MiNI RAG: A QA System for Faculty-related FAQs
Project Report for NLP Course, Winter 2024

Mikołaj Gałkowski, Mikołaj Piórczyński, Julia Przybytniowska
{firstname.lastname}.stud@pw.edu.pl
Warsaw University of Technology, Poland

Anna Wróblewska
Warsaw University of Technology, Poland
anna.wroblewska1@pw.edu.pl

Abstract

Information retrieval within academic settings often poses challenges for students seeking information about studies or faculty-related matters. Traditionally, this process involves navigating complex university websites or directly contacting administrative offices, which can be time-consuming and inefficient. This project introduces an innovative solution: a chatbot system based on a Large Language Model (LLM) and powered by Retrieval-Augmented Generation (RAG). This approach streamlines information access while enhancing user experience. Our case study focuses on leveraging state-of-the-art pre-trained models for language generation and retrieval in Polish. The resulting system, *MiNI* RAG, is designed to provide accurate, context-aware, and coherent responses to academic queries, such as faculty office hours or contact details. To assess its performance, we evaluate the system both qualitatively and quantitatively, emphasizing accuracy and factual correctness. Through this work, we investigate the possibilities and limitations of developing university-specific question-answering applications using currently available solutions.

1 Introduction

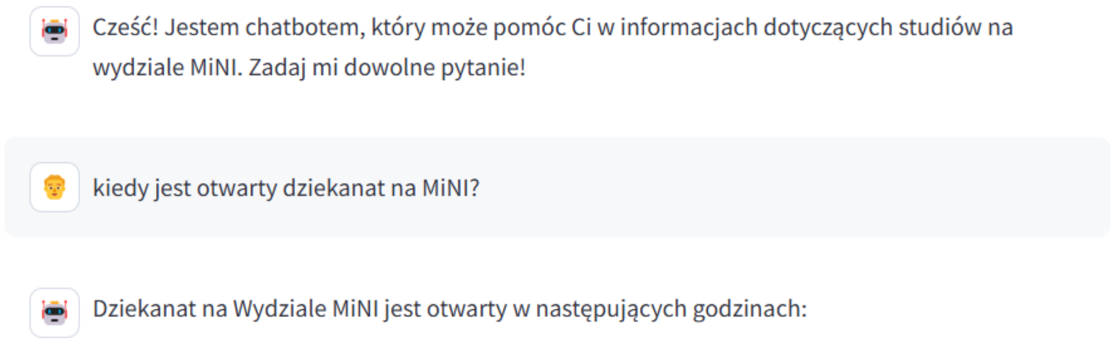
Universities invest significant resources in developing comprehensive websites to communicate essential information about academic programs, services, and resources. Despite these efforts, students often face challenges in efficiently accessing specific information, requiring them to navigate through multiple pages or wait for responses from administrative offices during limited working hours. This inefficiency can particularly impact time-sensitive matters and potentially affect student engagement with the institution.

Recent advances in Large Language Models (LLMs) and Natural Language Processing (NLP) have enabled the development of sophisticated chatbot systems that can understand and respond to user queries in natural language. However, traditional LLMs often struggle with domain-specific information, potentially generating plausible but incorrect responses when dealing with content outside their training data. Additionally, most existing solutions focus on English-language content, leaving a significant gap in non-English academic settings.

The emergence of Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) presents an opportunity to address these limitations by combining the generative power of language models with precise information retrieval. Building on this technology, we present *MiNI* RAG, a Polish-language chatbot system that leverages RAG to provide accurate, context-aware responses to frequently asked faculty-related questions from students in the Polish language. The system leverages data scraped from the Faculty of Mathematics and Information Science website[1] and other university-related PDF documents that may provide additional helpful information. The system aims to streamline information access while improving the user experience by providing context-aware and coherent responses to academic queries, such as faculty office hours or contact details. In-Figure 1 we present an example interaction showcasing the *MiNI* RAG answering a query about the dean's office hours, demonstrating its ability to efficiently and accurately respond to real-world student inquiries.

Concurrently to our work, (Neupane et al.,

[1] https://mini.pw.edu.pl/



Cześć! Jestem chatbotem, który może pomóc Ci w informacjach dotyczących studiów na wydziale MiNI. Zadaj mi dowolne pytanie!

kiedy jest otwarty dziekanat na MiNI?

Dziekanat na Wydziale MiNI jest otwarty w następujących godzinach:

Figure 1: Example interaction showcasing the *MiNI* RAG answering a query about dean's office hours.

2024) investigate the RAG system in the domain of academic-centric question-answering systems. However, our study focuses on the Polish language, exploring the challenges and opportunities of implementing RAG techniques in a non-English academic environment.

The contributions of our work are as follows:

1. Development of a Polish-language chatbot system that integrates RAG for academic queries.

2. Evaluation of embedding models for effective retrieval of Polish-language academic data.

3. Comparative analysis of Polish LLMs and multilingual models within the context of a RAG-powered chatbot.

4. Investigation of the potential for creating a reliable Polish-language question-answering system using state-of-the-art RAG techniques and open-source models.

The rest of the report is organized as follows: Section 2 provides an overview of related work, including recent advancements in RAG systems and Polish-centric language models. Section 3 details the methodology used to develop and evaluate the *MiNI* RAG system, including data collection, model selection, and evaluation metrics. In Section 4, we present the experimental setup and results of our performance evaluation, both qualitatively and quantitatively. Finally, Section 5 concludes the report with a summary of the key findings and directions for future research.

2 Related Work

2.1 Retrieval Augmented Generation

Retrieval-Augmented Generation (RAG)-based systems (Lewis et al., 2020) represent the forefront of conversational AI, particularly in domains requiring accurate and efficient information retrieval (Gao et al., 2023). These systems integrate retrieval modules with large language models (LLMs), enabling them to retrieve relevant context from external knowledge bases and generate coherent, context-aware responses. Recent advancements in RAG have focused on optimizing both the retrieval and generation components.

LLMs for RAG General-purpose models like LLaMA (Dubey et al., 2024) and Mistral (Jiang et al., 2023) are often utilized for RAG tasks, providing a strong foundation for retrieval and generation. Recently, models specifically designed for large-scale enterprise workloads, such as Cohere's Command R+ [2], have shown exceptional performance in RAG applications.

Embedding Models and Retrieval Embedding models play a critical role in the retrieval process of RAG systems. Retrieval is typically achieved by calculating similarity metrics (e.g., cosine similarity) between stored chunks and user queries. Sparse retrieval models like BM25 (Robertson et al., 2009) excel in capturing term-based relevance, while dense retrievers, often powered by pretrained language models, provide rich semantic representations. Recent advancements, including multi-task instruct-tuned models like BGE (Xiao et al., 2024) and GTE (Li et al., 2023) or NV-Embed (Lee et al., 2024) have further enhanced

[2] https://huggingface.co/CohereForAI/c4ai-command-r-v01

performance across various tasks. The the Massive Text Embedding Benchmark (MTEB) (Muennighoff et al., 2022) highlights the strengths of embedding models on diverse datasets, showcasing their effectiveness in specific applications.

Vector stores Vector stores are essential for rapid similarity searches in RAG systems. Open-source solutions like `LLamaIndex`[3], `Chroma`[4], and `FAISS`[5] offer varying trade-offs in scalability and latency, making them suitable for different retrieval scenarios.

RAG evaluation Evaluating RAG systems presents challenges due to the reliance on domain experts, making it time-consuming and costly. The RAGAS framework (Es et al., 2023) was proposed to address this issue, measuring context relevance, answer faithfulness, and answer relevance. RAGAS uses an LLM for automated evaluations. A more recent framework, ARES (Automated RAG Evaluation System) (Saad-Falcon et al., 2023), leverages LLM judges trained on queries, passages, and answers. It evaluates the same metrics as RAGAS but is more accurate, cost-effective, and reduces human annotation needs. However, its high computational requirements may pose scalability challenges.

2.2 Polish-centric LLMs

While significant research has focused on English-language models, building Polish-centric LLMs remains underexplored. Recent models like Qra[6], Krakowiak (Ruciński, 2023), and Bielik (Ociepa et al., 2024) are fine-tuned in a supervised way from models LLaMa and Mistral but struggle with domain-specific applications. PLLuM[7], that continued pretraining on Polish data shows promising results. However, it is still under development and may be open-sourced in the future, though it is not currently available for broader research and application.

3 Approach & Methodology

Data Preprocessing To lay the groundwork for the MiNI-RAG chatbot, we initiated a systematic approach to data acquisition, processing, and embedding storage. Using the BeautifulSoup library, we scraped HTML data from all accessible pages of the Faculty of Mathematics and Information Science (MiNI) website[8], ensuring the inclusion of comprehensive and relevant content covering faculty-related FAQs. The scraped HTML files were then transformed into text documents. During this transformation, HTML-specific formatting, such as tags and inline scripts, was removed while retaining the most meaningful and informative content. The goal was to optimize text extraction by preserving semantically significant information and eliminating unnecessary noise.

Chunking In line with common practices in the literature (Gao et al., 2024), various chunking techniques are used to split the data into manageable pieces. Typically, documents are divided into fixed-size chunks based on a specific token limit (e.g., 100, 256, 512), as larger chunks capture more context but also introduce noise and increase computational costs. Smaller chunks reduce noise but may fail to convey complete context, often leading to sentence truncation. To address this, advanced techniques such as recursive splitting and sliding windows have been proposed to optimize chunking and enhance retrieval performance by merging globally related information across multiple retrieval steps.

Building on these approaches, our project evaluated and utilized **Text-Structure-Based Chunking** (Recursive Character Splitter) with a chunk size of 1000 characters and an overlap of 200 characters. This method is particularly advantageous as it not only allows for the preservation of context across adjacent segments but also ensures that overlapping sections bridged gaps, enabling coherent understanding during response generation. This method retained structural integrity by recursively splitting text at logical boundaries (e.g., paragraphs, sentences) until the chunks met size constraints. By maintaining a balance between chunk size and context retention, it provided an effective strategy for managing and analyzing textual data.

Embedding & Retrieval In our project, we focused on semantic search using dense retrieval methods. However, hybrid approaches that combine sparse and dense retrieval, as suggested in the literature, show promise for enhanced robustness and efficiency. In such systems, sparse retrievers

[3] `https://www.llamaindex.ai/`

[4] `https://www.trychroma.com/`

[5] `https://faiss.ai/`

[6] `https://huggingface.co/OPI-PG/Qra-13b`

[7] `https://pllum.org.pl/`

[8] `https://ww2.mini.pw.edu.pl/`

can improve zero-shot performance and handle rare entities, complementing dense models. This approach could be potentially useful for our use case, where keyword-based initial searches could refine dense retrieval results.

We compared Polish-centric embedding models in our solution (we took the best perfoming ones from MTEB[9] for Polish data):

- **gte-Qwen2-7B-instruct:** The latest in the gte (General Text Embedding) family, ranking No.5 on the MTEB benchmark for English and No.6 for Chinese as of the end of November 2024. (Yang et al., 2024)

- **gte-Qwen2-1.5B-instruct:** A smaller version of the gte-Qwen2-7B, built on the Qwen2-1.5B LLM, following the same training methodology as its larger counterpart.

- **jina-embeddings-v3:** A multilingual, multitask embedding model based on the Jina-XLM-RoBERTa architecture. It supports long input sequences and task-specific embedding generation using LoRA adapters.

In order to store chunks in a framework well-suited for this purpose, we chose Chroma as our vector store. Chroma was selected over FAISS and Milvus, as both of these frameworks are designed for large-scale architectures and big storage requirements, which exceeded the needs of our project. While FAISS utilizes GPU-based memory optimization for high-performance retrieval and Milvus excels in managing vast datasets with distributed architectures, our data size did not demand such advanced capabilities. Instead, Chroma stood out with its lightweight design, better integration with LLM applications, and ease of deployment, making it an optimal choice for our smaller-scale, semantic search-focused project.

Generation We tested Polish-centric LLMs and experimented with prompt templates to enhance response quality. As part of the project, we assessed how the Polish large language model Bielik (Ociepa et al., 2024) performed in the context of a chatbot integrated with a RAG pipeline. Additionally, we evaluated multilingual LLMs, such as Meta Llama-3.1-8B (Dubey et al., 2024), for their ability to handle Polish queries and generate relevant, coherent responses within this setup.

Evaluation Based on insights from the recent survey on Retrieval-Augmented Generation (RAG) evaluation methodologies (Yu et al., 2024), we concluded that the *RAGAS framework* (Es et al., 2023) offers the most comprehensive and suitable approach for our project. This framework provides both quantitative and qualitative metrics to assess precision, recall, relevance, and coherence, which align well with our evaluation needs.

To complement this, we created an evaluation dataset containing questions that we identified as intriguing, challenging, or frequently searched on the MiNI website. All the questions, along with their relevant contexts, were submitted to the well-known ChatGPT, and its answers after human refinement were saved as our reference for evaluation. This approach allowed us to assess the performance of our chatbot in realistic, user-centered scenarios.

We assessed the performance of our chatbot using metrics commonly applied in retrieval-augmented generation literature (Yu et al., 2024), such as:

- **Factual Correctness:** This metric compares and evaluates the factual accuracy of the generated response with a reference. The score ranges from 0 to 1, with higher values indicating better performance. The alignment between the response and the reference is assessed by first breaking down the response and reference into claims, then using natural language inference (NLI) to determine the factual overlap. Precision, recall, and F1 score are used to quantify factual overlap.

- **Context Precision:** This metric measures the proportion of relevant chunks in the retrieved contexts. It is calculated as the mean of precision@k for each chunk in the context. Precision@k is the ratio of relevant chunks at rank k to the total number of chunks at rank k.

$$\text{Context Precision} =$$
$$= \sum_{k=1}^{n} \frac{\text{Number of Relevant Nodes At Position K} \cdot r_k}{k \cdot \text{Total Number Of Relevant Nodes}}$$

- **Context Recall:** Measures how many of the relevant documents were successfully retrieved. Higher recall means fewer relevant

[9] https://huggingface.co/spaces/mteb/leaderboard

documents were missed. Recall is important to ensure no key information is overlooked. It is calculated using reference data for comparison.

$$\text{Context Recall} =$$

$$= \frac{|\text{GT Claims That Can Be Attributed To Context}|}{|\text{Claims In GT}|}$$

, where `GT` stands for Ground Truth

- **Faithfulness:** Measures the factual consistency of the generated answer against the given context. The answer is scaled to a (0,1) range, with higher values indicating better performance. The answer is considered faithful if all claims made in the answer can be inferred from the context. To calculate this, a set of claims from the generated answer is first identified, then each of these claims is cross-checked with the context to see if it can be inferred.

$$\text{Faithfulness Score} =$$

$$= \frac{|\text{Claims In The Answer That Can Be Inferred From Context}|}{|\text{Claims In The Generated Answer}|}$$

Tools & Equipment

- **Hardware:** We utilized Eden, the university's GPU cluster, to provide additional computational power. Additionally, we developed the foundation of our REST API, which included endpoints for models and embedders.

- **Software:** We used Python for scraping and preprocessing data from the MiNI webpage and for integrating the model. Additionally, we utilized LangChain for RAG pipeline integration, Hugging Face Transformers embedding models, vLLM for LLM deployment for generation part, Chroma for vector storage, FastAPI for deploying the embedding model as an API, and Streamlit for creating the user interface.

Described architecture can be presented using a diagram illustrated in the Figure 2:

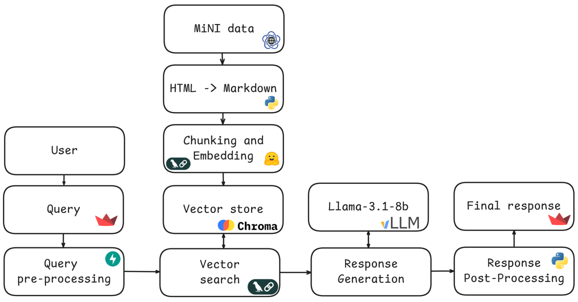


Figure 2: Architecture diagram.

4 Experiments

The experiments were conducted to evaluate and compare the performance of various embedding models integrated within the RAG framework, using the Bielik model as a baseline. Three Polish-centric embedding models selected based on the MTEB leaderboard, `gte-Qwen2-7B-instruct`, `gte-Qwen2-1.5B-instruct`, and `jina-embeddings-v3`, were assessed across four key evaluation metrics: Context Recall, Context Precision, Factual Correctness, and Faithfulness. For generation model, two configurations were tested: one with default greedy decoding and the other with Min-p sampling with temperature (Nguyen et al., 2024). (`{"temperature": 0.2, "min_p": 0.1}` parameters in vLLM). Evaluation results are summarized in Table 1.

All embedding and generation models were deployed on machines with GPU support to ensure efficient processing. Embedding models were hosted as REST APIs using the FastAPI framework, while generation models were served using the vLLM library. The vLLM library [10] is a cutting-edge inference and serving engine for large language models (LLMs). It provides an HTTP server that implements OpenAI's Completions [11] and Chat [12] APIs, ensuring seamless integration and reliable performance throughout the evaluation.

Significant effort was devoted to prompt engineering to optimize the performance of the retrieval and generation components. After testing various prompt formulations, the following prompt was finalized for use in the experiments:

[10] https://docs.vllm.ai/en/latest/index.html

[11] https://platform.openai.com/docs/api-reference/completions

[12] https://platform.openai.com/docs/api-reference/chat

Prompt

Jesteś pomocnym asystentem AI odpowiedzialnym za pomoc studentom w ich pytaniach związanych ze studiowaniem na wydziale MiNI (Matematyki i Nauk Informacyjnych) i Politechnice Warszawskiej.
Odpowiedz na pytanie użytkownika na podstawie pobranych fragmentów, a jeśli nie jesteś w stanie tego zrobić, poinformuj, że przy obecnej wiedzy nie jesteś w stanie odpowiedzieć na pytanie.
Pytanie użytkownika: {user_question}

Pomocnicza wiedza:
{chunks}

Prompt was designed to ensure that responses were contextually accurate, leveraging retrieved knowledge chunks effectively. The integration of this prompt into the RAG pipeline provided a consistent framework for evaluating the system's performance across all tested embedding models.

The evaluation results reveal that different models excel in specific metrics depending on the presence of sampling during decoding. The `gte-Qwen2-1.5B-instruct` model outperforms others in context recall and factual correctness, with its recall significantly improving from 0.5243 to 0.6895 when sampling is used. This model also achieves the highest factual correctness score (0.65) under the same condition. On the other hand, `jina-embeddings-v3` shows notable improvements in context precision and faithfulness when `temperature` and `min_p` are modified for generation, with its precision climbing from 0.4356 to 0.7401 and faithfulness reaching 0.6562. These results highlight the impact of fine-tuning parameters on retrieval accuracy and consistency. However, the `gte-Qwen2-7B-instruct` model exhibited relatively lower faithfulness with the use of additional params, and `jina-embeddings-v3` demonstrated the lowest recall. Overall, these findings suggest that model performance can be significantly influenced by the use of additional parameters such as `temperature` and `min_p`, with further optimization possible through fine-tuning and larger datasets.

Next, we compared the performance of two generation models `Llama-3.1-8B-Instruct` and `Bielik-11B-v2.3-Instruct` across the same evaluation metrics: Context Recall, Context Precision, Factual Correctness, and Faithfulness. Similarly to the previous experiment, each model was tested both with greedy decoding and Min-p sampling, to assess the impact of fine-tuning on model performance. `Llama-3.1-8B-Instruct` demonstrated notable improvements in Context Recall and Precision when additional params were applied, while `Bielik-11B-v2.3-Instruct` showed relatively consistent performance across metrics, with a slight increase in Recall and Faithfulness with modified `temperature` and Min-p sampling. Based on these results presented in Table 2, and the fact that Llama often returned responses in mixed polish-english language, we chose to utilize Bielik in the final architecture.

After an extensive evaluation process, we can address all the hypothesis questions formulated at the beginning of the project. First, `Qwen 1.5B` embedding model yielded the best performance for Polish university-related text data in retrieval tasks, providing the most relevant and coherent responses. We also found that prompt engineering significantly improved performance—by iterating on prompt designs, we saw progressively more accurate and contextually relevant responses. Regarding Polish LLMs, `Bielik` outperformed multilingual models like `Llama-3.1-8B` when handling Polish-language queries, showing superior relevance and accuracy. As for the effectiveness of embedding models identified as "best" on the Massive Text Embedding Benchmark (MTEB), we leave it to the reader to judge, though we believe that combining top-performing models in a hybrid approach may yield the best results. Finally, our results demonstrate that it is indeed possible to create a reliable question-answering system for Polish data using current open-source models and state-of-the-art RAG techniques, delivering satisfactory answers to user queries in real-world scenarios.

We developed a Streamlit application that utilizes the `gte-Qwen2-1.5B` and `Bielik-11B-v2.3-Instruct` models, that aims to integrate these models into a practical solution and evaluate their effectiveness in handling user queries, including follow-up questions.

In the Appendix B, the screenshots from the Streamlit app illustrate how the models respond to given queries, demonstrating the ability to maintain context, provide relevant answers, and adapt to new questions within a conversation.

Embedding Model	Sampling	Context Recall	Context Precision	Factual Correctness	Faithfulness
gte-Qwen2-7B-instruct	-	0.4211	0.5282	0.4000	0.5833
	+	0.3567	0.5762	0.5000	0.4286
gte-Qwen2-1.5B-instruct	-	0.5243	0.7194	-	0.6500
	+	**0.6895**	0.5024	**0.5850**	0.5303
jina-embeddings-v3	-	0.2958	0.4356	-	0.5407
	+	0.4398	**0.7401**	0.3350	**0.6562**

Table 1: Comparison of Embedding Models with `Bielik-11B-v2.3-Instruct`. Bold indicates the highest score, and underlined values indicate the second-highest score.

Model	Sampling	Context Recall	Context Precision	Factual Correctness	Faithfulness
Llama-3.1-8B-Instruct	-	0.3304	0.6152	0.3350	0.5093
	+	0.5702	0.6744	-	0.2847
Bielik-11B-v2.3-Instruct	-	0.5243	**0.7194**	-	**0.6500**
	+	**0.6895**	0.5024	**0.5850**	0.5303

Table 2: Comparison of Generation Models. Bold indicates the highest score, and underlined values indicate the second-highest score.

5 Conclusion

In this project, we developed a Polish-language chatbot using a Retrieval-Augmented Generation (RAG) framework to provide context-aware answers to frequently asked questions from students. The system utilized data scraped from the Faculty of Mathematics and Information Science website, as well as additional university-related documents, addressing the challenges of extracting and organizing domain-specific information from unstructured HTML data.

We applied advanced techniques such as Text-Structure-Based Chunking for efficient document splitting, and used dense retrieval methods with Polish-centric embedding models, including `gte-Qwen2-7B-instruct`, `gte-Qwen2-1.5B-instruct`, and `jina-embeddings-v3`. Chroma was selected as the vector store due to its lightweight design and integration capabilities.

For the generation component, we evaluated both Polish-optimized models, such as `Bielik-11B-v2.3-Instruct`, and multilingual models like Meta `Llama-3.1-8B-Instruct`, using prompt templates to enhance response quality. Performance was evaluated using the *RAGAS framework*, assessing metrics like factual correctness, context precision, recall, and faithfulness, and comparing results to reference answers from ChatGPT.

For future work, we would like to expand our dataset to cover all data from the PW domain, including the LEON platform, USOS, and Teams, to ensure comprehensive answers to all student-related queries. Additionally, we found collaboration with the team that developed RAG for lecture-related questions very prospective, that would enhance the model's capabilities even further. This collaboration would help ensure that the chatbot can provide more detailed, domain-specific information and handle a wider range of student inquiries. Furthermore, we believe it would be beneficial to consider conducting a human evaluation or user study to better understand the chatbot's effectiveness and gather valuable insights for potential refinements.

6 Contribution

Person	Contribution Area
Mikołaj Gałkowski	Requirements collection, Data Scrapping, API setup, UI, Report
Mikołaj Piórczyński	Requirements collection, Chunking, Vector Storage Setup, UI, Report
Julia Przybytniowska	Requirements collection, Data preprocessing, Evaluation, UI, Report

Table 3: Work Division

References

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. 2023. Ragas: Automated evaluation of retrieval augmented generation. *arXiv preprint arXiv:2309.15217*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. Nv-embed: Improved techniques for training llms as generalist embedding models. *arXiv preprint arXiv:2405.17428*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

Subash Neupane, Elias Hossain, Jason Keith, Himanshu Tripathi, Farbod Ghiasi, Noorbakhsh Amiri Golilarz, Amin Amirlatifi, Sudip Mittal, and Shahram Rahimi. 2024. From questions to insightful answers: Building an informed chatbot for university resources. *arXiv preprint arXiv:2405.08120*.

Minh Nguyen, Andrew Baker, Clement Neo, Allen Roush, Andreas Kirsch, and Ravid Shwartz-Ziv. 2024. Turning up the heat: Min-p sampling for creative and coherent llm outputs. *arXiv preprint arXiv:2407.01082*.

Krzysztof Ociepa, Łukasz Flis, Krzysztof Wróbel, Adrian Gwoździej, and Remigiusz Kinas. 2024. Bielik 7b v0.1: A polish language model – development, insights, and evaluation. *arXiv preprint arXiv:2410.18565*.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Szymon Ruciński. 2023. Krakowiak-v2-7b. `https://huggingface.co/szymonrucinski/krakowiak-v2-7b/`.

Jon Saad-Falcon, Omar Khattab, Christopher Potts, and Matei Zaharia. 2023. Ares: An automated evaluation framework for retrieval-augmented generation systems. *arXiv preprint arXiv:2311.09476*.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Hao Yu, Aoran Gan, Kai Zhang, Shiwei Tong, Qi Liu, and Zhaofeng Liu. 2024. Evaluation of retrieval-augmented generation: A survey.

A Reproducibility

A.1 Code

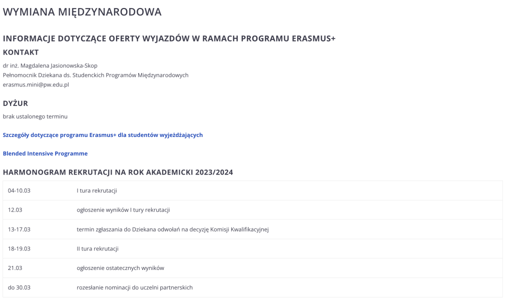
We have included all the necessary code and detailed instructions in the `README` file to set up the environment, run the project, and reproduce the results.

A.2 Test cases

We provide exemplar test cases in the table below, used in our study to systematically evaluate the performance of the *MiNI* RAG system. The complete test set is included in the published code repository.

Question	**Answer**
Kiedy jest otwarty dziekanat na MiNI?	Dziekanat na Wydziale MiNI Politechniki Warszawskiej jest otwarty w następujących godzinach: • Poniedziałek, Wtorek, Czwartek, Piątek: 11:00–14:00 • Środa: Zamknięte
Do kiedy powinno się wgrać do APD napisaną pracę magisterską ze studiów kierunku 'Data Science' i do kiedy jest możliwość przedłużenia terminu?	• Dla studiów 3-semestralnych: termin wgrania pracy do systemu APD to 15 września, z możliwością przedłużenia do 15 grudnia. • Dla studiów 4-semestralnych: termin wgrania pracy również wynosi 15 września, z możliwością przedłużenia do 15 grudnia.
Kto jest przewodniczącym komisji egzaminu dyplomowego dla kierunku Inżynieria i Analiza Danych?	Przewodniczącym komisji egzaminu dyplomowego dla kierunku Inżynieria i Analiza Danych jest dr hab. inż. Marek Gągolewski, prof. PW.
Czym zajmuje się ośrodek badań dla biznesu?	Ośrodek Badań dla Biznesu na Wydziale MiNI PW zajmuje się współpracą z przedsiębiorstwami, realizując projekty matematyczno-informatyczne, wspierając innowacje oraz łącząc studentów z biznesem. Szczegółowe informacje można znaleźć na stronie internetowej Ośrodka: `obb.mini.pw.edu.pl`.
Kto jest obecnym dziekanem wydziału?	Obecnym dziekanem Wydziału Matematyki i Nauk Informacyjnych (MiNI) Politechniki Warszawskiej jest prof. dr hab. Grzegorz Świątek.
Kto pełni rolę prodziekana do spraw studenckich?	Prodziekanem ds. studenckich Wydziału MiNI PW jest dr hab. inż. Agata Pilitowska.
Gdzie mogę znaleźć osobę z samorządu uczniowskiego, najlepiej przewodniczącego, na uczelni?	Przewodniczącego Samorządu Wydziału MiNI, Łukasza Grabarskiego, można znaleźć w biurze samorządu (pokój 45 'nogi' budynku) lub kontaktować się przez e-mail: MiNI@samorzad.pw.edu.pl. Więcej informacji dostępne jest na stronie Facebook: `https://www.facebook.com/wrsminipw/`.
Ile wynosi kwota stypendium rektora?	Wysokość stypendium rektora wynosi 850 zł.
Do kiedy trwa pierwsza tura rekrutacji do programu Erasmus+?	W 2024 roku pierwsza tura rekrutacji do programu Erasmus+ trwała od 4 do 10 marca.
Ile wynosi opłata za wznowienie studiów na MiNI?	Opłata za wznowienie studiów na Wydziale Matematyki i Nauk Informacyjnych wynosi 80 zł.

B Interface



WYMIANA MIĘDZYNARODOWA

INFORMACJE DOTYCZĄCE OFERTY WYJAZDÓW W RAMACH PROGRAMU ERASMUS+

KONTAKT

dr inż. Magdalena Jasionowska-Skop
Pełnomocnik Dziekana ds. Studenckich Programów Międzynarodowych
erasmus.mini@pw.edu.pl

DYŻUR

brak ustalonego terminu

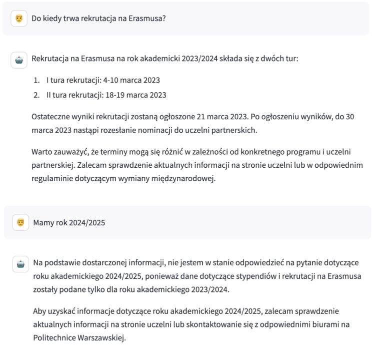
Szczegóły dotyczące programu Erasmus+ dla studentów wyjeżdżających

Blended Intensive Programme

HARMONOGRAM REKRUTACJI NA ROK AKADEMICKI 2023/2024

04-10.03	I tura rekrutacji
12.03	ogłoszenie wyników I tury rekrutacji
13-17.03	termin zgłaszania do Dziekana odwołań na decyzję Komisji Kwalifikacyjnej
18-19.03	II tura rekrutacji
21.03	ogłoszenie ostatecznych wyników
do 30.03	rozesłanie nominacji do uczelni partnerskich

(a) Reference from MiNI website.



Do kiedy trwa rekrutacja na Erasmusa?

Rekrutacja na Erasmusa na rok akademicki 2023/2024 składa się z dwóch tur:

1. I tura rekrutacji: 4-10 marca 2023
2. II tura rekrutacji: 18-19 marca 2023

Ostateczne wyniki rekrutacji zostaną ogłoszone 21 marca 2023. Po ogłoszeniu wyników, do 30 marca 2023 nastąpi rozesłanie nominacji do uczelni partnerskich.

Warto zauważyć, że terminy mogą się różnić w zależności od konkretnego programu i uczelni partnerskiej. Zalecam sprawdzenie aktualnych informacji na stronie uczelni lub w odpowiednim regulaminie dotyczącym wymiany międzynarodowej.

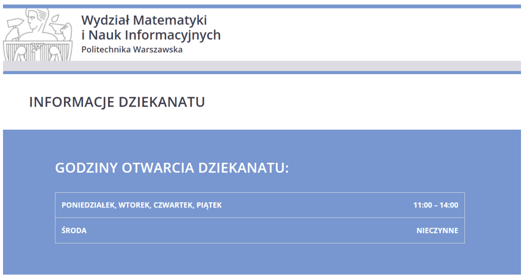
Mamy rok 2024/2025

Na podstawie dostarczonej informacji, nie jestem w stanie odpowiedzieć na pytanie dotyczące roku akademickiego 2024/2025, ponieważ dane dotyczące stypendiów i rekrutacji na Erasmusa zostały podane tylko dla roku akademickiego 2023/2024.

Aby uzyskać informacje dotyczące roku akademickiego 2024/2025, zalecam sprawdzenie aktualnych informacji na stronie uczelni lub skontaktowanie się z odpowiednimi biurami na Politechnice Warszawskiej.

(b) ChatBot's response.

Figure 3: Example interaction showcasing the *MiNI* RAG answering a query about ERASMUS+ recruitment



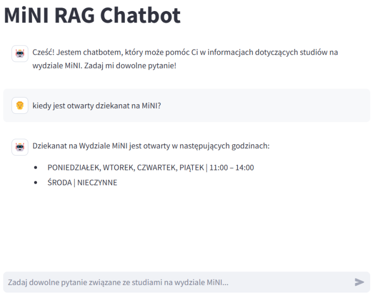
Wydział Matematyki
i Nauk Informacyjnych
Politechnika Warszawska

INFORMACJE DZIEKANATU

GODZINY OTWARCIA DZIEKANATU:

PONIEDZIAŁEK, WTOREK, CZWARTEK, PIĄTEK	11:00 – 14:00
ŚRODA	NIECZYNNE

(a) Reference from MiNI website.



MiNI RAG Chatbot

Cześć! Jestem chatbotem, który może pomóc Ci w informacjach dotyczących studiów na wydziale MiNI. Zadaj mi dowolne pytanie!

kiedy jest otwarty dziekanat na MiNI?

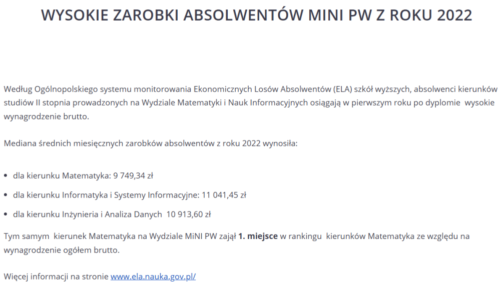
Dziekanat na Wydziale MiNI jest otwarty w następujących godzinach:

- PONIEDZIAŁEK, WTOREK, CZWARTEK, PIĄTEK | 11:00 – 14:00
- ŚRODA | NIECZYNNE

Zadaj dowolne pytanie związane ze studiami na wydziale MiNI...

(b) ChatBot's response.

Figure 4: Example interaction showcasing the *MiNI* RAG answering a query about the dean's office open hours.



WYSOKIE ZAROBKI ABSOLWENTÓW MINI PW Z ROKU 2022

Według Ogólnopolskiego systemu monitorowania Ekonomicznych Losów Absolwentów (ELA) szkół wyższych, absolwenci kierunków studiów II stopnia prowadzonych na Wydziale Matematyki i Nauk Informacyjnych osiągają w pierwszym roku po dyplomie wysokie wynagrodzenie brutto.

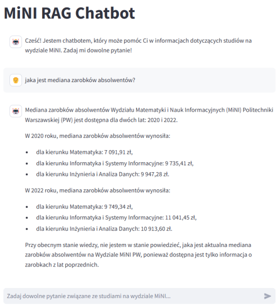
Mediana średnich miesięcznych zarobków absolwentów z roku 2022 wynosiła:

- dla kierunku Matematyka: 9 749,34 zł
- dla kierunku Informatyka i Systemy Informacyjne: 11 041,45 zł
- dla kierunku Inżynieria i Analiza Danych 10 913,60 zł

Tym samym kierunek Matematyka na Wydziale MiNI PW zajął **1. miejsce** w rankingu kierunków Matematyka ze względu na wynagrodzenie ogółem brutto.

Więcej informacji na stronie www.ela.nauka.gov.pl/

(a) Reference from MiNI website.



MiNI RAG Chatbot

Cześć! Jestem chatbotem, który może pomóc Ci w informacjach dotyczących studiów na wydziale MiNI. Zadaj mi dowolne pytanie!

jaka jest mediana zarobków absolwentów?

Mediana zarobków absolwentów Wydziału Matematyki i Nauk Informacyjnych (MiNI) Politechniki Warszawskiej (PW) jest dostępna dla dwóch lat: 2020 i 2022.

W 2020 roku, mediana zarobków absolwentów wynosiła:

- dla kierunku Matematyka: 7 091,91 zł,
- dla kierunku Informatyka i Systemy Informacyjne: 9 735,41 zł,
- dla kierunku Inżynieria i Analiza Danych: 9 947,28 zł.

W 2022 roku, mediana zarobków absolwentów wynosiła:

- dla kierunku Matematyka: 9 749,34 zł,
- dla kierunku Informatyka i Systemy Informacyjne: 11 041,45 zł,
- dla kierunku Inżynieria i Analiza Danych: 10 913,60 zł.

Przy obecnym stanie wiedzy, nie jestem w stanie powiedzieć, jaka jest aktualna mediana zarobków absolwentów na Wydziale MiNI PW, ponieważ dostępna jest tylko informacja o zarobkach z lat poprzednich.

Zadaj dowolne pytanie związane ze studiami na wydziale MiNI...

(b) ChatBot's response.

Figure 5: Example interaction showcasing the *MiNI* RAG answering a query about the graduate salaries.

C Reviews

C.1 Hubert Bujakowski, Jan Kruszewski, Łukasz Tomaszewski group

We greatly appreciate the reviewers' feedback on our project and the constructive suggestions provided. Below, we address the key points raised:

Weaknesses and Addressed Concerns

1. **Misleading Naming Convention (Chatbot)**
   Reviewer Concern: The current naming convention is misleading, as the PoC lacks a chat history feature and does not retain conversation context.
   Response: This observation was valid for the PoC stage. In the final project deliverable, we implemented a history feature that enables multi-turn interactions and supports follow-up questions, making the system more akin to a conversational chatbot.

2. **Reliance on Self-Written REST APIs**
   Reviewer Concern: Custom REST APIs add boilerplate code, reducing maintainability.
   Response: We respectfully disagree with this concern for the following reasons:

   (a) Our REST API is minimal, consisting solely of a single `/embeddings` endpoint, with the API code totaling fewer than 40 lines.

   (b) Using the FastAPI framework allowed us to seamlessly deploy the API on a machine with GPU access, simplifying the integration and deployment process.

3. **No Quantization of the Model**
   Reviewer Concern: Model quantization is not implemented, potentially leading to high resource usage.
   Response: Quantization was unnecessary in our case, as the generation models used in our system fit within our available GPU resources and operate with low latency, meeting performance requirements effectively.

Additional Suggestions

1. **Use of Newer Models (e.g., Llama 3.2)**
   Reviewer Suggestion: Consider using newer models that may yield better results with fewer parameters.
   Response: While we acknowledge this suggestion, we opted to continue using our current generation models due to their larger parameter sizes, which yield superior results compared to Llama 3.2. Additionally, Llama 3.2's smaller models (1B and 3B) do not meet our performance needs, while its larger models (11B and 90B) include multimodal capabilities that are unnecessary for our project.

2. **Translation of Data to English**
   Reviewer Suggestion: Translating scraped data to English could improve performance with English-centric embeddings and LLMs.
   Response: While translation could enhance model performance due to better support in English-based tools, it would risk compromising the **accuracy of culturally nuanced content** and add unnecessary preprocessing complexity. Our research focused on evaluating the system's performance on Polish-language data to ensure relevance to the target audience.

3. **Periodic Scraping of Data**
   Reviewer Suggestion: Ensure periodic data scraping to maintain up-to-date information.
   Response: We agree that periodic data scraping is essential. In a production scenario, we would implement a dedicated data pipeline for periodic scraping, chunking, and embedding updates into the vector store. Tools such as **Airflow** would be used for orchestration to ensure a robust and automated process.

C.2 Tomasz Siudalski, Weronika Plichta, Michał Taczała group

We greatly appreciate the reviewers' detailed feedback and constructive suggestions. Below, we address the key points raised:

Weaknesses and Addressed Concerns

1. **Limited Evaluation Metrics** *Reviewer Concern*: The evaluation is limited to predefined metrics from the RAGAS framework. Expanding the evaluation to include real user feedback would provide more practical insights.
   Response: We agree with this suggestion and acknowledge the importance of incorporating human feedback. In production scenario, we wouldconduct user studies involving both students and faculty members to evaluate the chatbot's practical utility and user satisfaction.

2. **Reliance on a Single Data Source** *Reviewer Concern*: Dependence on web-scraped data from the MiNI website limits the chatbot's knowledge base and might reduce accuracy due to outdated information.
   Response: This is a valid observation. In production scenario, we would propose integrating additional data sources such as Microsoft Teams, the Leon platform and PW website. These sources will enhance the chatbot's knowledge base and improve its ability to handle diverse and less common queries. Moreover, incorporating a periodic scraping pipeline will ensure that the knowledge base remains up-to-date.

Additional Suggestions

1. **Comparison of Retrieval and Embedding Configurations** *Reviewer Suggestion*: Conduct comparison studies on different retrieval and embedding configurations.
   Response: We appreciate this suggestion and we performed an in-depth comparative analysis of retrieval and embedding configurations for final project deliverable to identify the most effective combinations for Polish-language data.