Project Report: RAG Chatbot for CNN/DailyMail Articles

Component Choices and Justifications

1. Embeddings

- **Dense Embeddings:**
  - **Model:** all-MiniLM-L6-v2 (Sentence Transformers)
  - **Justification:**
    - Fast, lightweight, and provides strong semantic representations for sentence-level tasks.
    - Well-suited for semantic search and retrieval-augmented generation (RAG) pipelines.
- **Sparse Embeddings:**
  - **Model:** BM25 (Qdrant/bm25)
  - **Justification:**
    - Classic IR method, excels at exact keyword matching and recall.
    - Complements dense embeddings by capturing lexical overlap.
- **Late Interaction Embeddings:**
  - **Model:** ColBERT v2.0 (colbert-ir/colbertv2.0)
  - **Justification:**
    - State-of-the-art for token-level matching and entity-centric queries.
    - Enables hybrid search by capturing both semantic and fine-grained token interactions.

2. LLM (Large Language Model)

- **Model:** Google Gemini 2.0 Flash (via LangChain)
- **Justification:**
  - Modern, high-quality generative model for answering questions and summarizing context.
  - Integrates easily with LangChain and supports streaming responses.

3. Vector Database

- **Database:** Qdrant Cloud
- **Justification:**
  - Supports hybrid search (dense, sparse, late interaction) natively.
  - Scalable, fast, and easy to use with Python SDK.
  - Allows multi-vector and multi-modal search for advanced RAG workflows.

Chunking Approach and Novel Features

- **Chunking:**
  - Each article is treated as a single chunk (no further splitting), as the CNN/DailyMail dataset consists of relatively short news articles.
  - This approach preserves context and coherence for each document.
- **Novel Features:**
  - **Hybrid Search:** Combines dense, sparse, and late interaction embeddings for robust retrieval across query types.
  - **Prefetching:** Uses Qdrant's prefetch to retrieve candidates from multiple embedding spaces in a single query, improving recall and diversity.
  - **RAG Pipeline:** Retrieved documents are passed as context to the LLM for answer generation, ensuring responses are grounded in the source material.
  - **Streamlit Frontend:** Interactive chat interface with source document display and adjustable retrieval settings.

Additional Notes

- The system is designed for extensibility: new embedding models or LLMs can be swapped in with minimal code changes.
- All API keys and configuration are managed securely via environment variables.