Sai Pranav Madineni
smadinen

Comprehensive Usage Documentation

1) Tool Inventory (GenAI tools, models, APIs, and libraries)

How versions were/should be captured: I recorded versions at run time using `pip show` and library `__version__` attributes. (If any cell didn't print versions, I can rerun a short environment-report cell to fill gaps.)

Inventory Table

Category	Name / Identifier	Version (fill)	Local / Cloud	Primary Purpose	Where Used (Notebook Step)
Text Generation (LLM)	HuggingFace Transformers — **FLAN-T5-base**		Local	Context-grounded answer generation	Naïve (2&3), Advanced (5), Eval (6)
Embeddings	SentenceTransformers — **all-MiniLM-L6-v2**		Local	384-dim dense vector encoding for retrieval	Naïve (2&3), Advanced (5)
Reranker (Cross Encoder)	**cross-encoder/ms-marco-MiniLM-L-6-v2**		Local	Pairwise reranking of candidate passages	Advanced (5)
ANN Index (Exact)	**FAISS** (IndexFlatIP)		Local	Vector similarity search (exact, IP metric)	Advanced (5), Eval (6)
Vector DB (Baseline)	**Milvus Lite** (pymilvus)		Local	Vector storage & search (FLAT/IP)	Naïve (2&3)
Evaluation (optional)	**ragas** (≥0.1.9) + local HF LLM		Local	Faithfulness, answer-relevancy, context metrics	Eval (6)
Data Loading	HuggingFace **datasets**		Local	`rag-datasets/rag-mini-wikipedia` loader	EDA (1), 2–6
Utilities	**numpy**, **pandas**, **nbconvert**		Local	Arrays/dataframes; notebook→.py conversion	All
Authoring/Assistance (AI)	**ChatGPT (GPT-5 Thinking)**	N/A	Cloud	Documentation help, planning, troubleshooting advice	Planning, orchestration, report text

Sai Pranav Madineni
smadinen

Note: I ran fully **offline** for model inference (no OpenAI API usage in pipelines). The authoring assistant (ChatGPT) was used for documentation and non-core code scaffolding only; details are logged below.

---

2) AI Usage Log (Specific purposes, inputs, and output utilization)

Logged Entries

Entry A — Path Resolution & Conversion

- **Tool**: ChatGPT (GPT-5 Thinking)
- **Purpose**: Make nbconvert robust to Google Drive path variants (`MyDrive` vs `My Drive`) and find notebooks by filename.
- **Input**: "My nbconvert step can't find `NAIVE_RAG(Steps_2_and_3).ipynb`. Provide a resilient approach to mount Drive, search both roots, and rglob for files by name without changing core logic."
- **Output Usage**: I adopted a small discovery cell to resolve absolute paths and list directory contents when files weren't found.
- **Verification**: Confirmed the function returned correct absolute paths and that conversion produced `.py` files; downstream imports succeeded.
- **Date / File/Cell**: Orchestrator, Cell "Mount & Path Discovery".

Entry B — Normalizing Chunk Exports Across Notebooks

- **Tool**: ChatGPT (GPT-5 Thinking)
- **Purpose**: Standardize differing data shapes (e.g., `chunks[*].content`) into a canonical `{id, text}` list for FAISS/Milvus.
- **Input**: "My Step-2 notebook uses `chunks` with a `content` key; Step-5 may export `docs` or a DataFrame. Give me a minimal, safe normalizer (no core logic changes)."
- **Output Usage**: I used the suggested normalizer to bind `chunk_store = [{id, text}]` regardless of source shape.
- **Verification**: Checked length equality (num chunks == num vectors after embedding), printed sample entries, and successfully ran retrieval smoke tests.
- **Date / File/Cell**: Orchestrator, Cell "Bind resources".

Entry C — Report Prose & Structure

- **Tool**: ChatGPT (GPT-5 Thinking)
- **Purpose**: Improve readability and flow of the Phase-5 technical report (Executive Summary, Architecture, Results, Production).
- **Input**: "Rewrite for a technical audience, first-person singular, with a narrative flow and clear trade-offs; no turnkey insights."

- **Output Usage**: I used the edited paragraphs as a draft, then revised for accuracy and alignment with my logs.
- **Verification**: I cross-checked every number against printed notebook outputs/CSVs; I edited wording where needed.
- **Date / File/Cell**: Report document.

I will append additional entries if I obtain further AI assistance (e.g., debugging ideas or citation formatting). Each entry includes verification steps.

3) Code Generation (AI-assisted code with attribution)

Policy: I did **not** use AI to implement core RAG components (chunking, embeddings, indexing, retrieval, reranking, generation). Where I accepted small helper snippets, I clearly attributed them and verified correctness.

Attribution Template

- **Location**: [File/Cell reference]
- **What AI generated**: [Short description of non-core helper]
- **My modifications**: [Edits/refactors made by me]
- **Why acceptable**: [Does not implement core logic; I understand and can reproduce it]
- **Verification**: [Assertions, smoke tests, sample prints]

Attribution 1 — Path Discovery Helper

- **Location**: Orchestrator, "Mount & Path Discovery" cell
- **What AI generated**: A small function that tries `/content/drive/MyDrive` and `/content/drive/My Drive`, then `rglob`-searches for the target notebooks by filename.
- **My modifications**: Reduced branches, added explicit error messaging and directory listing on failure.
- **Why acceptable**: This only locates files; it does not affect RAG logic.
- **Verification**: Verified paths printed correctly; nbconvert succeeded; module import/run worked end-to-end.

Attribution 2 — Export Normalizer

- **Location**: Orchestrator, "Bind resources" cell
- **What AI generated**: A normalizer to map `{docs | chunks | DataFrame | list[str]}` into a canonical list of `{id, text}`.
- **My modifications**: Enforced key order (`text → body → content`), added strict assertions (non-empty, same count as vectors), and preview logging.
- **Why acceptable**: Pure data-shape adaptation; no retrieval/scoring logic.
- **Verification**: Length checks, sample inspection, and successful retrieval smoke tests.

Sai Pranav Madineni
smadinen

4) Content Creation (AI involvement in documentation, analysis, reporting)

Policy: AI assistance was limited to editing/structuring documentation and proposing clearer ways to present results. All technical claims, interpretations, and metrics originated from my notebooks and were independently verified.

Attribution A — Technical Report Flow

- **Section**: Executive Summary, Architecture, Results, Production Considerations
- **AI Contribution**: Improved narrative flow, tightened phrasing, suggested how to present recall–precision trade-offs without overclaiming.
- **My Verification**: I cross-checked every figure (chunk counts, dimensions, precision/recall, EM/F1) against the notebook outputs; edited wording where necessary to exactly match my runs.

Attribution B — Academic Integrity Appendix

- **Section**: Enhanced Integrity & AI Collaboration Framework
- **AI Contribution**: Provided the skeleton for the AI Usage Log and attribution format; refined wording for clarity.
- **My Verification**: Ensured the log entries reflect my actual usage; added verification steps and file/cell references.

Signature
I affirm that I independently implemented, executed, and analyzed the core RAG components in this project. AI assistance, where used, is documented in this section and did not replace my own understanding or decision-making.