Project Report: RAG-based Chatbot for Masaryk University Information System

Branislav Palúch, Ján Labuda
Masaryk University, Faculty of Informatics

July 13, 2025

Abstract

This report describes the design, implementation and evaluation of a RAG-based chatbot for IS MUNI (ISbot) in its first stage. The ISbot is designed to assist users of the Masaryk University Information System by providing natural language answers based on official documentation (Nápověda). This project originated within a course PA026 Artificial Intelligence Project. The source code is at `www.github.com/pranislav/ISbot`.

Contents

1 Introduction **3**
1.1 About the Project . . . 3
1.2 Motivation . . . 3
1.3 Evolution . . . 3
1.4 Current Status and Roadmap . . . 3

2 Theoretical Background **4**
2.1 Retrieval-Augmented Generation (RAG) . . . 4
2.2 Query Augmentation . . . 4
2.3 Similar Projects . . . 4

3 Implementation **5**
3.1 Document Indexing . . . 5
3.2 Prompt Engineering and Response Generation . . . 5
3.3 Interface . . . 5

4 Installation and Startup **5**

5 Examples **6**
5.1 Chatbot Answer Example . . . 6
5.2 UI Screenshot . . . 6

6 Evaluation **6**
6.1 Methodology . . . 6
6.2 Results . . . 7
6.3 Failure Analysis . . . 8

7 Conclusion **8**

1 Introduction

1.1 About the Project

This project aims to create a chatbot assistant for the Information System of Masaryk University (IS MU), using a retrieval-augmented generation (RAG) approach. The assistant draws information from official IS Help pages (Nápověda), making it easier for users to get relevant support without manually searching extensive documentation.

1.2 Motivation

The IS MU system can be complex and overwhelming, especially for new users. General-purpose language models cannot reliably assist with IS-specific tasks due to their lack of domain knowledge. This project bridges that gap by combining a language model with official IS documents, making system navigation more user-friendly.

1.3 Evolution

The first implementation attempt was using fine-tuning as the way to integrate the relevant information to the LLM. The upside of this approach is lower latency, but it would require very high effort to transfer the knowledge of the Help pages to the LLM reliably while also keeping the model's communicative abilities. Therefore we decided to go for Retrieval-Augmented Generation (RAG) which allows to reach reliable information transfer more easily while keeps model's experssive qualities untouched.

1.4 Current Status and Roadmap

As of July 2025, the chatbot (ISbot) is in its initial phase. It:

- responds to single-turn questions (no conversation memory),
- primarily supports Czech language,
- is not yet publicly accessible.

Planned next steps:

- Add chat mode with multi-turn memory,
- Improve English support (currently limited due to untranslated help content),
- Expand source documents to include:
  - Study and Examination Regulations (Studijní a zkušební řád),
  - Term Calendars by Faculties (Přehled harmonogramu období fakult),
  - and possibly more.

2 Theoretical Background

2.1 Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) is a hybrid approach that combines traditional information retrieval with generative language models. Instead of relying solely on pretrained knowledge within a language model, RAG retrieves relevant context from external documents and provides it as input for answer generation.

In our system, the source documents consist of official help pages (*nápověda*) from the Masaryk University Information System. These documents are first segmented into small chunks based on their hierarchical structure. Each chunk is then embedded into a high-dimensional vector space using a sentence embedding model. These vector representations are linked to their text-form origins so once the relevant chunks are found, the source text can be passed to the model.

During inference, a user query is transformed into a vector representation, and a similarity search is performed against the precomputed document embeddings. The query is embedded in a slightly different way than the document chunks because of its different nature - this is handled inside the relevant library. Retrieved chunks are then passed to the language model along with the query to guide the generation process.

2.2 Query Augmentation

To improve retrieval robustness and recall, we employ a query augmentation technique. The original user query is paraphrased by the language model into multiple alternative formulations. For each paraphrase, a separate similarity search is conducted. The deduplicated union of all retrieved document chunks is then assembled and provided to the language model along with an engineered prompt and the original user query.

This process increases the chance of capturing relevant information, especially when the original query contains ambiguous or uncommon phrasing.

2.3 Similar Projects

RAG-based systems have gained popularity in both academia and industry for building intelligent assistants that can answer factual questions grounded in external sources. Several notable projects and tools relevant to this topic are:

- **Facebook AI's RAG (2020):** The original RAG architecture introduced by Facebook AI combines dense passage retrieval with a generative model (BART). It laid the foundation for integrating retrieval into generation pipelines for open-domain QA.

- **Haystack:** An open-source framework for building question-answering systems that supports RAG, dense/sparse retrieval, and a variety of backends including Hugging Face models and OpenAI APIs.

- **LlamaIndex** A popular Python library for constructing RAG pipelines with local or remote language models. It provides tools for indexing documents, retrieving relevant chunks, and composing responses using prompts and it is also used in this project.

- **IS MU Search Tools:** Existing search functionalities within the IS (e.g., help search) are based on keyword matching and manual navigation. Our project extends this with natural language understanding and dynamic context integration.

3 Implementation

The chatbot system is implemented in Python using the LlamaIndex framework and Hugging Face models. It combines a vector-based retrieval component with a locally running language model to form a Retrieval-Augmented Generation (RAG) pipeline.

3.1 Document Indexing

Official help pages (*nápověda*) were preprocessed into structured JSON files, where each record corresponds to a small, self-contained text block (e.g., a paragraph or section). These blocks are embedded into a vector space using the multilingual model `intfloat/multilingual-e5-base`.

The embedded chunks are indexed using LlamaIndex's `VectorStoreIndex`, which by default uses FAISS for similarity search. The index is persisted to disk and reloaded when available.

3.2 Prompt Engineering and Response Generation

The prompt includes:

- A system message describing the assistant's role and stating the source of retrieved content.
- A delimiter-separated list of relevant text chunks.
- The original user query.

The final prompt is fed into a text-generation pipeline using a local instance of the `google/gemma-3-4b-it` model via Hugging Face Transformers.

3.3 Interface

The chatbot can run in two modes:

- **CLI mode:** a simple terminal interface for querying, used for testing and debugging.
- **Web interface:** built using Gradio, allowing browser-based access.

4 Installation and Startup

```
git clone https://github.com/pranislav/ISbot.git
cd ISbot
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
python src/RAG.py
```

5 Examples

5.1 Chatbot Answer Example

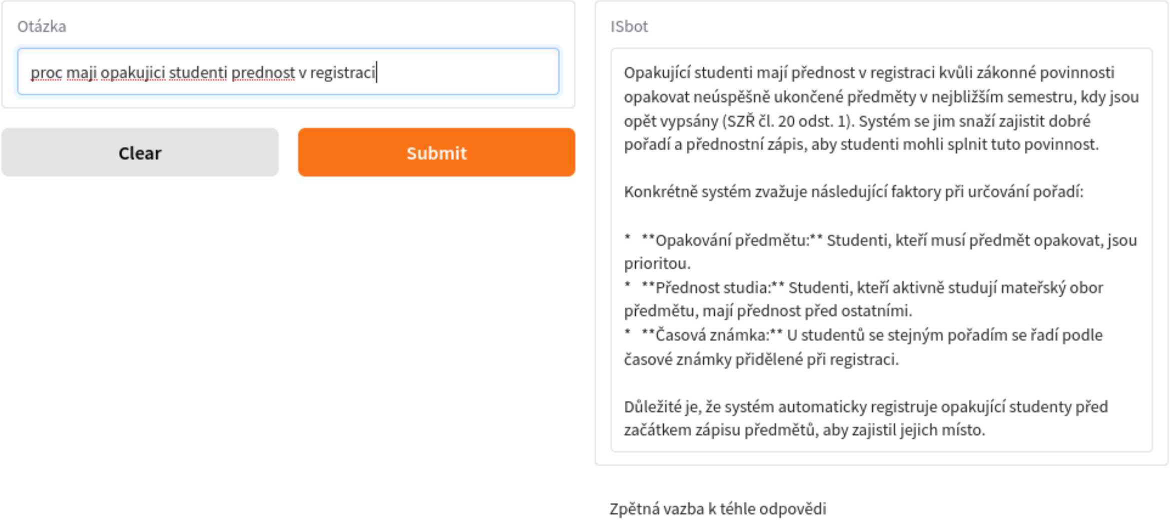
jak ziskam ubytovaci stipendium

Pro získání ubytovacího stipendia postupujte podle těchto kroků:

1. **Zažádejte o stipendium:** V aplikaci **Informační systém > Stipendia > Stipendium na podporu ubytování** zkontrolujte, zda splňujete podmínky pro přiznání koleje/stipendia.
2. **Kontrola údajů:** Zkontrolujte a potvrďte svoje údaje v Matrici studentů: **Informační systém > Personální > Kontrola a změna osobních údajů**.

Dále doporučuji navštívit web SKM pro další informace o ubytování: `https://www.skm.muni.cz/`

5.2 UI Screenshot



proc maji opakujici studenti prednost v registraci

Opakující studenti mají přednost v registraci kvůli zákonné povinnosti opakovat neúspěšně ukončené předměty v nejbližším semestru, kdy jsou opět vypsány (SZŘ čl. 20 odst. 1). Systém se jim snaží zajistit dobré pořadí a přednostní zápis, aby studenti mohli splnit tuto povinnost.

Konkrétně systém zvažuje následující faktory při určování pořadí:

* **Opakování předmětu:** Studenti, kteří musí předmět opakovat, jsou prioritou.
* **Přednost studia:** Studenti, kteří aktivně studují mateřský obor předmětu, mají přednost před ostatními.
* **Časová známka:** U studentů se stejným pořadím se řadí podle časové známky přidělené při registraci.

Důležité je, že systém automaticky registruje opakující studenty před začátkem zápisu předmětů, aby zajistil jejich místo.

Zpětná vazba k téhle odpovědi

Figure 1: Example interface of the chatbot

6 Evaluation

6.1 Methodology

The system was evaluated based on human feedback collected through a Google Form. Each respondent was presented with three rating questions, each using a 1–5 star scale:

- Was the response factually correct?
- Was it helpful for solving your problem?

- Was it clearly written and understandable?

An open-ended question followed:

- Was something missing, unclear, or incorrect? (If so, please describe the issue below.)

Respondents were also asked to indicate their role (Teacher/Staff, IS technical support, Student, Other) to help contextualize their feedback.

Each form submission was linked to a unique timestamp, which corresponded to a log file containing the original user query, the chatbot's answer, all paraphrased (augmented) queries, and the retrieved document chunks. This made it possible to trace back problematic responses and analyze their cause in more detail.

6.2 Results

The total number of responses was 17 and distribution of roles was following:

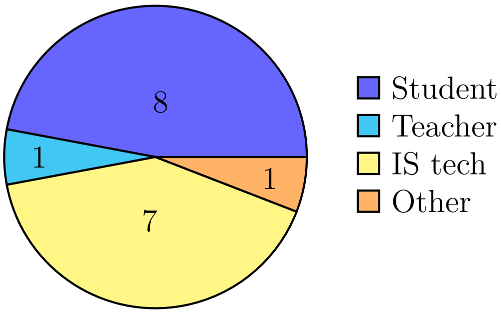


Figure 2: Distribution of respondent roles

Average values for rating questions with corresponding standard deviations are in table 1.

Question	Average Rating	Std. Dev.
Correctness	2.29 / 5	1.4
Helpfulness	2.47 / 5	1.7
Clarity	2.94 / 5	1.9

Table 1: Average user ratings from feedback form

Open-text responses and interaction logs analysis showed several important limitations in the current system:

- **Incorrect or misleading information:** In multiple cases, the chatbot produced inaccurate responses (e.g., one can not get electronic diploma or or referencing wrong agendas such as "Školitel" instead of "Učitel").

- **Hallucinated or not relevant links:** Some answers contained invalid URLs or pointed to irrelevant IS MU pages, which undermined trust in the response.

- **Overgeneralized output**: The system often returned verbose or irrelevant information that failed to directly address the user's query.

- **Ambiguity and unclear phrasing:** Several responses were noted as hard to understand or ambiguous in wording, which is probably due to relatively small model trying to speak czech.
- **Missing key steps:** Important steps or interface elements (e.g., combo-boxes) were sometimes omitted, resulting in incomplete guidance.
- **Off-topic content:** sometimes the chatbot provides information from irrelevant search results, mostly when it can not find the answer to user's question in the retrieved documents.
- **Weak role recognition:** The chatbot sometimes mixes together advices for different roles (student/teacher).

6.3 Failure Analysis

The specific flaws mentioned above could be considered to be caused by these factors:

- **Insufficient retrieval:** Relevant documents were not retrieved for certain paraphrased queries, possibly due to paraphrasing drift or poor semantic matching.
- **Overreliance on generation:** In several cases, the model "hallucinated" plausible-sounding but incorrect information rather than acknowledging a lack of sufficient evidence in the retrieved context.
- **Language limitations of the model:** As the underlying language model was primarily trained on English, its performance in Czech is inherently weaker. This likely affected not only the fluency of generated answers, but also the model's comprehension and contextual understanding of Czech queries and documents.

7 Conclusion

The evaluation confirms that while the chatbot can provide useful assistance in many scenarios, it is currently not reliable enough for unsupervised use. The most helpful answers were those where the correct document was retrieved, and the prompt was specific enough to minimize hallucination.

The following areas require improvement:

- Improving retrieval precision
- Better prompt engineering to control output verbosity
- Adding role-awareness and disambiguation mechanisms
- Including fallback answers when confidence is low
- better document preprocessing, mainly recovering ">" symbols in click-paths

The evaluation feedback is being used to drive future iterations of the system.