AI Research Assistant - Phase 1 Report

Student Name: Muhammad Rafay

Student ID: 22i-0948

Course: CS-4015 Agentic AI

Date: February 12, 2026

1. Introduction

The goal of this assignment was to build an AI Research Assistant capable of semantic search over a collection of AI-related documents.

The system loads documents in various formats, generates embeddings using state-of-the-art models, indexes them in a vector store,

and provides a user-friendly GUI for querying and retrieving relevant information.

The implementation leverages LangChain for document processing and embedding, supports multiple HuggingFace embedding models,

and allows users to choose between FAISS and ChromaDB as the vector store. The GUI, built with Tkinter, enables dataset selection,

configuration of embedding/vector store, and interactive semantic search.

2. System Architecture

Components:

- Document Loader: Loads .txt, .md, .pdf, .docx files using PyPDF2 and python-docx.
- Embedding Engine: Uses HuggingFace sentence-transformers via LangChain.
- Vector Store: FAISS (in-memory) and ChromaDB (persistent).
- Search Interface: Tkinter GUI for configuration and semantic search.

Technology Stack:

Embedding Models: all-MiniLM-L6-v2, all-mpnet-base-v2, multi-qa-MiniLM-L6-cos-v1, paraphrase-multilingual-MiniLM-L12-v2, all-distilroberta-v1

Vector Databases: FAISS, ChromaDB

Framework: LangChain

GUI: Tkinter

3. Experiments and Analysis

Embedding Model Comparison:

- all-mpnet-base-v2 performed best overall.

- all-MiniLM-L6-v2 was fast and generally relevant.

- multi-qa-MiniLM-L6-cos-v1 was good for QA but less technical depth.

Vector Store Comparison:

- FAISS: Faster, in-memory.

- ChromaDB: Persistent, slightly slower.

Dataset:

- 13 documents (~120KB)

- Topics: AI, ML, DL, NLP, CV, healthcare, ethics

4. Challenges and Solutions

- Handling multiple formats → Implemented robust loaders with error handling.

- Slow indexing → Used chunking and allowed store selection.

5. Retrieval Quality Assessment

Strengths:

- Modular architecture

- GUI-based configuration
- High retrieval accuracy

Limitations:

- Slower indexing for large datasets
- Some models struggle with ambiguous queries

Improvements:

- Add more file type support
- Integrate domain-specific embedding models

6. Conclusion

The AI Research Assistant demonstrates effective semantic search using multiple embedding models and vector databases.

Model selection significantly impacts retrieval quality, and vector store choice affects speed and persistence trade-offs.

7. References

LangChain: https://python.langchain.com/

Sentence Transformers: https://www.sbert.net/

FAISS: https://github.com/facebookresearch/faiss

ChromaDB: https://docs.trychroma.com/