Project proposal - Ask My Docs - RAG Chatbot

Name: Vidushi Raval
Date: February 2, 2026
Project Title: Ask My Docs – Retrieval-Augmented Generation (RAG) Chatbot with Citations
Category: Generative AI, NLP, Retrieval-Augmented Generation (RAG), Information Retrieval, AI Evaluation

Problem Statement

Large volumes of technical documents such as research papers, documentation, and internal knowledge bases are difficult to navigate efficiently. Traditional keyword search often fails to provide precise, contextual answers and does not synthesize information across multiple sources.

Large Language Models (LLMs) can generate fluent answers but are prone to **hallucinations** and lack traceability, making them unreliable for research, enterprise, or decision-critical use cases.

This project aims to build **AskMyDocs**, a Retrieval Augmented Generation (RAG) chatbot that allows users to ask natural-language questions over a corpus of machine learning and NLP research papers and receive **grounded answers with explicit citations** to the source documents.

The goal is to demonstrate how retrieval + generation can improve accuracy, trust, and transparency in AI systems.

Context

RAG systems are widely adopted in industry to power applications such as internal knowledge assistants, customer support bots, and research tools.

Companies like Google, Meta, Microsoft, and OpenAI use RAG pipelines to combine LLMs with external knowledge sources.

This project mirrors a real-world **LLM application development workflow**, focusing on:

- Document preprocessing and chunking
- Vector embeddings and semantic search
- Retrieval quality
- Citation-aware answer generation

It represents the work of a **Data Scientist / ML Engineer / AI Engineer** building reliable, production-oriented LLM systems.

Criteria for Success

- Build a clean, structured document corpus from raw research papers
- Implement a complete RAG pipeline:
  - Document loading and preprocessing
  - Chunking with metadata
  - Embedding generation
  - Vector storage and retrieval
  - Answer generation grounded in retrieved context
- Ensure **every response includes citations** referencing the original document source
- Demonstrate reduced hallucination through retrieval-based grounding
- Provide qualitative evaluation of answer relevance and citation accuracy
- Deliver a clear, well-documented project suitable for portfolio and interviews

Scope of Solution Space

In-Scope

- Data ingestion and preprocessing (JSON → structured format)
- Filtering ML and NLP research papers
- Text chunking and metadata enrichment
- Vector embeddings and similarity search
- RAG-based question answering
- Citation tracking (paper ID, section, or abstract)
- Evaluation of retrieval relevance and response quality
- Optional lightweight UI (Streamlit)

Out-of-Scope

- Fine-tuning large language models
- Real-time document ingestion
- Multi-modal (image/table) retrieval
- Full production deployment at scale

Constraints

- Dataset size may require filtering to manage memory and performance
- Abstract-only text may limit depth of answers compared to full papers
- Chunk size and overlap require careful tuning for retrieval quality
- Citation granularity is limited to available metadata
- LLM responses must remain strictly grounded in retrieved context

Stakeholders

Primary Stakeholder (Hypothetical)

AI Platform or Research Tools Team building internal or customer-facing knowledge assistants

Secondary Stakeholders

- Researchers: Efficiently explore ML/NLP literature
- Data Scientists & ML Engineers: Access grounded technical explanations
- Product Teams: Build trustworthy AI assistants
- Leadership: Ensure transparency and reliability in AI systems

Data Sources

Dataset

arXiv Research Papers Dataset (Kaggle):
https://www.kaggle.com/datasets/Cornell-University/arxiv?resource=download

Description

A publicly available dataset containing metadata and abstracts of academic research papers from arXiv, including machine learning and natural language processing domains.

Key Fields

- id (arXiv paper ID)
- title
- abstract
- categories (e.g., cs.CL, cs.LG, cs.AI)
- authors
- update_date

This dataset is well-suited for citation-based RAG due to stable paper identifiers and structured metadata.

Tools & Technologies

Category	Tools / Libraries
Data Processing	Python, Pandas, NumPy

NLP & RAG	LangChain or LlamaIndex
Embeddings	OpenAI or open-source embedding models
Vector Store	FAISS or Chroma
Evaluation	Manual relevance checks, retrieval inspection
Interface (Optional)	Streamlit
Environment	Jupyter Notebook, VS Code
Version Control	GitHub

Project Workflow

1. Data Exploration (EDA)

- Inspect raw arXiv metadata
- Analyze distribution of categories and paper counts
- Identify relevant ML/NLP subsets
- Validate data completeness and text quality

2. Data Transformation

- Filter papers by relevant categories (cs.CL, cs.LG, cs.AI)
- Normalize and clean text fields
- Create structured document objects with metadata
- Prepare data for chunking

3. Document Chunking & Embeddings

- Split abstracts into semantically meaningful chunks
- Attach metadata (paper ID, title, category)
- Generate embeddings for each chunk
- Store embeddings in a vector database

4. Retrieval-Augmented Generation Pipeline

- Accept user questions
- Retrieve top-k relevant chunks using similarity search
- Pass retrieved context to an LLM

- Generate answers strictly grounded in retrieved text
- Attach citations referencing source documents

5. Evaluation & Analysis

- Test question-answer relevance
- Verify citation accuracy
- Analyze failure cases (irrelevant retrieval, vague answers)
- Document design trade-offs and improvements

Deliverables

- Jupyter Notebooks covering EDA, preprocessing, and RAG implementation
- Modular Python scripts for retrieval and citation logic
- Optional Streamlit chatbot interface
- Final Project Report summarizing:
  - Methodology
  - RAG architecture
  - Evaluation findings
  - Limitations and future improvements
- GitHub repository with clear documentation

Expected Impact

This project will demonstrate my ability to:

- Design and implement an end-to-end RAG system
- Work with unstructured text data at scale
- Build trustworthy, citation-aware LLM applications
- Evaluate LLM outputs for relevance and grounding
- Communicate complex AI systems clearly to technical and non-technical audiences