LeoChat

Automation Track

Report Project

Hackaton IBM X Hackathon - 2025

ANDACIC Lucie, GUENNOU Evan, HADIFE Iyad, HUET Valentin, JAUNAY Iliana, MACHADO MONTEIRO Tiago

Table of Contents

I. Introduction

The IBM Data/AI hackathon, held from November 5th to 7th, 2025, aims to design innovative artificial intelligence solutions using the IBM WatsonX platform. This project is open to ESILV students in their fifth year of the DIA program.

During this intensive three-day event, participants must work in teams to tackle real-world challenges presented by IBM and the cluster.

Our team has chosen the "Automation" track, which focuses on creating intelligent systems capable of optimizing workflows and resource management within companies using artificial intelligence.

The *Intelligent Help Center for PLV Students* project aims to create a conversational assistant capable of automatically responding to student inquiries using an internal database of frequently asked questions. The system serves as a proof of concept (POC) to demonstrate how artificial intelligence can improve information accessibility, reduce manual workload, and enhance the overall user experience for students and staff at PLV.

Through a user-friendly web interface inspired by modern chat assistants, students can type their questions in natural language and instantly receive relevant answers. The assistant uses semantic search and contextual understanding to identify the best possible response from a structured dataset. When no suitable answer is found, users are redirected to a contact form or support email for human follow-up. The system is designed to continuously improve over time by learning from user feedback and integrating new validated data into the knowledge base.

II. Data description

The project's dataset consists of an **Excel file with three sheets**, serving as the foundation for the intelligent assistant's knowledge base.

1. **Main Sheet (Q&A Dataset):**
    a. This sheet contains **495 entries**, each representing a question-and-answer pair.
    b. The main columns include:
        i. **ID:** Unique identifier for each record.
        ii. **Title:** The question or topic submitted by a user.
        iii. **Content:** The corresponding answer or solution.
        iv. **Date:** The date the entry was created or last updated.
        v. **Post Type:** The type of content.
        vi. **Language:** Indicates whether the entry is in French or English.
        vii. **Theme:** Thematic category.
        viii. **User:** The author or contributor of the post.
        ix. **School:** The relevant PLV institution.
        x. **Status:** Indicates whether the entry is for example publish or private.

This sheet constitutes the **core dataset** used to train and query the assistant. It enables semantic matching between a student's question and the most contextually appropriate answer.

2. **Video Export Sheet:**
    a. Contains a limited number of entries that include links related video resources to specific student queries.
    b. These videos are used to enrich the chatbot's answers with multimedia content when relevant.
3. **Tutorial Export Sheet:**
    a. Includes short tutorials or links.
    b. These items serve as complementary materials to enhance the chatbot's responses with practical, actionable information.

Together, these three sheets form a **structured and diverse data source** enabling both textual and multimedia-based support for PLV students. After that, a data preprocessing has been processed.

III. Technical Approach

The architecture is based on a Python module that interacts with the LLaMA 3.2 11B Vision Instruct model through the IBM watsonx.ai API. The system's goal is to automatically generate responses based on a predefined dataset.

The data_pretraitee.csv file contains approximately 400 pairs of questions and answers. The script reads this file using the csv module and extracts all relevant data. This information serves as examples for the model to learn how to formulate coherent responses to new questions.

The extracted data is then integrated into a structured prompt, where each example is presented as follows:

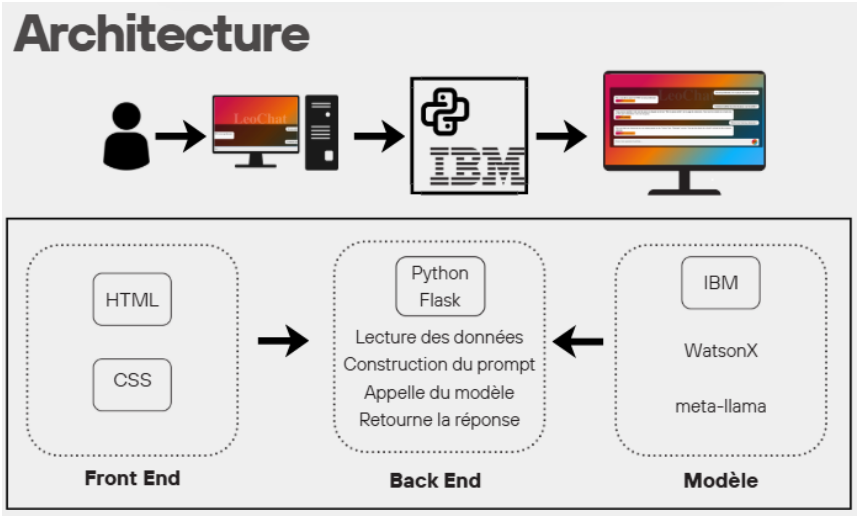
Input: <question>
Output: <answer>

This format allows the model to clearly understand the relationship between each question and its corresponding answer.

The script uses the ibm_watsonx_ai library to interact with the API. The selected model, meta-llama/llama-3-2-11b-vision-instruct, is a multimodal language model capable of understanding both textual and visual instructions. The generation parameters such as the decoding method, maximum number of tokens, and repetition penalty are configured to optimize the coherence and relevance of the produced answers. The model is trained on the question-answer pairs contained in the file, and then the user's new question is added so that the model can generate the appropriate output.

The text generated by the model is then cleaned and filtered to remove unwanted elements, such as repeated prompts or incomplete answers. If the result is incoherent or empty, a default message is returned. Finally, the final response is displayed to the user, directly in a simple interface, ensuring a smooth and intuitive interaction.

Furthermore, when a user is satisfied with the generated response, they can click a dedicated button in the interface. This action sends the question and its corresponding answer to the data_pretraitee.csv file, adding them as a new entry. This mechanism allows the system to continuously improve itself by incorporating user feedback to enhance the quality and accuracy of future responses. In this way, the model benefits from incremental self-learning, gradually strengthening its knowledge base with each validated interaction.



IV. Challenges Faced

One of the main challenges at the beginning of the project was the use of the IBM watsonx.ai platform. Even though the environment is powerful, it required some time to get familiar with before it could be used efficiently. Setting up the credentials, understanding how project spaces work, and managing API calls all took some effort. Several configurations had to be tested before achieving a stable workflow between the Python code and the Watsonx-hosted model.

Another important challenge was choosing the most suitable AI model. Several models were available on the platform, each with different capabilities, sizes, and execution costs. After several tests and comparisons, the LLaMA 3.2 11B Vision Instruct model was chosen for its strong performance and versatility, especially its ability to handle complex textual instructions. This decision required some analysis to find the right balance between power, execution speed, and response quality, while also meeting the technical and time constraints of the project.

- **Data Structure & Quality:** The initial Excel dataset contained unstandardized entries with inconsistent question phrasing and answer formats, which complicated preprocessing and semantic matching.

- **Response Ranking:** Selecting the most relevant answer required balancing lexical similarity and semantic relevance.

- **Integration Constraints:** The solution had to align with the PLV IT department's existing stack.

- **Feedback & Learning Mechanism:** Designing a self-learning approach that safely incorporates user feedback into the database without degrading answer quality posed an additional challenge.

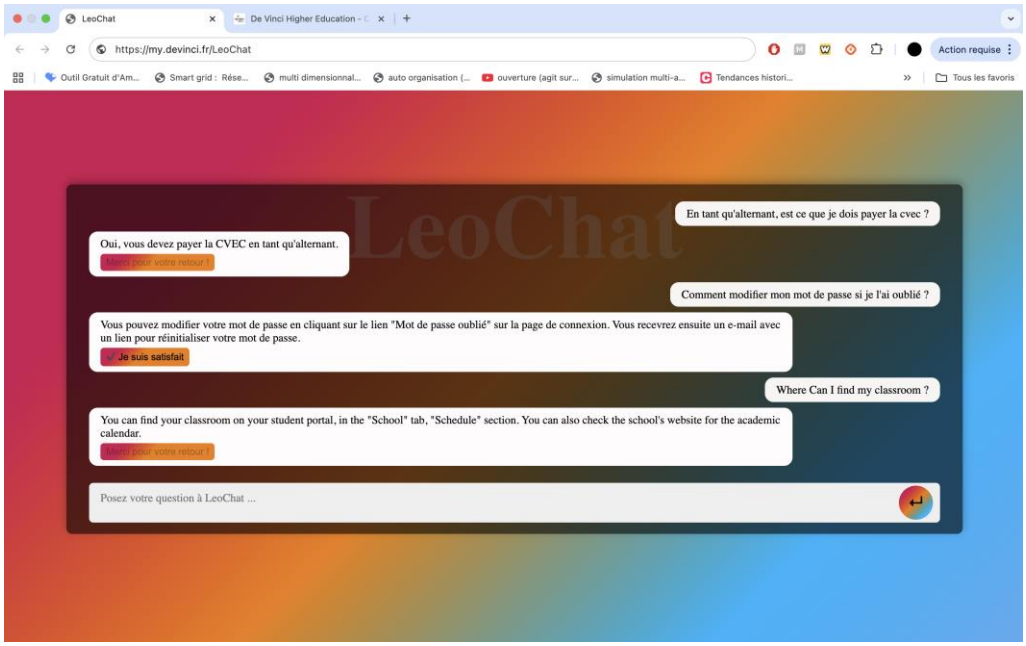
V. Solutions implemented

We realized a preprocessing pipeline for content and video data. It first loads and merges two Excel sheets containing question and video metadata into a single unified DataFrame. Then, it filters out non-final entries (where status is "draft") to retain only published or valid items for further processing.

Next, we isolate rows corresponding to video posts, extracts YouTube video IDs from embedded URLs or iframes using regular expressions, and exports this subset for transcription. Then we run an automated transcription pipeline: downloading audio from each video via `yt-dlp`, transcribing it using OpenAI's Whisper (with GPU acceleration and multilingual fallback), and cleaning up temporary files.

The transcribed text is **saved and merged back** into a structured CSV. The dataset undergoes additional data cleaning, handling missing values, removing empty or invalid text fields, and normalizing key columns to ensure consistency between the Excel export and the transcribed results. The final merged CSV, thus provides a well-structured, cleaned, and text-enriched dataset ready for deeper NLP or semantic processing.

- **Semantic Search Engine:** A lightweight NLP model was integrated to enable contextual similarity matching between user queries and stored questions.
- **Fallback Mechanism:** When no suitable match is found above a confidence threshold, the system gracefully redirects the user to an embedded contact support email.
- **Feedback Loop:** User interactions are logged, and liked responses are flagged for integration into the knowledge base, supporting continuous self-improvement.



Example of the website and prompts.

VI. Future Improvements

Future iterations of the project aim to transform the POC into a fully intelligent help center integrated into the PLV digital ecosystem. Planned enhancements include:

- **Advanced NLP Models:** Integration of new transformer-based language models for deeper semantic understanding.
- **Enhanced Analytics Dashboard:** Implementing dashboards for administrators to monitor question frequency, unanswered queries, and model performance.
- **Mobile Integration:** Create and extend the Vue.js interface to a mobile-friendly application for broader accessibility.

By combining intelligent search, user feedback, and the IT department's established infrastructure (Vue.js + MariaDB), the *Intelligent Help Center for PLV Students* demonstrates a scalable, future-ready approach to digital academic support.

VII. Conclusion

The *Intelligent Help Center for PLV Students* demonstrates how artificial intelligence can be effectively integrated into academic support systems to enhance accessibility, responsiveness, and self-service capabilities. By combining a structured dataset with a semantic search engine, the system provides students with fast, accurate, and user-friendly assistance.

Beyond its role as proof of concept, this project lays the foundation for a scalable, continuously improving support platform. Ultimately, the solution aims to empower the PLV IT department with a sustainable, intelligent tool for student engagement and knowledge management.