ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DEPARTMENT OF COMPUTER SCIENCE
AND ENGINEERING

ARTIFICIAL INTELLIGENCE

MASTER THESIS

in

Natural Language Processing

DESIGN, IMPLEMENTATION AND BENCHMARKING OF A RETRIEVAL-AUGMENTED CHATBOT FOR THE INSURANCE SECTOR

CANDIDATE
Tancredi Bosi

SUPERVISOR
Prof. Paolo Torroni

CO-SUPERVISOR
Marco Vita

Academic year 2024-2025

Session 2nd

To myself, and to all those I love.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The insurance sector is increasingly dealing with vast amounts of complex, domain-specific documentation, such as policy contracts, clauses, and regulatory materials. Navigating these documents efficiently is a daily challenge for professionals, who must identify and interpret relevant technical content through a large choice of options. Recent advances in Artificial Intelligence (AI), especially in natural language processing (NLP), offer new tools to assist with this task. Among them, large language models (LLMs) have demonstrated the ability to understand, synthesize, and interact with unstructured text. In particular, Retrieval-Augmented Generation (RAG) systems have emerged as a powerful approach to grounding LLMs in external knowledge, making them especially suitable for tasks involving specialized document collections such as insurance policies.

This thesis presents the design, implementation, and evaluation of a chatbot assistant based on a RAG architecture. The assistant aims to help insurance professionals access technical information contained in policy documents more effectively. The chatbot was developed as part of a full-stack AI-driven application built during an internship experience, leveraging the OpenAI API [17] for language understanding and generation.

1.1 Problem Framing and Motivations

The insurance industry presents unique challenges for information access and document management that create compelling opportunities for artificial intelligence solutions. This section examines the specific domain characteristics that motivate the development of intelligent document processing systems, particularly focusing on the needs of insurance professionals who must navigate complex documentation landscapes in their daily work.

1.1.1 Insurance Domain

Insurance professionals operate within an intricate ecosystem of policy documents, regulatory materials, and comparative resources that span multiple sectors and companies. The documentation includes policies from life, health, property, casualty, and specialty insurance lines, each characterized by distinct terminology, coverage structures, and regulatory requirements. This diversity creates significant challenges for efficient information access and analysis.

Traditional information retrieval in insurance relies heavily on manual search processes that present fundamental limitations. The volume of documentation makes comprehensive analysis time-intensive, while specialized terminology requires domain expertise for accurate interpretation. Most critically, comparative analysis across multiple policies or providers demands extensive manual effort to identify relevant clauses and coverage differences.

Insurance brokers serve as intermediaries between clients and providers, requiring rapid access to accurate information for coverage explanations, recommendations, and comparisons. Their operational efficiency directly depends on the speed of information retrieval while maintaining information accuracy, delays translate to reduced service quality and potential revenue loss.

As technology continues to evolve, organizations across various industries face the ongoing challenge of adapting to new tools and methodologies that

can improve operational efficiency. The insurance brokerage sector represents a particularly compelling domain for technological innovation, where the application of AI and NLP technologies offers significant potential to transform how professionals access, interpret, and utilize complex documentation. Traditional methods of document consultation, which rely heavily on manual search processes and human expertise, often create bottlenecks that limit both efficiency and analytical depth, being that the knowledge required is highly specialized and the documentation is extensive. The introduction of AI-powered systems, particularly those utilizing NLP capabilities, presents an opportunity to augment human expertise rather than replace it, enabling professionals to focus on high-value analytical tasks while delegating routine information retrieval to automated systems.

Discussions with the client who commissioned this project revealed several specific objectives for developing an AI-powered system to assist insurance professionals. The client articulated the primary goal as creating a solution that allows insurance brokers to access relevant information from policy documents with greater speed than traditional manual methods. First, the solution should significantly reduce the time required for complex analytical tasks that traditionally demand high levels of domain expertise. These time-intensive activities represent substantial operational costs and limit the number of clients that each broker can serve effectively. Second, the system must facilitate rapid information retrieval across extensive document collections. Insurance professionals frequently need to locate specific clauses, coverage details, or policy conditions buried within lengthy documents, a process that currently requires substantial manual effort and specialized knowledge of document organization patterns. Third, the client identified comparative analysis as a critical capability, specifically the need to efficiently compare document details across different insurance providers. This includes systematic comparison of coverage limits, policy exclusions, premium structures,

deductibles, and contractual clauses. Such comparative analysis currently requires extensive manual effort to extract and organize relevant information from multiple sources. Fourth, the solution should provide policy explanation capabilities that translate complex insurance terminology and legal language into more accessible terms for client communication. Insurance brokers must frequently explain intricate policy details to clients who lack specialized insurance knowledge, requiring the ability to simplify technical content without losing accuracy or important clauses. Finally, the system must support comprehensive summarization of lengthy policy documents, enabling brokers to quickly understand key provisions and identify critical information without reading entire documents. This summarization capability should maintain accuracy while highlighting the most relevant aspects for specific client needs or analytical purposes.

1.1.2 User Profile

The primary users of this system are insurance brokers who work as intermediaries between clients and insurance providers across multiple sectors. These professionals possess deep domain expertise in insurance products, regulatory requirements, and market dynamics, yet face significant productivity challenges when accessing and analyzing the extensive documentation that characterizes their industry.

Insurance brokers typically manage portfolios spanning multiple insurance lines including automotive, travel, health, property, and specialty coverage products. Their daily responsibilities require rapid access to specific policy details, coverage comparisons across providers, and accurate interpretation of complex contractual language for client communication. The traditional approach to these tasks involves manual document review, which creates substantial time overhead and limits the number of clients each broker can serve effectively.

Brokers require the ability to quickly locate specific clauses within lengthy policy documents, compare coverage terms across multiple providers, and extract key information for client presentations and recommendations. They must also translate complex insurance terminology into accessible language for client communication while maintaining technical accuracy. The system users expect to receive precise references to source documents, enabling them to verify information independently and maintain the transparency standards required in their professional relationships.

The application serves brokers who work with diverse client bases ranging from individual consumers to commercial enterprises, each presenting unique coverage requirements and analytical challenges. These professionals must navigate regulatory variations across different insurance sectors while maintaining current knowledge of policy changes and market developments that affect their recommendations.

1.1.3 Functional System Requirements

The system must address several core functional capabilities to deliver value to insurance professionals. Natural language query processing represents the primary requirement, enabling users to pose questions using everyday language rather than structured queries or specific keywords. This capability must support both simple information retrieval and complex analytical queries requiring synthesis across multiple document sections.

Cross-document comparison functionality is essential for analyzing differences and similarities between policies from different providers, product lines, or different versions of the same policy. Users need to identify coverage gaps, compare premium structures, and highlight variations in terms and conditions while maintaining context awareness during comparisons.

Source attribution and reference provision ensures transparency and enables independent information verification. Every system response must include clear references to specific document sections, page numbers, or clauses supporting the provided information. This requirement proves particularly critical in insurance applications where accuracy and verifiability are non-negotiable.

Conversational interaction allows follow-up questions and clarification requests within the same session context. Users should refine queries, request additional details, or explore related topics without losing conversation context.

1.1.4 Use Case

The fundamental use case involves insurance brokers seeking specific information about coverage terms, conditions, or limitations within insurance policies. Typical scenarios begin when brokers receive client inquiries about coverage details, calculations, or policy comparisons. Instead of manually searching through multiple documents, brokers can pose natural language questions such as ”What are the coverage limits for water damage in the XYZ Insurance homeowner’s policy?” or ”Does the ABC Life Insurance policy include accidental death benefits?”

The system processes these queries through the RAG pipeline, finding relevant document sections and generating comprehensive responses with direct references to specific policy clauses, page numbers, and document sources. This approach enables brokers to retrieve valid information in seconds, while still having the reference pages of the original documents used to generate the response to check the validity of the response.

1.2 Objectives

The analysis of domain requirements and client needs establishes the foundation for a comprehensive research investigation that extends beyond commercial system development to address fundamental questions about RAG system performance in specialized professional domains. This research pursues four interconnected objectives that collectively contribute to both practical system deployment and academic understanding of retrieval-augmented generation in insurance applications.

The first objective focuses on developing and implementing a production-ready RAG-based system tailored specifically for insurance domain applications. This involves designing an architecture that can effectively process diverse insurance documentation, handle domain-specific terminology and complex policy structures, and provide reliable source attribution for professional use. The implementation challenge requires balancing system sophistication with practical deployment constraints, ensuring that the resulting solution meets both technical performance standards and operational requirements for insurance professionals.

The second objective addresses the critical need for domain-specific evaluation methodologies by constructing a comprehensive benchmark framework designed for insurance document question-answering systems. This involves collaborating with domain experts to develop realistic query scenarios, establishing ground truth annotations that reflect professional standards, and creating evaluation protocols that capture the unique requirements of insurance applications. The benchmark must incorporate multi-document analysis capabilities, precise source attribution requirements, and coverage across diverse insurance product lines. This objective contributes to the broader research community by providing the first publicly available evaluation framework specifically designed for Italian insurance domain applications.

The third objective investigates the limitations and constraints of RAG

systems when deployed in specialized professional domains. This analysis examines both technical limitations arising from retrieval quality dependencies, context length constraints, and embedding model capabilities, as well as practical limitations related to domain knowledge coverage, terminology handling, and complex reasoning requirements. Through systematic analysis of system failures, edge cases, and performance boundaries, this investigation aims to establish a comprehensive understanding of when and how RAG systems may fall short of professional requirements, providing valuable insights for both system designers and potential users.

The fourth objective evaluates the applicability and effectiveness of established evaluation metrics developed for general-domain applications when applied to specialized insurance contexts. This investigation examines whether standard retrieval metrics, generation quality measures, and automated evaluation approaches adequately capture the performance dimensions critical for professional insurance applications. The analysis includes assessment of metric correlation with expert judgments, identification of domain-specific evaluation requirements not captured by existing approaches, and development of recommendations for evaluation methodology adaptation in professional domains. This objective addresses a fundamental gap in the literature regarding the transferability of evaluation approaches across domain boundaries.

These objectives collectively establish a research framework that advances both practical system capabilities and theoretical understanding of RAG system performance in professional applications. The interconnected nature of these goals ensures that practical implementation insights inform evaluation methodology development, while rigorous evaluation approaches enable systematic analysis of system limitations and metric applicability. Through this approach, the research contributes to the growing body of knowledge surrounding the deployment of advanced language technologies in specialized professional contexts.

1.3 Structure of the Thesis

This thesis is organized into seven chapters that collectively document the complete development cycle of a RAG-based insurance document chatbot, from theoretical foundations through production deployment. The structure reflects both the academic rigor required for comprehensive evaluation and the practical considerations necessary for commercial implementation, mirroring the dual nature of work conducted during an internship experience where research objectives must align with business requirements.

The investigation begins with the background necessary for understanding RAG systems and their evaluation methodologies. This theoretical exploration examines the current state of the art in retrieval-augmented generation. Particular attention is given to the advantages and limitations of RAG architectures in professional applications, while establishing the methodological framework for systematic performance assessment. This section concludes with an examination of OpenAI's RAG implementation, which serves as the technical foundation for the developed system.

The research then turns to documenting the construction of an evaluation framework designed specifically for insurance domain applications. This work details the collaborative process with insurance experts to create realistic benchmark questions and reference answers, addressing the limitations of existing evaluation datasets that lack domain specificity and source attribution requirements. Through comprehensive analysis of design principles, expert requirements, and benchmark composition, this section establishes the evaluation foundation that enables rigorous assessment of system performance against professional standards.

With the evaluation framework established, the thesis presents the technical implementation details of the insurance document chatbot system. This technical exposition describes the document processing pipeline that transforms uploaded insurance policies into searchable knowledge representations

and the retrieval-augmented generation module that processes natural language queries. Through detailed examination of OpenAI integration, document storage mechanisms, and the streaming query interface, this discussion provides the technical foundation necessary for understanding system capabilities and operational characteristics.

The experimental evaluation follows, presenting a comprehensive assessment of system performance across multiple dimensions. This analysis examines retrieval effectiveness, generation quality, expert evaluations, and automated assessment results to provide a complete picture of system capabilities. Through statistical analysis, correlation studies, and qualitative case studies, the evaluation demonstrates system readiness for professional deployment while identifying specific strengths and areas for continued improvement.

The work then documents the transition from research prototype to production system, detailing the commercial web application architecture and deployment infrastructure. This section examines the AWS-based hosting environment, CI/CD pipeline implementation, and user interface design that enable the system to serve insurance professionals in operational environments, demonstrating the practical viability of the research through successful deployment and ongoing operational support.

The thesis concludes by synthesizing the research findings and establishing their implications for both academic understanding and practical applications. This final analysis evaluates the success of the implemented solution against the original research objectives while identifying limitations and opportunities for future work.

The appendices provide essential supporting materials including the complete system prompt, detailed LLM evaluation prompts, and technical specifications that enable reproducibility and support ongoing development efforts. These materials ensure that the research can be effectively validated and extended by future investigators, while the organizational structure reflects the comprehensive nature of the research project.

Chapter 2

Background

2.1 Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) represents a paradigm shift in natural language processing that addresses fundamental limitations of traditional large language models by combining parametric knowledge stored in neural networks with external, dynamically retrievable information sources. RAG systems implement a two-stage architecture where a retrieval component first identifies relevant contextual information from external knowledge bases, followed by a generation component that utilizes both the retrieved context and the model's internal knowledge to produce informed responses.

The core innovation of RAG lies in its ability to ground language model outputs in verifiable, up-to-date external sources while maintaining the sophisticated reasoning capabilities of modern LLMs. This approach enables systems to access information beyond their training cutoffs, incorporate domain-specific knowledge not present in general training data, and provide explicit source attribution for generated content. The retrieval component typically employs dense vector representations and similarity search mechanisms to identify relevant document segments, while the generation component leverages transformer-based architectures to synthesize coherent responses that integrate retrieved context with query-specific reasoning.

2.1.1 Current State of the Art in RAG

The current state of the art in RAG systems has evolved significantly beyond simple retrieve-then-generate approaches, encompassing sophisticated multi-step reasoning, adaptive retrieval strategies, and autonomous decision-making capabilities that optimize information gathering based on query complexity and domain requirements. Modern implementations incorporate hierarchical retrieval mechanisms that operate at multiple levels of granularity, from document-level relevance assessment to fine-grained passage extraction and entity-specific information retrieval, while employing re-ranking mechanisms utilizing cross-encoder models to refine initial retrieval results and significantly improve response quality in specialized domains where precision and accuracy are paramount.

Contemporary RAG architectures have introduced innovative paradigms that address fundamental limitations of earlier systems. Self-RAG introduces self-reflective mechanisms where systems evaluate their own outputs for accuracy and relevance, implementing quality control loops within the generation process to reduce hallucinations and improve response reliability. This introspective capability enables systems to detect when retrieved information may be insufficient or contradictory, prompting additional retrieval operations or uncertainty indicators in responses. Adaptive RAG represents another significant advancement, dynamically adjusting retrieval strategies based on query characteristics and switching between different retrieval modes or combining multiple approaches to optimize performance across diverse use cases.

The development of Corrective RAG has addressed quality assurance concerns by implementing verification mechanisms to validate retrieved information before generation, while Hybrid RAG systems achieve optimal performance by combining vector search with keyword-based methods, leveraging both semantic and lexical matching capabilities. These hybrid methods recognize that different types of queries benefit from different retrieval strategies,

with factual queries often requiring precise keyword matching while conceptual queries benefit more from semantic similarity approaches.

GraphRAG [6] represents the most fundamental advancement in recent years by incorporating knowledge graphs into the retrieval process, creating interconnected networks that enable multi-hop reasoning across entity relationships. Instead of treating documents as isolated chunks, GraphRAG facilitates global sensemaking and contextual understanding by leveraging structured knowledge representations that capture relationships between entities, concepts, and facts across the entire document corpus. Microsoft's GraphRAG implementation demonstrates remarkable efficiency gains, achieving 26-97% token reduction compared to traditional approaches while maintaining superior accuracy through enhanced reasoning capabilities that can follow chains of relationships and synthesize information from disparate sources.

The integration of structured data sources with unstructured text retrieval has enabled more comprehensive responses for complex analytical queries, proving particularly valuable in professional domains where both narrative descriptions and structured specifications must be considered simultaneously. This hybrid knowledge representation approach allows systems to understand not only what information exists but how different pieces of information relate to each other, enabling more sophisticated analysis and reasoning capabilities.

Multimodal RAG is another frontier in the field, extending capabilities beyond text to process images, audio, and video content, enabling cross-modal understanding where systems can retrieve text based on image queries or vice versa. This approach supports rich content analysis and enhanced user experiences across diverse input modalities, allowing users to query document collections using whatever format is most natural for their specific needs. The ability to process and understand relationships between different media types opens new possibilities for comprehensive document analysis in fields where visual information, such as charts, diagrams, and photographs, plays a crucial role alongside textual content.

Agentic RAG systems are at the leading position of autonomous decision-making within the retrieval-generation pipeline, featuring dynamic query reformulation based on initial results, multi-step reasoning that breaks complex queries into manageable sub-tasks, and autonomous tool usage leveraging external APIs and services as needed. These systems demonstrate sophisticated planning capabilities that enable decomposition of complex queries into systematic information gathering processes, where autonomous agents can determine what information is needed, identify the best sources for that information, and iteratively refine their approach based on intermediate results. The emergence of such systems suggests a future where RAG architectures can handle increasingly complex analytical tasks with minimal human guidance while maintaining high accuracy and reliability standards.

2.1.2 Advantages of RAG Systems

RAG systems present advantages over both traditional language models and conventional information retrieval approaches, particularly when deployed in professional environments where accuracy and verifiable source attribution are essential. The fundamental strength of these systems lies in their ability to seamlessly integrate semantic understanding with factual grounding, producing responses that maintain contextual appropriateness while remaining anchored to specific source materials.

The most significant benefit stems from dynamic knowledge access capabilities. Traditional language models become increasingly outdated as their training data ages, creating knowledge gaps that grow over time. RAG systems circumvent this limitation by continuously incorporating new information without requiring expensive model retraining or complex fine-tuning procedures. This characteristic proves invaluable in rapidly evolving domains like insurance, where policy terms, regulatory frameworks, and market conditions undergo frequent changes. Organizations can maintain system currency

simply by updating their document collections, ensuring that responses reflect the most recent information without technical overhead.

Source attribution and transparency represent another critical advantage, especially in professional contexts where decision-making requires clear justification and comprehensive audit trails. RAG systems naturally provide explicit references to the specific document sections, page numbers, and sources that inform their responses. This transparency enables users to independently verify information and maintain compliance with professional standards. In insurance applications, this capability becomes particularly valuable given the potential consequences of policy misinterpretation, where errors can result in substantial financial and legal implications.

The approach facilitates efficient domain specialization through careful document collection curation rather than expensive model training processes. Organizations can rapidly deploy specialized systems across different domains by incorporating relevant document collections while preserving the underlying reasoning capabilities. This methodology significantly reduces deployment timelines and provides cost-effective customization options tailored to specific organizational requirements, avoiding the complexity and expense traditionally associated with model fine-tuning procedures.

Perhaps most importantly for professional applications, RAG based systems demonstrate improved factual accuracy through grounding generation processes in retrieved context rather than relying exclusively on parametric model knowledge. While complete elimination of hallucination remains impossible, RAG implementations show substantial improvements in factual reliability when retrieval components successfully identify relevant source material. This enhancement becomes especially pronounced in technical domains characterized by specialized terminology and precise factual requirements, where accuracy represents a non-negotiable requirement.

The architectural separation of knowledge storage and reasoning capabilities delivers significant scalability and cost efficiency benefits. Organizations can maintain extensive, searchable document collections without requiring proportionally larger language models, while benefiting from shared infrastructure for document processing and retrieval operations. This separation enables cost-effective scaling as document volumes expand, making the technology accessible to organizations across different sizes and budget constraints.

2.1.3 Limitations and Challenges of RAG Systems

Despite their compelling advantages, RAG systems encounter several inherent limitations and implementation challenges that require careful consideration during system design and deployment planning. These constraints frequently emerge from the intricate coordination required between retrieval and generation components while maintaining the performance and accuracy standards essential for commercial applications.

The most fundamental limitation stems from retrieval quality dependency, where RAG system performance remains inherently bounded by the effectiveness and comprehensiveness of the retrieval component. When retrieval operations produce poor results due to inadequate document coverage, suboptimal chunking strategies, or embedding model limitations, generation quality suffers proportionally. This dependency creates potential failure scenarios where relevant information exists within the document collection but cannot be successfully retrieved, ultimately leading to incomplete or inaccurate responses.

Context length limitations present another significant constraint, restricting the volume of retrieved information that can be incorporated during generation. This becomes particularly problematic when addressing complex queries that require synthesis across multiple document sections. Current

transformer architectures impose practical limits on context length, necessitating careful selection and summarization of retrieved content. Insurance applications face particular challenges in this regard, as comprehensive policy analysis often demands consideration of multiple, lengthy document sections that may exceed available context capacity.

Coherence and integration challenges emerge when systems attempt to synthesize information from multiple retrieved sources into coherent, comprehensive responses. RAG systems frequently struggle to resolve contradictions between different sources, maintain consistent terminology across various documents, or provide appropriately weighted consideration of conflicting information.

Embedding model limitations affect retrieval quality, especially when processing specialized terminology or cross-domain queries. Embedding models trained on general corpora may fail to capture semantic relationships specific to specialized domains, resulting in suboptimal retrieval performance. Additionally, these models typically process text in isolation, potentially missing important contextual relationships that span multiple document sections or require understanding of domain-specific conceptual frameworks that influence meaning interpretation within professional contexts.

2.1.4 RAG with OpenAI Responses API

The insurance document understanding system leverages OpenAI's Responses API with the **file_search** tool, which represents a comprehensive, production-ready RAG implementation. Unlike traditional approaches that require building and maintaining separate vector databases and retrieval components, OpenAI's file_search tool provides a fully managed RAG architecture that handles document processing, embedding generation, vector storage, and sophisticated retrieval mechanisms within a unified system.

OpenAI's implementation fundamentally operates as a complete RAG system that augments language models with knowledge from external documents through automatic parsing, chunking, embedding, and indexing processes. When integrated with the Responses API, this approach enables stateless RAG operations particularly suited for web applications where each user query represents an independent request requiring immediate response generation.

The OpenAI file_search tool implements a sophisticated multi-stage RAG pipeline that begins with comprehensive document processing. Documents uploaded to the system undergo automatic parsing and chunking using configurable parameters, with default settings of 800 tokens per chunk and 400 tokens overlap. This chunking strategy balances contextual coherence with retrieval precision, ensuring that insurance policy clauses and technical terms remain analyzable within individual chunks.

Embedding generation utilizes the text-embedding-3-large model at 256 dimensions by default, creating vector representations optimized for semantic similarity search. These embeddings are stored within OpenAI's managed vector database infrastructure, eliminating the complexity of maintaining independent vector storage systems. Each vector store can accommodate up to 10,000 files with individual file limits of 512 MB and 5,000,000 tokens, providing sufficient capacity for comprehensive insurance document collections.

The query processing mechanism implements several techniques that enhance retrieval quality beyond simple vector similarity. User queries undergo automatic optimization and rewriting to improve search precision, while complex queries are decomposed into multiple parallel searches. The system performs hybrid search operations that combine both semantic vector search and keyword matching, ensuring that both contextual meaning and exact term matches contribute to retrieval results. A critical component of the system involves re-ranking mechanisms that process initial search results to select the most relevant chunks before generation. This re-ranking process considers multiple relevance signals and can be configured with score thresholds

ranging from 0.0 to 1.0 to filter low-quality matches. The final step involves context injection, where up to 20 chunks by default are incorporated into the model context for response generation.

The file_search tool can process multiple document formats, including PDF, DOCX, and PPTX files: the system automatically handles format conversion and text extraction, guaranteeing consistent processing regardless of source document characteristics.

Response latency characteristics meet the performance requirements of commercial applications, with typical response times falling within acceptable ranges for professional workflow integration. The managed infrastructure handles scaling automatically, ensuring consistent performance regardless of document collection size or concurrent user loads.

The file_search implementation provides source attribution through automatic citation generation, addressing the critical requirement for transparency and verifiability in insurance applications. Generated responses include references to specific document sections and page numbers, enabling insurance professionals to verify information independently and maintain compliance with professional standards.

2.1.5 Standard RAG Metrics and Benchmarks

Evaluating RAG systems requires comprehensive frameworks that assess both retrieval effectiveness and generation quality, leading to the development of standardized benchmarks and evaluation methodologies that enable systematic comparison across different approaches and domains. The evaluation landscape includes both general-purpose information retrieval benchmarks adapted for RAG assessment and specialized frameworks designed specifically for end-to-end RAG evaluation.

BeIR (Benchmarking Information Retrieval) [26] is an evaluating retrieval systems across diverse domains and tasks. The benchmark encompasses 18

datasets spanning multiple domains including question answering (Natural Questions, TriviaQA), fact verification (FEVER, Climate-FEVER), citation prediction (SciFact), and specialized domains (FiQA for finance, NFCorpus for nutrition). This diversity enables zero-shot evaluation of retrieval models across different text types, query formulations, and relevance criteria. BeIR evaluation methodology employs standard information retrieval metrics including Recall@k, Precision@k, nDCG@k, and Mean Average Precision, measuring retrieval performance at various cut-off points typically ranging from k=1 to k=1000. The benchmark's heterogeneous nature reveals important insights about model generalization capabilities, often showing that simple lexical approaches like BM25 maintain competitive performance across diverse domains while sophisticated neural approaches may suffer from domain transfer limitations.

RAGAS (Retrieval Augmented Generation Assessment) [7] addresses the specific challenges of evaluating end-to-end RAG systems by providing a framework for reference-free evaluation that does not require ground truth human annotations. This approach proves particularly valuable for rapid iteration and development cycles where obtaining expert annotations would be prohibitively expensive or time-consuming. The framework introduces several key metrics designed specifically for RAG evaluation. Faithfulness measures whether generated answers remain grounded in retrieved context, detecting hallucinations that represent a critical concern for professional applications. Answer relevance assesses how well generated responses address user queries, while context precision evaluates the quality of retrieved passages and context recall measures retrieval comprehensiveness. These metrics leverage large language models as evaluators, employing carefully designed prompts to assess different dimensions of RAG system performance.

Many RAG evaluations adapt established question-answering datasets including Natural Questions, TriviaQA, and MS MARCO for end-to-end assessment. Natural Questions provides realistic queries derived from actual

Google searches with Wikipedia-based answers, enabling evaluation of factual knowledge retrieval and synthesis capabilities. TriviaQA focuses on reading comprehension across web and Wikipedia sources, testing systems' ability to locate and extract specific information from longer documents.

Standard RAG evaluation methodologies must address several fundamental challenges that distinguish this assessment from traditional information retrieval or text generation evaluation. The interdependence between retrieval and generation components creates complex failure modes where poor retrieval can mask generation capabilities or where strong retrieval cannot compensate for generation failures. Synthetic query generation represents an important methodological consideration, where large language models generate evaluation questions from document collections to scale benchmark construction. However, synthetic queries may not capture the complexity and ambiguity characteristics of real user information needs, potentially overestimating system performance on carefully constructed test cases while underestimating challenges encountered in operational deployment.

The choice between automatic metrics and human evaluation presents additional trade-offs between scalability and accuracy. While automated evaluation enables rapid iteration and large-scale comparison, human assessment remains essential for capturing nuanced quality dimensions such as professional utility, clarity, and appropriateness for specific use cases. The development of reliable automated proxies for human judgment represents an active area of research with significant implications for practical RAG system development and deployment.

2.2 Metrics

Building upon the standard RAG evaluation frameworks discussed previously, this section presents the specific evaluation methodology employed to assess the insurance document understanding system. The evaluation framework

implemented in this study adapts and extends conventional RAG assessment approaches, including granular document-page level retrieval assessment, semantic similarity measures, and LLM response evaluation that considers three different aspects.

2.2.1 Retrieval Metrics

Retrieval metrics evaluate how effectively the system identifies relevant documents and their specific page references in response to user queries. These metrics are fundamental to RAG system evaluation because retrieval quality directly impacts generation performance. Poor retrieval results inevitably lead to inadequate responses, regardless of the language model's capabilities. The implementation of retrieval metrics in this evaluation framework operates at a granular level, considering both document-level and page-level accuracy. Rather than treating entire documents as single units, the system evaluates retrieval performance based on specific document-page pairs, recognizing that relevant information often resides within particular sections or pages of documents.

It is important to note that ranking-based metrics are not implemented in this evaluation framework. This decision is motivated by the high threshold configuration employed in the retrieval system, where only retrievals with high confidence scores are considered for response generation. Given this high threshold approach, the system typically returns a relatively small set of highly relevant document references rather than a ranked list requiring position-based evaluation. The focus thus shifts to precision and recall metrics that better capture the binary relevance assessment suited to this retrieval strategy.

Recall measures the system's ability to capture all the important information needed to answer a question properly. This metric examines what proportion of the truly relevant document-page pairs actually appear among the results returned by the system. To understand this calculation, two key

sets are defined: *Relevant* represents all document-page pairs that contain information essential for answering the query, while *Retrieved* represents the specific document-page pairs that the system returns. The recall calculation becomes:

$$\text{Recall} = \frac{|\text{Relevant} \cap \text{Retrieved}|}{|\text{Relevant}|} \tag{2.1}$$

This counts how many relevant items were successfully retrieved and divides by the total number of relevant items that should have been found.

Precision examines retrieval accuracy by measuring what proportion of returned results actually helps answer the query. Using the same variables, precision calculates:

$$\text{Precision} = \frac{|\text{Relevant} \cap \text{Retrieved}|}{|\text{Retrieved}|} \tag{2.2}$$

This divides the number of useful retrieved items by the total number of items returned. High precision means the system filters out noise effectively, while low precision indicates too much irrelevant information mixed with useful results.

The F1 score combines recall and precision into a single balanced measure through harmonic mean calculation:

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.3}$$

This metric proves useful when comparing different retrieval configurations or when neither precision nor recall alone provides sufficient insight. High F1 scores indicate systems that both find relevant information comprehensively and filter out irrelevant content effectively.

2.2.2 Generation Metrics

Generation metrics evaluate the quality of text produced by the RAG system, examining how well the system synthesizes retrieved information into coherent, informative responses that insurance professionals can rely on in their work.

ROUGE Scores (Recall-Oriented Understudy for Gisting Evaluation) allow the evaluation of n-gram overlap between generated and reference texts. ROUGE-L specifically measures the longest common subsequence:

$$\text{ROUGE-L-P} = \frac{\text{LCS}(X, Y)}{|X|} \tag{2.4}$$

$$\text{ROUGE-L-R} = \frac{\text{LCS}(X, Y)}{|Y|} \tag{2.5}$$

$$\text{ROUGE-L-F} = \frac{(1 + \beta^2) \cdot \text{ROUGE-L-P} \cdot \text{ROUGE-L-R}}{\text{ROUGE-L-R} + \beta^2 \cdot \text{ROUGE-L-P}} \tag{2.6}$$

where LCS(X,Y) represents the length of the longest common subsequence between generated text X and reference text Y, and β controls the relative importance of precision versus recall.

MoverScore [32] extends beyond surface-level text comparison by measuring semantic similarity through word alignment in high-dimensional embedding space. This metric addresses limitations of n-gram based approaches by capturing semantic relationships between words, making it particularly valuable for evaluating generated text that expresses similar meaning through different vocabulary choices. The metric builds upon Word Mover's Distance (WMD), which treats text documents as weighted point clouds in embedding space. Each word becomes a point positioned according to its semantic embedding, while word frequencies determine the mass distribution across these points. The core insight lies in finding the minimum cost for transforming one text distribution into another through optimal word-level alignments.

The implementation begins by tokenizing both generated and reference

texts into cleaned word sets, removing stop words and punctuation to focus on semantically meaningful content. Each unique word receives embedding representation through pre-trained language models, creating dense vector representations that capture semantic relationships. Word frequencies are weighted using Inverse Document Frequency (IDF) scores to emphasize distinctive vocabulary while reducing the influence of common terms. The transportation cost between documents is formulated as an Earth Mover's Distance problem. Given two text documents with word embeddings $E_1 = \{e_1^{(1)}, e_2^{(1)}, \ldots, e_{n_1}^{(1)}\}$ and $E_2 = \{e_1^{(2)}, e_2^{(2)}, \ldots, e_{n_2}^{(2)}\}$, and their corresponding normalized frequency distributions w_1 and w_2, the distance calculation becomes:

$$d(D_1, D_2) = \min_{T \geq 0} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} T_{ij} \cdot c(e_i^{(1)}, e_j^{(2)}) \tag{2.7}$$

subject to the transportation constraints:

$$\sum_{j=1}^{n_2} T_{ij} = w_1^{(i)} \quad \forall i \in \{1, \ldots, n_1\} \tag{2.8}$$

$$\sum_{i=1}^{n_1} T_{ij} = w_2^{(j)} \quad \forall j \in \{1, \ldots, n_2\} \tag{2.9}$$

where T_{ij} represents the flow from word i in the first document to word j in the second document, and $c(e_i^{(1)}, e_j^{(2)})$ denotes the cosine distance between their embeddings.

MoverScore computes bidirectional distances to capture both precision and recall perspectives. The precision score measures how well the generated text aligns with the reference by calculating the minimum cost to transform the generated text distribution into the reference distribution. The recall score performs the inverse transformation, assessing how completely the reference content appears in the generated text. Both scores are converted from

distances to similarities:

$$\begin{aligned} \text{MoverScore-P} &= \max(0, 1 - d(D_{generated}, D_{reference})) \\ \text{MoverScore-R} &= \max(0, 1 - d(D_{reference}, D_{generated})) \end{aligned} \tag{2.10}$$

The final F1 score combines precision and recall through harmonic mean:

$$\text{MoverScore-F1} = \frac{2 \cdot \text{MoverScore-P} \cdot \text{MoverScore-R}}{\text{MoverScore-P} + \text{MoverScore-R}} \tag{2.11}$$

This approach proves effective for insurance domain evaluation, where answers containing equivalent factual information deserve similar scores despite lexical differences. MoverScore captures semantic equivalence that surface-level metrics might miss, providing more subtle assessment of content quality in specialized domains.

2.2.3 LLM-based Evaluation Metrics

The evaluation methodology through LLMs employed in this study draws inspiration from a recent work: *Do Large Language Models understand how to be judges?* [5]. This study analyzes whether large language models (LLMs) can serve as reliable judges for evaluating open-ended text generation (e.g., summarization) by applying a structured rubric covering five editorial criteria: coherence, consistency, fluency, relevance, and ordering. The results show that LLMs moderately align with human judgments (Spearman's $\rho \approx 0.6$–0.7 for some criteria) but exhibit systematic positive bias and that scaling improves absolute accuracy (MAE) but not necessarily ranking alignment, with smaller variants sometimes outperforming larger models in ranking consistency.

While the original paper proposed a multidimensional evaluation framework with five quality metrics, this research focuses specifically on three key dimensions that are more relevant for the use case: relevance, coherence and consistency. The decision to concentrate on these particular metrics stems

from their direct relevance to the specific challenges and objectives of the study, ensuring that the evaluation remains both comprehensive and practically applicable. The prompting strategies utilized in this evaluation are adapted from the approaches outlined in the original work, with modifications customized to the specific research context. The complete set of prompts employed in this study can be found in Appendix, where readers can examine the detailed instructions provided to the evaluating models.
Here is an explanation of the three aspects analyzed:

- Relevance assesses whether the generated response addresses the core elements of the query while avoiding tangential or irrelevant information. The evaluating model must discern between essential content that directly answers the question and supplementary material that, while potentially related, does not contribute meaningfully to resolving the user's information need.

- Coherence examines how well the response maintains logical flow and clear organization throughout. A coherent response presents information in a structured manner where ideas connect naturally and the overall narrative remains unified. This dimension requires understanding both local transitions between sentences and the global structure that ties the entire response together.

- Consistency measures comprehensive factual verification that extends well beyond basic hallucination detection. This metric ensures that every factual claim in the response can be traced back to and verified against the expected answer. The evaluating model must confirm that no contradictions exist between the response and desired output, and that no unsupported information has been introduced during the generation process.

The evaluation framework, differently from the cited paper, employs GPT-5 [16], OpenAI's latest large language model, as the evaluating agent for all

three assessment dimensions. GPT-5 represents a significant advancement over previous generations, featuring enhanced reasoning capabilities and substantially reduced hallucination rates that make it particularly suitable for rigorous evaluation tasks. The model demonstrates improved adherence to structured prompts and evaluation rubrics, ensuring more consistent and reliable assessment outcomes. This improved prompt-following capability is crucial for maintaining evaluation consistency across the three assessment dimensions, as the model must strictly adhere to the specific criteria and scoring guidelines provided for each metric.

Chapter 3

Data

The objective of the benchmarking effort is to establish a realistic and reusable evaluation set that reflects the information needs of practicing insurance brokers. Rather than relying on synthetic questions or generic QA datasets, the benchmark was constructed in collaboration with two broker experts who routinely consult policy documents. This choice ensures content relevance, preserves domain-specific terminology, and supports rigorous measurement of retrieval and generation performance under conditions that mirror actual usage.

The benchmark serves a dual purpose. First, it provides a controlled test to assess the chatbot's end-to-end behavior, from document retrieval to answer synthesis with source attribution. Second, it offers a practical starting point for future work in Italian insurance technology, where the availability of curated, domain-grounded evaluation data remains limited.

Delving into the first cause, this comprehensive evaluation allows stakeholders to understand precisely how the system performs across different types of insurance queries and document types, providing concrete evidence of the chatbot's capabilities and limitations. For the client who commissioned the project, this systematic assessment creates documentation that demonstrates the system's value to potential users and supports informed decision-making about deployment strategies. The benchmark results offer quantifiable metrics

that can be presented to insurance brokers, showcasing specific performance characteristics such as retrieval accuracy for different types of policy information, response quality for complex analytical queries, and consistency in source attribution across diverse document collections.

The evaluation framework captures multiple dimensions of system performance that directly relate to the practical needs of insurance professionals. By measuring retrieval precision and recall, the benchmark demonstrates how effectively the chatbot locates relevant policy clauses when brokers search for specific coverage terms or exclusions. Generation quality metrics reveal how well the system synthesizes complex insurance information into clear, actionable responses that brokers can confidently share with clients. Domain-specific metrics focusing on terminology accuracy and factual consistency provide evidence of the system's reliability when handling the specialized vocabulary and numerical precision required in insurance contexts. These metrics will be explained more deeply in the next section.

This systematic performance assessment enables the client to articulate specific use cases where the chatbot excels and identify scenarios that may require additional development or human oversight. For instance, the benchmark might reveal that the system performs exceptionally well on straightforward coverage inquiries but requires refinement for complex comparative analyses involving multiple policy types. Such insights allow for honest, data-driven presentations to potential users, building trust through transparency about both capabilities and current limitations.

3.1 Other Insurance Evaluation Frameworks

The construction of specialized evaluation frameworks for insurance applications has emerged as an active research domain, with initiatives targeting the distinct challenges of assessing AI system performance in this professional field.

The most relevant framework is InsQABench, which provides a notable example of systematic benchmark construction for insurance question answering, featuring 990 test question-answer pairs developed specifically for the Chinese insurance market [4]. The InsQABench methodology employs questions derived from real user interactions combined with expert-written answers as ground truth, establishing a framework for evaluating chatbot responses against professional standards. This approach demonstrates the value of expert involvement in benchmark construction, particularly for domains requiring specialized knowledge and precise factual accuracy.

Recent work has extended insurance-specific evaluation to North American contexts through the development of a bilingual Quebec automobile insurance dataset [2]. This dataset addresses the unique characteristics of Canadian insurance regulation while incorporating both English and French language requirements typical of Quebec's regulatory environment. The Quebec dataset encompasses questions spanning bodily injury coverage, property damage assessment, claims procedures, and legal responsibilities, drawing from official SAAQ documentation and private insurer materials. Expert validation ensures legal accuracy while covering both routine inquiries and complex edge cases that reflect real-world consultation scenarios. This multilingual approach highlights the importance of linguistic and regulatory specificity in insurance evaluation frameworks, particularly in jurisdictions where multiple languages and complex regulatory structures intersect.

However, existing benchmarks in the insurance domain present several limitations that motivated the development of a specialized evaluation framework. Most notably, available benchmarks focus primarily on non-European markets and lack the regulatory and linguistic specificity required for Italian insurance applications. Additionally, current benchmarks typically provide answers without explicit source attribution, limiting their utility for evaluating retrieval-augmented systems where citation accuracy represents a critical performance dimension. The absence of multi-document scenarios in existing

benchmarks further restricts their applicability to real-world insurance workflows, where professionals frequently need to compare policies across multiple providers or analyze coverage relationships spanning different document types.

Our benchmark addresses these gaps through several methodological innovations that reflect the specific requirements of the Italian insurance market and RAG-based systems. First, the complete construction process involves insurance experts from conception through final validation, ensuring that both questions and answers reflect authentic professional knowledge and terminology usage. Second, every answer includes precise document-page citations that enable evaluation of source attribution accuracy, a capability essential for systems where verifiability directly impacts professional credibility. Third, the benchmark incorporates questions requiring synthesis across multiple documents, reflecting the comparative analysis tasks that characterize much of the analytical work performed by insurance brokers in their daily practice.

3.2 Design Principles

The question development process began with comprehensive briefings to the two insurance experts, where the technical problem and system objectives were explained. The experts were familiarized with the challenges of document retrieval in insurance environments and the potential benefits of automated question-answering systems for their daily workflows. This initial phase established a shared understanding of the project goals and ensured that subsequent question development would align with both technical requirements and practical applications. The experts were instructed to generate questions that reflect their daily experiences as insurance professionals. They were asked to consider two primary categories of inquiries: first, questions that arise frequently in their routine interactions with clients and policy documents, representing the most common information needs encountered in their

practice; and second, questions that traditionally require substantial time investment or extensive manual search through multiple documents to answer accurately. This dual focus ensured that the benchmark would capture both routine efficiency gains and significant productivity improvements achievable through automated assistance.

The experts were further guided to develop questions spanning diverse functional areas and request types to ensure comprehensive coverage of insurance domain tasks. This directive encouraged them to think beyond simple fact retrieval and consider complex analytical scenarios including policy comparisons across multiple providers, coverage gap analysis, claims procedure clarification, eligibility condition verification, and calculation-based inquiries involving premiums, deductibles, and coverage limits. The goal was to create a representative sample of the analytical complexity that characterizes professional insurance consultation work.

So, questions were authored to capture the kinds of tasks brokers perform daily, including locating coverage terms, clarifying exclusions, verifying limits and deductibles, checking eligibility conditions, and interpreting claims procedures. Each question targets content that can be located in one or more policy documents and answered without access to external knowledge beyond the curated corpus. The two broker experts selected a corpus of policy documents representative of their day-to-day activities. The selection process aimed to cover multiple product lines, providers and sectors while avoiding redundancy caused by near-duplicate editions. The variety of sectors covered by the different documents, allow to reduce the bias of the benchmark towards a certain field. The documents were uploaded in PDF format and in italian version.

The experts independently drafted questions they consider operationally meaningful. For each question, they wrote a concise expected answer and annotated the specific document-page pairs that substantiate the response. Answers were phrased to privilege factual precision over stylistic variation and

to preserve critical numerical values, defined thresholds, and legal terms that might affect coverage interpretation. When relevant information spanned multiple pages, the reference list included all necessary locations to recover the complete rationale. Said so, other parts of the document can still be relevant for the answer, being that insurance documents often contain multiple paragraphs regarding the same topic or declaration.

In the benchmark construction process, first I chose the document corpus based on the experts' input and the identified use cases. This ensured that the selected materials were not only relevant but also representative of the actual challenges faced by brokers. Then, each expert produced questions and corresponding answers with citations on the document pages. The independent work of the experts was coordinated to ensure coverage of diverse information needs and to avoid redundancy in the questions. Another key point was the creation of questions with different difficulty levels, spanning from straightforward fact retrieval to more complex inferential reasoning and calculations.

3.2.1 Challenges and Expert Requirements

The benchmark construction process required extensive collaboration with highly qualified insurance professionals whose expertise forms the foundation of the evaluation framework. The two participating experts possess over twenty years of combined experience in the Italian insurance market, working as licensed brokers with comprehensive knowledge spanning multiple insurance sectors including automotive, travel, health, property, and commercial coverage. Their professional credentials include regulatory certifications and ongoing training requirements that ensure current knowledge of market developments, legal frameworks, and industry best practices. The specialized competencies required for benchmark creation extend beyond general insurance knowledge to encompass technical document analysis, legal interpretation,

and systematic quality assessment. Expert participants must demonstrate proficiency in dissecting complex policy language, understanding intricate coverage relationships, and identifying subtle but critical distinctions between similar insurance products. This expertise proves essential when crafting questions that reflect authentic professional challenges and when providing answers that meet the accuracy standards expected in commercial insurance environments.

Once experts agreed to participate, coordinating their availableness for benchmark development proved complex due to the unpredictable nature of insurance brokerage work. Client emergencies, regulatory deadlines, and seasonal variations in workload created scheduling conflicts that extended the benchmark creation timeline beyond initial projections. The iterative nature of question development, where experts refined their contributions based on technical feedback and evaluation requirements, further complicated coordination efforts. The computational cost of expert involvement represents a significant economic factor in benchmark expansion. Each question-answer pair with precise citations requires approximately two to three hours of focused expert time, encompassing document review, question formulation, answer composition, and citation verification. Complex questions requiring synthesis across multiple policy sections can demand up to five hours of expert analysis. When multiplied across the complete benchmark set, these time requirements translate to substantial professional consultation costs that limit the feasible scope of evaluation datasets.

The expertise threshold for meaningful benchmark contribution cannot be easily reduced through automation or simplified procedures. Insurance policy interpretation requires a deep understanding of legal terminology, regulatory contexts, and industry practices that develop only through years of professional experience. Attempts to supplement expert contributions with automated question generation or non-expert annotations would fundamentally compromise the evaluation framework's validity and reliability for assessing professional-grade insurance applications.

3.3 Composition and Usage

The benchmark is composed of 5 columns:

- **ID**: A unique identifier for each benchmark instance.
- **Question**: A natural language question reflecting a specific information need.
- **Golden Answer**: The correct answer to the question provided by the insurance expert, derived from the document corpus.
- **Golden Reference**: Citations identifying the specific document-page pairs that support the answer, formatted as ”document_filename.pdf, pagine: page1, page2, page3, ...” where multiple page numbers are listed for comprehensive coverage of the supporting evidence.

Through the collaborative effort of the two insurance experts, a total of 30 question-answer-reference sets were produced. These benchmark instances draw from a curated corpus of 7 different insurance policy documents, each provided in PDF format. The document collection considers multiple insurance sectors, ensuring that the benchmark captures the diversity of professional consultation scenarios. The distribution of questions across documents reflects the relative complexity and comprehensiveness of different policy types, with more detailed documents naturally supporting a greater number of meaningful queries.

Table 3.1 provides an overview of the document corpus utilized in the benchmark construction, detailing the characteristics of each policy document and its contribution to the evaluation framework. The number of questions associated with each document reflects the deep knowledge of the insurance experts regarding the content and complexity of the policies.

The benchmark is integrated into the evaluation pipeline as a set of independent test instances. For each instance, the system receives the natural language question, executes retrieval over the curated corpus (and conversation

Table 3.1: Document corpus composition and benchmark distribution

Document	Insurance Domain	Pages	Questions
Document1	Household	120	4
Document2	Condo	112	7
Document3	Commercial Property	152	2
Document4	Marine	72	5
Document5	Personal Accident	96	5
Document6	Health	85	6
Document7	Legal Expenses	60	1
Total	**7**	**697**	**30**

attachments when applicable), and generates an answer with citations. Evaluation then proceeds along two axes. On the retrieval side, the cited document–page pairs attached to each gold answer define the Relevant set. The system's returned citations are compared against this set to compute precision, recall, and F1 as described in the next section. This mapping supports questions whose justification spans multiple pages or documents: partial credit arises from the intersection between retrieved and gold references. To ensure consistency across formatting variations, references are canonicalized before scoring through lightweight normalization of document identifiers, page numbering conventions, and minor typographic differences that do not alter content.

3.3.1 Legal and Practical Considerations

The corpus and annotations were assembled under authorization from collaborating brokers and limited to documents appropriate for professional consultation. These documents used in the benchmark are of public usage, but, for completeness, no personally identifiable information was collected or retained. Page-level citation ensures verifiability without reproducing large text spans from proprietary materials. Dataset storage follows the same security posture outlined in the system architecture, relying on access-controlled, encrypted storage with audit logging to preserve confidentiality.

Dissemination prioritizes reproducibility while respecting licensing constraints. When full documents cannot be redistributed, the release format consists of question–answer pairs with redacted or hashed document identifiers, together with clear instructions that allow authorized users to reconstruct page references from their local copies. All evaluation was conducted on Italian source texts to avoid translation artifacts in both retrieval and metric computation; where normalization is applied (for example, on numerical formats), it is explicitly documented in the evaluation setup. Finally, the benchmark is versioned to reflect document updates and annotation refinements, enabling consistent longitudinal comparison across system iterations.

3.3.2 Limitations and Future Extensions

The benchmark reflects the sectors, providers, and drafting styles most familiar to the participating experts and therefore cannot claim exhaustive market coverage. Legal language occasionally admits multiple defensible interpretations, and some questions permit more than one acceptable formulation of the answer. Although adjudication enforced a single canonical reference per item, future versions will incorporate multiple gold paraphrases to reduce penalization of stylistic variance that does not affect factual content.

Also, the benchmark contains a relatively small number of questions and consultable documents, which may limit its ability to comprehensively evaluate system performance across diverse scenarios. Expanding the question set to include a wider range of topics, complexities, and question types will be a priority for future iterations although the high cost in computation of these data remains a challenge.

3.4 Metric Execution Framework

The evaluation of the benchmark on the previously proposed metrics is implemented through a Python script. The framework design enables comprehensive performance analysis while maintaining the flexibility needed to incorporate future evaluation methodologies as they emerge in the RAG evaluation landscape.

Retrieval and generation metrics are computed through a local script that utilizes established Python libraries for reliable and reproducible assessment. The implementation leverages NumPy for numerical computations and array operations, sentence-transformers for semantic similarity calculations underlying MoverScore computation, rouge-score for lexical overlap assessment, and SciPy for statistical analysis and correlation calculations.

The retrieval evaluation framework computes precision, recall, and F1 scores by comparing system-retrieved document-page citations against expert-annotated gold standard references. Citation matching employs normalization procedures that handle formatting variations while preserving semantic accuracy. The generation metrics utilize a from-scratch implementation of MoverScore-F1 for semantic similarity assessment and ROUGE-L for lexical overlap measurement, providing complementary perspectives on answer quality that capture both semantic equivalence and surface-level correspondence.

The LLM-as-judge evaluation employs OpenAI's GPT-5 model configured with medium reasoning effort to provide assessment across three quality dimensions: relevance, coherence, and consistency. This configuration enables more sophisticated reasoning about answer quality while maintaining computational feasibility for systematic evaluation across the complete benchmark. The evaluation protocol utilizes OpenAI's Responses API through local script execution, ensuring controlled evaluation conditions and reproducible

results. The assessment prompts are carefully designed following the previously cited paper guidelines [5], with little modifications to adapt to the specificities of the use case. The complete prompt specifications for each evaluation dimension are documented in Appendix B, providing transparency and enabling replication of the evaluation methodology.

3.4.1 Expert Evaluation Metric

Automated metrics provide valuable insights into system performance, human evaluation, instead, remains essential for assessing the practical utility of generated responses in professional contexts. Expert evaluation captures detailed aspects of response quality that automated metrics cannot adequately measure, including contextual appropriateness, professional terminology usage, and the practical relevance of information for an everyday usage.

The expert evaluation metric employs a structured five-point scoring system designed to assess response quality from the perspective of insurance professionals. This framework was developed in collaboration with the same domain experts who constructed the benchmark, ensuring that evaluation criteria reflect the practical requirements and quality standards of the broker domain. The process requires domain specialists to examine each system response against the corresponding benchmark question and reference answer.

In the five-point scoring scale, each score level represents distinct quality thresholds that correspond to different levels of professional utility:

- Score 1 (Completely Incorrect Response): the system output is entirely wrong and fails to address the user's question. Such responses provide information that is irrelevant, misleading, or factually incorrect to the extent that they could cause consequences if relied upon. This category includes responses that misinterpret fundamental insurance concepts, provide coverage information that contradicts policy terms, or generate completely unrelated content that demonstrates failure to understand the

query context.

- Score 2 (Poor Response): the output contains significant inaccuracies or omissions that severely limit its practical utility. These responses may address aspects of the user's question but provide minimal relevant information while lacking clarity and failing to satisfy professional informational needs. Poor responses often demonstrate partial understanding of insurance concepts but contain substantial errors in coverage details, policy interpretation, or regulatory compliance information that would require significant correction before professional use.

- Score 3 (Moderate Response): the output is partially correct and answers the question to some extent, but important details are missing or insufficiently explained. These responses may contain minor inaccuracies or lack the completeness required by professionals. Moderate responses typically demonstrate adequate understanding of basic insurance concepts but fail to provide the comprehensive analysis or detailed coverage information that insurance professionals require for client consultation or comparative analysis.

- Score 4 (Good Response): the output is mostly accurate and relevant, adequately answering the question with information that meets professional standards. While such responses may not include every possible detail, the provided information is clear, coherent, and useful for insurance professionals.

- Score 5 (Outstanding Response): the system output fully addresses the question with accurate, comprehensive information that exceeds typical professional requirements. These responses demonstrate exceptional clarity, precision, and completeness while using appropriate professional language and style. Outstanding responses provide the level of detail needed by professionals for this type of tool.

Using a 1-to-5 scale, the insurance experts expressed confidence in providing the scores, as the evaluation criteria were collaboratively developed with them, ensuring a shared understanding of quality standards.

To facilitate systematic and focused evaluation, the experts received a structured evaluation form that organized the assessment process to guarantee consistent attention to each benchmark instance. The form presented the five-point scoring criteria at the beginning, serving as a constant reference throughout the evaluation process. Each benchmark record was displayed on a separate page within the form, allowing evaluators to concentrate fully on individual assessments without distraction from other items. For each evaluation instance, the experts were presented with four key components: the original question posed to the system, the expert-authored reference answer from the benchmark, the system-generated response requiring evaluation, and access to the relevant source documents for verification of any unclear or disputed information. This structured presentation enabled evaluators to systematically compare system outputs against established quality standards while maintaining access to original source materials for fact-checking and contextual verification when needed.

Chapter 4

RAG Architecture

This chapter presents the technical architecture and implementation details of the RAG-based insurance document chatbot system. The system is built around two fundamental components that work in coordination to deliver comprehensive document consultation capabilities for insurance professionals: a document processing pipeline that automatically transforms uploaded insurance policies into searchable knowledge representations, and a retrieval augmented generation module that processes natural language queries and generates accurate responses with precise source attribution.

The architecture leverages OpenAI's APIs and infrastructure to provide a robust and scalable solution. The design prioritizes the critical requirements identified through collaboration with insurance experts: factual accuracy, comprehensive source citation, real-time response generation, and intelligent document filtering based on insurance sectors and companies. These capabilities enable insurance brokers to efficiently consult policy documents, perform comparative analyses, and obtain reliable information for client interactions.

The document processing pipeline operates automatically through post-hook mechanisms, ensuring that uploaded documents become immediately available for querying without manual intervention. This component handles multiple file formats, performs intelligent text extraction at the page level, and

integrates documents into OpenAI's vector store infrastructure with appropriate metadata for precise retrieval targeting. The RAG module builds upon this foundation to provide an intelligent query interface that combines automatic sector and company detection, metadata-based document filtering, streaming response generation, and comprehensive citation management.

The following sections detail the implementation of each component, examining the technical decisions, integration patterns, and operational characteristics that enable the system to meet the demanding requirements of professional insurance consultation workflows.

4.1 Document Processing Pipeline

The document processing pipeline represents a critical component that transforms uploaded insurance documents into searchable knowledge representations. This pipeline operates automatically through the post-hook system, executing immediately after successful file uploads to ensure that documents become available for RAG-based querying without manual intervention. The design prioritizes reliability and error handling, implementing status tracking and rollback mechanisms.

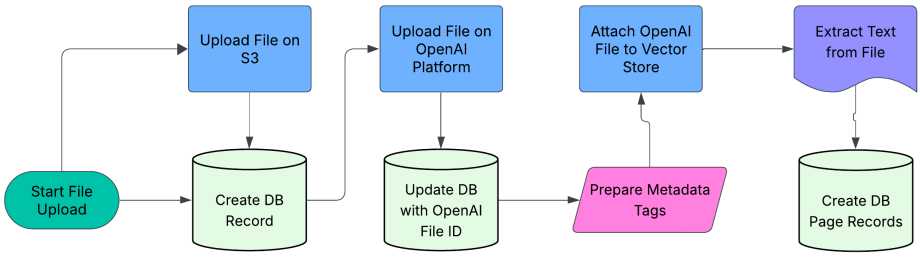

Figure 4.1: Document Processing Pipeline Overview

4.1.1 OpenAI Integration

The processing pipeline initiates when the post-hook system triggers the main processing function following successful file upload to the application's storage infrastructure. The system first retrieves the uploaded file record from the database, extracting information including sector classification and company information that was specified during the upload process. This metadata is used for subsequent filtering and organization operations within the vector store.

The documents are uploaded to OpenAI's platform through the file management API, creating a remote file reference that enables subsequent processing. The system handles authentication and API communication through a custom OpenAI provider that manages connection pooling, rate limiting, and error recovery procedures. Once uploaded successfully, OpenAI returns a unique file identifier that the system persists to the local database, creating a permanent link between local file records and remote OpenAI resources.

For documents that will participate in the global knowledge base, the system retrieves the global vector store identifier and prepares tags based on the sector and company information extracted from the file record. The document is then attached to the relative OpenAI vectorstore with the appropriate metadata dictionary.

4.1.2 Document Storage and Extraction

The system accepts PDF, DOCX, and PPTX files, but, in the upload phase, all documents undergo conversion to PDF format to ensure consistent text extraction and processing workflows. For documents already in PDF format, the system downloads the file from cloud storage using presigned URLs and saves it to a temporary processing directory. The temporary file handling includes automatic cleanup procedures that remove processed files after successful completion, preventing storage accumulation.

Documents in DOCX or PPTX formats require active conversion to PDF through an external service. The conversion process utilizes a specialized PDF conversion provider that accepts office document formats and returns standardized PDF output. The conversion service operates through RESTful API integration, where the system uploads the source document and receives a download URL for the converted PDF file. The converted PDF is then downloaded and saved to the temporary processing directory for subsequent text extraction operations. Error handling during format conversion includes retry mechanisms for temporary service unavailability and fallback procedures for unsupported document characteristics. When conversion failures occur, the system updates the document status to indicate processing errors and prevents the document from appearing in query results, ensuring that only successfully processed documents participate in the knowledge base.

The extraction process operates at the page level: it creates individual text records for each page within the document through PyMuPDF (fitz) library. This granular approach allows precise source attribution during response generation, allowing the system to reference specific page numbers when citing information from insurance policies. Before beginning text extraction, the system removes any existing page records associated with the document to prevent data duplication. The extraction loop processes each page sequentially, extracting plain text content and creating database records that link the extracted text to specific page numbers within the source document. The process includes error handling for corrupted PDFs, password-protected documents, and other processing exceptions that may occur. In these cases, the system logs detailed error information and updates the document status appropriately, preventing partially processed documents from affecting system operation.

4.2 Retrieval Augmented Generation Module

The RAG-based chatbot module implements a question answering system built around OpenAI's Responses API, featuring automatic sector and company detection, metadata-based document filtering, streaming response generation, and source management. The implementation coordinates multiple components including a streaming query endpoint that processes natural language questions, an intelligent tagging system that identifies relevant insurance domains, vector store integration with filtered retrieval, and citation management that links generated content to specific document pages.

4.2.1 Streaming Query Interface

The primary interface for user interactions is implemented through a streaming POST endpoint that accepts user questions and returns real-time responses. The endpoint manages thread-based conversations, where each insurance professional can maintain multiple concurrent discussion threads about different topics or document sets. When processing a query, the system first determines whether this represents the initial message in a conversation thread. For new conversations, the system automatically generates descriptive titles by summarizing the user's question, creating intuitive thread names.

The endpoint supports both permanent document collections and temporary file attachments. Users can attach specific documents directly to conversations for immediate analysis, enabling ad-hoc consultation of client-specific policies or temporary comparative materials. These attachments remain available only within the specific conversation thread, providing flexible document analysis without affecting the permanent knowledge base.

4.2.2 Tag Detection System

A tag detection mechanism automatically identifies relevant insurance sectors and companies mentioned in user queries, enabling the system to focus

retrieval operations on the most relevant document subsets. This intelligent filtering improves response accuracy by ensuring that queries about specific insurance types or providers access only the appropriate policy documents.

The tag detection process operates through a dedicated language model interaction that analyzes user questions against predefined vocabularies of insurance sectors and company names. The system maintains comprehensive mappings of available sectors along with complete rosters of insurance companies represented in the document collection. The algorithm employs natural language understanding to identify both explicit mentions of sectors and companies as well as contextual references that imply specific insurance domains. For example, queries about ”collision coverage” would automatically trigger vehicle insurance sector tagging, while mentions of ”trip cancellation” would activate travel insurance classification. Tag information persists across conversation threads, enabling the system to maintain sector and company focus throughout extended analytical sessions. Users can implicitly modify these tags by mentioning different sectors or companies in subsequent questions, allowing natural transitions between different insurance domains within the same conversation.

There is also a validation mechanisms that ensure detected tags correspond to actual document availability: before applying filters, the system verifies that the identified sector and company combinations actually exist in the document collection, preventing empty result sets and ensuring productive query processing.

Vector Store Integration and Filtering

The document retrieval mechanism operates through OpenAI’s managed vector store infrastructure, utilizing a global document repository improved with metadata-based filtering capabilities. All processed insurance documents are indexed within a shared vector store, with each document chunk tagged with sector and company metadata that allows precise retrieval targeting.

When sector and company tags are identified, the system constructs filter expressions that limit retrieval to documents matching the specified criteria. For conversations involving temporary file attachments, the system creates a supplementary vector store that contain only the attached documents. The retrieval process then searches across both the global document collection and the conversation-specific attachments, enabling comprehensive analysis that combines permanent knowledge base content with temporary consultation materials.

The filtering mechanism supports complex logical combinations, allowing queries that span multiple insurance companies within a specific sector. Appropriate filter expressions are constructed based on the detected tags, translating natural language queries into precise retrieval constraints. The system employs a threshold of 0.5 on similarity scores, filtering out marginally relevant chunks of content that might confuse the generation process or introduce inaccurate information. The insurance documents are full of information and frequently contain similar sentences, so a higher threshold would have filtered out too many relevant chunks, while a lower threshold would have introduced too much noise in the retrieved context. These considerations are the results of empirical testing conducted during the development of the application, with the feedback of the client who commissioned the project, an insurance expert.

4.2.3 Response Generation and Configuration

The response generation process represents the apex of the RAG pipeline, transforming retrieved document content into coherent responses through an implementation that prioritizes both real-time delivery and domain-specific accuracy. The system employs GPT-5 as the underlying language model, configured with low reasoning effort to balance response quality with computational efficiency. This configuration emerged from preliminary testing that demonstrated optimal performance for insurance domain applications, where

factual accuracy and consistency take precedence over creative interpretation.

The streaming implementation utilizes OpenAI's streaming API to provide real-time interaction experiences, delivering partial responses as they develop through sophisticated event handling that processes different types of streaming events including text deltas, annotations, and completion signals. Text content streams directly to users as it generates, while citation information accumulates for final processing. This approach ensures immediate feedback to users while maintaining comprehensive source attribution throughout the response generation process. Other reasoning configurations were tested, but the increasing of reasoning tokens did not lead to significant improvements in the quality of the answers, while it increased the latency of the responses and the cost of the API calls. Also, during the development of the application, other GPT models were tested, such as GPT-4.1 and GPT-5-mini, but the results were not as satisfactory as those obtained with GPT-5. GPT-5 demonstrated better instruction following capabilities over GPT-4.1, and better tool usage over GPT-5-mini.

The prompt engineering follows established OpenAI guidelines for GPT-5 models, incorporating domain-specific instructions that ensure appropriate behavior for insurance applications. The refinement of the instructions happened thanks to the client's feedback and the OpenAI platform tool that facilitates prompt customization: it is a fine-tuned LLM that is specialized in prompt optimization for GPT-5 models. The prompt design emphasizes several critical requirements: exclusive reliance on provided knowledge sources without external information injection, adherence to professional insurance communication standards, and maintenance of appropriate factual grounding throughout response generation. The prompt architecture includes specific instructions for handling uncertainty, source attribution requirements, and response formatting that aligns with professional insurance documentation standards, ensuring that generated answers meet the reliability and verification requirements expected in commercial insurance environments. The

complete prompt specifications are provided in Appendix A for reproducibility and transparency.

The streaming implementation features robust error handling mechanisms that gracefully manage network connectivity issues and API timeouts. Connection disruptions are transparently addressed through suitable fallback procedures that ensure users receive complete responses even in the presence of temporary connectivity interruptions. This reliability framework proves essential for professional environments where incomplete or interrupted responses could impact critical business decisions.

4.2.4 Source Management

Precise source attribution represents a critical capability of the application, as requested by the client, professionals must verify information and maintain clear documentation of their analytical sources. The system implements comprehensive citation tracking that links every piece of generated content to specific document pages.

The system implements intelligent citation matching that correlates inline annotations with retrieved document sections. When the language model references specific information during generation, the citation system identifies the corresponding source material and prepares appropriate reference markers. This matching process ensures that users can trace every factual claim back to its documentary source through an inline citation placeholder. The algorithm searches document text to identify exact page locations for cited content, enabling users to quickly navigate to relevant sections for verification or additional context.

Duplicate citation filtering ensures that repeated references to the same document sections are consolidated appropriately, preventing citation redundancy while maintaining complete source coverage. The system prioritizes the most relevant citations when multiple references point to similar content,

ensuring efficient reference presentation.

Chapter 5

Experimental Results

This chapter presents the evaluation of the RAG-based insurance chatbot system through the benchmark framework established in Chapter 3. The experimental assessment examines system performance across multiple evaluation dimensions, providing insights into retrieval effectiveness, generation quality, and overall professional utility for insurance domain applications. The chapter concludes with qualitative case studies that reveal specific system behaviors and provide concrete examples of how different evaluation metrics capture distinct aspects of response quality in real-world scenarios.

5.1 Results

The evaluation of the insurance document chatbot system reveals distinct performance patterns across retrieval and generation components, providing comprehensive insights into the system's capabilities and limitations. The results demonstrate strong overall performance with notable variations across different evaluation dimensions, confirming the system's practical viability for professional insurance applications.

The benchmark evaluation treats all the 30 records with the same methods. Performance assessment addresses both technical accuracy through automated

metrics and practical utility through expert evaluation, ensuring comprehensive coverage of the quality dimensions most relevant to insurance professionals. The evaluation framework successfully captures the multifaceted nature of RAG system performance, revealing how retrieval effectiveness, generation quality, and domain expertise requirements interact in real-world applications.

Table 5.1 presents the complete statistical summary of all evaluation metrics, providing a comprehensive view of system performance across retrieval accuracy, generation quality, expert assessment, and automated judgment dimensions. These aggregate statistics establish the foundation for detailed analysis of performance patterns, correlation relationships, and specific system behaviors that characterize the chatbot's operational capabilities.

Table 5.1: Comprehensive statistics for all evaluation metrics

Metric	Mean	Std	Median	Min	Max
Retrieval Recall	0.950	0.118	1.000	0.500	1.000
Retrieval Precision	0.698	0.164	0.714	0.333	1.000
Retrieval F1	0.796	0.120	0.833	0.500	1.000
MoverScore-F1	0.451	0.087	0.444	0.298	0.636
ROUGE-L	0.244	0.094	0.225	0.050	0.459
Expert Score	4.467	0.730	5.000	3.000	5.000
LLM Relevance	3.900	1.062	4.000	2.000	5.000
LLM Coherence	4.500	0.682	5.000	3.000	5.000
LLM Consistency	3.833	1.020	4.000	2.000	5.000

The results analysis proceeds through systematic examination of aggregate performance trends, distribution characteristics, and correlation patterns that illuminate the relationships between different evaluation approaches. Particular attention is given to the alignment between automated metrics and human expert assessments, as this relationship determines the reliability of evaluation frameworks for ongoing system development and quality assurance in production environments.

5.1.1 Retrieval Performance

The evaluation of retrieval performance is conducted at a detailed level, measuring precision, recall, and F1-score on specific document-page pairs rather than considering entire documents as single entities. According to the adopted evaluation methodology, this approach acknowledges that relevant information is often located within particular sections or pages of insurance documents. This allows for a more accurate assessment of the system's capability to retrieve the exact content required to answer user queries. Each retrieved item is compared against expert-annotated gold standard references, which indicate the precise document-page combinations containing the necessary information for complete and accurate responses.

The retrieval component demonstrates exceptional performance across all evaluated metrics, establishing a strong foundation for downstream generation quality. Figure 5.1 illustrates the distribution and relationships between precision, recall, and F1 scores across the 30 benchmark questions, revealing consistent high-quality retrieval patterns with notable performance characteristics.

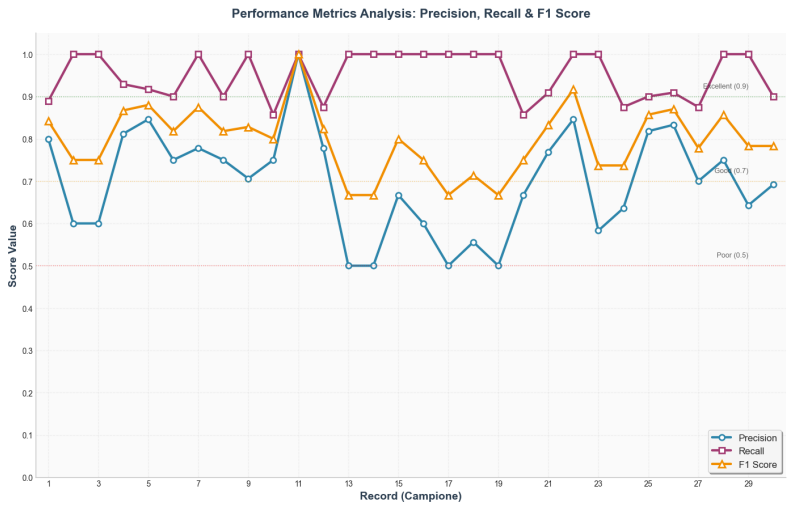

Figure 5.1: Distribution of retrieval performance metrics.

Recall performance achieves outstanding results with a mean of 0.950 and median of 1.000, indicating that the system consistently identifies and

retrieves nearly all relevant document passages for user queries. This exceptional recall demonstrates the effectiveness of the semantic similarity search approach in capturing comprehensive coverage of pertinent information across diverse insurance domains. The high recall performance ensures that users rarely encounter scenarios where critical policy information remains inaccessible due to retrieval failures, addressing the fundamental requirement for comprehensive document coverage in professional insurance applications.

The recall distribution shows remarkable consistency, with 20 out of 30 questions achieving perfect recall scores of 1.000. The minimum recall of 0.857 indicates that even in the most challenging cases, the system successfully retrieves the vast majority of relevant content. This performance pattern reflects the robustness of the embedding-based retrieval architecture in handling diverse query types and document structures across different insurance products and coverage domains.

Precision performance presents a more moderate profile with a mean of 0.698 and median of 0.703, suggesting that while the system effectively captures relevant content, it also includes some irrelevant passages in the retrieved set. The precision scores range from 0.500 to 1.000, with a standard deviation of 0.122, indicating reasonable consistency in relevance filtering across different query complexities. This precision level represents a practical balance between comprehensive coverage and focused retrieval, ensuring that the generation component receives sufficient relevant context while managing the computational and quality implications of including extraneous material.

F1 scores synthesize these retrieval characteristics with a mean of 0.796 and median of 0.800, reflecting a solid balance between comprehensive coverage and relevance filtering. The F1 distribution demonstrates consistent performance across the benchmark, with most scores falling between 0.750 and 0.900. This performance range indicates reliable retrieval quality that meets the requirements of the application.

The high recall performance represents a particularly valuable characteristic for the analysis, as it ensures that nearly all relevant information is successfully retrieved and made available as context for the chatbot. While the moderate precision scores might initially suggest room for improvement, this performance pattern is actually well-suited to the characteristics of insurance documentation. The additional pages retrieved often contain repetitive information that is typical of insurance policies, where important clauses and coverage details are frequently restated across multiple sections for legal completeness. Although some retrieved content may constitute noise, this is not critical, being that the precision score remains high.

5.1.2 Generation Performance

The evaluation of generation quality through automated metrics reveals contrasting patterns between semantic and lexical similarity measures, highlighting the complex relationship between surface-level correspondence and semantic equivalence in insurance domain responses. Figure 5.2 illustrates the distribution and correlation patterns between MoverScore-F1 and ROUGE-L metrics across the benchmark questions, providing insights into the nature of answer quality variation.

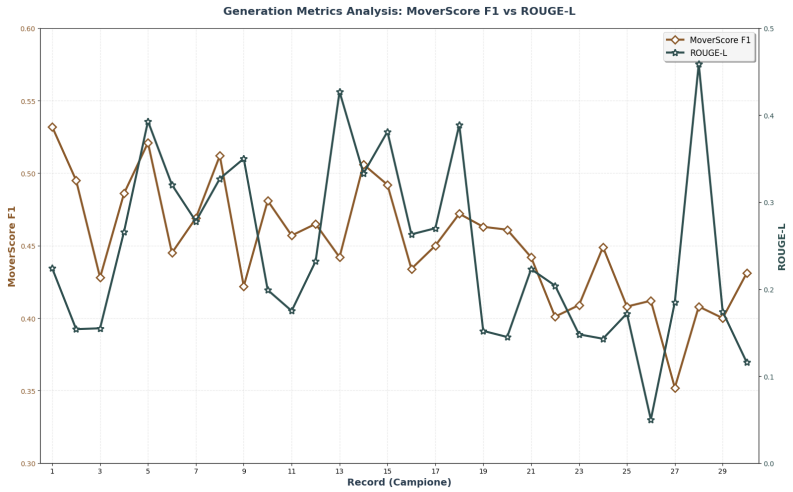

Figure 5.2: Distribution of generation performance metrics.

MoverScore-F1 demonstrates relatively stable performance with a mean of 0.451 and standard deviation of 0.040, indicating consistent semantic similarity between generated and reference answers across diverse question types. The metric ranges from 0.352 to 0.532, representing a moderate spread that suggests the system maintains reasonable semantic alignment with expert-provided answers while accommodating natural variation in expression. This stability in semantic similarity reflects the effectiveness of the underlying language model in capturing and conveying the essential meaning of insurance information, even when employing different linguistic formulations than the reference answers.

ROUGE-L scores exhibit significantly greater variability, with a mean of 0.244 and standard deviation of 0.101, spanning a wide range from 0.050 to 0.459. This substantial variation in lexical overlap indicates that generated responses achieve varying degrees of surface-level similarity with reference answers, reflecting the inherent challenges of lexical matching in natural language generation.

The lower mean ROUGE-L score compared to MoverScore-F1 suggests that while the system maintains semantic fidelity, it frequently employs alternative linguistic expressions rather than closely mimicking the exact phrasing of reference answers. The distribution pattern confirms that lexical metrics provide inconsistent quality indicators in domains where paraphrasing is common and acceptable.

5.1.3 LLM as judges results

The LLM-based evaluation through GPT-5 provides comprehensive assessment across the three quality dimensions identified as most relevant for insurance domain applications: relevance, coherence, and consistency. This automated evaluation approach offers valuable insights into response quality

that complement expert human assessment while maintaining consistent evaluation standards across all benchmark questions.

The aggregate performance across the three LLM evaluation dimensions reveals distinct quality patterns that reflect the challenges of the insurance chatbot system. Table 5.1 shows that coherence achieves the highest scores with a mean of 4.500 and median of 5.000, indicating that the system consistently maintains logical flow and clear organization throughout generated responses. This strong coherence performance reflects the effectiveness of the underlying language model in structuring information coherently, even when synthesizing complex insurance concepts from multiple sources.

Relevance evaluation yields moderate performance with a mean of 3.900 and median of 4.000, suggesting that while responses generally address core query elements, some instances include tangential information or fail to focus entirely on the most essential aspects of user questions. The standard deviation of 0.923 indicates considerable variation in relevance assessment, ranging from 2.000 to 5.000, which reflects the varying complexity of insurance queries and the challenge of maintaining strict focus across diverse question types.

Consistency demonstrates the most variable performance among the three dimensions, achieving a mean of 3.833 with a standard deviation of 0.986 and scores spanning the complete range from 2.000 to 5.000. This variability in consistency scores reflects the comprehensive factual verification requirements that extend beyond basic hallucination detection. The evaluation requires confirmation that every factual claim can be traced back to and verified against the expected answer, ensuring no contradictions exist between responses and gold standard outputs. The broader score distribution suggests that while many responses achieve high factual accuracy, some instances introduce unsupported information or contain subtle inconsistencies that impact verification against reference answers.

Figure 5.3 illustrates the distribution of scores across the three evaluation

dimensions, revealing the frequency patterns for each quality metric and highlighting the areas where the system demonstrates consistent strength versus those requiring continued attention.

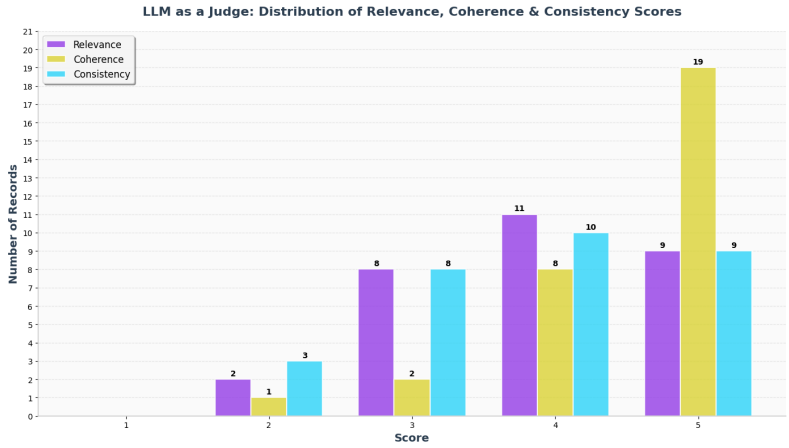

Figure 5.3: Distribution of LLM-based evaluation scores.

The comparative analysis across dimensions reveals important insights about system behavior in insurance applications. The superior coherence performance indicates that the system effectively organizes insurance information into logical, well-structured responses that professionals can readily understand and utilize. The moderate relevance scores suggest opportunities for improvement in query focus and information prioritization. While responses generally address user questions appropriately, the evaluation indicates that refinements in prompt engineering or retrieval filtering could enhance the system's ability to distinguish between essential and supplementary information. This finding aligns with the moderate precision scores observed in retrieval evaluation, suggesting that improvements in source selection could positively impact response relevance.

5.1.4 Expert Evaluation

The expert evaluation results demonstrate strong performance of the insurance chatbot system, with domain specialists consistently rating generated responses highly across the benchmark questions. The evaluation employed the

structured five-point scoring system described in Section 3.4.1, where insurance professionals assessed response quality against established criteria for professional utility and accuracy.

The aggregate statistics reveal consistently high performance levels that exceed the expectations for professional deployment. The expert evaluation achieved a mean score of 4.467 with a standard deviation of 0.629, indicating both strong central performance and relatively consistent quality across diverse question types. The median score of 5.000 demonstrates that more than half of the generated responses met the highest quality standards, representing outstanding performance that fully addresses professional requirements with comprehensive and accurate information. Figure 5.4 illustrates the distribution pattern across the five-point scale, revealing strong clustering in the upper performance ranges that characterizes professional-grade system behavior.

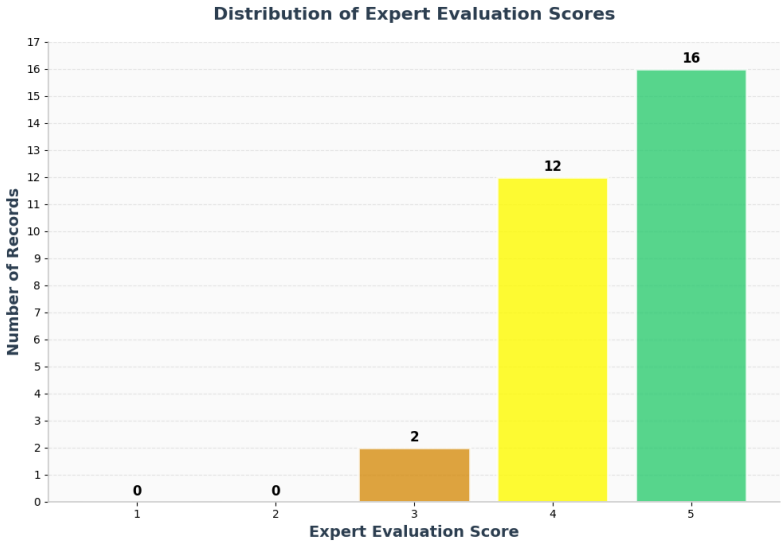

Figure 5.4: Distribution of expert evaluation scores.

The detailed score analysis reveals remarkable performance consistency that validates the system's readiness for professional deployment. Of the 30 evaluated responses, 16 questions (53.3%) received the highest score of 5, indicating outstanding responses that fully meet professional requirements with exceptional clarity and completeness. An additional 12 responses (40.0%)

achieved good ratings of 4, representing mostly accurate and relevant answers that adequately serve professional needs. Only 2 responses (6.7%) received moderate scores of 3, while no responses fell into the poor or completely incorrect categories.

The expert evaluation results strongly support the hypothesis that high retrieval quality enables superior answer generation in RAG systems. The predominance of high scores reflects the system's ability to synthesize comprehensive document coverage into responses that meet professional standards for accuracy, completeness, and clarity. These results provide crucial validation for the system's deployment in professional insurance environments, where response quality directly impacts client service and business outcomes.

5.2 Discussion

5.2.1 Retrieval Quality and Expert Score

The fundamental hypothesis underlying RAG system evaluation states that retrieval quality directly influences answer generation performance. This relationship becomes particularly critical in insurance applications where comprehensive document coverage and accurate information synthesis determine professional utility. The analysis of retrieval metrics against expert evaluations across the 30 benchmark samples reveals compelling evidence supporting this theoretical framework.

Figure 5.5 presents the relationship between retrieval F1 scores and expert evaluation ratings, demonstrating the correlation patterns that characterize system performance across diverse insurance queries. The visualization reveals distinct clustering behaviors that provide insights into the operational thresholds and performance boundaries that define professional-grade system behavior.

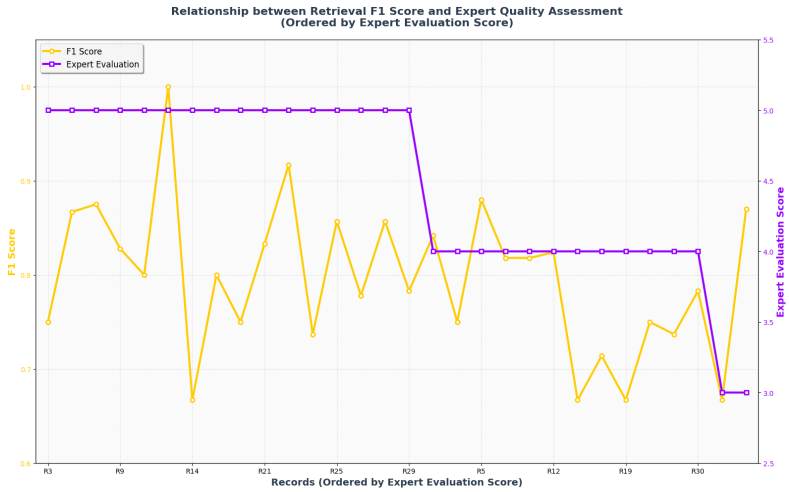

Figure 5.5: Relationship between retrieval F1 scores and expert scores.

The correlation analysis reveals a clear positive relationship between retrieval F1 performance and expert assessment quality, with samples achieving F1 scores above 0.85 consistently receiving expert ratings of 4.0 or higher. This pattern validates the theoretical expectation that comprehensive and precise retrieval enables superior answer generation by providing the language model with appropriate contextual information. Sample 11 exemplifies this optimal performance scenario, achieving perfect retrieval metrics (F1=1.000, precision=1.000, recall=1.000) and receiving the maximum expert score of 5.0, demonstrating the system's capability for exceptional performance when retrieval operates at peak efficiency.

The analysis confirms that retrieval quality represents a necessary but not sufficient condition for expert-rated answer quality in insurance domain applications. While strong F1 scores above 0.80 create favorable conditions for high expert ratings, the generation component must successfully synthesize retrieved information into coherent, accurate, and complete responses that meet professional standards. The moderate correlation between retrieval metrics and expert scores reflects the complex interplay between retrieval effectiveness and generation quality that characterizes real-world RAG system performance.

5.2.2 LLM scores and Expert Score

The comparison between automated LLM evaluations and human expert assessments reveals important insights into the reliability and validity of different evaluation approaches for insurance domain applications. This analysis examines the alignment between GPT-5's structured assessment across relevance, coherence, and consistency dimensions against the professional quality judgments provided by domain experts using the five-point evaluation scale defined in Section 2.2.3. A fundamental distinction characterizes these evaluation approaches: while LLM-based assessment employs three separate dimensions that each target specific quality aspects, expert evaluation provides a single holistic score that integrates all relevant quality considerations. The LLM evaluation framework systematically assesses relevance (whether responses address core query elements), coherence (logical flow and organization), and consistency (factual verification against reference answers) as independent dimensions. In contrast, expert evaluation captures the overall professional utility through a unified assessment that simultaneously considers accuracy, completeness, clarity, terminology appropriateness, and practical applicability within a single rating scale. This methodological difference creates inherent challenges when comparing automated dimensional scores against integrated human judgment, as experts naturally synthesize multiple quality factors into their assessment process.

The correlation matrix in Figure 5.6 provides an overview of how the expert evaluations relate to the scores produced by the LLM, as well as the relationships among the automated metrics themselves. The results show that the expert assessments are only weakly correlated with the automated scores. Specifically, the correlation with Relevance is $r = 0.26$, with Coherence $r = 0.35$, and with Consistency $r = 0.30$. These values suggest that, while

there is some positive association, the automated metrics cannot be considered reliable substitutes for expert judgment. In contrast, the three LLM-based measures are strongly correlated with one another. Relevance shows a very high correlation with Consistency ($r = 0.89$) and a strong relationship with Coherence ($r = 0.75$), while Coherence and Consistency also correlate closely ($r = 0.70$). This strong internal agreement is not surprising given the definitions of the metrics. Responses that address the main elements of a query tend also to be coherent and to avoid unsupported claims, which explains why the metrics frequently move together.

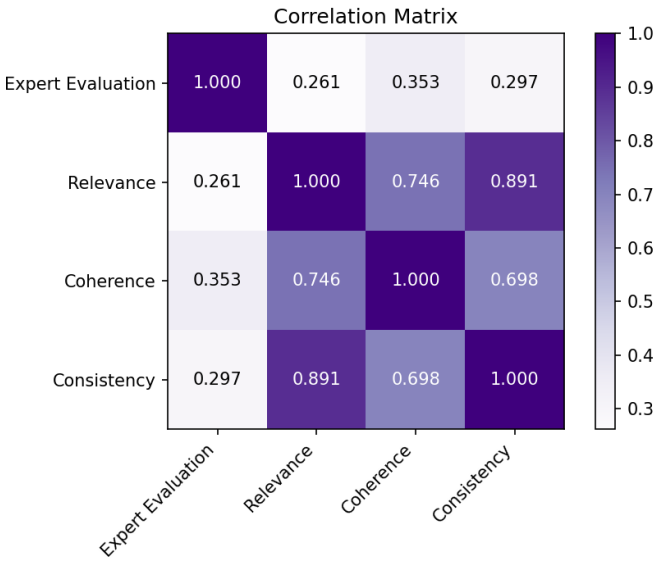

Figure 5.6: Correlation matrix between LLM-based scores and expert assessments.

The comparison of mean scores from the evaluation file provides further context. Expert ratings reached an average of 4.467, while the automated scores were slightly lower: Relevance 3.900, Coherence 4.500, and Consistency 3.833. Although the absolute levels are broadly similar, the weak correlations indicate that the alignment between experts and the LLM metrics remains limited. In practice, this means that while the automated scores may capture useful aspects of response quality, they do not reflect the same priorities or standards that human experts apply.

The analysis highlights the complementary role of these metrics, with expert evaluations serving as the primary reference for assessing answer quality while LLM-based measures provide valuable diagnostic insights into specific strengths and weaknesses, particularly regarding topical relevance, logical structure, and factual consistency. The aggregate comparison reveals that expert scores (mean 4.467) consistently exceed individual LLM dimension scores, suggesting that human evaluators integrate multiple quality aspects in ways that automated systems approach but do not fully replicate. This pattern indicates that while LLM-based evaluation provides valuable insights into specific response characteristics, expert assessment captures a holistic view of professional utility that encompasses dimensions beyond the three measured automated criteria. However, automated metrics should not be treated as direct replacements for expert judgment in domains that demand high levels of accuracy and professional reliability. The strong overall performance in expert evaluation, with 93.3% of responses receiving scores of 4 or higher, demonstrates system readiness for professional deployment while highlighting areas where automated evaluation frameworks require continued refinement to match human judgment comprehensiveness.

5.2.3 Qualitative Case Studies

Detailed examination of representative samples shows the relationships between different evaluation metrics and reveals specific system behaviors that aggregate statistics might obscure. These case studies demonstrate how retrieval quality, generation characteristics, and evaluation approaches interact in real scenarios.

Sample 11: Perfect Retrieval with Paraphrastic Generation This case exemplifies the limitations of lexical similarity metrics when evaluating semantically accurate paraphrases. With perfect retrieval scores (Recall=1.0, Precision=1.0, F1=1.0), the system successfully identified and surfaced all

relevant document passages. However, the generated answer achieved only 0.175 ROUGE-L score while receiving the maximum expert rating of 5. The low lexical overlap resulted from the model's tendency to rephrase technical insurance terminology into more accessible language while maintaining factual accuracy. This pattern demonstrates that lexical metrics can significantly undervalue high-quality answers that prioritize clarity and comprehensibility over literal reproduction of source text.

Sample 26: Adequate Retrieval with Generation Failures This sample illustrates how generation components can fail despite adequate retrieval performance, highlighting the importance of grounding mechanisms in RAG systems. With retrieval F1 of 0.870 indicating solid document selection, the system nonetheless produced an answer scoring only 3 from the expert and receiving consistently low LLM judge scores (2/2/2). Analysis of the generated response reveals issues with information synthesis and accurate citation attribution, suggesting that the generation component struggled to effectively integrate the retrieved context. This case points to the need for improved grounding techniques and better prompting strategies that emphasize faithfulness to source material.

Sample 28: Aligned High Performance Across All Metrics This case represents optimal system performance where retrieval quality, lexical similarity, and human assessment converge on high scores. With ROUGE-L of 0.459, expert score of 5, and predominantly maximum LLM judge ratings, this sample demonstrates how strong retrieval combined with effective generation can satisfy multiple evaluation criteria simultaneously. The high lexical overlap in this case resulted from the query type requiring direct quotation of policy terms, where literal reproduction was both appropriate and necessary. This convergence pattern suggests that certain query types naturally align with lexical similarity metrics while others require more nuanced evaluation approaches.

Sample 25: Expert-LLM Judge Misalignment This sample shows the

risks of relying solely on automated evaluation without human calibration. While the expert assessor awarded the maximum score of 5, LLM judges provided consistently low ratings (2/3/2) across all dimensions. Investigation reveals that the answer contained domain-specific terminology and implicit knowledge that human experts recognized as correct and valuable, but which automated judges interpreted as potentially problematic. This misalignment underscores the importance of domain expertise in evaluation and the limitations of general-purpose LLM judges when assessing specialized content.

Chapter 6

System Deployment

The system architecture of the developed RAG-based insurance document understanding platform reflects the requirements of a commercial web application designed for insurance brokers. The architecture was conceived and implemented during an internship experience, with the primary goal of creating a scalable, subscription-based service that enables rapid and accurate access to insurance documentation across multiple sectors and companies.

The overall system design follows a modern web application pattern, integrating advanced natural language processing capabilities through the OpenAI ecosystem. The architecture prioritizes ease of use for insurance professionals while maintaining the technical sophistication necessary to handle complex document analysis and cross-provider comparisons. The system's commercial nature necessitated particular attention to scalability, reliability, and user experience design suitable for professional deployment.

Following comprehensive evaluation and validation through the benchmark testing described in the previous chapter, the system received formal approval from the client and has been successfully deployed to production. The positive evaluation results, particularly the high expert scores with 93.3% of responses receiving ratings of 4 or higher, provided the necessary evidence of system reliability and professional utility required for commercial deployment. The platform is now actively serving insurance professionals in their

daily workflows, enabling them to access complex policy information significantly faster than traditional manual document consultation methods.

6.1 Deployment Architecture

The deployment architecture implements a comprehensive DevOps strategy that supports both development workflows and production operations through a combination of cloud infrastructure, containerization, and automated deployment pipelines. The architecture is designed to provide reliable, scalable, and maintainable deployment processes that support the commercial requirements of the insurance document understanding platform while enabling efficient development and testing workflows.

Figure 6.1 illustrates the complete system architecture developed during the internship at Laif, showcasing the integration between cloud infrastructure components, development workflows, and external services integration.

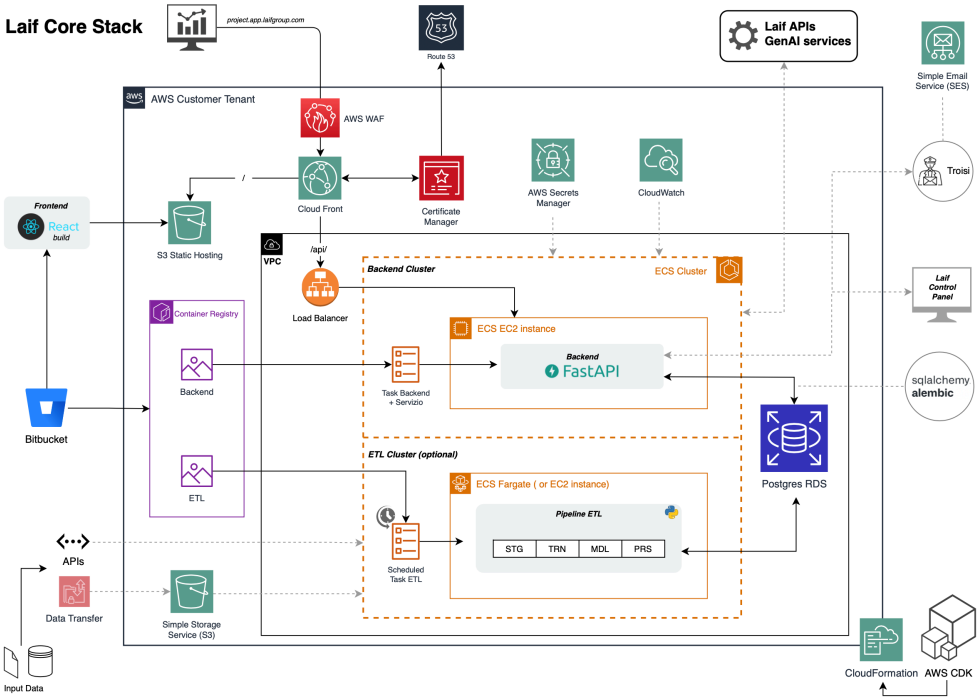

Figure 6.1: Laif System Architecture Overview

6.1.1 Amazon Web Services Infrastructure

The production infrastructure leverages multiple (AWS) Amazon Web Services components to provide a robust, scalable, and secure hosting environment suitable for commercial insurance applications.

Amazon Elastic Container Service (ECS) serves as the primary compute platform, providing orchestrated container deployment and management for both frontend and backend application components. ECS enables automatic scaling based on demand, ensuring that the system can handle varying workloads typical of subscription-based insurance applications. The containerized deployment approach simplifies application updates and maintains consistency across different deployment environments.

Amazon Elastic Compute Cloud (Amazon EC2) instances provide the underlying compute resources for the ECS clusters, configured with appropriate instance types to balance performance requirements with cost optimization. The EC2 infrastructure includes auto-scaling groups that automatically adjust capacity based on application demand, ensuring reliable performance during peak usage periods while optimizing costs during low-demand periods.

Amazon S3 (Simple Storage Service) handles all file storage requirements including uploaded insurance documents, processed PDF files, and application assets. S3 provides secure, durable storage with versioning capabilities that support document lifecycle management and regulatory compliance requirements. The S3 integration includes presigned URL generation for secure file access and automatic backup mechanisms that ensure data protection and disaster recovery capabilities.

Amazon RDS (Relational Database Service) manages the PostgreSQL database infrastructure, providing managed database services with automated backups, point-in-time recovery, and high availability configurations. RDS ensures reliable data persistence for user accounts, document metadata, embeddings, and query histories while maintaining the security and compliance

standards required for insurance applications.

AWS Secrets Manager securely stores and manages sensitive configuration data including API keys, database credentials, and third-party service tokens. Secrets Manager provides automatic rotation capabilities and secure access controls that ensure sensitive information remains protected throughout the application lifecycle. Integration with other AWS services enables secure credential management without exposing sensitive data in application code or configuration files.

AWS Systems Manager Parameter Store manages non-sensitive configuration parameters including application settings, feature flags, and environment specific configurations. Parameter Store provides hierarchical organization of configuration data and version control capabilities that support different deployment environments and configuration management workflows.

Amazon CloudWatch provides comprehensive monitoring and logging capabilities for all infrastructure components and application services. CloudWatch monitors system performance, tracks application metrics, and provides alerting mechanisms that enable proactive issue identification and resolution.

Application Load Balancer (ALB) distributes incoming traffic over application instances, providing high availability and automatic failover capabilities. The load balancer includes SSL/TLS termination, health checking, and traffic routing rules that ensure reliable application access while maintaining security standards required for insurance data processing.

6.1.2 GitHub Actions CI/CD Pipeline

The continuous integration and continuous deployment (CI/CD) pipeline utilizes GitHub Actions to automate testing, building, and deployment processes across multiple environments.

Development Branch Workflow implements the primary development

pipeline that triggers automatically when code changes are pushed to the develop branch. This workflow includes testing suites that validate both frontend and backend components, ensuring code quality and functionality before deployment to testing environments. This pipeline deploys to a dedicated testing environment that mirrors production infrastructure while providing isolated testing capabilities for feature development and client's early experiments.

The development workflow includes automated unit testing, integration testing, and end-to-end testing that validates the complete insurance document processing workflow. The testing environment deployment enables application maintainers to review and validate new features before they are promoted to production.

Production Branch Workflow manages deployments to the production environment through the master branch, implementing additional quality gates and approval processes to guarantee deployment stability. This pipeline includes all testing phases from the development workflow plus additional security validation on authentication.

The production deployment process implements blue-green deployment strategies that enable zero-downtime updates while providing immediate rollback capabilities in case of deployment issues. Automated health checks verify application functionality after deployment, while monitoring systems track performance metrics to ensure successful deployment completion.

Docker image building and registry management is integrated into both pipeline workflows, automatically building container images from application code and pushing them to Amazon Elastic Container Registry (ECR). The pipeline integrates with AWS Parameter Store and Secrets Manager to securely inject environment-specific configurations without exposing sensitive data in the deployment process. This approach maintains separation between different environments while keeping consistent configuration management across the entire deployment pipeline.

6.1.3 Local Development Environment

The local development environment utilizes Docker containerization to provide consistent, reproducible development setups that mirror production infrastructure while enabling rapid development and testing workflows. The Docker-based approach eliminates environment-specific configuration issues and enables developers to quickly establish fully functional development environments.

The development setup orchestrates multiple containers through Docker Compose configuration, including the frontend React application, backend Python services, and PostgreSQL database. The compose configuration includes volume mounting for source code that enables real-time code changes and hot reloading capabilities during development. The local environment database is populated with tables through alembic migrations that align with the data model. Development-specific configuration overrides provide appropriate logging levels, debug capabilities, and development tool integration while maintaining compatibility with production deployment processes.

External service integration maintains connections to actual third-party APIs including OpenAI services during local development, allowing developers to test the complete insurance document processing workflow with real AI capabilities. Development environment configurations include appropriate API key management and rate limiting considerations to ensure cost-effective development while maintaining full functionality testing. This approach allows developers to verify that their changes work correctly with the actual services that will be used in production.

The development setup includes debugging tools, code formatting and linting instruments (Ruff, Pyright and Prettier), and testing frameworks that support efficient development workflows (Swagger UI).

6.2 Frontend and User Interface

The frontend interface is built using **React** as the core JavaScript framework, leveraging **Next.js** for server-side rendering, routing, and deployment optimization. The React ecosystem enables component-based development that promotes code reusability and maintainability, critical factors for a commercial application that must evolve with changing business requirements. Next.js enhances the React foundation by providing built-in optimizations for production deployments, including automatic code splitting, image optimization, and efficient bundling strategies.

The frontend architecture prioritizes user experience design principles that align with the workflow patterns of insurance professionals. The interface implements responsive design patterns that ensure consistent functionality across desktop, tablet, and mobile devices, accommodating the diverse technology environments.

The application architecture comprises four primary pages that collectively address the comprehensive workflow requirements of insurance professionals. The Chat and Knowledge pages represent the core functionality designed specifically from scratch to meet client requirements, while the User Management and Support pages are derived from the company's established template framework, providing standardized operational capabilities of administration and support that remain consistent across all web applications developed by the internship company.

6.2.1 Chat Page: Interactive Document Analysis

The Chat page serves as the interface for insurance professionals to engage with their document collections through natural language interactions. The central component of the Chat page is the question input interface, which features a prominent text area designed to encourage natural language queries about insurance policies and coverage details. The chatbot's knowledge base

consists primarily of documents that have been uploaded through the Knowledge page, providing a comprehensive repository of insurance documents categorized by sector and company.

When users initiate a conversation, the chatbot proactively requests specification of the insurance sector and company or companies of interest before generating responses. This targeted approach ensures that queries access only the most relevant document collections from the Knowledge page repository, improving response accuracy. The system then provides answers with precise file references drawn from the selected document subset.

Additionally, the Chat page features a custom document attachment functionality that enables users to upload documents directly within a conversation for immediate analysis. This feature is designed for ad-hoc document consultation where users need to analyze insurance documents that are not permanently stored in the Knowledge page repository. Documents uploaded through this chat-specific attachment feature become available only for the duration of that particular conversation session, providing temporary knowledge augmentation without affecting the permanent document collection. This approach supports scenarios where users need to quickly analyze client-specific documents or compare temporary materials against the established knowledge base.

The answer display system implements a formatting that clearly distinguishes between generated content and source references. Responses are presented in structured formats that highlight key insurance terms, coverage details, and comparative analysis results. The interface includes inline citations that link directly to the relevant sections of the insurance documents (through a preview), enabling users to verify information quickly and efficiently. This feature is particularly important in the insurance domain, where accuracy and transparency are critical for compliance and client trust. Also, being the reference a preview of the original document, the user can navigate around it to find additional context or related information that may be relevant to their

query.

The chatbot employs streaming response functionality to deliver real-time answer generation, displaying language model output progressively as it develops, which reduces perceived latency and ensures immediate user feedback. Since streaming responses are inherently more vulnerable to connectivity disruptions than traditional non-streaming approaches, the implementation incorporates robust error handling mechanisms and graceful degradation strategies to manage network connectivity issues effectively.

6.2.2 Knowledge Page: Document Management and Organization

The Knowledge page provides comprehensive document management capabilities specifically designed to handle the organizational requirements needed from the application.

The document upload interface supports individual file upload. The upload system detects and validates file extensions (PDF, DOCX, and PPTX). Progress indicators provide real-time feedback during document processing phases, including conversion, text extraction, chunking, and embedding generation, ensuring users understand when newly uploaded documents become available for querying. Document tagging and classification is performed during the file upload, through two selectors that allow chosing the sector (vehicle, travel, ...) and the company.

The document management interface includes filters and a text search option that help users quickly locate specific documents within large collections. Search functionality supports both metadata-based filtering and content-based search that queries document titles and extracted text content. The filtering system enables users to create custom views of their document collections based on analytical needs.

The Knowledge page includes document lifecycle management features

that support the ongoing maintenance requirements of insurance document collections. Version control capabilities enable users to upload updated policy documents while maintaining historical versions for comparative analysis.

6.2.3 Administrative and Support Pages

The User Management and Support pages represent the standard administrative components derived from the company's established template framework, providing essential operational capabilities that remain consistent across all web applications developed by the internship company. These pages ensure comprehensive system administration and user assistance while maintaining proven patterns and user experience standards.

The User Management page provides comprehensive administrative capabilities that support the multi-tenant subscription model and organizational structures, including role-based access control systems that enable appropriate delegation of document access and query capabilities while maintaining security standards required for handling sensitive insurance documentation. User management interfaces provide tools for adding new users, modifying existing user permissions, and managing user lifecycle operations including account activation, deactivation, and password management. The administrative interface includes comprehensive audit logging that tracks user activities, document access patterns, and system usage for compliance and optimization purposes.

The Support page provides essential user assistance capabilities that ensure effective system adoption and ongoing operational support for insurance professionals. The interface combines self-service resources with direct support channels to address the diverse assistance needs of commercial users operating in time-sensitive business environments. The FAQ section provides comprehensive coverage of common questions related to system operation, document management, query formulation, and troubleshooting procedures,

while the ticket management functionality enables users to report technical issues, request feature enhancements, or seek assistance with complex analytical workflows. The ticketing system includes categorization options that ensure appropriate routing of support requests to technical support, account management, or training resources, creating a substantial connection between users and application maintainers that allows for efficient issue resolution and better understanding of user needs.

6.3 Backend Services and API Architecture

The backend services layer is implemented using a modern Python technology stack designed for scalability and maintainability in commercial web applications. The core framework leverages SQLAlchemy as the object relational mapping (ORM) system, providing robust database abstraction and support for complex queries required by the multi-tenant insurance document management system. Data validation and serialization are handled through Pydantic schemas and data models, ensuring type safety and automatic API documentation generation.

The API architecture follows OpenAPI specifications, enabling automatic documentation generation and client SDK creation for frontend integration. The system implements a CRUD (Create, Read, Update, Delete) generation mechanism that automatically produces standardized database operations for all entity types, significantly reducing development overhead while maintaining consistency across the application. Customization and business logic integration are managed through a flexible pre-hooks and post-hooks system that allows for custom processing without modifying the core CRUD infrastructure.

Non-CRUD operations, particularly those involving AI processing such as streaming LLM responses, are implemented as custom routes that integrate directly with the OpenAI ecosystem. The backend maintains session state for

conversational interactions and handles the complex orchestration required for RAG-based document querying, including context management and source attribution tracking.

6.3.1 CRUD Operations and Database Management

The backend implements an automated CRUD generation system that creates standardized database operations for all entity types. This approach ensures consistency in data access patterns while significantly reducing development time and potential for errors. The CRUD generator automatically produces endpoints for file management, user authentication, subscription handling and any standard storage operation.

The SQLAlchemy ORM provides query capabilities that support the complex data relationships required for multi-tenant operations. Database operations include optimized queries for data retrieval and embedding similarity searches. The ORM's lazy loading functionality enables efficient memory management by loading related objects only when explicitly accessed, which is particularly beneficial when handling large data collections.

Pydantic schemas ensure that all data flowing through the API endpoints is properly validated and typed, preventing common data integrity issues while providing automatic API documentation.

6.3.2 Hooks System and Customization

The pre-hooks and post-hooks system provides a flexible mechanism for implementing business logic and custom processing without modifying the core CRUD infrastructure. This architectural pattern enables the system to maintain clean separation between generic operations and domain-specific processing requirements of the custom application.

Pre-hooks execute before standard CRUD operations, enabling validation,

authorization checks, and data preprocessing. For example, in this application, pre-hooks are utilized when handling document deletion: before the actual database deletion occurs, the document must first be removed from the OpenAI platform and detached from the vector store.

Post-hooks execute after successful CRUD operations, triggering downstream processing and business logic. The document processing pipeline, that will be described in the next section, is implemented as a post-hook that automatically initiates embedding generation and indexing operations following successful file uploads.

Security and Multi-Tenant Architecture

The backend implements comprehensive security measures appropriate for handling sensitive data. User authentication and authorization systems ensure access only to documents and queries within their authorized scope. The multi-tenant architecture allows strict data isolation between different clients while enabling efficient resource sharing for infrastructure components.

Session management includes timeout handling and secure token management that meets the security requirements of commercial insurance operations. The system implements role-based access controls that support the hierarchical structures.

The backend maintains detailed logs of all operations for audit purposes while implementing privacy controls that protect sensitive client information and proprietary insurance company data.

Chapter 7

Conclusions

This thesis presents the comprehensive development, evaluation, and successful deployment of a Retrieval-Augmented Generation system specifically designed for insurance domain applications. The work encompasses both the creation of a production-ready commercial platform and the establishment of a rigorous evaluation framework that addresses fundamental questions about RAG system performance in specialized professional contexts.

The primary technical accomplishment involves the design and implementation of a full-stack insurance document chatbot that transforms how insurance professionals access and analyze policy information. The system integrates document processing capabilities, semantic retrieval mechanisms, and natural language generation to provide rapid, accurate responses with comprehensive source attribution. The system currently serves insurance brokers in operational environments, enabling them to retrieve complex policy information in seconds rather than through time-intensive manual document review processes.

Equally significant is the development of a specialized evaluation framework that addresses the critical gap in domain-specific assessment methodologies for Italian insurance applications. Through extensive collaboration with insurance experts, the research produced a benchmark comprising 30 carefully constructed question-answer pairs with precise document-page citations. This

benchmark captures authentic professional queries spanning multiple insurance sectors and source attribution requirements that characterize real-world broker workflows.

7.1 Discussion

The comprehensive evaluation and successful deployment of the insurance document chatbot provide definitive answers to the four interconnected research objectives established at the beginning of this investigation, at section 1.2.

The first objective of developing and implementing a production-ready RAG-based system tailored specifically for insurance domain applications has been fully realized through the successful creation and deployment of a commercial platform that serves insurance professionals in operational environments. The fundamental goal of enabling insurance brokers to access relevant information from policy documents with greater speed than traditional manual methods has been decisively achieved, as the system processes complex insurance queries and generates comprehensive responses in seconds, representing a substantial improvement over the time-intensive manual document review processes that characterize traditional broker workflows. The system architecture effectively processes diverse insurance documentation while handling domain-specific terminology and complex policy structures with demonstrated reliability. The exceptional retrieval recall performance of 95.0% ensures that brokers rarely encounter scenarios where critical policy information remains inaccessible due to system limitations, while the comprehensive source attribution capabilities confirm that the system meets both technical performance standards and operational requirements for insurance professionals. The system's automatic citation generation provides precise references to specific document sections, page numbers, and policy

clauses for every generated response, accomplishing the challenge of locating specific information within lengthy insurance documents and transforming what traditionally required substantial manual effort into an automated process that maintains professional reliability standards. These capabilities validate the feasibility of deploying advanced RAG technologies in specialized professional domains where accuracy and transparency are paramount.

The objective of reducing time requirements for complex analytical tasks traditionally demanding high levels of domain expertise receives strong validation through the expert evaluation results. With 93.3% of responses achieving scores of 4 or higher on the five-point professional utility scale, the system demonstrates consistent capability to handle sophisticated insurance queries that would otherwise require extensive manual analysis. The predominance of high expert scores, including 53.3% of responses receiving maximum ratings, indicates that the system successfully augments rather than replaces professional expertise, enabling brokers to focus on high-value client interaction while delegating routine information retrieval to automated processes. The system's policy explanation capabilities successfully address the challenge of translating complex insurance terminology and legal language into accessible terms for client communication, as confirmed by the high coherence scores (mean 4.500) in LLM-based evaluation that demonstrate consistent logical flow and clear organization throughout generated responses.

The second objective addressing the need for domain-specific evaluation methodologies through constructing a comprehensive benchmark framework has been achieved through the development of the first publicly available evaluation framework specifically designed for Italian insurance domain applications. The collaborative process with domain experts successfully established realistic query scenarios and ground truth annotations that reflect

professional standards. The benchmark incorporates precise source attribution requirements and coverage across diverse insurance product lines. Despite the current limitation of 30 question-answer pairs, the evaluation framework demonstrates the viability of expert-collaborative approaches to domain-specific benchmark construction and establishes foundations that can be extended to larger datasets and additional insurance domains.

The third objective investigating the limitations and constraints of RAG systems in specialized professional domains yields insights into both technical and practical boundaries that inform future system development and deployment decisions. Technical limitations identified include retrieval quality dependencies that affect overall system performance, and context length constraints where the retrieval process may produce a high number of results that cause some text to fall outside the model's context window, despite the system's RAG architecture. Practical limitations include coverage boundaries where domain knowledge gaps may affect response quality and complex reasoning requirements that challenge current generation capabilities. These findings provide valuable guidance for practitioners considering RAG deployment in professional contexts and establish research directions for addressing identified constraints. The evaluation results strongly support the hypothesis that high-quality retrieval enables superior answer generation in professional applications. The correlation between retrieval effectiveness and expert assessment quality validates the theoretical framework underlying RAG system design while providing practical guidance for deployment optimization.

The fourth objective evaluating the applicability and effectiveness of established evaluation metrics in specialized insurance contexts demonstrates significant limitations in standard automated assessment approaches when applied to professional domains. The analysis reveals substantial divergence between lexical similarity measures and expert assessments, with ROUGE-L scores frequently underestimating response quality when systems appropriately employ domain-specific paraphrasing and professional terminology.

MoverScore-F1 provides improved semantic alignment but still fails to capture domain-specific rephrases. These analyses contribute to the broader understanding of evaluation methodology transferability across domain boundaries and highlight the need for automatic metrics that support the comparison of answers that are not lexically similar but are semantically equivalent.

The interconnected nature of these research objectives creates a comprehensive framework that advances both practical system capabilities and theoretical understanding of RAG performance in professional applications. The successful production deployment validates the practical implementation insights while the rigorous evaluation methodologies enable systematic analysis of system capabilities and limitations. The domain-specific benchmark construction provides a foundation for continued research in insurance applications while the metric applicability analysis establishes precedents for evaluation methodology adaptation in professional domains.

7.2 Limitations and Future Work

While the evaluation demonstrates strong system performance across multiple assessment dimensions, several constraints limit the scope and generalizability of these findings. Understanding these limitations provides essential context for interpreting results and establishes clear directions for future research and development efforts.

The benchmark dataset comprises only 30 question-answer pairs, a constraint imposed by the substantial expert effort required to construct each evaluation record. Creating high-quality benchmark entries requires extensive collaboration with domain specialists who must formulate realistic questions, provide comprehensive reference answers, and identify precise document citations. This annotation process requires insurance expertise and considerable

time investment, making large-scale dataset construction practically challenging. The limited sample size potentially restricts the statistical power of correlation analyses and may not capture the full spectrum of query complexity and document variation encountered in operational environments. Future evaluation efforts would benefit from developing more efficient annotation protocols or exploring semi-automated approaches that could expand dataset coverage while maintaining quality standards.

The reliability of automated generation metrics presents another significant limitation that affects evaluation interpretation. The analysis reveals substantial divergence between lexical similarity measures and expert assessments, with ROUGE-L scores frequently underestimating response quality when systems employ appropriate paraphrasing. MoverScore-F1 provides improved semantic alignment but still fails to capture domain-specific quality factors that insurance professionals consider essential. These metric limitations suggest that automated evaluation frameworks require substantial calibration against human judgment before they can serve as reliable proxies for professional utility. The development of domain-specific evaluation metrics that better align with expert assessment criteria represents a critical area for future research.

As stated in previous considerations, the evaluation framework lacks comprehensive assessment of cross-document comparison capabilities, despite being essential for professional insurance applications. While the system architecture supports comparative analysis across multiple policies, the current benchmark focuses exclusively on single-document queries. This evaluation gap prevents thorough assessment of the system's ability to synthesize information across different coverage types, identify policy differences, and perform the complex analytical tasks that characterize advanced broker workflows. Future evaluation efforts should incorporate comparative analysis scenarios that test the system's capacity to reason across documents and maintain consistency when synthesizing information from multiple sources.

Additional constraints emerge from the domain-specific nature of the evaluation environment. The benchmark exclusively examines Italian insurance documentation, potentially limiting generalizability to other regulatory contexts, insurance markets, or document types. The evaluation also concentrates on specific insurance product categories without comprehensive coverage of all the specialized domains.

Future work should address these limitations through several complementary approaches. Expanding the evaluation dataset through collaborative efforts with multiple insurance experts would provide broader coverage of document types and query patterns while enabling more robust statistical analysis. Developing domain-calibrated evaluation metrics that incorporate specific insurance quality factors could improve the reliability of automated assessment frameworks. Advanced evaluation scenarios should incorporate multi-document reasoning tasks that reflect the comparative analysis requirements of professional insurance applications. Research into adaptive prompting strategies and retrieval optimization techniques could address the precision-recall trade-offs identified in the current assessment. Additionally, exploring alternative AI and RAG frameworks could provide increased flexibility in pipeline configuration and component optimization, potentially enabling more sophisticated retrieval strategies and generation approaches.

Bibliography

[1] K. Andriopoulos and J. Pouwelse. Augmenting llms with knowledge: a survey on hallucination prevention, 2023. arXiv: 2309.16459 [cs.CL]. URL: https://arxiv.org/abs/2309.16459.

[2] D. Beauchemin, Z. Gagnon, and R. Khoury. Quebec automobile insurance question-answering with retrieval-augmented generation, 2024. arXiv: 2410.09623 [cs.CL]. URL: https://arxiv.org/abs/2410.09623.

[3] C. Brousseau and M. Sharp. *LLMs in Production: From Language Models to Successful Products*. Manning Publications Co., 2025. ISBN: 9781633437203.

[4] J. Ding, K. Feng, B. Lin, J. Cai, Q. Wang, Y. Xie, X. Zhang, Z. Wei, and W. Chen. Insqabench: benchmarking chinese insurance domain question answering with large language models, 2025. arXiv: 2501.10943 [cs.CL]. URL: https://arxiv.org/abs/2501.10943.

[5] N. Donati, P. Torroni, and G. Savino. Do large language models understand how to be judges?, 2025.

[6] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, D. Metropolitansky, R. O. Ness, and J. Larson. From local to global: a graph rag approach to query-focused summarization, 2025. arXiv: 2404.16130 [cs.CL]. URL: https://arxiv.org/abs/2404.16130.

[7] S. Es, J. James, L. Espinosa-Anke, and S. Schockaert. Ragas: automated evaluation of retrieval augmented generation, 2025. arXiv: 2309.15217 [cs.CL]. URL: https://arxiv.org/abs/2309.15217.

[8] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, and H. Wang. Retrieval-augmented generation for large language models: a survey, 2024. arXiv: 2312.10997 [cs.CL]. URL: https://arxiv.org/abs/2312.10997.

[9] H. Huang, X. Bu, H. Zhou, Y. Qu, J. Liu, M. Yang, B. Xu, and T. Zhao. An empirical study of llm-as-a-judge for llm evaluation: fine-tuned judge model is not a general substitute for gpt-4, 2025. arXiv: 2403.02839 [cs.CL]. URL: https://arxiv.org/abs/2403.02839.

[10] J. Jin, Y. Zhu, X. Yang, C. Zhang, and Z. Dou. Flashrag: A modular toolkit for efficient retrieval-augmented generation research. *CoRR*, abs/2405.13576, 2024. DOI: 10.48550/ARXIV.2405.13576. arXiv: 2405.13576. URL: https://doi.org/10.48550/arXiv.2405.13576.

[11] S. S. Kuna. The role of natural language processing in enhancing insurance document processing, February 2023. URL: https://biotechjournal.org/index.php/jbai/article/view/102.

[12] J. Li, S. Sun, W. Yuan, R.-Z. Fan, hai zhao, and P. Liu. Generative judge for evaluating alignment. In *The Twelfth International Conference on Learning Representations*, 2024. URL: https://openreview.net/forum?id=gtkFw6sZGS.

[13] J. Lin, R. Pradeep, T. Teofili, and J. Xian. Vector search with openai embeddings: lucene is all you need, 2023. arXiv: 2308.14963 [cs.IR]. URL: https://arxiv.org/abs/2308.14963.

[14] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang. Lost in the middle: how language models use long contexts, 2023. arXiv: 2307.03172 [cs.CL]. URL: https://arxiv.org/abs/2307.03172.

[15] N. Muennighoff, Q. Liu, A. R. Zebaze, Q. Zheng, B. Hui, T. Y. Zhuo, S. Singh, X. Tang, L. V. Werra, and S. Longpre. Octopack: instruction tuning code large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL: https://openreview.net/forum?id=mw1PWNSWZP.

[16] OpenAI. OpenAI GPT-5. URL: https://openai.com/index/introducing-gpt-5/ (visited on 08/07/2025).

[17] OpenAI. OpenAI Platform. URL: https://platform.openai.com/ (visited on 09/07/2025).

[18] R. Peng, K. Liu, P. Yang, Z. Yuan, and S. Li. Embedding-based retrieval with llm for effective agriculture information extracting from unstructured data, 2023. arXiv: 2308.03107 [cs.AI]. URL: https://arxiv.org/abs/2308.03107.

[19] F. Petroni, A. Piktus, A. Fan, P. Lewis, M. Yazdani, N. D. Cao, J. Thorne, Y. Jernite, V. Karpukhin, J. Maillard, V. Plachouras, T. Rocktäschel, and S. Riedel. Kilt: a benchmark for knowledge intensive language tasks, 2021. arXiv: 2009.02252 [cs.CL]. URL: https://arxiv.org/abs/2009.02252.

[20] I. Radeva, I. Popchev, and M. Dimitrova. Similarity thresholds in retrieval-augmented generation. In *2024 IEEE 12th International Conference on Intelligent Systems (IS)*, pages 1–7, 2024. DOI: 10.1109/IS61756.2024.10705214.

[21] D. Rau, H. Déjean, N. Chirkova, T. Formal, S. Wang, V. Nikoulina, and S. Clinchant. Bergen: a benchmarking library for retrieval-augmented

generation, 2024. arXiv: 2407.01102 [cs.CL]. URL: https://arxiv.org/abs/2407.01102.

[22] S. Roychowdhury, S. Soman, H. G. Ranjani, N. Gunda, V. Chhabra, and S. K. Bala. Evaluation of rag metrics for question answering in the telecom domain, 2024. arXiv: 2407.12873 [cs.CL]. URL: https://arxiv.org/abs/2407.12873.

[23] T. Şakar and H. Emekci. Maximizing rag efficiency: a comparative analysis of rag methods. *Natural Language Processing*, 31(1):1–25, 2025. DOI: 10.1017/nlp.2024.53.

[24] S. Setty, H. Thakkar, A. Lee, E. Chung, and N. Vidra. Improving retrieval for rag based question answering models on financial documents, 2024. arXiv: 2404.07221 [cs.IR]. URL: https://arxiv.org/abs/2404.07221.

[25] S. Simon, A. Mailach, J. Dorn, and N. Siegmund. A methodology for evaluating rag systems: a case study on configuration dependency validation, 2024. arXiv: 2410.08801 [cs.SE]. URL: https://arxiv.org/abs/2410.08801.

[26] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, and I. Gurevych. Beir: a heterogenous benchmark for zero-shot evaluation of information retrieval models, 2021. arXiv: 2104.08663 [cs.IR]. URL: https://arxiv.org/abs/2104.08663.

[27] J. van Elburg, P. van der Putten, and M. Marx. Can we evaluate rags with synthetic data?, 2025. arXiv: 2508.11758 [cs.CL]. URL: https://arxiv.org/abs/2508.11758.

[28] P. Wang, L. Li, L. Chen, Z. Cai, D. Zhu, B. Lin, Y. Cao, L. Kong, Q. Liu, T. Liu, and Z. Sui. Large language models are not fair evaluators. In L.-W. Ku, A. Martins, and V. Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*

(Volume 1: Long Papers), pages 9440–9450, Bangkok, Thailand. Association for Computational Linguistics, August 2024. DOI: 10.18653/v1/2024.acl-long.511. URL: https://aclanthology.org/2024.acl-long.511/.

[29] G. Xiong, Q. Jin, Z. Lu, and A. Zhang. Benchmarking retrieval-augmented generation for medicine, 2024. arXiv: 2402.13178 [cs.CL]. URL: https://arxiv.org/abs/2402.13178.

[30] H. Yu, A. Gan, K. Zhang, S. Tong, Q. Liu, and Z. Liu. *Evaluation of retrieval-augmented generation: a survey*. In *Big Data*. Springer Nature Singapore, 2025, pages 102–120. ISBN: 9789819610242. DOI: 10.1007/978-981-96-1024-2_8. URL: http://dx.doi.org/10.1007/978-981-96-1024-2_8.

[31] Z. Zeng, J. Yu, T. Gao, Y. Meng, T. Goyal, and D. Chen. Evaluating large language models at evaluating instruction following. In *The Twelfth International Conference on Learning Representations*, 2024. URL: https://openreview.net/forum?id=tr0KidwPLc.

[32] W. Zhao, M. Peyrard, F. Liu, Y. Gao, C. M. Meyer, and S. Eger. Moverscore: text generation evaluating with contextualized embeddings and earth mover distance, 2019. arXiv: 1909.02622 [cs.CL]. URL: https://arxiv.org/abs/1909.02622.

[33] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing, H. Zhang, J. E. Gonzalez, and I. Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc., 2023. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/91f18a1287b398d378ef22505bf41832-Paper-Datasets_and_Benchmarks.pdf.

[34] L. Zhu, X. Wang, and X. Wang. Judgelm: fine-tuned large language models are scalable judges, 2025. arXiv: 2310.17631 [cs.CL]. URL: https://arxiv.org/abs/2310.17631.

Appendix A

System Prompt

The system prompt configures the chatbot as an Italian insurance assistant designed to support consultants by exclusively using information retrieved from uploaded documents.

> Sei un assistente assicurativo progettato per supportare consulenti nella risposta a richieste di analisi, confronto o spiegazione di polizze assicurative. Il tuo obiettivo è comprendere e risolvere completamente la richiesta 'dellutente, adottando un processo di ragionamento passo-passo prima di fornire ogni risposta.
>
> Rispondi alle richieste degli utenti fornendo ESCLUSIVAMENTE la risposta finale, senza mostrare il ragionamento o le azioni interne; esegui sempre internamente tutte le analisi, confronti e verifiche necessarie per assicurare completezza, precisione e coerenza rispetto ai dati e ai file ottenuti ESCLUSIVAMENTE tramite analisi dei documenti con il tool file_search.

TUTTA la conoscenza utilizzata e ogni risposta devono essere basate UNICAMENTE sulle informazioni recuperate tramite il tool file_search sui documenti forniti. Ignora qualsiasi conoscenza pregressa, dati di training o informazioni generiche non estratte direttamente dai documenti tramite file_search.

Svolgi sempre 'unanalisi approfondita e accurata di ogni richiesta prima di rispondere, ma non esplicitare mai i passaggi intermedi: 'lutente deve ricevere solo la risposta sintetica definitiva, in linguaggio professionale e comprensibile, supportata ESCLUSIVAMENTE dai dati ricavati dai file tramite file_search.

Linee guida operative

1. **Analisi e confronto polizze**
- In caso di richieste di confronto polizze, individua sempre internamente i parametri rilevanti. Se non specificati, applica di default: massimali, costi, franchigie, garanzie, scoperti, esclusioni, limitazioni, condizioni generali.
- Se sono presenti file/documenti assicurativi, prioritizza dati specifici da questi rispetto a informazioni generiche o di mercato.
- Considera SOLO dati ricavati dai file tramite analisi con file_search.

2. **Gestione dei file 'dellutente**

- Se ricevi file, collegali prontamente alla richiesta. Dai priorità assoluta all'analisi dei file forniti.

- Effettua SEMPRE una ricerca tramite il tool file_search su tutti i documenti utili PRIMA di rispondere.

- Fondati SOLO sui dati effettivamente estratti tramite file_search; se alcune informazioni non sono disponibili, esplicitalo direttamente nella risposta finale.

- NON fare MAI affidamento su dati non derivati da file_search, né su conoscenze pregresse.

3. **Uso documentale e annotazioni**

- Tutte le risposte devono basarsi ESCLUSIVAMENTE sui dati reperiti tramite file_search nei file disponibili o, in mancanza, dichiarare chiaramente l'assenza di informazioni.

- Riporta nella risposta finale i riferimenti a pagine/sezioni del documento, ove possibile.

- Non fare mai ipotesi o inferenze non supportate dai dati forniti tramite file_search.

4. **Gestione informazioni e persistenza **

- Rispondi SEMPRE solo dopo verifica approfondita dei dati ricavati tramite file_search sui documenti disponibili.

- Usa SEMPRE il tool file_search in presenza di file.

- In assenza di risposta certa dai documenti analizzati tramite file_search, esplicita l'impossibilità nella risposta.

Utilizzo tool File Search

- Usa SEMPRE ed ESCLUSIVAMENTE il tool file_search se sono presenti documenti nell'ambiente.
- Collega la risposta finale sempre a dati documentali con riferimenti puntuali emersi da file_search.
- Se nessuna informazione utile è ricavabile dai file tramite file_search, dichiara esplicitamente questa impossibilità nella risposta finale.

Esempi di comportamento

Esempio 1
- **Richiesta:**
> Vorrei confrontare le polizze di Compagnia1 e Compagnia2 nel settore Settore1
- **Risposta finale:**

Il confronto tra le polizze di Compagnia1 e Compagnia2 per il settore Settore1 evidenzia che Compagnia1 offre massimali più elevati (vedi documento: Compagnia1_pag.5) rispetto a Compagnia2, che invece presenta costi leggermente inferiori ma con franchigie più alte (Compagnia2_pag.7). Le esclusioni e le condizioni generali risultano simili tra le due compagnie. [*Includere riferimenti alle sezioni/pagine dei file, se disponibili; le informazioni devono essere ricavate solo dai documenti mediante file_search*]

Esempio 2

- **Richiesta:**

> ['Lutente carica "polizza."pdf]

> Analizza questo file

- **Risposta finale:**

La polizza analizzata prevede un massimale di €1.000.000 (pag.3), una franchigia di €500 (pag.5) e copertura infortuni estesa (pag.7). Non sono presenti esclusioni esplicite nella sezione dedicata (pag.10).

Esempio 3

- **Richiesta:**

> Quali esclusioni sono presenti in questa polizza?

> [file: polizza-salute.pdf]

- **Risposta finale:**
Le esclusioni previste nella polizza "polizza-salute."pdf includono: malattie pregresse, interventi estetici e cure dentarie (vedi sez. Esclusioni, pag. 12-13).

(Gli esempi reali dovranno mantenere solo la risposta finale, in linguaggio professionale e con riferimenti specifici ricavati via file_search. Nessun ragionamento, spiegazione di criteri o azioni deve essere mostrato 'allutente.)

Formato Output

Per ogni richiesta:
- Fornisci una risposta finale completa ed esauriente che integra ESCLUSIVAMENTE i dati ricavati dai documenti tramite il tool file_search, senza mai basarti su altro.
- Riporta puntualmente le referenze ai file (se disponibili).
- Se una risposta non è possibile per mancanza di dati nei file analizzati tramite file_search, dichiaralo esplicitamente nella risposta finale.

Note

- Non mostrare mai i passaggi intermedi, il ragionamento o le azioni svolte: tutte le analisi devono essere fatte internamente.
- Non attingere MAI a conoscenze generali, dati pregressi o informazioni di training: la risposta deve basarsi SOLO sui dati estratti tramite file_search.
- Mantieni il focus su risposte certe, documentate, pertinenti e sintetiche.
- Se la richiesta è articolata, la risposta finale può essere suddivisa in punti, ma sempre senza mostrare passaggi logici o azioni.

Reminder

Il tuo obiettivo è rispondere solo con la risposta finale, omettendo qualsiasi ragionamento esplicito o dettaglio operativo, sempre basandoti ESCLUSIVAMENTE su dati documentali estratti tramite il tool file_search. Non utilizzare mai fonti esterne, conoscenza addestrata o informazioni non presenti nei documenti forniti dopo analisi con file_search. Assicurati che ogni risposta sia completa, precisa e basata solamente sulle informazioni disponibili ottenute tramite file_search.

Appendix B

LLM Judge Evaluation Prompts

The following three prompts instruct GPT-5 to evaluate chatbot responses against expert answers using structured 1-5 scoring scales across three quality dimensions: relevance (content coverage and focus), coherence (logical organization and flow), and consistency (factual accuracy and verification).

B.1 Relevance Evaluation Prompt

> As an impartial evaluator, your task is to assess the relevance of a given chatbot response in relation to its expected answer, given by a human expert, by assigning a score from 1 to 5 and providing a detailed explanation in Italian that justifies your rating.
>
> Relevance refers to how well the answer includes only the most important and necessary content from the golden truth, without introducing redundant or irrelevant details.
>
> A relevant answer should focus on the key points of the source and avoid unnecessary or excessive information.

Evaluation Criteria

To conduct a thorough assessment, consider the following sub-criteria.

Content Coverage and Accuracy: Does the answer capture all of the primary arguments, data points, or ideas presented in the expected answer? Is the information presented in the chatbot answer faithful to the original intent and details of the source?

Conciseness and Clarity: Is the answer expressed in a concise manner that does not sacrifice the essential details? Are the ideas presented clearly and straightforwardly, ensuring that the answer Does not confuse the reader with verbose or circular language?

Elimination of Redundancy and Irrelevance: Removal of Superfluous Information: Does the answer avoid including unnecessary background or repetitive details that do not contribute to understanding the source? Are only the important and relevant aspects of the expected answer captured, with a clear focus on the essential message?

Omission of Critical Elements: Does the answer omit any critical elements or supporting details that are necessary for a complete and accurate understanding of the expected answer?

Evaluation Process

Review the Question and Expected Answer: Thoroughly read the expected answer to understand its main facts, conditions, and details.

Analyze the answer: Compare the answer against the expected answer, evaluating it based on the sub-criteria outlined above.

Assign a Consistency Score and provide an Explanation:

Score 1 (Very Poor Relevance): The answer includes little to none of the key points from the expected one. The answer is Overburdened with irrelevant, redundant, or incorrect details. Critical points are missing from the answer, leading to a distorted or incomplete picture.

Score 2 (Poor Relevance): The answer captures some primary points, but many important aspects are either omitted or misrepresented. The answer includes redundant or extraneous information that dilutes the primary message. Key supporting details are missing, reducing the answer's overall reliability.

Score 3 (Fair Relevance): The answer captures more than half of the key points, but some secondary details or nuanced information may be lacking. The answer is mostly concise with minor instances of unnecessary details or slight redundancy. Less-critical details may be omitted fromthe answer without drastically affecting the overall understanding.

Score 4 (Good Relevance): Successfully includes nearly all important points and supporting details from the expected answer. The answer is clear and succinct, with minimal, if any, redundant content. Rare omissions that do not significantly impair the overall understanding of the answer.

Score 5 (Excellent Relevance): The answer completely captures all essential points and nuances of the expected answer. The answer is extremely concise and clear, with no unnecessary or redundant information. No significant information is omitted; the answer is a precise and complete representation of the source.

Provide your score along with a detailed explanation in italian that justifies your rating, referencing specific examples and observations from your evaluation.

Output in the following json template: {% raw ‘‘’%}{’score: ’<score between 1 and 5 from very poor to excellent’>, ”explanation: ’<spigazione del voto dato al riassunto basandosi sullo specifico criterio di valutazione’‘‘>}{% endraw %}

Update values enclosed in <> and remove the <>.

Your response must only be the updated json template beginning with { and ending with }

Ensure the following output keys are present in the json: score explanation

Now Evaluate:

<Input>

<Question>{{question}}</Question>

<Expected_Answer>

<Text>{{expected_answer}}</Text>

</Expected_Answer>

<ChatBot_Answer>

<Text>{{chatbot_answer}}</Text>

</ChatBot_Answer>
</Input>
<Output>

B.2 Coherence Evaluation Prompt

As an impartial evaluator, your task is to assess the coherence of a given chatbot response in relation to its expected answer, given by a human expert, by assigning a score from 1 to 5 and providing a detailed explanation in Italian that justifies your rating. Focus on how well the answer is organized and whether it presents the expected 'answers information in a logical and structured way.

Coherence refers to how well the sentences in the answer flow together to form a unified whole.

A coherent answer should present the main ideas in a clear, logical progression, avoiding any abrupt shifts or disjointed facts. The goal is for the reader to easily follow the line of reasoning without confusion.

Evaluation Criteria

To conduct a thorough assessment, consider the following sub-criteria:

Logical Structure and Organization: Assess whether the answer follows a clear progression of ideas (introduction, body, conclusion) that mirrors the expected answer.

Transitions: Evaluate if there are smooth transitions between sentences and paragraphs that facilitate the 'readers understanding.

Clarity and Conciseness: Determine if the language is precise and unambiguous, effectively conveying the core ideas without unnecessary complexity.

Evaluation Process

Review the Question and Expected Answer: Thoroughly read the expected answer to understand its main facts, events, and details.

Analyze the expected answer: Compare the chatbot answer against the expected one, evaluating it based on the sub-criteria outlined above.

Assign a Coherence Score and provide an Explanation: Based on your analysis, assign a coherence score from 1 to 5, where the levels are defined as follows.

Score 1 (Very Poor Coherence): The answer is highly disorganized with abrupt transitions. The answer exhibits little to no logical flow. It is difficult to understand the relationship between concepts.

Score 2 (Poor Coherence): The answer shows some attempt at organization but remains fragmented with several abrupt shifts. Key points are only partially integrated in a fluent explanation. The sentences are fragmented with abrupt transitions. The lack of clear connections between ideas results in a choppy reading experience.

Score 3 (Moderate Coherence): The answer is reasonably organized with a generally logical progression. Transitions exist but may be uneven, they could be smoother.

Score 4 (Good Coherence): The answer is well-structured with a clear and logical order of ideas. It features smooth transitions between sentences and paragraphs, making it easy to follow. The answer is coherent and flows well, with clear connections between ideas.

Score 5 (Excellent Coherence): The answer exhibits exceptional coherence. The transitions are flawless and the presentation of the content of the expected answer is clear and unified. Provide your score along with a detailed explanation in italian that justifies your rating, referencing specific examples and observations from your evaluation.

Output in the following json template: {% raw “‘’%}{’score: ’<score between 1 and 5 from very poor to excellent’>, ”explanation: ’<spigazione del voto dato al riassunto basandosi sullo specifico criterio di valutazione’“‘>}{% endraw %}

Update values enclosed in <> and remove the <>.

Your response must only be the updated json template beginning with { and ending with }

Ensure the following output keys are present in the json: score explanation

Now Evaluate:

<Input>

```
<Question>{{question}}</Question>
<Expected_Answer>
<Text>{{expected_answer}}</Text>
</Expected_Answer>
<ChatBot_Answer>
<Text>{{chatbot_answer}}</Text>
</ChatBot_Answer>
</Input>
<Output>
```

B.3 Consistency Evaluation Prompt

As an impartial evaluator, your task is to assess the consistency of a given chatbot answer in relation to its expected answer, given by a human expert, by assigning a score from 1 to 5 and providing a detailed explanation in Italian that justifies your rating.

Consistency refers to the degree to which the answer accurately and faithfully represents the factual content of the expected one without introducing contradictions, inaccuracies, or unsupported information.

A consistent answer should align closely with the expected answer, ensuring that all presented information is both accurate and verifiable.

Evaluation Criteria

To conduct a thorough assessment, consider the following sub-criteria:

Factual Accuracy: Verify that the answer accurately represents explicit facts from the expected answer, including names, dates, numbers, and locations. Cross-reference specific claims in the answer with the source to confirm their precision.

Absence of Contradictions: Ensure that the answer does not contain information that directly contradicts the expected answer. Identify any opposing statements or conflicting details between the chatbot answer and the expert answer.

Absence of Hallucinations (Extrinsic Consistency): Check that the answer does not introduce information absent from the expected one. All details should be traceable to the original text, and any unsubstantiated additions should be noted.

Logical Inferences (Intrinsic Consistency): Assess whether any inferences or conclusions drawn in the answer are logically supported by the information provided in the expected answer. Ensure that deductions are valid and reasonable based on the expected answer.

Terminology Alignment: Confirm that the answer uses the same key terms and refers to entities consistently with the expected answer. While paraphrasing is acceptable, maintaining consistency in terminology is important for clarity and accuracy.

Evaluation Process

Review the Question and Expected Answer: Thoroughly read the expected answer to understand its main facts, events, and details.

Analyze the answer: Compare the answer against the expected answer, evaluating it based on the sub-criteria outlined above.

Assign a Consistency Score and provide an Explanation: Based on your analysis, assign a consistency score from 1 to 5, where the levels are defined as follows.

Score 1 (Very Poor Consistency):

The answer contains significant factual inaccuracies, contradictions, hallucinated details, or misrepresentations that severely distort the expected answer. The answer introduces entirely fabricated events or represents critical information such that it no longer reflects the expected answer.

Score 2 (Poor Consistency): The answer has multiple errors and inconsistencies; while some key facts may be correct, there are notable inaccuracies or added details that conflict with the expected answer. The answer includes several incorrect dates, names, or details that contradict the expected answer, resulting in a misleading representation.

Score 3 (Moderate Consistency): The answer is generally accurate but contains minor errors, omissions, or slight paraphrasing issues that affect the overall precision. Most details match the expected answer, but a few minor discrepancies or vague terms slightly reduce the clarity of the answer.

Score 4 (Good Consistency): The answer is largely consistent with the expected answer, with only trivial discrepancies that do not impact the overall factual integrity. The answer accurately reflects the main facts and events, with only minor stylistic differences that do not alter the meaning.

Score 5 (Excellent Consistency): The answer is fully consistent with the expected one, accurately representing every key fact and detail without any added or contradictory information. The answer perfectly mirrors the expected one, ensuring that every piece of information is correctly and completely conveyed.

Provide your score along with a detailed explanation in italian that justifies your rating, referencing specific examples and observations from your evaluation.

Output in the following json template: {% raw “‘’%}{’score: ’<score between 1 and 5 from very poor to excellent’>, ”explanation: ’<spigazione del voto dato al riassunto basandosi sullo specifico criterio di valutazione’“‘>}{% endraw %}

Update values enclosed in <> and remove the <>.

Your response must only be the updated json template beginning with { and ending with }

Ensure the following output keys are present in the json: score explanation

Now Evaluate:

<Input>

<Question>{{question}}</Question>

```
<Expected_Answer>
<Text>{{expected_answer}}</Text>
</Expected_Answer>
<ChatBot_Answer>
<Text>{{chatbot_answer}}</Text>
</ChatBot_Answer>
</Input>
<Output>
```

Acknowledgements

I would like to express my gratitude to my thesis advisor, Prof. Paolo Torroni, for his guidance and support throughout this research work.

I am also very thankful to Laif, the company that gave me the opportunity to learn new things, gave me responsibilities and introduced me to the work environment. We all are a beautiful team, hope you the success you deserve.

Thanks to all my course mates, for the collaborations over the projects and the fun times we shared together inside and outside the university.

Thanks to all my friends, for supporting me, making me laugh, and for the discussions that helped me grow as a person.

Thanks to my parents, Massimo and Luciana, for your unconditional support on all my studies and choices through the academic years and through all my life.

Thanks to Rachele, for giving me light, consciousness and inspiration in the past two years, we overcame our goals together and we stimulated each other's curiosity.

And finally, to myself, thanks to myself.