XRayAnalyzer: Agente Radiologo Intelligente

Integrazione di Visione Artificiale e RAG Semantico (FAISS)

Autori del Progetto

30 novembre 2025

Indice

Capitolo 1

Introduzione al Progetto

1.1 Cos'è XRayAnalyzer?

XRayAnalyzer è un sistema diagnostico avanzato che supera i limiti dei classificatori tradizionali. Integra un modello di visione profonda (Faster R-CNN) con un motore di ricerca semantico (RAG) per fornire non solo una diagnosi, ma un consulto completo basato su letteratura medica ufficiale.

Spiegazione Semplice: Il Concetto

Immagina un radiologo che ha letto migliaia di libri. Il sistema prima individua visivamente la polmonite. Poi, invece di dare una risposta standard, "sfoglia" mentalmente (tramite Ricerca Semantica) i manuali medici per trovare esattamente il trattamento per quei sintomi specifici, citando la fonte.

Capitolo 2

Modulo 1: Il Cervello Visivo (Deep Learning)

Questa parte del progetto si occupa di "insegnare al computer a vedere" le anomalie radiologiche.

2.1 Il Modello: Faster R-CNN

Per l'analisi delle immagini è stata scelta l'architettura **Faster R-CNN**, addestrata specificamente sul dataset RSNA Pneumonia utilizzando l'infrastruttura HPC (High Performance Computing) dell'università. Il modello localizza le opacità polmonari disegnando Bounding Box precisi con un punteggio di confidenza.

Capitolo 3

Modulo 2: RAG Semantico Avanzato

Il cuore innovativo del sistema risiede nel modulo `RagTool`, che implementa una ricerca vettoriale avanzata.

3.1 Architettura del RAG (Retrieval-Augmented Generation)

A differenza dei sistemi basati su parole chiave ("Keyword Search"), il nostro sistema utilizza **FAISS (Facebook AI Similarity Search)** per comprendere il *significato* del testo.

Il processo è gestito dalla classe `RagTool` e avviene in 4 fasi:

1. **Ingestion:** Caricamento di documenti PDF e TXT dalla cartella `knowledge_base`.
2. **Chunking:** Suddivisione del testo in frammenti più piccoli (`RecursiveCharacterTextSplitter`) per non perdere il contesto.
3. **Embedding:** Trasformazione del testo in vettori numerici ad alta dimensionalità usando il modello `sentence-transformers/all-MiniLM-L6-v2`.
4. **Indicizzazione:** Creazione di un indice FAISS per la ricerca ultra-rapida.

```
# Caricamento Modello di Embedding (HuggingFace)
self.embeddings = HuggingFaceEmbeddings(model_name="sentence-transformers/all-MiniLM-L6-v2")

# Creazione Database Vettoriale FAISS
vector_db = FAISS.from_documents(chunks, self.embeddings)
```

Listing 3.1: Costruzione dell'Indice Vettoriale (RagTool)

Spiegazione Semplice: Cos'è un Embedding?

I computer non capiscono le parole, ma capiscono i numeri. L'*Embedding* trasforma la frase "Il paziente ha la tosse" in una lunga lista di numeri (es. [0.1, 0.5, -0.9...]) che rappresenta il *significato*. Grazie a questo, il sistema capisce che "Tosse" e "Sintomi respiratori" sono concetti vicini, anche se le parole sono diverse.

3.2 Logica dell'Agente Medico

L'Agente (`agent.py`) utilizza il `RagTool` per rispondere a domande cliniche specifiche. Invece di cercare parole esatte, l'agente pone domande in linguaggio naturale:

- "Quali sono i sintomi tipici della polmonite?"
- "Qual è il protocollo terapeutico antibiotico consigliato?"

Il sistema cerca nel database vettoriale i frammenti di testo che rispondono meglio a queste domande "semantiche".

```
# L'agente interroga il RAG su argomenti specifici
info_trattamento = self.rag.search("Qual    il protocollo terapeutico antibiotico
    consigliato?")

# Il RAG restituisce il passaggio pi  pertinente trovato nei manuali PDF/TXT
```

Listing 3.2: Interrogazione Semantica dell'Agente

Capitolo 4

Integrazione e Docker

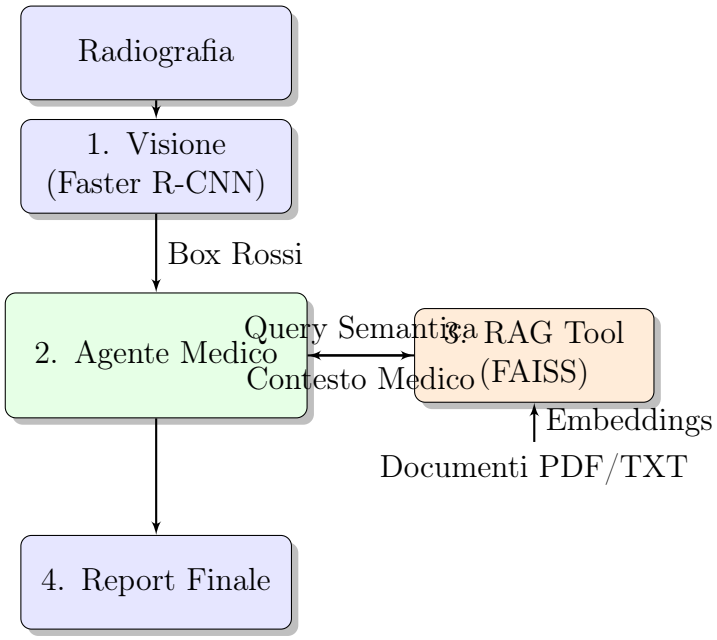
L'intero sistema (Modello HPC + RAG + Interfaccia Web) è containerizzato con Docker.

4.1 Dipendenze Critiche

Il file `requirements.txt` include librerie specializzate per il supporto AI:

- `faiss-cpu`: Per la gestione efficiente del database vettoriale.
- `langchain-huggingface`: Per gestire i modelli di embedding open-source.
- `pypdf`: Per l'estrazione del testo dai documenti ufficiali.

4.2 Flusso Completo (Workflow)



Capitolo 5

Conclusioni

L'implementazione di un sistema RAG basato su vettori (FAISS) rappresenta un salto di qualità rispetto ai sistemi basati su regole. XRayAnalyzer non si limita a "incollare" testo, ma "comprende" la richiesta diagnostica e recupera le informazioni più pertinenti dalla letteratura medica, offrendo un supporto decisionale di alto livello.