

Configuration Manual

MSc Research Project
MSc in Artificial Intelligence

Hassan Ozair
Student ID: 23410663

School of Computing
National College of Ireland

Supervisor: Ade Fajemisin / Paul Stynes

National College of Ireland
MSc Project Submission Sheet
School of Computing



Student Name: Ozair Hassan

Student ID: 23410663

Programme: MSc Artificial Intelligence **Year:** 2025

Module: MSc (Research) Practicum Part 2

Lecturer: Ade Fajemisin / Paul Stynes

Submission

Due Date: 11th December 2025

Project Title: A RAG Framework for Legal Text Retrieval and Generation in Irish Statutory Instruments

Word Count: 1024 **Page Count:** 5 Pages

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: Ozair Hassan

Date: 10/12/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

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

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

Hassan Ozair
Student ID: 23410663

1 System Overview and Requirements

The following is an in depth step by step guide written to allow future researchers to fully recreate the experiments of “A RAG Framework for Legal Text Retrieval and Generation in Irish Statutory Instruments”. Section 1 will cover the hardware requirements along with the software requirements. Section 2 will go through the environment setup and section 3 will be detailed step by step guide for running all experiments. For any individual with a familiar background and looking for a quick guide they can skip directly to Section 4

1.1 System Overview

The goal of this configuration manual is to give future researchers clear instructions on how to install, configure and execute the full experimental RAG pipeline. The experiment can be broken down into the following:

- Scrape Irish Statutory Instrument PDFs
- Text extraction and preprocessing
- Executing baseline on Irish SIs dataset
- Running improve Weaviate based RAG pipeline
- Running LLM answer generation
- Evaluating generated answers using RePASS evaluation framework

1.2 Hardware Requirements

Minimum:

- 16 GB RAM
- CPU: AMD Ryzen 5 5600 6-Core Processor
- 20 GB free disk space

Recommended:

- 32 GB RAM
- CPU: AMD Ryzen 5 5600 6-Core Processor
- NVIDIA GPU with CUDA for embedding generation
- SSD Storage

1.3 Software Requirements

Required:

- Python 3.10+
- Git

- Docker Desktop

2 Environment Setup

2.1 Clone the Repository

Use the following command in any terminal to clone the repository to your local system

```
git clone https://github.com/Ozair-Hassan/A-RAG-Framework-for-Legal-Text-Retrieval-and-Generation-in-Irish-Statutory-Instruments
```

And then cd into the clone repository using

```
cd A-RAG-Framework-for-Legal-Text-Retrieval-and-Generation-in-Irish-Statutory-Instruments
```

2.2 Create and activate the Python virtual environment

On windows (powershell) use the following commands:

```
python -m venv .venv
.\.venv\Scripts\Activate.ps1
```

On macOS / Linux use the following commands:

```
python3 -m venv .venv
source .venv/bin/activate
```

2.3 Install required Python packages

On windows use the following commands to install the required dependencies to run the project:

```
pip install requests beautifulsoup4 tqdm unidecode pdfminer.six pytesseract pdfplumber
Pillow nltk datasets pandas scikit-learn rank-bm25 sentence-transformers contractions groq
python-dotenv nest_asyncio weaviate-client matplotlib backoff openai
```

2.4 Setup the environment variables

You will need to create a .env file in both the Benchmark and Weaviate_Implementation folders respectively.

After creating the .env file in the root of these two folders you can then add the relevant API keys

```
GROQ_API_KEY_1=your_api_key_here
```

If using OpenAI:

```
OPENAI_API_KEY=your_key
```

3 Running the Pipeline

3.1 Step 1: Scrape Irish Statutory Instruments PDFs

The scrape.py script downloads all the SI PDFs from the URLs that are stored in the Links.txt.

Run the following command:

```
python scrape.py --download-path ./downloads --links-file Links.txt --delay 1.0
```

Argument	Description
--download-path	Location where you want to save the PDF files
--links-file	Location of text file containing the URLs
--delay	Delay between requests in order to not overload or crash the server

3.2 Step 2: Preprocess PDFs into JSONL form

This script will process the pdf files into their jsonl format dividing each document into its relevant sections. NOTE: if you may have to adjust the input path inside the script at the top in the configuration section as per your directory layout.

Run:

Python preprocess.py

3.3 Step 3: Run baseline system

This is quite straightforward as all logic is handled through the Jupyter notebooks.

Simply execute them in the following order:

1. Benchmark → DP_Au_Irish.ipynb
2. Benchmark → Eval_Au_Irish.ipynb
3. Benchmark → QnA_Au_Irish.ipynb

At the end of the executions, you will have the a data output folder containing the .trec files, the answers-llama3.1.json file and at the root of the benchmark folder you will have png files for with the graphs for all the various retriever techniques.

3.4 Step 4: Start your local instance of Weaviate (Docker)

You will need to use the following command from the root of the repository to move to the Weaviate_local folder

```
cd Weaviate_local
```

After which you will run the following docker command to bring the build for the Weaviate instance and execute it:

```
docker compose up -d
```

Once it has finished running and you can see the docker instance running the next step is to simply run the following in any terminal of choosing to verify the status of the docker container

```
curl http://localhost:8080/v1/.well-known/ready
```

with the expected output along the lines of
{ "status": "ready" }

3.5 Step 5 Execute Weaviate RAG system

Again similar to the baseline this will all be run through the Jupyter notebooks

Simply execute them in the following order:

1. Weaviate_Implementation → DP_My_Irish.ipynb
2. Weaviate_Implementation → Eval_My_Irish.ipynb
3. Weaviate_Implementation → QnA_My_Irish.ipynb

Similar to the benchmark at the end of the executions, you will have the data output folder containing the .trec files, the answers-llama3.1.json file and at the root of the Weaviate Implementation folder you will have png files for with the graphs for all the various retriever techniques.

3.6 Step 6 RePASs Evaluation

You will need to clone the RePASs repository and cd into it using the following commands:

```
git clone https://github.com/RegNLP/RePASs.git && cd RePASs
```

Next step is to install the requirements

```
pip install -r RePASs/requirements.txt
```

Then run the evaluation script:

```
python scripts/evaluate_model.py --input_file answers_weaviate.json --group_method_name hybrid-llama
```

This will finally result in two files one result.csv that gives the breakdown answer by answer and the result.txt that gives a summary of the results.

More information can be found at the RePASs repository.

4 Quick Guide

This is only for experience users who want to setup and recreate the experiment as quickly as possible.

Clone repo

```
git clone https://github.com/Ozair-Hassan/A-RAG-Framework-for-Legal-Text-Retrieval-and-Generation-in-Irish-Statutory-Instruments
```

```
cd A-RAG-Framework-for-Legal-Text-Retrieval-and-Generation-in-Irish-Statutory-Instruments
```

Create environment

```
python -m venv .venv  
./venv/Scripts/Activate.ps1
```

or

```
source .venv/bin/activate
```

Install dependencies

```
pip install requests beautifulsoup4 tqdm unicode pdfminer.six pytesseract pdfplumber  
Pillow nltk datasets pandas scikit-learn rank-bm25 sentence-transformers contractions groq  
python-dotenv nest_asyncio weaviate-client matplotlib backoff openai
```

Add API keys

Create .env in Benchmark and Weaviate_Implementation folders:

```
GROQ_API_KEY_1=your_key
```

```
OPENAI_API_KEY=your_key
```

Scrape PDFs

```
python scrape.py --download-path ./downloads --links-file Links.txt --delay 1.0
```

Preprocess PDFs

```
python preprocess.py
```

Run baseline notebooks

Jupyter notebook

1. Benchmark → DP_Au_Irish.ipynb
2. Benchmark → Eval_Au_Irish.ipynb
3. Benchmark → QnA_Au_Irish.ipynb

Start Weaviate

```
cd Weaviate_local  
docker compose up -d
```

Run Weaviate notebooks

1. Weaviate_Implementation → DP_My_Irish.ipynb
2. Weaviate_Implementation → Eval_My_Irish.ipynb
3. Weaviate_Implementation → QnA_My_Irish.ipynb

RePASs evaluation

```
git clone https://github.com/RegNLP/RePASs.git
```

```
cd RePASs
```

```
pip install -r requirements.txt
```

```
python scripts/evaluate_model.py --input_file answers_weaviate.json --group_method_name  
hybrid-llama
```