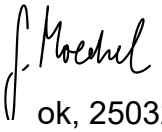

ok, 250325

Master Thesis Expose

Development of an AI-Powered Question Answering System for PDF and Web Content using Retrieval-Augmented Generation

Ketan Kishor Darekar

Matriculation number: 11037367

Official Start Date: 01/04/2025

SRH University Heidelberg

School of Information, Media and Design

Internal Supervisor 1:

Prof. Dr. Gerd Moeckel

Internal Supervisor 2:

Mr. Paul Tanzer

Table of Contents

1 Introduction

Retrieval-Augmented Generation (RAG) has emerged as a transformative approach in AI-driven knowledge retrieval, enabling language models to provide more accurate responses by incorporating external knowledge sources. Traditional RAG implementations primarily rely on static document retrieval, such as PDFs, which limits their ability to access dynamic and real-time knowledge. This gap hinders the effectiveness of AI systems in scenarios where up-to-date and diverse sources of information are required.

This research aims to develop a Hybrid RAG system that integrates both PDF-based retrieval and web-based HTML data sources to improve information access and response accuracy. By leveraging PostgreSQL with the pgvector extension as a vector database, the study will also explore efficient strategies for metadata management, chunking techniques, and retrieval optimization.

2 Problem Statement

Current RAG implementations face several challenges:

1. **Limited knowledge access:** Most RAG systems depend on pre-existing static documents (PDFs, Word files), restricting the retrieval of real-time information from web-based sources.

2. **Inefficient metadata handling:** Existing solutions often lack structured metadata tagging, which can significantly impact search accuracy and ranking.

3. **Scalability concerns:** With the increasing volume of data, an optimized and scalable vector database (PostgreSQL with pgvector) is essential to efficiently handle storage and retrieval operations.

4. **Lack of conversation memory:** RAG systems struggle to maintain multi-turn conversational context, which is crucial for applications requiring long-form interactions.

To address these challenges, this research will develop a Hybrid RAG system that combines PDF and HTML-based knowledge sources, enhances metadata management, optimizes chunking strategies, and implements conversation memory for improved AI-driven interaction.

3 Research Objectives & Research Questions

3.1 Research Objectives

- Develop a hybrid RAG system that retrieves information from both PDFs and HTML-based sources.
- Implement PostgreSQL with the pgvector extension as a scalable vector database for managing large data volumes.
- Enhance retrieved text with metadata, including timestamps, categories, and relevance scores.
- Optimize chunking and vectorization strategies to improve retrieval efficiency and semantic search accuracy.
- Integrate conversational memory, allowing the model to maintain context across multi-turn interactions.

3.2 Research Questions

- **RQ1:** How can large-scale text data (PDFs & HTML) be efficiently processed and stored for Retrieval-Augmented Generation?
- **RQ2:** What metadata strategies can improve information retrieval relevance in hybrid RAG systems?
- **RQ3:** How can conversation history be incorporated to enhance user interactions?
- **RQ4:** What are ideal chunking strategies if the content structure is known before?

4 Technologies & Tools

This research will leverage modern AI and database technologies to implement an efficient Hybrid RAG system.

4.1 Retrieval & Processing

- **LangChain** – Text chunking, extraction, and retrieval pipeline (LangChain, 2025)
- **PyMuPDF & PDF.js** – Extracting text from PDF files. (PyMuPDF, 2025)
- **BeautifulSoup & Scrapy** – Web scraping for HTML-based document retrieval
- **Newspaper3k** – Processing online articles dynamically. (Newspaper3k., 2025)

4.2 Vector Database & Indexing

- **PostgreSQL (pgvector)** – Storing and retrieving embeddings for large-scale search.
- **Full-text search (GIN & GIST indexing)** – Optimizing keyword-based search within PostgreSQL.

4.3 Language Model & Backend Development

- **OpenAI's GPT-4 API** – Generating context-aware answers. (OpenAI, 2023)
- **Python (FastAPI or Flask)** – Backend implementation for API handling and retrieval processing. (Pallets., 2025)

5 Methodology

This study will follow a structured experimental approach to test and validate the Hybrid RAG system.

5.1 Data Collection & Preprocessing

- **PDF Processing:** Extracting text, segmenting into chunks, and vectorizing.
- **HTML Processing:** Extracting content, removing noise, and structuring data.
- **Metadata Tagging:** Enhancing documents with timestamps, categories, and keywords.

5.2 Vectorization & Storage

- Embedding documents using OpenAI & Sentence Transformers.
- Storing embeddings in PostgreSQL's pgvector extension.
- Indexing for fast retrieval using HNSW & full-text search.

5.3 Hybrid Retrieval Process

- **Query processing:** Matching user input with stored document vectors.
- **Keyword & semantic search combination:** Using PostgreSQL's full-text search + vector search.
- **Ranking & context-awareness:** Prioritizing results based on relevance, freshness, and document type.

5.4 Conversational Memory Integration

- Storing previous interactions for multi-turn conversations.
- Retrieving relevant past queries to maintain context.
- Generating responses that consider conversation history.

6 Expected Outcomes & Evaluation metrics

6.1 Expected Outcomes

- Improved RAG accuracy by combining static (PDFs) and dynamic (HTML) sources.
- Faster retrieval times with optimized indexing and metadata tagging.
- Context-aware conversations with memory-enhanced retrieval.
- Scalable & cost-effective alternative to proprietary File Search solutions (e.g., OpenAI's expensive API).

6.2 Evaluation Metrics

- **Response Accuracy:** BLEU & ROUGE scores to compare system responses with human-validated answers.
- **Retrieval Efficiency:** Query response time, measured in milliseconds.
- **Metadata Impact:** Measuring improvements in retrieval relevance with and without metadata.
- **Chunking Strategy Performance:** Comparing retrieval speed & accuracy with different chunking methods.

7 References

LangChain, 2025. *Build a Retrieval Augmented Generation (RAG) App: Part 1,* s.l.: https://python.langchain.com/docs/tutorials/rag/.

Newspaper3k., 2025. *Newspaper3k Documentation,* s.l.: https://newspaper.readthedocs.io/en/latest/.

OpenAI, 2023. *GPT-4,* s.l.: https://openai.com/index/gpt-4/.

Pallets., 2025. *Flask Documentation (Stable Release),* s.l.: https://flask.palletsprojects.com/en/stable/.

PyMuPDF, 2025. *PyMuPDF Tutorial,* s.l.: https://pymupdf.readthedocs.io/en/latest/tutorial.html.