LAIwyer
Project Report for NLP Course, Winter 2025

Adam Majczyk
Warsaw University of Technology
adam.majczyk.stud@pw.edu.pl
Szymon Matuszewski
Warsaw University of Technology
szymon.matuszewski.stud@pw.edu.pl
supervisor: Anna Wróblewska
Warsaw University of Technology
anna.wroblewska1@pw.edu.pl

Abstract

The LAIwyer project addresses a significant gap in the legal domain: the lack of a fast, accessible, and reliable AI tool tailored to the complexities of Polish law. While general-purpose AI systems exist, they often fail to deliver precise legal guidance due to language nuances, jurisdiction-specific requirements, and the need for reliable citations. This project introduces a novel Retrieval-Augmented Generation (RAG) system capable of answering legal questions in Polish with high accuracy and contextual understanding.

1 Introduction

1.1 Problem Proposal

In this section the problem that the project will solved is described. It is defined from the user's perspective. The end goal of the task is stated.

1.2 Problem Definition

Currently, a common problem mentioned by lawyers is the task of collecting information. They spend many hours looking for pieces of law that may help them in their endeavours. Aside from that, lawyers often have to analyse already passed, related legal sentences and compare them with their case. That may also be an arduous task. Another important aspect of the task is the fact, that law changes continuously. Therefore it is important that the lawyer has access to the up-to date knowledge. The problem may therefore be summarised as:

1. looking for related pieces of law,
2. analysing them and drawing conclusions,
3. constantly updating the knowledge base.

1.3 Aim

Based on the stated problem definition the aim is to develop a solution that supports lawyers in looking for important pieces of law, sentences and analysing them. It should also be regularly updated with the newest legal information.

2 Literature

Artificial Intelligence has steadily influenced the legal domain, beginning with rule-based expert systems in the 1980s. These systems aimed to automate routine legal tasks like document review and compliance checks. Over the years, AI capabilities evolved, enabling more sophisticated applications such as LLM-based agents configured for delivering law tasks such as court views generation or legal judgement prediction. Despite this progress, challenges like "hallucinations" (false information confidently presented) and biases in large language models (LLMs) persist, especially in high-stakes contexts like litigation and policy drafting.

LEGALBENCH: A COLLABORATIVELY BUILT BENCHMARK FOR MEASURING LE-

GAL REASONING IN LARGE LANGUAGE MODELS [1]

LegalBench is a comprehensive benchmark designed to evaluate the legal reasoning abilities of LLMs. It comprises 162 tasks that encompass diverse legal reasoning scenarios, from rule recall to conclusion generation. Built collaboratively by legal professionals, it assesses the practical utility of AI in legal reasoning without aiming to replace human expertise. LegalBench has been instrumental in identifying the strengths and weaknesses of LLMs in handling domain-specific tasks. Most importantly this benchmark summarises both open-source and commercial models.

LLM	Issue	Rule	Conclusion	Interpretation	Rhetorical
GPT-4	82.9	59.2	89.9	75.2	79.4
GPT-3.5	60.9	46.3	78.0	72.6	66.7
Claude-1	58.1	57.7	79.5	67.4	68.9
Flan-T5-XXL	66.0	36.0	63.3	64.4	70.7
LLaMA-2-13B	50.2	37.7	59.3	50.9	54.9
OPT-13B	52.9	28.4	45.0	45.1	43.2
Vicuna-13B-16k	34.3	29.4	34.9	40.0	30.1
WizardLM-13B	24.1	38.0	62.6	50.9	59.8
BLOOM-7B	50.6	24.1	47.2	42.8	40.7
Falcon-7B-Instruct	51.3	25.0	52.9	46.3	44.2
Incite-7B-Base	50.1	36.2	47.0	46.6	40.9
Incite-7B-Instruct	54.9	35.6	52.9	54.5	45.1
LLaMA-2-7B	50.2	33.7	55.9	47.7	47.7
MPT-7B-8k-Instruct	54.3	25.9	48.9	42.1	44.3
OPT-6.7B	52.4	23.1	46.3	48.9	42.2
Vicuna-7B-16k	3.9	14.0	35.6	28.1	14.0
BLOOM-3B	47.4	20.6	45.0	45.0	36.4
Flan-T5-XL	56.8	31.7	52.1	51.4	67.4
Incite-3B-Instruct	51.1	26.9	47.4	*49.6*	40.2
OPT-2.7B	53.7	22.2	46.0	44.4	39.8

Table 1: Table from [1]: *Average performance for each LLM over the different LEGALBENCH categories. The first block of rows corresponds to large commercial models, the second block corresponds to models in the 11B-13B range, the third block corresponds to models in the 6B-7B range, and the final block corresponds to models in the 2B-3B range. The columns correspond to (in order): issue-spotting, rule-recall, rule-conclusion, interpretation, and rhetorical-understanding. For each class of models (large, 13B, 7B, and 3B), the best performing model in each category of reasoning is underlined.*

The results from Table 1 indicate that commercial models are undeniably the best in terms of understanding law logics. Open-source models perform considerably worse.

LegalBench-RAG: A Benchmark for Retrieval-Augmented Generation in the Legal Domain [2]

LegalBench-RAG extends the capabilities of LegalBench by focusing on Retrieval-Augmented Generation (RAG). While LegalBench evaluates generative reasoning, LegalBench-RAG emphasizes the retrieval component, essential for legal-specific RAG systems. Its benchmark includes 6,858 query-answer pairs derived from real-world legal documents, highlighting the importance of precision in retrieving relevant legal text segments. By addressing gaps in context retrieval, LegalBench-RAG ensures higher accuracy and utility in RAG pipelines.

Evaluating AI for Law: Bridging the Gap with Open-Source Solutions [3]

The paper highlights the risks of overreliance on general-purpose AI tools. These obstacles are described as follow:

- Hallucinations and confident regurgitation,
- Lack of diversity in generated responses,
- Linear Reasoning,
- Lack of temporal dimension in legal applications,
- Static training databases.

It advocates for open-source legal AI models to improve accessibility, accuracy, and narrative diversity. Notably, the study introduces benchmarks for assessing bias, fact-checking, and legal reasoning abilities, which are essential for deploying AI in high-stakes legal scenarios. The authors used 2 commercial models (*gpt-3.5-turbo* and *gpt-4-turbo-1106*) and one ope-source model (*mixtral-8x7B*).

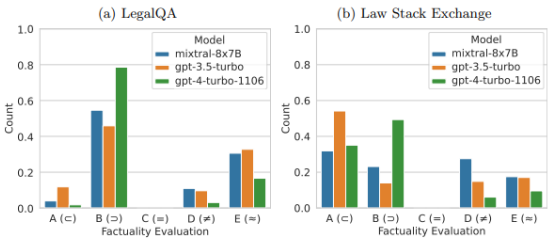

Figure 1: Plots from [3]. Automatic evaluation results for two different datasets.

In the Figure 1 we can see that once again commercial models outperforms open-source options. Sadly, *mixtral-8x7B* is inaccurate quite often (categories D and E on the plots).

LAWGPT: A Chinese Legal Knowledge-Enhanced Large Language Model [4]

LawGPT is a Chinese open-source model specifically designed for legal applications, with significant enhancements over general-purpose LLMs. It incorporates legal-oriented pre-training using a large corpus of legal documents and supervised fine-tuning with a knowledge-driven dataset. Experimental results show that *LawGPT* outperforms other open-source models like *LLaMA* in downstream legal tasks, including legal question answering and document analysis. Its focus on privacy and adaptability makes it a preferred choice for jurisdictions requiring data confidentiality.

Models	case analysis	criminal damages calculation	charge prediction
GPT-3.5 Turbo	27.4	61.2	35.5
GPT-4	48.6	77.6	42.0
LLaMA	0.2	14.4	7.0
LaWGPT	**6.2**	**15.4**	**15.7**

Table 2: Shortened table from [4]. Performance comparison between LAWGPT, proprietary models including GPT3.5 Turbo and GPT-4, and 7B open-source model LLaMA on the zero-shot setting. The best performance among LAWGPT and open-source models is in bold.

3 Tools

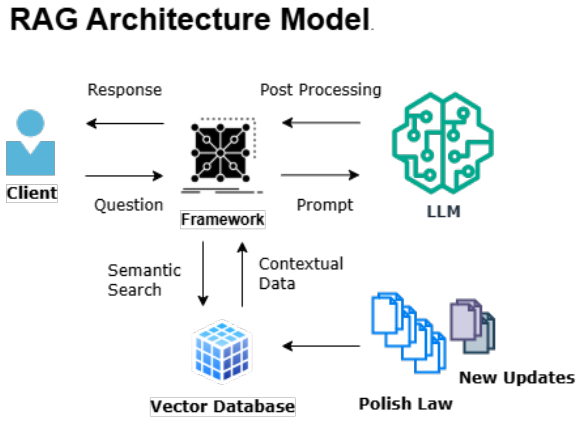

Figure 2: RAG Architecture Model which is planned to be implemented

Our project is based on the RAG Architecture Model 2. Hence, in we need tools to fulfill the requirements for modules:

1. Data Aquisition
2. Semantic Search
3. Vector Database
4. LLM
5. User Interface

Data Aquisition As a Data Aquisition modelu we will use *Python's scripts* using handy APIs for the Polish Law.

Semantic Search We need to use the embedding model for semantic search within the vector database. We have two parallel path which can be followed: *Azure Cloud* or *local solution*. Thus, we have checked multiple options described in table 3.

Model	Type	Parameters	Embedding Dimensions	Advantages
paraphrase-multilingual-MiniLM-L12-v2	Open-Source (Hugging Face)	118M	384	Highly effective for semantic similarity and retrieval tasks; Top-1 Trening within Similarity Search models with Polish on Hugging Face.
multi-qa-MiniLM-L6-cos-v1	Open-Source (Hugging Face)	33M	384	Lightweight and efficient; optimized for question-answering and document retrieval.
gte-multilingual-base	Open-Source (Hugging Face)	305M	768	Achieves state-of-the-art (SOTA) results in multi-lingual retrieval tasks and multi-task representation model evaluations when compared to models of similar size.
text-embedding-ada-002	Azure OpenAI	Proprietary	1,536	State-of-the-art embeddings; versatile across tasks like search, clustering, and recommendations.
text-embedding-3-large	Azure OpenAI	Proprietary	1,536	Superior accuracy for complex semantic tasks; supports nuanced understanding of text.
text-embedding-3-small	Azure OpenAI	Proprietary	1,536	Optimized for cost and latency; retains high-quality embedding for efficient deployment.

Table 3: Comparison of Open-Source and Azure OpenAI Embedding Models

Vector Database Firstly, we will attempt to create the vector store locally. Problems with memory may occur, then we will turn to the cloud-based vector store, which is very often an expensive way of storing the documents (*non-preferable*). Hence, considered vector databases are presented in Table 4.

Database	Type	Compatibility	Advantages
Chroma	Open-Source	Python API, LangChain	Easy-to-use for Retrieval-Augmented Generation (RAG); seamless integration with LangChain for building conversational AI applications; provides high-level abstractions for managing embeddings.
PGVector	Open-Source	PostgreSQL	Combines the power of a relational database with vector storage; allows efficient querying of embeddings while leveraging the robustness of PostgreSQL for transactional operations.
Qdrant	Open-Source	REST API, Python SDK, gRPC	Scalable and efficient; supports filtering and payload search for contextual queries; integrates easily into ML pipelines with support for hybrid queries.

Table 4: Comparison of Chroma, PGVector, and Qdrant Databases

LLM Described in section 4.

User Interface To fulfill compatibility and maintain the programming environment our default user interface will resist on *Streamlit* [5], which is an open-source *Python* framework to deliver interactive data apps including user-friendly chatbots and dashboards.

4 Models

In our task, we are focused on the **Polish** language. Unfortunately, the available literature predominantly tests models in other languages, primarily English. Therefore, when selecting open-source

models, we must concentrate on multilingual models or those specifically designed for the Polish language.

CohereForAI/aya-expanse-8b [6]

Aya Expanse, developed by CohereForAI, is an open-source large language model specifically designed to cater to a wide range of natural language processing tasks. The model, named Aya Expanse 8B, features **8 billion parameters** and is optimized for high-quality text understanding and generation. Aya excels in multilingual capabilities, making it a strong candidate for tasks involving diverse languages, **including Polish**. Its pretraining utilises a diverse corpus, enabling robust performance in tasks such as summarisation, question answering, and text classification. The model's availability on *Hugging Face* ensures accessibility and adaptability for various downstream applications.

It was introduced less than a month ago and gathered nearly **44,000 downloads** as of 20th of November 2024.

MaLA-LM/emma-500-llama2-7b [7]

EMMA-500 is a state-of-the-art multilingual language model designed to enhance language representation, particularly in low-resource languages. It **builds upon the Llama 2 7B architecture [8]** through continual pre-training, leveraging the MaLA Corpus, which encompasses over 500 languages and 74 billion tokens. This extensive training enables EMMA-500 to excel in tasks such as commonsense reasoning, machine translation, open-ended generation, and text classification. Notably, it outperforms other Llama 2-based models across diverse multilingual settings while maintaining robustness in specialized tasks.

The model supports **546 languages**, each with substantial training data exceeding 100,000 tokens. Its data mix includes a diverse array of texts from domains like code, books, and instructional materials. Key tasks for EMMA-500 encompass *commonsense reasoning, machine translation, text classification, natural language inference, code generation, and open-ended generation*.

The paper of the model was submitted *on 26th of September 2024* so it is an unexamined area of as well as Aya Expanse.

speakleash/Bielik-7B-v0.1 [9]

Bielik 7B v0.1 is a cutting-edge **Polish language model** developed as a collaborative effort between the open-science project *SpeakLeash and the ACK Cyfronet AGH High-Performance Computing center*. The model, built on the *Mistral 7B v0.1* foundation [10], has **7 billion parameters** and is specifically designed for high-performance Polish language processing tasks.

Key innovations in Bielik 7B include *Weighted Instruction Cross-Entropy Loss*, which balances the learning of different instruction types, and *an Adaptive Learning Rate mechanism* that dynamically adjusts training parameters to optimize performance. These techniques enable Bielik to excel in Polish-specific NLP benchmarks, such as the Open PL LLM Leaderboard and the Polish MT-Bench, achieving significant improvements in areas like reasoning and role-playing tasks.

The model leverages a diverse corpus of Polish texts combined with high-quality English data, resulting in a robust training dataset of **36 billion tokens**.

WARNING In case open-source models fail to meet the requirements for our task, we can consider transitioning to commercial models such as *ChatGPT-4* or *ChatGPT-3.5-turbo*.

5 Open Datasets

As of now, there are no curated, open datasets containing the Polish legal basis and sentences. There are databases such as the **Wolters Kluwer LEX** [11] database, they are however expensive and not open for free use.

Therefore scraping of official governmental websites is required.

5.1 Polish Constitution

The constitution of Poland is available at `https://trybunal.gov.pl/o-trybunale/akty-normatywne/konstytucja-rzeczypospolitej-polskiej`. It shall be scraped and stored. This will be done once without deliberation on constant updates, as changes to the constitution are an exception.

5.2 Legal Acts

The current acts are being published on the website of the Polish parliament: `https://isap.sej`

`m.gov.pl/`. The content of each act shall be scraped and stored. In the case of this data, since it new pieces of legal text appear regularly, a scraper that obtains the content periodically (e.g. daily) will be developed.

5.3 Sentences

The sentences passed in courts are published to: `https://orzeczenia.warszawa.so.gov.pl/`. Similarly to the legal acts, this shall be scraped and updated regularly, as new sentences appear daily.

6 Proof of Concept

The LAIwyer Proof of Concept is a convenient tool designed for users seeking advice based on **the Polish Constitution** [12], as of the version dated *2015-01-07*. The solution is built upon the integration of a **`pgvector`** [13] vector database, **the `paraphrase-multilingual-MiniLM-L12-v2` [14] embedding model**, and **the `Bielik-11B-v2.3-Instruct-GGUF:Q4_K_M` [9] language model**.

Considering the opinions of legal professionals, who consistently emphasize *the importance of using language models locally*, we opted to deploy the language model **locally** using the **`Ollama`** [15] environment.

This approach ensures greater control, security, and alignment with the specific needs of Polish legal applications.

The final user-facing application was developed using **the `Streamlit` [5] library in `Python`**.

The graphical view of the POC is presented in the Figure 3.

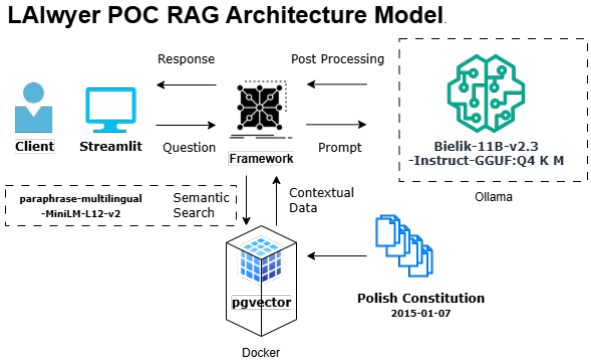

Figure 3: POC RAG Architecture Model which is now implemented as a result of the milestone.

6.1 Exploratory Data Analysis

In the Proof of Concept solution, we focused on **the Polish Constitution** to determine whether classical legal concepts in the Polish language could be effectively interpreted and cited by the language model.

To achieve this, we obtained the Polish Constitution in its **2015-01-07** version and parsed it using a custom parser. As a result, we produced a `.json` file named `parsed_constitution.json`, with the following structure:

```
[
    {
        "section": "I",
        "section_title":
        ↪ "RZECZPOSPOLITA",
        "article": "1",
        "text": "Rzeczpospolita Polska
        ↪ jest dobrem wspólnym
        ↪ wszystkich obywateli."
    },
    {
        "section": "I",
        "section_title":
        ↪ "RZECZPOSPOLITA",
        "article": "2",
        "text": "Rzeczpospolita Polska
        ↪ jest demokratycznym państwem
        ↪ prawnym,
            urzeczywistniajacym
            ↪ zasady
            ↪ sprawiedliwości
            ↪ spo␣lecznej."
    },
    {
        "section": "I",
        "section_title":
        ↪ "RZECZPOSPOLITA",
        "article": "3",
        "text": "Rzeczpospolita Polska
        ↪ jest państwem jednolitym."
    },
    ...
]
```

Next, we conducted Exploratory Data Analysis to verify the completeness of the parsed data—this was indeed the case. The Constitution consists of **13 sections** numbered in Roman numerals from I to XIII, and **243 articles**. Additionally, the section names, sorted alphabetically, are as follows:

- FINANSE PUBLICZNE
- ORGANY KONTROLI PAŃSTWOWEJ I OCHRONY PRAWA
- PREZYDENT RZECZYPOSPOLITEJ POLSKIEJ

- PRZEPISY PRZEJŚCIOWE I KOŃCOWE
- RADA MINISTRÓW I ADMINISTRACJA RZADOWA
- RZECZPOSPOLITA
- SAMORZAD TERYTORIALNY
- SEJM I SENAT
- STANY NADZWYCZAJNE
- SADY I TRYBUNAŁY
- WOLNOŚCI, PRAWA I OBOWIAZKI CZŁOWIEKA I OBYWATELA
- ZMIANA KONSTYTUCJI
- ŹRÓDŁA PRAWA

We also checked the number of articles per section to ensure that no sections were omitted. The results are shown in the Figure 4. Furthermore, we analysed the distribution of article lengths 5 to determine whether a chunking mechanism would be necessary before populating the vector database. It turns out that the articles have an average length of **411 characters**, with a median of **300**, so we concluded that chunking was not required.

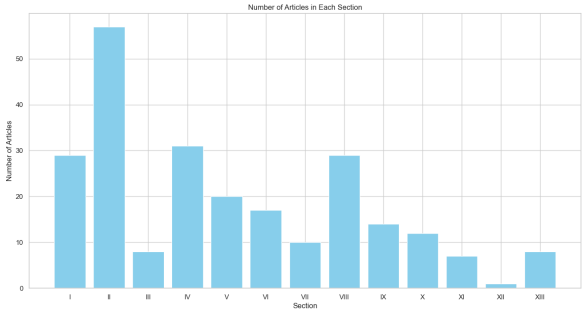

Figure 4: Number of Articles per each Section in the Polish Constitution.

6.2 Vector Database

For the vector database, we decided to use **pgvector**. This choice was based on its integration with `PostgreSQL` and its excellent compatibility with `Docker`.Additionally, we were attracted by its low latency in reads which are crucial in business solutions. Our goal was to ensure easy scalability and maintain the local nature of the solution.

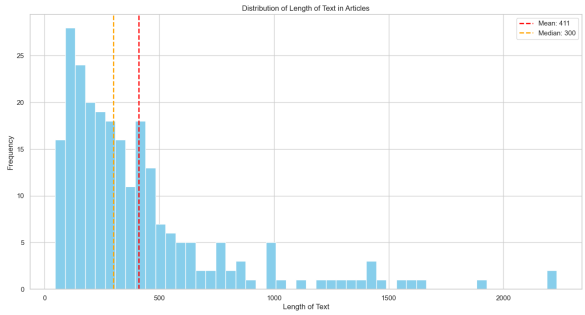

Figure 5: The distribution of the lengths of the articles in the Polish Constitution.

To implement your version of the database, you need to have `the Docker Desktop` application installed and use `a Docker image` provided in a `'.tar' file`. If you are using an application that communicates with the database, make sure the container is running.

Connection parameters:

- **Database Name:** `pgvector_nlp_db`
- **User:** `postgres`
- **Password:** `yourpassword`
- **Host:** `localhost`
- **Port:** `56434`

`pgvector` `Database Parameters`

1. **Embedding Model Name:** `sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2`
2. **Similarity Function:** Inner Product

$$\text{Similarity}(A, B) = \sum_{i=1}^{n} A_i \cdot B_i$$

, where

- A and B: The embedding vectors of two text entries.
- n: The dimensionality of the embedding vectors.
- A_i: The i-th component of vector A.
- B_i: The i-th component of vector B.
- $\cdot$: The multiplication operator.
- $\sum_{i=1}^{n}$: The summation over all dimensions of the embedding vectors.

3. **Number of Articles Returned in the Application:** 10

The inner product calculates the similarity between two embedding vectors, with higher values indicating greater similarity. This function is efficient and widely used in natural language processing tasks.

6.3 Model

6.3.1 Embedding model

`sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2` has been chosen as the embedding model because of its tight integration with `sentence-transformers` and hence, ease of use. It generates dense embedding vectors of dim **384** and is easily adapted for semantic search.

6.3.2 LLM model

`hf.co/speakleash/Bielik-11B-v2.3-Instruct-GGUF:Q4_K_M` has been chosen for two reasons.

1. Its good performance on the Polish language, especially as a Role-Playing model
2. This quantization was the largest that was viable given the GPU available

It is a 11B parameter model based on Mistral 7B v0.2 [10] (Bielik has additional layers, hence the additional 4B parameters). The quantized version used mixed precision GGUF quantization to Int4, reducing the size of the model from $\approx$ 21 GB (in FP16) to 6.72 GB. It supports a context length 32k, but contrary to the 7B version of Bielik it does not use Sliding Window Attention. Given a context length of that size, the usage of additional context from the Retrieval part of RAG and context from the previous parts of the conversation is viable.

6.4 Application

We decided to use the `Streamlit` library in `Python` due to its high-level design and excellent integration with language models. As a result, a web application was developed, allowing users to ask questions about the Polish Constitution. Additionally, *to make the experience more engaging and to produce more entertaining (and humorous) effects*, we decided to let the user choose one of seven constitutional expert personalities. These are:

- **Professor of Constitutional Law**
- **Priest**
- **Egocentric**
- **Silesian Speaker**
- **Rapper**
- **The Best Lawyer**
- **Shrek**

Each personality answers questions truthfully but in its own unique style.

We conducted beta tests with a lawyer who teaches the subject *Legal Risks and Compliance in ICT Industry*. The lawyer confirmed that the system answers his questions correctly and appreciated the humorous side of the available personalities.

REMARK: This solution is intended solely for Polish-speaking users due to the nature of the project. Conversations are conducted exclusively in Polish.

The application is presented on Figures 6 8 7 9.

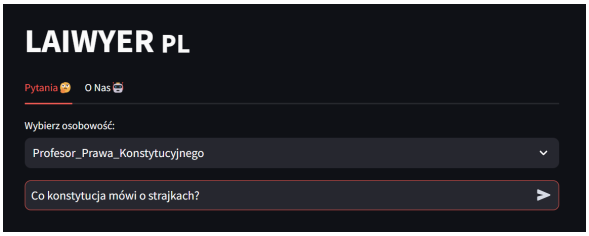

Figure 6: Main Page

6.5 Evaluation

To assess the effectiveness of our system in the context of answering *Polish constitutional law questions*, we conducted a comprehensive evaluation using a dataset of **41 questions**. Each question included multiple-choice answers **(A, B, or C)**, and the evaluation was performed under three distinct configurations:

- **Base Model (No Retrieval):** In this configuration, the system relied solely on the language model without any additional context retrieval.

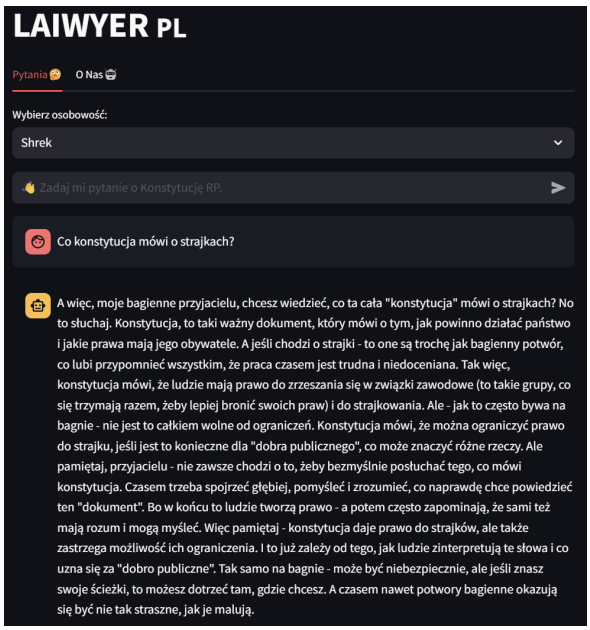

Figure 7: Answer of Shrek Persona

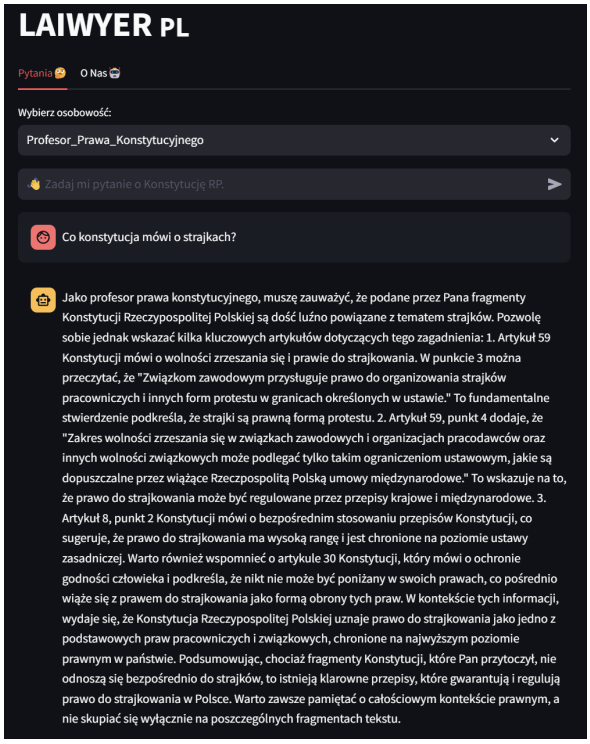

Figure 8: Answer of Professor Persona

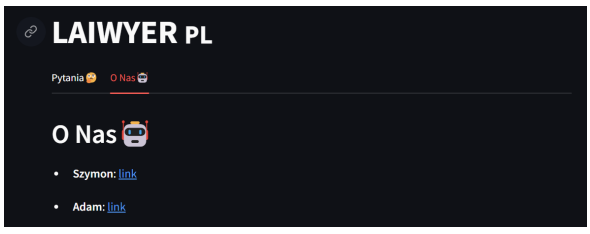

Figure 9: About Us Page

The model was prompted with the question and the possible answers, and it was required to select the correct option based solely on its pre-trained knowledge.

- **Retriever with Context:** Here, the system incorporated a retrieval component that fetched relevant articles from a vector database. The retrieval process was configured to fetch up to **20 articles**. The retrieved articles were processed and formatted as additional context for the model. This configuration aimed to evaluate how the inclusion of relevant external information impacted the accuracy of the answers.

- **Retriever with Reranker:** This configuration extended the previous setup by incorporating a reranker. The reranker, implemented using the `BAAI/bge-reranker-large` model [16] because of its highest reranking score on the `MTEB` benchmark, utilized a cross-encoder approach to reorder the retrieved articles based on their relevance to the query. The top-ranked **5 articles** (with retrieved top 20 potential candidates) were then provided as context to the language model. This configuration aimed to determine whether the reranking step improved performance by ensuring the most relevant information was prioritized.

The evaluation was repeated **5 times** to ensure statistical robustness. For each question, a prompt was generated, combining the question text, potential answers, and optionally the retrieved or reranked context. The questions were gathered from renowned polish law testing sites such as `https://arslege.pl/` and their correctness was validated by a lawyer. The system's output was compared against the correct answer to measure accuracy.

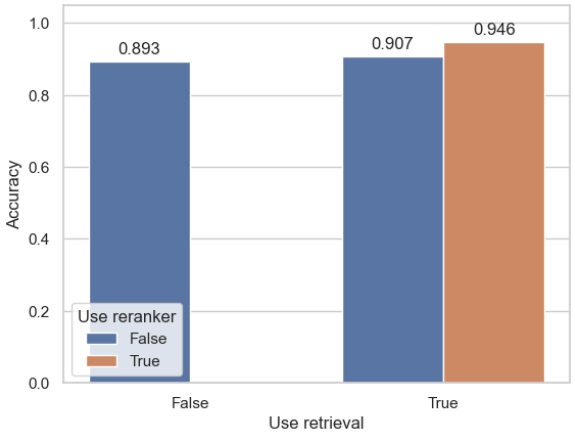

Figure 10: Accuracy for each configuration. The colour corresponds to the use of the reranker.

NOTE 1: A dedicated Polish reranker `sdadas/polish-reranker-large-ranknet` [17] has also been tested, however it achieved worse results (93.2% accuracy). It was chosen because of it high performance in benchmarks (NDCG@10 of 62.65 in Polish reranking).

NOTE 2: Retrieval of only top 5 articles was also tested. It achieved accuracy of 91.7%.

To quantify the performance of each configuration, we calculated the accuracy as the percentage of correctly answered questions presented on Figure 10. Additionally, we analyzed which questions were the most challenging for the model under different configurations (see Table 5). This analysis provided deeper insights into the impact of context retrieval and reranking on model performance.

The key outcomes of the evaluation is that the retrieval algorithm in combination with the reranker boosts the performance of the language model. In our case, we managed to increase the mean accuracy of multiple choice test in **nearly 5 percentage points** by introducing the RAG system with reranker. Moreover, there are some questions, like: *According to the Consti- tution of the Republic of Poland, anyone against whom criminal proceedings are conducted has the right to defense at all stages of the proceedings. They may, in particular, choose a defender or:* that causes great problems even for the RAG system with reranker. It can be treated as a proof for the humanlike prone to mistakes. It is always advisable to fact-check models output.

Table 5: Questions that were always incorrect for particular configuration.

Polish Question	English Translation	retrieval	reranker
Zgodnie z Konstytucja Rzeczypospolitej Polskiej wybrany do Sejmu może być obywatel polski majacy prawo wybierania, który najpóźniej w dniu wyborów kończy:	According to the Constitution of the Republic of Poland, a person elected to the Sejm must be a Polish citizen with voting rights, who turns this age at the latest on election day:	False	False
Zgodnie z Konstytucja Rzeczypospolitej Polskiej za naruszenie ustawy, w zwiazku z zajmowanym stanowiskiem lub w zakresie swojego urzedowania, odpowiedzialność konstytucyjna przed Trybunałem Stanu ponosi:	According to the Constitution of the Republic of Poland, for violating the law related to their position or duties, constitutional responsibility is borne before the Tribunal of State by:	False	False
Zgodnie z Konstytucja Rzeczypospolitej Polskiej, każdy przeciw komu prowadzone jest postépowanie karne, ma prawo do obrony we wszystkich stadiach postépowania. Może on w szczególności wybrać obrońce lub:	According to the Constitution of the Republic of Poland, anyone against whom criminal proceedings are conducted has the right to defense at all stages of the proceedings. They may, in particular, choose a defender or:	False	False
Zgodnie z Konstytucja Rzeczypospolitej Polskiej, każdy przeciw komu prowadzone jest postépowanie karne, ma prawo do obrony we wszystkich stadiach postépowania. Może on w szczególności wybrać obrońce lub:	According to the Constitution of the Republic of Poland, anyone against whom criminal proceedings are conducted has the right to defense at all stages of the proceedings. They may, in particular, choose a defender or:	True	True
Zgodnie z Konstytucja Rzeczypospolitej Polskiej ważność wyboru Prezydenta Rzeczypospolitej stwierdza:	According to the Constitution of the Republic of Poland, the validity of the election of the President of the Republic of Poland is confirmed by:	True	True
Zgodnie z Konstytucja Rzeczypospolitej Polskiej, każdy przeciw komu prowadzone jest postépowanie karne, ma prawo do obrony we wszystkich stadiach postépowania. Może on w szczególności wybrać obrońce lub:	According to the Constitution of the Republic of Poland, anyone against whom criminal proceedings are conducted has the right to defense at all stages of the proceedings. They may, in particular, choose a defender or:	True	False
Zgodnie z Konstytucja Rzeczypospolitej Polskiej Siły Zbrojne Rzeczypospolitej Polskiej służa:	According to the Constitution of the Republic of Poland, the Armed Forces of the Republic of Poland serve:	False	False

7 Answers to Reviewers

No objective evaluation We left the model evaluation stage for the final phase of the project. Consequently, the evaluation was added in the last milestone, and it can be found here 10.

Limited legal sources We decided to focus solely on the Polish Constitution due to its relatively simple evaluation through tests available on the website https://arslege.pl/. Additionally, we were constrained by the requirements of implementing the entire solution locally, from the

vector database to the models. In the past (when we explored a similar project on our own), we downloaded a small subset of legal acts from https://isap.sejm.gov.pl/ covering the last two years, and even then, populating the vector database required many gigabytes of storage. In this project, we concentrated on the correctness of the solution and assessing whether the use of retrieval and reranker improved model quality.

Ablation studies The system without the RAG solution was tested. The results are available here 10.

Error analysis Examples of questions that the model struggles with are provided here 5.

Choice of the size of chunks The length of the text input to the embedding model is derived from Figure 5. We concluded that there was no need to use a character-based text splitter because the articles are approximately 300–400 tokens in length.

Personalities The personalities were designed through appropriate prompt engineering.

8 Division of work

- **Adam Majczyk** - models configuration, personalities, QnA questions acquisition, model evaluation, application enhancements, report
- **Szymon Matuszewski** - the Polish Constitution scraper, EDA, vector database configuration, application fundamentals, report

References

[1] Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N. Rockmore, Diego Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory M. Dickinson, Haggai Porat, Jason Hegland, Jessica Wu, Joe Nudell, Joel Niklaus, John Nay, Jonathan H. Choi, Kevin Tobia, Margaret Hagan, Megan Ma, Michael Livermore, Nikon Rasumov-Rahe, Nils Holzenberger, Noam Kolt, Peter Henderson, Sean Rehaag, Sharad Goel, Shang Gao, Spencer Williams, Sunny Gandhi, Tom Zur, Varun Iyer, and Zehua Li. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models, 2023.

[2] Nicholas Pipitone and Ghita Houir Alami. Legalbench-rag: A benchmark for retrieval-augmented generation in the legal domain, 2024.

[3] Rohan Bhambhoria, Samuel Dahan, Jonathan Li, and Xiaodan Zhu. Evaluating ai for law: Bridging the gap with open-source solutions, 2024.

[4] Zhi Zhou, Jiang-Xin Shi, Peng-Xiao Song, Xiao-Wen Yang, Yi-Xuan Jin, Lan-Zhe Guo, and Yu-Feng Li. Lawgpt: A chinese legal knowledge-enhanced large language model, 2024.

[5] Streamlit Inc. Streamlit: The fastest way to build data apps, 2019.

[6] Ayomide Odumakinde, Daniel D'souza, Pat Verga, Beyza Ermis, and Sara Hooker. Multilingual arbitrage: Optimizing data pools to accelerate multilingual progress, 2024.

[7] Shaoxiong Ji, Zihao Li, Indraneil Paul, Jaakko Paavola, Peiqin Lin, Pinzhen Chen, Dayyán O'Brien, Hengyu Luo, Hinrich Schütze, Jörg Tiedemann, and Barry Haddow. EMMA-500: Enhancing massively multilingual adaptation of large language models. *arXiv preprint 2409.17892*, 2024.

[8] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan

Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[9] Krzysztof Ociepa, Łukasz Flis, Krzysztof Wróbel, Adrian Gwoździej, and Remigiusz Kinas. Bielik 7b v0.1: A polish language model – development, insights, and evaluation, 2024.

[10] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

[11] Baza informacji prawnej.

[12] The constitution of the republic of poland. https://www.sejm.gov.pl/prawo/konst/angielski/kon1.htm, 2015. Version dated January 7, 2015. Accessed on [insert date of access].

[13] pgvector contributors. pgvector: Open-source vector similarity search for postgresql. https://github.com/pgvector/pgvector, 2023.

[14] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019.

[15] Ollama contributors. Ollama: Local language models for secure and private ai. https://github.com/ollama/ollama/tree/main, 2023. Accessed on [insert date of access].

[16] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. C-pack: Packaged resources to advance general chinese embedding, 2023.

[17] Sławomir Dadas and Małgorzata Grebowiec. Assessing generalization capability of text ranking models in polish. 2024.