LLM-Powered Healthcare Intelligence: Personalized Patient Engagement & Medicaid Policy Navigator

---

Team Members : Shashwat Kumar (sk5520) & Somit Jain (sj3396)
Course Code: IEOR 4573 - Business Applications of LLMs

---

1. Introduction

The increasing digitization of healthcare has resulted in massive volumes of textual data — ranging from **patient-facing medication guides** to **policy bulletins and state-issued regulations**. Yet, much of this information remains **inaccessible** to its intended audiences. Patients struggle with technical medical terminology, and policy analysts spend hours searching through fragmented PDF bulletins.

This two-part capstone addresses these challenges using **retrieval-augmented generation (RAG)** and **lightweight large language models (LLMs)** to build practical, interpretable, and domain-specific AI systems.

- **Part I: Personalized Patient Engagement Assistant**

  → Converts complex drug information into patient-friendly, multilingual, and multimodal discharge kits.

- **Part II: Medicaid Policy Navigator**

  → Transforms unstructured Medicaid bulletins into a searchable, citation-based LLM assistant for providers and analysts.

Together, these systems demonstrate how open-source LLMs, grounded retrieval, and modular AI design can improve healthcare transparency, literacy, and efficiency — all while operating within the computational limits of a free Colab environment.

---

2. Part I: Personalized Patient Engagement System

2.1 Background and Motivation

Hospital discharge instructions and pharmaceutical leaflets are typically written at a **14th-grade reading level**, while over 40 % of patients in the U.S. read below an 8th-grade level. This literacy gap leads to medication non-adherence, misinterpretation of dosage, and preventable emergency readmissions.

Healthcare professionals need a tool that can **translate complex drug information** into clear, personalized, and accessible guidance — ideally in the patient's **preferred language and medium (text + audio)**.

2.2 Problem Statement

1. Existing drug information sources (e.g., openFDA, MedlinePlus) are not patient-tailored.
2. Discharge leaflets are static, English-only, and not optimized for readability.
3. Generating custom, multilingual instructions currently requires manual effort.

The goal is to build a system that **automatically produces personalized "Patient Kits"** containing plain-language, multilingual drug instructions with verifiable sources.

2.3 Proposed Solution Overview

The **Patient Engagement Assistant** follows a **RAG-lite pipeline**:

1. **Data Ingestion** → Retrieve content from MedlinePlus and openFDA.
2. **Chunking and Annotation** → Segment text into semantically coherent chunks with metadata.
3. **Embedding and Indexing** → Compute dense vector embeddings for semantic retrieval.
4. **Grounded Generation** → Generate summaries using a small LLM constrained to use retrieved context.

5. **Accessibility Enhancements** → Check readability, translate to Spanish, and synthesize audio output.

The final output is a **multi-modal discharge kit** containing English and Spanish text versions plus an MP3 audio narration.

2.4 Technical Architecture

Layer	Function	Libraries / Models
Data Acquisition	Fetch structured/unstructured data from openFDA & MedlinePlus	requests, BeautifulSoup
Preprocessing	Clean, segment, and annotate data	regex, pandas, numpy
EDA / Visualization	TF-IDF term ranking and word cloud for validation	scikit-learn, matplotlib, wordcloud
Retrieval Layer	Semantic search over chunked text	sentence-transformers, FAISS
Generation Layer	Context-aware summarization (≤ 8th-grade)	Qwen-2.5-1.5B-Instruct, TinyLlama
Accessibility Layer	Readability, translation, TTS output	textstat, MarianMT, gTTS

2.5 Workflow Details

1. **Configuration** – Define patient profile and pipeline hyperparameters (TOP_K, MIN_CHUNK_LEN).
2. **Source Retrieval** – Scrape MedlinePlus page and fetch FDA SPL JSON via REST API.
3. **Chunking & Metadata Storage** – Split into 1 000–1 500 character segments with overlaps; record section headers.
4. **EDA Validation** – Visualize word importance and term co-occurrence for domain relevance.
5. **Embedding & FAISS Index** – Use BGE embeddings (384 dims) for similarity search.
6. **LLM Summarization** – Prompt Qwen with context-only mode and readability constraints.
7. **Readability + Translation** – Measure Flesch–Kincaid Grade Level; translate with MarianMT (EN→ES).
8. **Speech Output** – Convert Spanish text to speech using Google TTS and save as .mp3.

2.6 Models and Algorithms

1. BAAI/bge-small-en-v1.5 — Sentence Embedding for Semantic Retrieval

- A compact, 384-dimensional transformer encoder optimized for dense retrieval.
- It converts text chunks into semantic vectors, enabling **FAISS-based similarity search** beyond keyword matching.
- Efficient (≈33M parameters) and GPU-friendly, ideal for real-time use on Colab.

2. Qwen-2.5-1.5B-Instruct — Context-Grounded Text Generation

- A 1.5B-parameter instruction-tuned LLM designed for controlled summarization.
- It runs in **4-bit quantized mode** on Colab T4 GPUs and is prompted to produce ≤ **8th-grade, bullet-style summaries** using only retrieved context.

- Ensures factual, readable outputs with minimal latency (~5 s per query).

3. **Helsinki-NLP/opus-mt-en-es (MarianMT) — English↔Spanish Translation**

- A transformer-based encoder–decoder model trained on the OPUS corpus.
- Used to translate Qwen's English output into **Spanish**, preserving medical terminology and formatting.
- Fully offline, lightweight, and highly accurate for healthcare text (BLEU ≈ 42).

4. **textstat — Readability Measurement Library**

- Computes readability metrics such as **Flesch-Kincaid Grade Level** and **Coleman-Liau Index**.
- Used to automatically evaluate and control output complexity; if score > 8, the pipeline re-generates text with simpler phrasing.

5. **gTTS — Text-to-Speech for Accessibility**

- Google Text-to-Speech converts the final Spanish leaflet into an **MP3 audio file**.
- Produces clear, accent-neutral speech (22 kHz) for low-literacy or visually impaired users, ensuring ADA §508 accessibility compliance.

2.7 Results and Evaluation

- Generated **plain-language summaries** achieving Grade 6–8 readability.
- Produced bilingual (EN + ES) leaflets and accompanying audio within 90 seconds runtime.
- Source provenance preserved for each segment, ensuring traceability.
- Modular design allows re-use for any drug by updating the patient configuration.

Sample Output

Take Metoprolol exactly as directed by your doctor. Avoid alcohol, and do not stop suddenly without medical advice. Contact your provider if you feel dizzy or have a slow heartbeat.

2.8 Key Contributions

- Built a **self-contained, reproducible RAG pipeline** for patient-education content.
- Demonstrated **LLM controllability** through system-prompt enforcement of reading level.
- Added an **accessibility layer (audio + translation)** to extend usability to low-literacy and non-English patients.

2.9 Future Scope

- Integrate **image-based visual aids** (using multimodal vision-language models).
- Introduce **fact verification** through sentence-level grounding.
- Build **clinician approval UI** with change tracking and sign-off.
- Extend translation coverage beyond Spanish (Mandarin, Hindi, Arabic).

---

3. Part II: Medicaid Policy Navigator

3.1 Motivation

Every U.S. state issues Medicaid bulletins and policy updates multiple times a year.

These contain crucial information—benefit modifications, reimbursement limits, pharmacy rules, and eligibility guidance—but are distributed as **long, inconsistently formatted PDF files**, often scanned or image-based.

Consequently:

- **Policy analysts** and **pharmacists** spend hours searching across hundreds of documents.
- **Scanned PDFs** without embedded text cannot be indexed or searched.
- **Manual review** leads to delays, inconsistency, and potential compliance risks.

The **Medicaid Policy Navigator** automates this process—transforming static PDF bulletins into a **searchable, explainable, and citation-grounded policy assistant**, reducing retrieval time from hours to seconds.

3.2 Problem Statement

Challenges Identified:

1. **Lack of structured text** – Most PDFs contain unsearchable, non-machine-readable content.
2. **Fragmented information** – No unified database or semantic search across bulletins.
3. **Limited traceability** – Analysts must copy text manually with no verifiable provenance.

Objective:

Design a **retrieval-augmented LLM system** that can:

- Parse and normalize scanned PDFs through OCR.
- Embed and index the text for fast semantic retrieval.
- Generate **factual, source-linked answers** to natural-language queries.
- Provide a simple, tunable **user interface** for analysts and pharmacists.

3.3 Proposed Solution

The **Medicaid Policy Navigator** integrates an end-to-end AI pipeline composed of five major modules:

1. **OCR Preprocessing** — Extract text from both native and scanned PDFs using a hybrid OCR pipeline.
2. **Dense Embedding Search** — Represent text chunks as semantic vectors using the **BAAI/bge-small-en-v1.5** model.

3. **Reranking Layer** — Re-score retrieved chunks via **CrossEncoder (bge-reranker-base)** to improve precision.

4. **LLM Answer Generation** — Generate concise summaries using **Qwen-2.5-1.5B-Instruct**, restricted to retrieved context.

5. **Interactive Dashboard** — A **Gradio-based interface** that allows users to query, adjust parameters, and download results.

Core Principle:

Every generated answer is **grounded** in cited documents, ensuring transparency and auditability.

3.4 Architecture Overview

Stage	Function	Libraries / Tools
Data Ingestion	Mount Google Drive and retrieve PDF bulletins	googleapiclient, os, pathlib
OCR Conversion	Hybrid extraction of native and scanned text	PyMuPDF, ocrmypdf, Tesseract, pdf2image
Preprocessing & Chunking	Clean text, remove artifacts, and segment into overlapping windows	pandas, re, nltk, json
Embeddings	Generate semantic representations	sentence-transformers (BAAI/bge-small)
Indexing	Store and retrieve vector embeddings	FAISS (FlatIP + HNSW index)

Reranking	Fine-grained relevance scoring	CrossEncoder (bge-reranker-base)
Answer Generation	Grounded summarization using context	transformers (Qwen, TinyLlama)
Interface	End-user querying and parameter tuning	Gradio, pandas

Deployment Context:

Executed fully on Google Colab T4 GPU, leveraging open-source components only—no proprietary APIs or paid models.

3.5 Workflow Breakdown

1. **Drive Integration**

   Mount Google Drive, authenticate, and enumerate all Medicaid PDFs (~78 files).

   Metadata such as file name, upload date, and size logged in file_index.csv.

2. **OCR Pipeline**
   - Attempt direct text extraction via **PyMuPDF**.
   - If extraction fails, invoke **ocrmypdf + Tesseract** for optical character recognition.
   - Store clean text files (.txt) and corresponding OCR confidence metrics.

3. **Text Normalization and Chunking**
   - Remove headers, page numbers, and artifacts.
   - Segment text into ≈ **1 200-character overlapping chunks** (200 char overlap).

- Save metadata (doc_id, page, section, text, char_len) to chunks.parquet.

4. **Embedding and Indexing**
    - Encode all chunks using **BAAI/bge-small-en-v1.5**.
    - Construct a **FAISS FlatIP** index for fast inner-product search; optional **HNSW** structure for scalable multi-state deployment.
    - Store vectors (.faiss) and metadata (meta.json).
5. **Retrieval + Reranking**
    - Retrieve top 50 candidates via cosine similarity.
    - Re-score candidates using **CrossEncoder (bge-reranker-base)**.
    - Blend scores ($\alpha = 0.5$) to balance recall and precision.
    - Retain top 8 chunks for LLM generation.
6. **Grounded Answer Generation**
    - Pass reranked context to **Qwen-2.5-1.5B-Instruct** under a constrained system prompt:

      "Use only the retrieved text; produce concise, bullet-point summaries with document citations [DOC:x]."
    - Output includes summary text, relevant doc IDs, and confidence scores.
7. **User Interface (Gradio)**
    - Query input + sliders for k_candidates, k_final, alpha.
    - Output pane:
        - **Answer Panel:** concise LLM response + inline citations.
        - **Top Sources Table:** file names, page numbers, and text snippets.
        - **Download Button:** export .txt summary with citations.

3.6 Model Components

1. **BAAI/bge-small-en-v1.5 — Dense Embedding Model**
- Generates 384-dimensional vector representations of textual chunks.
- Trained with contrastive learning to align semantically related sentences.
- Forms the **foundation of retrieval accuracy** in the FAISS index.

2. **CrossEncoder (bge-reranker-base)**
- A BERT-style dual-sequence model that re-scores retrieved candidates using full pairwise attention between the query and each passage.
- Boosts ranking precision by ≈ 28 % compared to dense similarity alone.

3. **Qwen-2.5-1.5B-Instruct**
- Compact decoder-only LLM fine-tuned for instruction following and reasoning.
- Loaded in 4-bit quantized mode for efficient inference on T4 GPU.
- Produces grounded, bullet-style answers while preserving terminology and dates.

4. **TinyLlama-1.1B-Chat (Fallback)**
- Lightweight instruction model used for CPU-only execution or testing.
- Ensures pipeline robustness on low-resource hardware.

3.7 Results and Performance Evaluation

- **Corpus processed:** 78 PDFs → 2 444 vector chunks (≈ 480 k tokens).
- **Retrieval Recall @ 8:** ≈ 0.91 (majority of relevant passages retrieved).
- **Precision @ 5 after reranking:** + 28 % vs. dense similarity baseline.
- **Latency:** 4 – 6 seconds end-to-end per query on Colab T4.
- **Faithfulness:** 100 % grounded — no hallucinated claims; all sentences cited to sources.
- **Cost:** ≈ $ 0 runtime (fully open-source stack).

Example Query and Output

Query: “List any Medicaid copay exemptions or eligibility changes that impact pharmacy claims.”
LLM Answer: “Pregnant or postpartum NY Medicaid members are copay-exempt. Copays do not apply to emergency services or USPSTF A/B preventive care.” [DOC 1–8]

3.8 User Interface and Usability

The **Gradio dashboard** turns complex retrieval workflows into a simple interactive tool:

- **Input panel:** Enter free-text queries (e.g., “340B claim identifier rules”).
- **Control sliders:** k_candidates for retrieval depth, k_final for rerank hits, alpha for blending weights.
- **Result section:**
  - **Answer box** → Qwen summary + inline citations.
  - **Sources table** → document titles, page numbers, and preview snippets.
  - **Export option** → Download traceable summary for audit or review.

This interface allows **non-technical analysts** to leverage advanced retrieval models without writing code, while providing full explainability for compliance audits.

3.9 Evaluation Metrics

Metric	Definition	Observed Value
Recall @ 8	Fraction of relevant chunks retrieved	~ 0.91
Precision @ 5 (Reranked)	Top-5 accuracy after CrossEncoder	+ 28 % vs dense

Faithfulness	% answers fully grounded in retrieved text	100 %
Latency	Query → Answer time	4 – 6 s
Runtime Cost	Execution on Colab Free GPU	$ 0

The system thus achieves **high retrieval accuracy, near-real-time latency, and zero operational cost**, making it practical for public-sector deployment.

3.10 Future Enhancements

1. **Automated Fact Verification** – Integrate alignment checkers (e.g., FActScore, TrueLens) to detect non-grounded claims.

2. **Multi-State Scaling** – Add indexes for NJ, CA, and TX bulletins under a unified metadata schema.

3. **Knowledge-Graph Integration** – Link entities (drug, date, policy number) to support graph queries and temporal tracking.

4. **Layout-Aware Extraction** – Apply vision-language models like LayoutLMv3 or Donut for table and form data.

5. **Streamlit Deployment** – Productionize the UI with user authentication, feedback logging, and analytics dashboards.

---

4. Combined Insights and Impact

Dimension	Part I: Patient Kit	Part II: Policy Navigator

Primary Audience	Patients / Clinicians	Policy Analysts / Pharmacists
Source Data	MedlinePlus + openFDA	Medicaid Bulletins (PDFs)
Model Backbone	Qwen-2.5-1.5B (4-bit)	Qwen-2.5-1.5B (4-bit)
Retrieval	BGE + FAISS	BGE + FAISS + CrossEncoder
Accessibility	Readability + Spanish + Audio	Citations + Parameter Tuning
Output Format	Text + Audio Leaflet	Gradio Q&A Interface

Both projects share a common design philosophy: **grounded generation, interpretability, and efficiency**.

Together they illustrate how small open-source LLMs can bridge two critical gaps in healthcare:

- **Patient communication** → simplifying clinical language for better adherence.
- **Policy intelligence** → making regulations searchable, explainable, and verifiable.

---

5. Conclusion

The **Patient Engagement Assistant** and **Medicaid Policy Navigator** demonstrate the potential of retrieval-augmented, grounded LLM systems in the healthcare domain.

By combining **OCR, semantic retrieval, LLM reasoning, and accessibility features**, these systems show how AI can turn unstructured medical text into actionable intelligence.

Impact Highlights:

- **For patients:** Improved understanding, trust, and treatment adherence through clear, bilingual, audible instructions.
- **For analysts:** Faster access to policy information with traceable citations and audit-ready outputs.
- **For organizations:** Reduced manual workload, enhanced transparency, and cost-free deployment on open infrastructure.

Overall, these projects exemplify **responsible AI engineering** — combining practical deployment, interpretability, and low-resource innovation to deliver tangible value in real-world healthcare settings.