

Project Assignment: Question-Answering Slackbot for a DSAN Slack Channel

1\ Project Overview

In this project, you will build a **Question-Answering Slackbot** that can assist users in a DSAN (Data Science and AI) Slack workspace by providing intelligent, context-aware responses based on **Slack conversation history**.

Your solution must be **Retrieval-Augmented Generation (RAG)-based**, leveraging **multiple advanced RAG techniques** to improve retrieval accuracy and answer relevance. You will deploy your Slackbot **within a Slack channel**, allowing users to interact with it seamlessly.

You will be provided with **anonymized Slack data**, but you may also generate **synthetic Slack conversations** (extra credit) in Slack's JSON format to ensure robustness.

2\ A Note on Synthetic Data

If you choose to generate **synthetic Slack data**, it should:

- * **Follow Slack's JSON format** so that your solution is compatible with real Slack exports.
- * **Mimic real-world conversations** around data science, AI, and relevant topics.
- * **Include diverse query patterns**, such as follow-ups, ambiguous questions, and references to past discussions.

This synthetic data will help test and fine-tune your bot's response quality in various scenarios.

* 📌 *Extra Credit: **Additional points** for generating high-quality synthetic Slack data.*

3\ Technical Components & Implementation

Your solution should consist of the following key components:

A. Data Processing & Storage

- * Load and preprocess **Slack JSON data** (real or synthetic).
- * Identify **message threads**, metadata (timestamps, user references), and structure the

data for retrieval.

- * Store processed data in a **vector database** (e.g., FAISS, Weaviate, Pinecone, or Chroma).

B. Retrieval-Augmented Generation (RAG) Pipeline

Your bot should implement **multiple RAG techniques** to enhance retrieval and response quality. **At least three of the following techniques are required**:

1. **Query Rewriting**:

- * Rewriting user queries to improve retrieval accuracy.
- * Example: "How do I use LangChain for document QA?" → "LangChain document question answering implementation examples."

2. **Query Decomposition**:

- * Breaking down **complex questions** into smaller sub-queries for better retrieval.
- * Example:
 - * "How do I use LlamaIndex with a knowledge graph and fine-tuning?"
 - * → Split into:
 1. "How does LlamaIndex work with knowledge graphs?"
 2. "How to fine-tune a retrieval model?"

3. **Graph RAG**:

- * Building **a knowledge graph** from Slack data (e.g., connecting concepts, FAQs, or recurring topics).
- * Using **graph-based retrieval** instead of just keyword-based search.

4. **Ensemble Retriever**:

- * Using **multiple retrievers** (e.g., **dense vector search** + **BM25**) to improve recall.
- * Combining results using ranking or fusion techniques.

5. **Hybrid Search (Dense + Sparse Retrieval)**:

- * Combining **vector embeddings (semantic search)** with **keyword-based BM25 retrieval** to improve search results.

C. Slackbot Integration

- * Implement a **Slack app** using Slack's API and event handling.
- * Handle **real-time user queries** and return responses from your RAG pipeline.
- * Ensure **multi-turn conversation support** (the bot should remember context within a thread).
- * Allow **user feedback collection** (e.g., thumbs up/down for responses).

D. Deployment

- * Deploy the bot using **a cloud or local environment** (e.g., AWS Lambda, FastAPI, or a Flask-based server).
- * Ensure **secure API integration** with Slack.

4\. Evaluation & Success Metrics

A. Core Functionality

- * Slackbot **correctly retrieves and answers** user questions based on Slack data.
- * Responses are **coherent, contextually relevant, and factually correct**.

**B. RAG Techniques

- * Uses **at least three advanced RAG techniques** from the list.
- * Effectiveness of retrieval and response generation improves with these techniques.

**C. Slackbot Integration

- * Bot is **fully functional in Slack** with real-time interaction.
- * Supports **multi-turn conversations** and context tracking.

**D. Extra Credit


- * High-quality **synthetic Slack data** is generated in JSON format.
- * Synthetic data improves **bot performance on edge cases**.


**E. Success Metrics


Your project will be evaluated based on:

- * **Response Accuracy:** Does the bot return relevant and correct answers?
- * **Retrieval Efficiency:** Are the retrieved documents and Slack messages useful?
- * **User Interaction Quality:** Does the bot handle follow-ups and ambiguous queries well?

**5\. Why This Project Matters

This project prepares students for **real-world AI applications** by combining:  **NLP & Retrieval-Augmented Generation (RAG):** Industry-standard techniques used in enterprise AI search solutions.

 **Generative AI & Question Answering:** Training models to generate responses based on retrieved documents.

 **Slackbot Integration & Real-World Deployment:** Building an **end-to-end AI assistant** that integrates into **enterprise workflows**.

✅ **Hands-On Work with Multiple RAG Variants:** Applying **Graph RAG, Hybrid Search, and Query Rewrite** to improve retrieval.

By completing this project, you'll gain **practical experience in AI development**, **retrieval techniques**, and **deploying AI assistants in business environments**.

Tools & Resources

💡 **Vector Databases:** FAISS, Pinecone, Weaviate, ChromaDB

💡 **RAG Frameworks:** LangChain, LlamaIndex

💡 **Slack API Docs:** Slack API

💡 **Cloud Deployment:** AWS Lambda, FastAPI, Flask