] Jon Saad-Falcon, Omar Khattab, Christopher Potts, and Matei Zaharia. **ARES: An Automated Evaluation Framework for Retrieval-Augmented Generation Systems**. *arXiv (Cornell University)* (Jan. 2024). DOI: 10.18653/v1/2024.naacl-long.20. (Visited on 04/22/2025) (see page 45).

[Sab23] Walid S Saba. **Stochastic LLMs do not understand language: towards symbolic, explainable and ontologically based LLMs**. In: *International conference on conceptual modeling*. Springer. 2023, 3–19 (see pages 12, 23).

[San23] Katharine Sanderson. **GPT-4 is here: what scientists think**. *Nature* 615:7954 (2023), 773 (see pages 1, 38).

[SAW15] Ali Seyed Shirkhorshidi, Saeed Aghabozorgi, and Teh Ying Wah. **A comparison study on similarity and dissimilarity measures in clustering continuous data**. *PloS one* 10:12 (2015), e0144059 (see page 35).

[SB09] Krysta M Svore and Christopher JC Burges. **A machine learning approach for improved BM25 retrieval**. In: *Proceedings of the 18th ACM conference on Information and knowledge management*. 2009, 1811–1814 (see page 36).

[SBS24] Matthias Stadler, Maria Bannert, and Michael Sailer. **Cognitive ease at a cost: LLMs reduce mental effort but compromise depth in student scientific inquiry**. *Computers in Human Behavior* 160 (2024), 108386 (see page 2).

[Sch+93] Jonathan Schaeffer, Paul Lu, Duane Szafron, and Robert Lake. **A re-examination of brute-force search**. In: *Proceedings of the AAAI Fall Symposium on Games: Planning and Learning*. AAAI Press Menlo Park, Calif. 1993, 51–58 (see page 32).

[Sch16] Marc Schukey. **Ein Blick in die Praxis – Software zur Dokumentenlenkung und zum Workflowmanagement im Zertifizierungsalltag**. *Zertifizierung als Erfolgsfaktor* (2016), 435–446. DOI: 10.1007/978-3-658-09701-1_32 (see pages iii, v, 1, 3, 14, 43, 52, 76, 79).

[Sch20] Katrin Schnetter. **Datenschutz Im Gesundheitswesen (Medizinische Daten)**. *Regulatorisches Wissen Für Medizinprodukte* (June 2020). Accessed: May 6, 2025 (see pages 43, 60, 83).

[She+21] Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. **How much can clip benefit vision-and-language tasks?** *arXiv preprint arXiv:2107.06383* (2021) (see page 83).

[She+24] Junhong Shen, Neil Tenenholtz, James Brian Hall, David Alvarez-Melis, and Nicolo Fusi. **Tag-LLM: Repurposing general-purpose LLMs for specialized domains**. *arXiv preprint arXiv:2402.05140* (2024) (see page 23).

[She20] Alex Sherstinsky. **Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network**. *Physica D: Nonlinear Phenomena* 404 (2020), 132306 (see page 12).

[Shi+24] Zhuo-Fan Shi, Kun Liu, Shan Bai, Yun-Tao Jiang, Tong Huo, Xiang Jing, Rui-Zhi Li, and Xin-Jian Ma. **Meta data retrieval for data infrastructure via RAG**. In: *2024 IEEE International Conference on Web Services (ICWS)*. IEEE. 2024, 100–107 (see page 33).

[Sho+25] Sina Shool, Sara Adimi, Reza Saboori Amleshi, Ehsan Bitaraf, Reza Golpira, and Mahmood Tara. **A systematic review of large language model (LLM) evaluations in clinical medicine**. *BMC Medical Informatics and Decision Making* 25:1 (2025), 117 (see pages iii, v, 12, 23).

[Sim+24] Sebastian Simon, Alina Mailach, Johannes Dorn, and Norbert Siegmund. **A Methodology for Evaluating RAG Systems: A Case Study On Configuration Dependency Validation**. *arXiv preprint arXiv:2410.08801* (2024) (see page 46).

[Sin+22] Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. **Large language models encode clinical knowledge**. *arXiv preprint arXiv:2212.13138* (2022) (see page 13).

[Sin+25] Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Mohamed Amin, Le Hou, Kevin Clark, Stephen R Pfohl, Heather Cole-Lewis, Darlene Neal, Qazi Mamunur Rashid, Mike Schaekermann, Amy Wang, Dev Dash, Jonathan H Chen, Nigam H Shah, Sami Lachgar, Philip Andrew Mansfield, Sushant Prakash, Bradley Green, Ewa Dominowska, Blaise, Nenad

Tomašev, Yun Liu, Renee Wong, Christopher Semturs, S Sara Mahdavi, Joelle K Barral, Dale R Webster, Greg S Corrado, Yossi Matias, Shekoofeh Azizi, Alan Karthikesalingam, and Vivek Natarajan. **Toward expert-level medical question answering with large language models**. *Nature Medicine* (Jan. 2025). DOI: 10.1038/s41591-024-03423-7 (see page 1).

[Sir+23] Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, and Suranga Nanayakkara. **Improving the domain adaptation of retrieval augmented generation (RAG) models for open domain question answering**. *Transactions of the Association for Computational Linguistics* 11 (2023), 1–17 (see pages 2, 3, 23, 24).

[SK10] Jason Sanders and Edward Kandrot. **CUDA by example: an introduction to general-purpose GPU programming**. Addison-Wesley Professional, 2010 (see pages 10, 57).

[SKH24] Heydar Soudani, Evangelos Kanoulas, and Faegheh Hasibi. **Fine tuning vs. retrieval augmented generation for less popular knowledge**. In: *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region (SIGIR-AP '24)*. Tokyo, Japan: Association for Computing Machinery (ACM), Dec. 2024, 12–22 (see page 27).

[Slo+24] Elizabeth A Sloss, Shawna Abdul, Mayfair A Aboagyewah, Alicia Beebe, Kathleen Kendle, Kyle Marshall, S Trent Rosenbloom, Sarah Rossetti, Aaron Grigg, Kevin D Smith, et al. **Toward alleviating clinician documentation burden: a scoping review of burden reduction efforts**. *Applied Clinical Informatics* 15:03 (2024), 446–455 (see pages 3, 15).

[SMS24a] Kunal Sawarkar, Abhilasha Mangal, and Shivam Raj Solanki. *Blended RAG: Improving RAG (Retriever-Augmented Generation) Accuracy with Semantic Search and Hybrid Query-Based Retrievers*. arXiv.org, Mar. 2024. DOI: 10.48550/arXiv.2404.07220. URL: https://arxiv.org/abs/2404.07220 (see pages 31, 73, 81).

[SMS24b] Kunal Sawarkar, Abhilasha Mangal, and Shivam Raj Solanki. **Blended rag: Improving rag (retriever-augmented generation) accuracy with semantic search and hybrid query-based retrievers**. In: *2024 IEEE 7th International Conference on Multimedia Information Processing and Retrieval (MIPR)*. IEEE. 2024, 155–161 (see pages 36, 37).

[SRL24] Mark G Siegel, Michael J Rossi, and James H Lubowitz. *Artificial intelligence and machine learning may resolve health care information overload*. 2024 (see page 7).

[Ste+22] Arnulf Stenzl, Cora N Sternberg, Jenny Ghith, Lucile Serfass, Bob JA Schijvenaars, and Andrea Sboner. **Application of artificial intelligence to overcome clinical information overload in urological cancer**. *BJU international* 130:3 (2022), 291–300 (see page 7).

[Sul+24] Dewang Sultania, Zhaoyu Lu, Twisha Naik, Franck Dernoncourt, David Seunghyun Yoon, Sanat Sharma, Trung Bui, Ashok Gupta, Tushar Vatsa, Suhas Suresha, Ishita Verma, Vibha Belavadi, Cheng Chen, and Michael Friedrich. *Domain-specific Question Answering with Hybrid Search*. Appeared at the AAAI-25 Workshop on Document Understanding and Intelligence (The arXiv page mentions "AAAI-25 Workshop on Document Understanding and Intelligence", implying future appearance at the time of its listing, but given the current date, this note reflects that it would have appeared if the workshop has passed or is still upcoming relevant to its 2024 submission date). 2024. DOI: 10.48550/arXiv.2412.03736. arXiv: 2412.03736 [cs.CL]. URL: https://arxiv.org/abs/2412.03736 (see page 36).

[SWK24] Miyu Sasaki, Natsumi Watanabe, and Tsukihito Komanaka. **Enhancing contextual understanding of mistral llm with external knowledge bases** (2024) (see page 38).

[Tea24a] Giskard Team. *Giskard-AI/giskard*. GitHub, Mar. 2024. URL: https://github.com/Giskard-AI/giskard (see pages 49, 50).

[Tea24b] Openweb UI Team. *open-webui/open-webui*. GitHub, May 2024. URL: https://github.com/open-webui/open-webui (see page 63).

[Tei+19] Yee-Yang Teing, Sajad Homayoun, Ali Dehghantanha, Kim-Kwang Raymond Choo, Reza M Parizi, Mohammad Hammoudeh, and Gregory Epiphaniou. **Private cloud storage forensics: Seafile as a case study**. *Handbook of big data and IoT security* (2019), 73–127 (see pages 58, 60).

[Tha+21] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. **Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models**. *arXiv preprint arXiv:2104.08663* (2021) (see pages 23, 46).

[TP10] Peter D Turney and Patrick Pantel. **From frequency to meaning: Vector space models of semantics**. *Journal of artificial intelligence research* 37 (2010), 141–188 (see pages 31, 78).

[TR16] Chen-Tse Tsai and Dan Roth. **Cross-lingual wikification using multilingual embeddings**. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2016, 589–598 (see pages 72, 73).

[Tru] TruLens. *RAG Triad*. https://www.trulens.org/getting_started/core_concepts/rag_triad/. TruLens Documentation. Accessed: May 6, 2025 (see pages 43, 46).

[TSB09] Duygu Tümer, Mohammad A. Shah, and Yiltan Bitirim. **An empirical evaluation on semantic search performance of keyword-based and semantic search engines: Google, yahoo, msn and hakia**. In: *2009 Fourth International Conference on Internet Monitoring and Protection (ICIMP)*. Washington, DC, USA: IEEE Computer Society, May 2009, 51–55. DOI: 10.1109/ICIMP.2009.16 (see pages 3, 15).

[UH05] Ranjith Unnikrishnan and Martial Hebert. **Measures of similarity**. In: *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05)-Volume 1*. Vol. 1. IEEE. 2005, 394–394 (see page 35).

[Uni24] University of Potsdam Data Protection Officer. *Leitfaden zur Nutzung von KI-gestützten Schreibtools wie ChatGPT an der Universität Potsdam*. Stand: 22.01.2024. Verantwortlich für den Inhalt: AG GPT Co. Kontakt Datenschutzbeauftragter: datenschutz@uni-potsdam.de. Universität Potsdam. Jan. 2024. URL: https://www.uni-potsdam.de/de/gptup/leitfaden-zur-nutzung (visited on 05/07/2025) (see page xxv).

[Uni25] Universitätsklinikum Essen. *Startseite | Universitätsklinikum Essen*. https://www.uk-essen.de/. Accessed: 2025-04-14. 2025 (see pages iii, v).

[Vas+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. **Attention is all you need**. *Advances in neural information processing systems* 30 (2017) (see page 9).

[Veg+19] A. van der Vegt, G. Zuccon, B. Koopman, and A. Deacon. **Impact of a Search Engine on Clinical Decisions Under Time and System Effectiveness Constraints: Research Protocol**. *JMIR Research Protocols* 8:5 (May 2019), e12803. DOI: 10.2196/12803. URL: https://www.researchprotocols.org/2019/5/e12803/ (see page 7).

[Ver+24] Pat Verga, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. *Replacing Judges with Juries: Evaluating LLM Generations with a Panel of Diverse Models*. arXiv.org, 2024. URL: https://arxiv.org/abs/2404.18796 (visited on 04/22/2025) (see pages iii, v, 47, 69).

[Vim+24] Sabrina De Capitani di Vimercati, Dario Facchinetti, Sara Foresti, Gianluca Oldani, Stefano Paraboschi, Matthew Rossi, and Pierangela Samarati. **Multi-dimensional flat indexing for encrypted data**. *IEEE Transactions on Cloud Computing* (2024) (see page 33).

[VPJ22] Abhishek Velankar, Hrushikesh Patil, and Raviraj Joshi. **Mono vs multilingual bert for hate speech detection and text classification: A case study in marathi**. In: *IAPR workshop on artificial neural networks in pattern recognition*. Springer. 2022, 121–128 (see page 72).

[VSP17] Ashish Vaswani, Noam Shazeer, and Niki Parmar. **Attention is All You Need. ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS : 31st annual conference on neural information process ...** Neural Info Process Sys F, 2017, 5998–6008 (see pages 10, 11, 31).

[Wan+19] Bin Wang, Angela Wang, Fenxiao Chen, Yuncheng Wang, and C-C Jay Kuo. **Evaluating word embedding models: Methods and experimental results**. *APSIPA transactions on signal and information processing* 8 (2019), e19 (see page 31).

[Wan+21] Jianguo Wang, Xiaomeng Yi, Rentong Guo, Hai Jin, Peng Xu, Shengjun Li, Xiangyu Wang, Xiangzhou Guo, Chengming Li, Xiaohai Xu, Kun Yu, Yuxing Yuan, Yinghao Zou, Jiquan Long, Yudong Cai, Zhenxiang Li, Zhifeng Zhang, Yihua Mo, Jun Gu, Ruiyi Jiang, Yi Wei, and Charles Xie. **Milvus: A Purpose-Built Vector Data Management System**. In: *Proceedings of the 2021 International Conference on Management of Data (SIGMOD '21)*. New York, NY, USA: Association for Computing Machinery (ACM), June 2021, 2633–2646. DOI: 10.1145/3448016.3457550. URL: https://doi.org/10.1145/3448016.3457550 (visited on 12/19/2021) (see pages 32, 33).

[Wan+23] Jiaqi Wang, Zhengliang Liu, Lin Zhao, Zihao Wu, Chong Ma, Sigang Yu, Haixing Dai, Qiushi Yang, Yiheng Liu, Songyao Zhang, et al. **Review of large vision models and visual prompt engineering**. *Meta-Radiology* 1:3 (2023), 100047 (see page 39).

[Wan+24a] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. *Multilingual E5 Text Embeddings: A Technical Report*. arXiv.org, 2024. URL: https://arxiv.org/abs/2402.05672 (see pages 57, 72, 79, 81, 83).

[Wan+24b] Wenmin Wang, Junpeng Ma, Peilin Zhang, Zhuolun Hu, Qi Jiang, and Yafei Liu. **Application of Multi-Way Recall Fusion Reranking Based on Tensor and ColBERT in RAG**. In: *2024 IEEE 7th International Conference on Information Systems and Computer Aided Education (ICISCAE)*. IEEE. 2024, 138–141 (see page 62).

[Wan+24c] Xiaohua Wang, Zhenghua Wang, Xuan Gao, Feiran Zhang, Yixin Wu, Zhibo Xu, Tianyuan Shi, Zhengyuan Wang, Shizheng Li, Qi Qian, et al. **Searching for best practices in retrieval-augmented generation**. *arXiv preprint arXiv:2407.01219* (2024) (see pages 3, 17, 24, 25, 28, 30, 33, 36, 39, 45, 59, 73, 81, 83).

[WC11] John S Whissell and Charles LA Clarke. **Improving document clustering using Okapi BM25 feature weighting**. *Information retrieval* 14 (2011), 466–487 (see pages 4, 37, 62, 65, 73, 79).

[Wei+22] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. **Emergent abilities of large language models**. *arXiv preprint arXiv:2206.07682* (2022) (see page 11).

[Wen+19] Andrew Wen, Sunyang Fu, Sungrim Moon, Mohamed El Wazir, Andrew Rosenbaum, Vinod C Kaggal, Sijia Liu, Sunghwan Sohn, Hongfang Liu, and Jungwei Fan. **Desiderata for delivering NLP to accelerate healthcare AI advancement and a Mayo Clinic NLP-as-a-service implementation**. *NPJ digital medicine* 2:1 (2019), 130 (see page 1).

[WFG24] Shaoyuan Weng, Zongwen Fan, and Jin Gou. **A fast DBSCAN algorithm using a bi-directional HNSW index structure for big data**. *International Journal of Machine Learning and Cybernetics* 15:8 (2024), 3471–3494 (see pages 34, 59).

[Whi+23] Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. **A prompt pattern catalog to enhance prompt engineering with chatgpt**. *arXiv preprint arXiv:2302.11382* (2023) (see pages 38, 39).

[WJ96] David A White and Ramesh C Jain. **Similarity indexing: Algorithms and performance**. In: *Storage and Retrieval for Still Image and Video Databases IV*. Vol. 2670. SPIE. 1996, 62–73 (see pages 32, 33, 36).

[WKR25] Zhengxiang Wang, Jordan Kodner, and Owen Rambow. **Exploring Limitations of LLM Capabilities with Multi-Problem Evaluation**. In: *The Sixth Workshop on Insights from Negative Results in NLP*. 2025, 121–140 (see pages 10, 23).

[Wu+24a] Junde Wu, Jiayuan Zhu, Yunli Qi, Jingkun Chen, Min Xu, Filippo Menolascina, and Vicente Grau. **Medical graph rag: Towards safe medical large language model via graph retrieval-augmented generation**. *arXiv preprint arXiv:2408.04187* (2024) (see pages 40, 78, 80).

[Wu+24b] Kevin Wu, Eric Wu, Ally Cassasola, Angela Zhang, Kevin Wei, Teresa Nguyen, Sith Riantawan, Patricia Shi Riantawan, Daniel E Ho, and James Zou. **How well do LLMs cite relevant medical references? An evaluation framework and analyses**. *arXiv preprint arXiv:2402.02008* (2024) (see page 25).

[Wu+24c] Siyu Wu, Alessandro Oltramari, Jonathan Francis, C Lee Giles, and Frank E Ritter. **Cognitive LLMs: Toward Human-Like Artificial Intelligence by Integrating Cognitive Architectures and Large Language Models for Manufacturing Decision-making**. *Neurosymbolic Artificial Intelligence* (2024) (see page 9).

[Wu+24d] Suhang Wu, Jialong Tang, Baosong Yang, Ante Wang, Kaidi Jia, Jiawei Yu, Junfeng Yao, and Jinsong Su. **Not All Languages are Equal: Insights into Multilingual Retrieval-Augmented Generation**. *arXiv (Cornell University)* (Oct. 2024). DOI: 10.48550/arxiv.2410.21970. (Visited on 04/19/2025) (see page 3).

[WW24] Bartosz Walkowiak and Tomasz Walkowiak. **Assessing Inference Time in Large Language Models**. In: *International Conference on Dependability of Computer Systems*. Springer. 2024, 296–305 (see page 58).

[Xia+93] Peng Xia, Kangyu Zhu, Haoran Li, Hongtu Zhu, Yun Li, Gang Li, Linjun Zhang, and Huaxiu Yao. *RULE: Reliable Multimodal RAG for Factuality in Medical Vision Language Models*. 1093. URL: https://aclanthology.org/2024.emnlp-main.62.pdf (visited on 04/22/2025) (see page 25).

[Xio+24] Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. **Benchmarking retrieval-augmented generation for medicine**. In: *Findings of the Association for Computational Linguistics ACL 2024*. 2024, 6233–6251 (see page 43).

[Yan+24] Hang Yang, Jing Guo, Jianchuan Qi, Jinliang Xie, Si Zhang, Siqi Yang, Nan Li, and Ming Xu. **A Method for Parsing and Vectorization of Semi-structured Data used in Retrieval Augmented Generation**. *arXiv preprint arXiv:2405.03989* (2024) (see pages 28, 29).

[Yan+25] Rui Yang, Yilin Ning, Emilia Keppo, Mingxuan Liu, Chuan Hong, Danielle S Bitterman, Jasmine Chiat, Daniel Shu, and Nan Liu. **Retrieval-augmented generation for generative artificial intelligence in health care**. *npj Health Systems* 2 (Jan. 2025), 1–5. DOI: 10.1038/s44401-024-00004-1. URL: https://www.nature.com/articles/s44401-024-00004-1 (visited on 01/25/2025) (see page 17).

[Yeh+23] Kai-Ching Yeh, Jou-An Chi, Da-Chen Lian, and Shu-Kai Hsieh. **Evaluating interfaced llm bias**. In: *Proceedings of the 35th Conference on Computational Linguistics and Speech Processing (ROCLING 2023)*. 2023, 292–299 (see page 13).

[Yep+24] Antonio Jimeno Yepes, Yao You, Jan Milczek, Sebastian Laverde, and Renyu Li. **Financial report chunking for effective retrieval augmented generation**. *arXiv preprint arXiv:2402.05131* (2024) (see pages 28, 30, 35, 82).

[YM98] Clement T. Yu and Weiyi Meng. **Principles of Database Query Processing for Advanced Applications**. The Morgan Kaufmann Series in Data Management Systems. San Francisco, CA: Morgan Kaufmann Publishers Inc., 1998. ISBN: 1558604340 (see page 25).

[Yu+24] Jun Yu, Yunxiang Zhang, Zerui Zhang, Zhao Yang, Gongpeng Zhao, Fengzhao Sun, Fanrui Zhang, Qingsong Liu, Jianqing Sun, Jiaen Liang, and Yaohui Zhang. **RAG-Guided Large Language Models for Visual Spatial Description with Adaptive Hallucination Corrector**. *Proceedings of the 32nd ACM International Conference on Multimedia* (Oct. 2024), 11407–11413. DOI: 10.1145/3664647.3688990. (Visited on 04/21/2025) (see page 37).

[Zem19] MgrT Zemčík. **A brief history of chatbots**. *DEStech Transactions on Computer Science and Engineering* 10 (2019), 14–18 (see page 9).

[Zha+24] Tianjun Zhang, Shishir G Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E Gonzalez. **RAFT: Adapting Language Model to Domain Specific RAG**. *arXiv (Cornell University)* (Mar. 2024). DOI: 10.48550/arxiv.2403.10131 (see pages 38, 84).

[Zhe+23] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. **Judging llm-as-a-judge with mt-bench and chatbot arena**. *Advances in Neural Information Processing Systems* 36 (2023), 46595–46623 (see pages 44, 47).

[Zho+14] Wei Zhou, Li Li, Min Luo, and Wu Chou. **REST API design patterns for SDN northbound API**. In: *2014 28th international conference on advanced information networking and applications workshops*. IEEE. 2014, 358–365 (see page 60).

[Zho+24a] Zijie Zhong, Hanwen Liu, Xiaoya Cui, Xiaofan Zhang, and Zengchang Qin. **Mix-of-granularity: Optimize the chunking granularity for retrieval-augmented generation**. *arXiv preprint arXiv:2406.00456* (2024) (see pages 29, 60, 78).

[Zho+24b] Yujia Zhou, Yan Liu, Xiaoxi Li, Jiajie Jin, Hongjin Qian, Zheng Liu, Chaozhuo Li, Zhicheng Dou, Tsung-Yi Ho, and Philip S Yu. **Trustworthiness in retrieval-augmented generation systems: A survey**. *arXiv preprint arXiv:2409.10102* (2024) (see pages 24, 25, 40, 84).

[Zhu+24] Kunlun Zhu, Yifan Luo, Dingling Xu, Ruobing Wang, Shi Yu, Shuo Wang, Yukun Yan, Zhenghao Liu, Xu Han, Zhiyuan Liu, and Maosong Sun. **RAGEval: Scenario Specific RAG Evaluation Dataset Generation Framework**. *arXiv (Cornell University)* (Aug. 2024). DOI: 10.48550/arxiv.2408.01262 (see pages 39, 43).

Declaration of Authorship

I hereby declare that this thesis is my own unaided work. In accordance with the University's guidelines[Uni24], I made limited use of OpenAI generative-AI models (GPT-4o and o3) [Ope20] solely to refine language and LaTeX formatting. All direct and indirect sources have been fully acknowledged in the references.

Munich, May 31, 2025

Fahad Deshmukh