

RAG-Based Knowledge Transformation Framework

A framework for transforming static documents into dynamic, user-friendly training guides and knowledge bases

By Patrick Diamitani

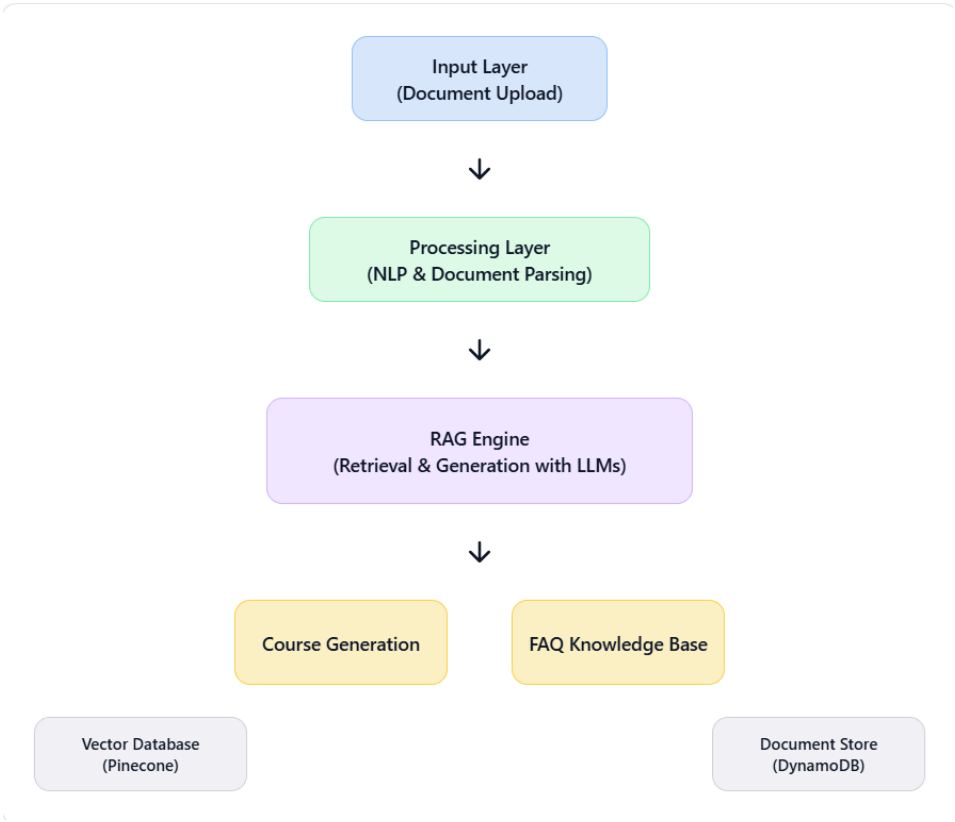
4.4.25

Abstract

This white paper introduces a Retrieval-Augmented Generation (RAG) framework designed to transform static documents into dynamic, user-friendly training guides and knowledge bases. By integrating advanced AI models with scalable cloud infrastructure, this solution addresses the growing demand for efficient, accessible, and tailored educational content.

Leveraging Amazon Web Services (AWS) and a modular architecture, the framework automates content processing, enhances scalability across diverse domains, and empowers users with actionable insights.

Architecture



Input Layer

- **Functionality:** Users upload documents (e.g., PDFs) through a web interface.
- **Technology:** AWS Amplify hosts a responsive frontend, with S3 buckets storing uploaded files.

Processing Layer

- **Functionality:** Documents are parsed, segmented, and indexed for retrieval.
- **Technology:** Python scripts (PyPDF2 for PDF extraction, spaCy for NLP) process content, storing structured data in DynamoDB and vector embeddings in Pinecone for similarity search.

RAG Engine

- **Functionality:** User queries trigger retrieval of relevant content, which is synthesized into training modules or FAQs.
- **Technology:** DeepSeek's language model provides the base for generation, with Lambda functions orchestrating retrieval and generation processes.

Output Layer

- **Functionality:** Generated content is delivered as HTML lessons or query responses.
- **Technology:** A React-based frontend serves dynamic content, hosted on AWS services.

Benefits

- **Efficiency:** Reduces manual content creation time from hours to minutes.
- **Scalability:** Adapts to diverse domains using cloud infrastructure.
- **Accessibility:** Delivers tailored, easy-to-understand insights for non-expert users.

Extensibility

- Policy Education: Transforming regulatory documents into public-facing guides.
- Health Data Analysis: Converting medical records into patient education tools.
- Technical Documentation: Creating interactive tutorials from technical manuals.

Implementation Example

Input:

- The author's resume (listing AWS skills: EC2, Lambda) and a Healthie Max Pro document (describing lab results).

Processing:

- The system extracts key entities (e.g., "EC2," "blood work") and stores them as indexed modules.

Output:

- Resume → "Lesson 1: AWS Basics - Learn EC2 setup with hands-on experience."
- Healthie Max Pro → "Lesson 2: Interpreting Lab Results - Understand blood work simply."

Technical Implementation: Document Processing Algorithm

Our document transformation process uses DeepSeek AI to analyze and structure content effectively:

Document Processing Flow

1. User uploads document (PDF, DOC, TXT)
2. Content extraction based on document type
3. DeepSeek API analyzes the document content:
 - Identifies key topics and concepts
 - Creates logical structure for curriculum
 - Generates learning objectives
 - Creates assessment questions
4. Results are formatted into:
 - Course with modules, objectives, and quizzes
 - FAQ with categorized questions and answers
5. Content is presented to user via interactive UI

Conclusion

The RAG-Based Knowledge Transformation Framework offers a scalable, AI-driven solution to bridge the gap between raw documents and actionable knowledge. By automating content synthesis, it saves time, enhances accessibility, and supports diverse applications—positioning it as a valuable tool for education, healthcare, and policy innovation.