

# **Enhancing Legal Research Efficiency through Retrieval-Augmented Generation (RAG) with Vector Search**

MSc Research Project  
Artificial Intelligence

**Keerthana Venkatesan**

Student ID: x23275243

School of Computing  
National College of Ireland

Supervisor: Arundev Vamadevan

**National College of Ireland**  
**MSc Project Submission Sheet**  
**School of Computing**



**Student Name:** Keerthana Venkatesan

**Student ID:** X23275243

**Programme:** MSc in Artificial Intelligence **Year:** 2024-2025

**Module:** Research Project (Configuration Manual)

**Supervisor:** Arundev Vamadevan

**Submission Due Date:** 11/08/2025

**Project Title:** Enhancing Legal Research Efficiency through Retrieval-Augmented Generation (RAG) with Vector Search

**Word Count:** 623

**Page Count :** 6

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

**Signature:** Keerthana Venkatesan

**Date:** 10/08/2025

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST**

|                                                                                                                                                                                           |                          |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| Attach a completed copy of this sheet to each project (including multiple copies)                                                                                                         | <input type="checkbox"/> |
| <b>Attach a Moodle submission receipt of the online project submission,</b> to each project (including multiple copies).                                                                  | <input type="checkbox"/> |
| <b>You must ensure that you retain a HARD COPY of the project,</b> both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. | <input type="checkbox"/> |

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

| Office Use Only                  |  |
|----------------------------------|--|
| Signature:                       |  |
| Date:                            |  |
| Penalty Applied (if applicable): |  |

# Configuration Manual

## 1. Introduction

All the information regarding the hardware requirements and software requirements that have been employed in the process of this legal chatbot research is contained in this configuration manual. The following provisions are steps that ought to be taken to establish the running environment of this research undertaking. It also has various applications that need to be set-up and used in the application of the retrieval-augmented generation system using LLaMA-3 as well as Flan-T5 models as the answers to legal questions.

## 2. System Configuration

The system specifications encompassed all the materials involved in the conduct of research study on implementation of legal chatbots. The local system requirements consist of high performance computing and hardware to run large language models with Jupyter Notebook as an environment to develop and test. All the hardware accelerators have been explored and warranted and developed to be used with a degree of caution because of the restrictions in usage and costs of computations involved in transformer model's inference.

|                                                                                                                              |                                                     |      |   |
|------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|------|---|
|                                           | Device specifications                               | Copy | ^ |
| Device name                                                                                                                  | Keerthana                                           |      |   |
| Processor                                                                                                                    | 12th Gen Intel(R) Core(TM) i7-1255U 1.70 GHz        |      |   |
| Installed RAM                                                                                                                | 16.0 GB (15.7 GB usable)                            |      |   |
| Device ID                                                                                                                    | A77A23A1-E35A-4250-9C71-186181CB6CCF                |      |   |
| Product ID                                                                                                                   | 00342-22283-41989-AAOEM                             |      |   |
| System type                                                                                                                  | 64-bit operating system, x64-based processor        |      |   |
| Pen and touch                                                                                                                | No pen or touch input is available for this display |      |   |
| Related links <a href="#">Domain or workgroup</a> <a href="#">System protection</a> <a href="#">Advanced system settings</a> |                                                     |      |   |
|                                           | Windows specifications                              | Copy | ^ |
| Edition                                                                                                                      | Windows 11 Home                                     |      |   |
| Version                                                                                                                      | 24H2                                                |      |   |
| Installed on                                                                                                                 | 17/07/2025                                          |      |   |
| OS build                                                                                                                     | 26100.4652                                          |      |   |
| Experience                                                                                                                   | Windows Feature Experience Pack 1000.26100.128.0    |      |   |
|                                                                                                                              | <a href="#">Microsoft Services Agreement</a>        |      |   |
|                                                                                                                              | <a href="#">Microsoft Software Licence Terms</a>    |      |   |

Figure 1. System Configuration

```
=== Runtime Environment ===
Python Version: 3.12.4 | packaged by Anaconda, Inc. | (main, Jun 18 2024, 15:03:56) [MSC v.1929 64 bit (AMD64)]
PyTorch Version: 2.7.1+cpu
Transformers Version: 4.54.0
CUDA Available: False
```

Figure 2. Runtime types

Software Used

- Python 3.8+: programming language for model implementation and evaluation
- Jupyter Notebook: Interactive environment for code execution, model training, and results visualization
- Anaconda: Package management and virtual environment setup
- Stored the API keys in environmental setup

3. Dataset Specifications

The respective use case of the current study involved an informed method of training and testing the outstanding performance of LLaMA-3 and Flan-T5 responses in querying legal questions. The main training include 10-17 selected legal PDFs on different fields of law such as contract law, tort law, criminal law, constitutional law, and civil procedure that will complete the whole legal knowledge to be used in the implementation of retrieval-augmented generation. Hugging Face dataset (dzungggg/legal-qa-v1) was utilized to best represent a standardized evaluation setting in comparison and comparison of models based on each respective ROUGE-L, BLEU, and response latency measure.

| Name                                                                                                                                                                                                                  | Date modified       | Type               | Size      |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|--------------------|-----------|
|  Administr Administrative Law Public Administr Public Administration and the Administr ation and the Administrative Conference ... | 16/07/2025 10:16 pm | Adobe Acrobat D... | 137 KB    |
|  Administrative Law Author Aberham Yohannes.pdf                                                                                    | 16/07/2025 10:16 pm | Adobe Acrobat D... | 1,328 KB  |
|  Administrative Law I Author Andres Molina Gimenez.pdf                                                                             | 16/07/2025 10:16 pm | Adobe Acrobat D... | 956 KB    |
|  Basic Principles of Administrative Law Author Robert Musumeci Avukat Perit.pdf                                                    | 16/07/2025 10:16 pm | Adobe Acrobat D... | 140 KB    |
|  Criminal Justice Assessment Toolkit Autor United Nations Office on Drugs and Crime (UNODC).pdf                                    | 16/07/2025 10:16 pm | Adobe Acrobat D... | 9,111 KB  |
|  Criminal Law General Provisions Course Notes Autor Pinar No Okur.pdf                                                              | 16/07/2025 10:16 pm | Adobe Acrobat D... | 431 KB    |
|  Criminal Law I Autor Mrs. Glory Nirmala y Ato Serkaddis Zegeye.pdf                                                                | 16/07/2025 10:16 pm | Adobe Acrobat D... | 1,442 KB  |
|  Criminal Law Outline Autor Rachel Barkow.pdf                                                                                      | 16/07/2025 10:16 pm | Adobe Acrobat D... | 627 KB    |
|  General Administrative Law ACT Author Legislationline.pdf                                                                         | 16/07/2025 10:16 pm | Adobe Acrobat D... | 159 KB    |
|  Halsbury_Laws_of_England_v1_1907.pdf                                                                                              | 16/07/2025 10:16 pm | Adobe Acrobat D... | 82,133 KB |
|  Introduction to Criminal Law Autor lardbucket.pdf                                                                                 | 16/07/2025 10:16 pm | Adobe Acrobat D... | 7,149 KB  |
|  Introduction to the Laws of Timor-Leste Criminal Law Autor Erik Jensen.pdf                                                        | 16/07/2025 10:16 pm | Adobe Acrobat D... | 363 KB    |
|  The Evolution of General Administrative Law and the Emergence of Postmodern Administrative Law Author Karl Heinz Ladeur.pdf       | 16/07/2025 10:16 pm | Adobe Acrobat D... | 775 KB    |
|  The Fundamentals of Criminal Law Autor Oxford University Press.pdf                                                                | 16/07/2025 10:16 pm | Adobe Acrobat D... | 1,525 KB  |
|  The Heart of the Criminal Justice System A critical Analysis of the Position of the Victim Autor Jana Bednarova.pdf               | 16/07/2025 10:16 pm | Adobe Acrobat D... | 553 KB    |
|  The Theory and Practice of Criminal Justice in Africa Autor Etannibi E O Alemika, Dr Richard Bowd, Simon Robins.pdf               | 16/07/2025 10:16 pm | Adobe Acrobat D... | 790 KB    |
|  Victim Participationin Criminal Law Proceedings Autor Redress, Iss.pdf                                                            | 16/07/2025 10:16 pm | Adobe Acrobat D... | 1,583 KB  |

Figure 3. List of PDFs

## 4. Project Development

Once all PDF documents that are legal are gathered and the dzunggg/legal-qa-v1 evaluation dataset created, the Jupyter notebook environment could be activated to develop a project. Launch the notebook environment and go to the project directory where all the code files would be in the .ipynb format that makes accessibility to previous run outputs and iterative development accessible. The development is a step upon step of configuring the pipeline of the retrieval-augmented generation pipeline or in other terms, the steps range between legal document processing, set up of the configuration of the vector database, loading of the model, and the implementation of the evaluation framework. Foreach stage of development can be accomplished cell by cell enabling it to debug its system or in parallel to facilitate batch operations and system wide testing.

### 4.1 Libraries imports

All necessary Python libraries and packages of legal NLP are installed. Anaconda comes with several versions and libraries that are by default installed. Wherever it is needed, load in your environment the necessary libraries. These are open libraries and are easily installable using Python sources of authority.

```
from langchain.document_loaders import PyPDFLoader, DirectoryLoader
from langchain.text_splitter import RecursiveCharacterTextSplitter
from typing import List
from langchain.schema import Document
from langchain_huggingface import HuggingFaceEmbeddings
from qdrant_client import QdrantClient
from langchain.prompts import PromptTemplate
from langchain.chains import RetrievalQA
from langchain_groq import ChatGroq
from huggingface_hub import InferenceClient
from datasets import load_dataset
import tqdm as notebook_tqdm
from langchain_qdrant import Qdrant
import time
import evaluate
from nltk.translate.bleu_score import sentence_bleu
from rouge import Rouge
import warnings
import numpy as np
```

Figure 4. Importing Libraries

### 4.2 Import PDF

PDF text extraction from 10-15 legal documents, text cleaning and normalization for legal content. Document chunking for optimal retrieval performance. Embedding generation using sentence-transformer models. Qdrant vector database initialization and document indexing

```
#Extract Data From the PDF File
def load_pdf_file(data):
    loader= DirectoryLoader(data,
                            glob="*.pdf",
                            loader_cls=PyPDFLoader)

    documents=loader.load()

    return documents
```

**Figure 5. Extracting Data from the PDF**

### 4.3 Legal Document Processing

```
#Split the Data into Text Chunks
def text_split(minimal_docs):
    text_splitter=RecursiveCharacterTextSplitter(chunk_size=500, chunk_overlap=20)
    text_chunks=text_splitter.split_documents(minimal_docs)
    return text_chunks
```

**Figure 6. Spitting the documents into chunks**

Embedding part is very important, Figure 7. Shows the embedding step

```
from langchain_huggingface import HuggingFaceEmbeddings

embeddings = HuggingFaceEmbeddings(model_name='sentence-transformers/all-MiniLM-L6-v2')
```

**Figure 7. Embedding using sentence transformer**

```
from langchain_qdrant import Qdrant

# Initialize new vector store for MiniLM
new_qdrant = Qdrant(
    client=qdrant_client,
    collection_name=new_collection_name,
    embeddings=embeddings # <- MiniLM embeddings
)

# Add your already split text chunks
new_qdrant.add_documents(text_chunks)
```

**Figure 8. Initializing vector DB**

## 4.3 Model Setup

```
def query_llama_legal_rag(question: str, retriever):
    llm = ChatGroq(
        groq_api_key=GROQ_API_KEY,
        model_name="llama3-70b-8192"
    )

    prompt = PromptTemplate(
        input_variables=["context", "question"],
        template="""You are a trustworthy legal AI assistant. Use only the provided legal context to answer the question below.

---

Context:
{context}

Question:
{question}

---

Answer:
- Do not offer legal advice or guesses.
- If context lacks the answer, say: "I'm sorry, but the provided context does not contain relevant legal information."
""")

    qa_chain = RetrievalQA.from_chain_type(
        llm=llm,
        chain_type="stuff",
        retriever=retriever,
        chain_type_kwargs={"prompt": prompt},
        return_source_documents=True
    )

    result = qa_chain.invoke(question)
    return result['result'], result['source_documents']
```

**Figure 9. Legal RAG Pipeline Using LLaMA-3 and LangChain's RetrievalQA**

```
from transformers import pipeline

# Load Flan model locally (only once)
flan_pipe = pipeline("text2text-generation", model="MBZUAI/LaMini-Flan-T5-783M")

def query_flan_legal_rag_local(question: str, retriever):
    docs = retriever.invoke(question)
    context = "\n".join(doc.page_content for doc in docs)

    prompt = f"""You are a legal assistant AI.

Use the following context to answer the question clearly and concisely. Avoid using numbered lists unless they already appear in the context.

Context:
{context}

Question:
{question}

Answer: """

    response = flan_pipe(prompt, max_new_tokens=300)[0]['generated_text']
    return response.strip(), docs
```

**Figure 10. Retrieval-Augmented Generation Locally with Flan-T5**

```

def evaluate_legal_models(dataset, retriever=None, sample_size=10):
    rouge = Rouge()
    smoothie = SmoothingFunction().method4

    llama_scores = []
    flan_scores = []

    for example in dataset.select(range(sample_size)):
        question = example['question']
        reference = example['answer']

        llama_response, _ = query_llama_legal_rag(question, retriever)
        flan_response, _ = query_flan_legal_rag_local(question, retriever)

        # ROUGE
        llama_rouge = rouge.get_scores(llama_response, reference)[0]['rouge-1']['f']
        flan_rouge = rouge.get_scores(flan_response, reference)[0]['rouge-1']['f']

        # BLEU
        llama_bleu = sentence_bleu([reference.split()], llama_response.split(), smoothing_function=smoothie)
        flan_bleu = sentence_bleu([reference.split()], flan_response.split(), smoothing_function=smoothie)

        llama_scores.append((llama_rouge, llama_bleu))
        flan_scores.append((flan_rouge, flan_bleu))

    return llama_scores, flan_scores

```

**Figure 11. Evaluate Legal QA Models Using ROUGE and BLEU Metrics**

## Reference

Nguyen, P.D. (2024). *legal-qa-v1*. [online] Huggingface.co. Available at: <https://huggingface.co/dzunggg/legal-qa-v>.