

An abstract graphic on the left side of the slide. It features a series of bright green points connected by thin, glowing green lines, forming a network or mesh structure that curves upwards and to the right. The background is dark green with some faint, out-of-focus light spots.

Final Course Project

Meta-data driven RAG Agent for insurance documents analysis

The Insurance RAG Agent System is an intelligent document processing and analysis system specifically designed for the insurance industry. It serves as a comprehensive solution for processing, understanding, and extracting insights from complex insurance documents including accident reports, policy documents, and payment reports.

Personal Background



My name is Golda Krishtalev

- B.Sc. in Computer Science, Statistics and Mathematics
- M.Sc. in Statistics and Computer Science
- 15 years of IT industry experience
- Senior Java and Python developer
- Primarily back-end development focus

Business objectives of the project

The Insurance RAG Agent can automatically extract key data points from various documents related to an insurance event. It can identify dates, times, locations, parties involved, actions taken, and damages incurred. The Insurance RAG Agent processing detailed insurance event timelines transform a complex, manual process into an efficient, data-driven operation. This directly translates into achieving critical business goals:

- Faster claims Reduced costs
- Enhanced fraud detection
- Better risk management
- Improved compliance

Insurance RAG Agent System Capabilities

1. Intelligent Document Processing

- Multi-format PDF Handling: Processes complex insurance documents containing text, tables, and diagrams
- Advanced Table Extraction: Uses pdfplumber for table extraction from PDFs
- Metadata Preservation: Automatically extracts and maintains document structure, page numbers, and case information
- Chunk Optimization: Intelligently segments documents while preserving context and relationships

2. Hybrid Retrieval System

- Dual Search Approach: Combines semantic search (OpenAI embeddings) with keyword search (BM25)
- Advanced Reranking: Uses cross-encoder with LLM-based fallback for optimal result ordering
- Metadata Filtering: Precise filtering capabilities for targeted document retrieval
- Context-Aware Retrieval: Understands query intent and retrieves most relevant information

3. Specialized Agent Tools:

- Summary Tool: Provides comprehensive overviews and general summaries of cases and incidents
- Needle Tool: Delivers precise, exact information with high fidelity and source attribution
- Table QA Tool: Handles structured data queries, calculations, and tabular analysis
- Statistics Tool: Performs advanced statistical analysis, correlation analysis, and cross-table comparisons
- Intelligent Query Routing
 - LLM-based Classification: Automatically determines the most appropriate tool for each query
 - Context Understanding: Analyzes query intent and content type for optimal routing
 - Fallback Mechanisms: Keyword-based routing when LLM routing encounters issues
 - Tool Specialization: Each tool optimized for specific types of insurance queries

4. Advanced Analytics

- Correlation Analysis: NumPy-powered correlation matrix calculation for numerical data relationships
- Pattern Recognition: Identifies trends and patterns across insurance data
- Cross-table Analysis: Combines data from multiple tables for comprehensive insights Statistical
- Comparisons: Comparative analysis across policies, cases, and payment data

5. Comprehensive Evaluation

- RAGAS Integration: Industry-standard evaluation using RAGAS metrics:
 - Answer Correctness: Measures accuracy of generated responses
 - Context Quality: Evaluates retrieval precision and recall
 - Faithfulness: Ensures responses are grounded in source documents
- Domain-specific Metrics: Specialized evaluation for insurance use cases

Evaluation Results

1. Table - QA Accuracy: 100% - Processed by run_table_qa_eval.py script
2. RAGAS Evaluation Results

86.28%

Answer Correctness

Measures factual accuracy of generated summaries compared to source document.

80.00%

Faithfulness

Assesses how well Q&A responses maintain fidelity to original document content.

83.39%

Context Precision

Evaluates relevance of information retrieved by Q&A tool.

90.00%

Context Recall

Measures comprehensiveness of information retrieved by Q&A tool.

Source Data for the Insurance RAG Agent

Data Folder (data/)

The data directory contains a comprehensive collection of 9 insurance-related PDF documents with text and table content that serve as the primary data source for the Insurance RAG Agent system and has the next directory structure:

1. Accident Reports/Cases (3 documents)

- Alex_Jones_Accident_Report.pdf - Detailed accident case report from Alex Jones' perspective
- Maria_Petrov_Accident_Case.pdf - Maria Petrov's accident case documentation
- Petr_Petrov_Accident_Case.pdf - Petr Petrov's accident case documentation

Content Structure:

- Timeline-based narratives with detailed incident descriptions
- Weather conditions and environmental factors
- Vehicle information (make, model, VIN, license plates)
- Driver details (age, license information, employment history)
- Emergency response and first responder actions
- Medical assessments and injury documentation
- Police investigation details and official reports

2. Insurance Policies (3 documents)

- Alex_Jones_Insurance_Policy.pdf - Alex Jones' commercial vehicle insurance policy
- Petr_Petrov_Insurance Policy_Toyota.pdf - Petr Petrov's Toyota vehicle policy
- Petr_Petrov_Insurance_Policy_Honda.pdf - Petr Petrov's Honda vehicle policy

Content Structure:

- Coverage details with per-person and per-accident limits
- Deductible amounts for different coverage types
- Premium information broken down by coverage type
- Policy terms and conditions
- Liability coverage (bodily injury, property damage)
- Physical damage coverage (collision, comprehensive)
- Special endorsements and additional coverage options

3. Payment Reports (3 documents)

- Alex_Jones_Payment_Report.pdf - Payment and settlement details for Alex Jones case
- Maria_Petrov_Payment_Report.pdf - Payment details for Maria Petrov case
- Petr_Petrov_Payment_Report.pdf - Payment details for Petr Petrov case

Content Structure:

- Settlement amounts and payment breakdowns
- Medical expenses (past and future)
- Lost wages calculations
- Pain and suffering compensation
- Home modifications and accessibility costs
- Payment dates and settlement timelines
- Insurance company payment details

Output Folder Content Description

The output folder contains the results and artifacts generated by the Insurance RAG Agent system during document processing, analysis, and evaluation phases. Here's a detailed breakdown of its contents Main Output Files and Evaluation Results (eval_results/) Folders.

1. Main Output Files

Document Processing Results

- `print_chunks_and_table._terminal_output/print_chunks_and_table._outputs.txt` - Processing logs by `print_chunks_and_tables.py`
- `extracted_chunks_20250907_200015.txt` - Comprehensive text extraction results, created by running utility script `print_chunks_and_tables.py`
- `extracted_tables_20250907_200015.txt` - Table extraction results, created by running utility script `print_chunks_and_tables.py`

The `print_chunks_and_tables.py` file is a comprehensive utility script designed to process PDF documents and generate detailed reports of extracted text chunks and tables.

2. Agent's Tools Execution Results (tools_output/)

The tools_output folder contains detailed execution logs and outputs from testing the four main agent tools in the Insurance RAG Agent system. Here's a comprehensive description of its contents:

- needle_tool_output.txt - Needle tool execution logs
- statistics_tool_output.txt - Statistics tool execution logs
- summary_tool_output.txt - Summary tool execution logs
- table_qa_agent_tool_output.txt - Table QA tool execution logs

3. Evaluation Results (eval_results/)

1. Table QA Evaluation

- run_table_qa_eval.txt - Table-specific query evaluation results

2. RAGAS Evaluation Screenshots

- Screenshot 2025-09-08 114925_RAGASoutput.png - RAGAS evaluation interface
- Screenshot 2025-09-08 121349_tableQA_eval_output.png - Table QA evaluation results

Next Steps



- Expand data source
- Performance improvements.
- Develop adaptive chunking algorithm.
- Expand statistical analysis
- Expand RAGAS metric tracking.
- Improving results of RAGAS evaluation.
- MCP Integration
- Graph DB Integration