Informatique et systèmes de communication ISC
Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften

RAG-Based Knowledge Access for Multimodal E-Commerce Assistant

Author: Ana Gomes Goncalves
Date: 24.11.2025
Version: v.2

Supervisor: Carrino Francesco

Abstract

(1/2 page) Short summary of the project. Describe the context, the pain points, the proposed solution, and the goals to achieve (expected results).

This project is a project semester for ISC at HES-SO Valais-Wallis under Professor Francesco Carrino. It addresses the need for AI-driven tools in e-commerce, where companies require systems to access internal knowledge bases (e.g., product details, FAQs, policies) to deliver accurate and context-aware responses.

Main points include inefficient knowledge retrieval and integration, produce explainable responses using an LLM and challenges with multimodal handling (text, audio, video).

The solution is a Python-based RAG pipeline using open-source tools: LangChain for orchestration, Hugging Face Transformers for embeddings and models, FAISS for vector storage, and a model like OpenAI Whisper for speech-to-text/text-to-speech. It retrieves documents, generates explainable responses with source traceability, and supports at least text and audio modalities.

1 Goals

Briefly define at least one mandatory goal (and no more than 2-3). These goals represent the final objectives of your project and should be defined using the SMART criteria (Specific, Measurable, Achievable, Relevant, Time-bound). At the end, you can add "nice to have" goals: additional objectives to be achieved once the main goals have been completed.

Goal 1 – Retrieve information using RAG

Build a simple retrieval component in the RAG pipeline that finds and pulls relevant information from an e-commerce knowledge base such as product recommendations, details, and help for usage. Using RAG, data from URLs, PDFs, and videos will be handled by converting them into vectors: embedding them with Hugging Face Transformers, storing them in a FAISS vector database, and running similarity searches through LangChain to enhance queries. Assistant should produce a response to the user.

Goal 2 – Response explainability

The second goal focuses on improving the pipeline, so the assistant can explain sources of given response. Using LangChain for orchestration, the system will prompt the model to produce coherent, context-aware answers in text or audio / image format, with embedded citations to original sources for transparency (source in documents or timestamps in video). Assistant must deliver trustworthy recommendations without hallucinations.

Goal 3 – Implement multimodal

The objective is to enable multimodal inputs and outputs, including speech-to-text for user queries, text-to-speech for assistant responses to make interactions more accessible and versatile. This enhances user experience in e-commerce scenarios, such as voice-assisted shopping.

Nice-to-have goal – Interface and images

If the main goals have been achieved, an interface can be implemented to facilitate the use of the assistant. Interface would help users communicate easily with the assistant through text or audio. The interface would be simple, user-friendly, and would have "prompt examples" so users know what type of questions the assistant can handle.

Images could also be shared by the assistant to help users with their needs.

2 Tasks

Define the tasks required to achieve the goals mentioned above. Tasks can be grouped into work packages (WP) or phases. See the example below for guidance.

WP1 – Analysis and hands-on project

- **T1.1 – Project organization.** Plan and organize project tasks using GANTT.
- **T1.2 – Hands-on RAG.** Review fundamentals of RAG using Mr. Carrino's lab.
- **T1.3 – Details on LLM.** Discuss the use of Calypso with Mr. Mudry.
- **T1.4 – Have a simple demo of an assistant.** Create a simple demo (on terminal/console) using data stored in vectors in simple LLM prompt.
- **T1.5 – Documentation.** Start documenting tasks that are done.

WP2 – Conception

- **T2.1 – Create architecture for the whole project.** Have an architecture displaying libraries used and links between clusters, dB, etc… And UML diagrams of main classes
- **T2.2 – Interface mockup.** Quick and dirty mockup of the interface
- **T2.3 – Prepare data for search.** Index, split and store data as vectors
- **T2.4 – Retrieve and generate answers.** Using RAG agent, retrieve relevant information and produce an answer (basic)
- **T2.5 - Pipeline evaluation.** Make test datasets and evaluate pipeline performance and accuracy. These same datasets can also be used later, to test overall performance
- **T2.6 – Documentation.** Complete documentation with new tasks done.

WP3 – Iterative implementation

- **T3.1 – Test.** Test previously made assistant (using test datasets, unit tests, …) and improve if needed.
- **T3.2 - Implement multimodals.** Implement speech-to-text and text-to-speech.
- **T3.3 - Improve LLM.** Add memory / chat history.
- **T3.4 - Create interface.** Simple interface, chat-like, possibility to use audio instead of text. Link backend to interface.
- **T3.5 – Documentation.** Complete documentation with new tasks done.

WP4 – Tests and Evaluation

- **T4.1 – Final testing.** Test the whole assistant.
- **T4.2 - Final improvements.** Finish / polish every feature. Re-confirm by testing new add-ons.
- **T4.2 - Document.** Polish documentation for client uses and overall project documentation.

- **T5.2 - Presentation.** Prepare the project's presentation as well as future improvements, etc...

3 Milestones

Define the milestones of your project. Milestones correspond to significant moments in your project where progress can be assessed. Examples include the completion of a work package (WP), the achievement of a major task, a major code release, or an imposed deadline (e.g., final presentation). Each milestone should be accompanied by measurable, tangible deliverables. Examples of deliverables include sections of a report, intermediary presentations, source code, etc.

M1 – Project Planning and Initial Hands-On, week 1

Description: This milestone marks the completion of the analysis phase, including project organization, foundational RAG exploration, and a basic proof-of-concept demo.

- **D1.1 – Project plan:** Gantt chart outlining tasks, dependencies, etc...
- **D1.2 – Simple RAG demo:** Console-based prototype demonstrating basic vector storage and retrieval from sample data.
- **D1.3 – Documentation:** Documentation for WP1 tasks.

M2 – Architecture Design and Basic RAG Implementation completed, week 2

Description: Conceptualizing the system, including architecture diagrams and initial RAG pipeline setup for retrieval and generation. It addresses main points like efficient knowledge integration and basic explainability.

- **D2.1 – System Architecture Documentation:** UML diagrams and component overviews showing libraries, data flows, and modularity, from T2.1.
- **D2.2 – Basic RAG Pipeline Code**: Functional Python scripts for data preparation, vector indexing, retrieval, and simple answer generation with source citations, covering T2.2-T2.3.
- **D2.3 - Evaluation metrics:** Metrics / graphs to represent project evaluation. List of improvements and solutions that could be made.

M3 – Multimodal interaction integrated into the system

Description: This milestone involves enhancing the assistant with multimodal features (speech-to-text, text-to-speech, potential image sharing) and initial testing for robustness and usability. It builds the basic pipeline to support at least text + audio outputs.

- **D3.1 - Multimodal Prototype:** Updated code integrating APIs like OpenAI Whisper for audio handling and chat history/memory, with tests on pre-made queries (T3.1-T3.4).
- **D3.2 - Interface Demo:** A simple chat-like text/audio interface (e.g., Streamlit or even React and Vite) demonstrating end-to-end functionality.
- **D3.3 - Documentation:** A Documentation of all tasks done for now and user guide.

-

M4 – Final modifications

Description: This final milestone represents the last modifications of the project. Main features will be tested, and minor fixes will be made. The project's documentation and presentation will also be made.

- **D3.1 – Final version:** Assistant finished and tested.
- **D3.2 - Presentation:** Presentation for the project of semester finished and ready to be presented.