Introduction to searching Data with a LLM

Inhaltsverzeichnis

[1 Web Scraping: 1](#_Toc177673273)

[2 Document Processing: 1](#_Toc177673274)

[3 Natural Language Processing (NLP): 1](#_Toc177673275)

[4 Information Retrieval: 2](#_Toc177673276)

[5 Machine Learning: 2](#_Toc177673277)

[6 Large Language Models (LLMs): 2](#_Toc177673278)

[7 Deep Learning Frameworks: 2](#_Toc177673279)

[8 Data Storage and Databases: 2](#_Toc177673280)

[9 Cloud Computing: 2](#_Toc177673281)

[10 Programming Languages: 2](#_Toc177673282)

[11 Software Development Practices: 3](#_Toc177673283)

[12 Ethics and Legal Considerations: 3](#_Toc177673284)

# Web Scraping:

## HTTP requests

* HTTP (Hypertext Transfer Protocol) is the foundation of data communication on the web.
* You need to understand:
  + HTTP methods (GET, POST, PUT, DELETE, etc.)
  + Request headers and parameters
  + Response status codes (200 OK, 404 Not Found, etc.)
  + How to send requests and handle responses
* In Python, the 'requests' library is commonly used for making HTTP requests.

## HTML parsing (e.g., BeautifulSoup, lxml)

* HTML (Hypertext Markup Language) is the standard markup language for web pages.
* You need to learn:
  + Basic HTML structure (tags, attributes, DOM tree)
  + How to navigate and search HTML documents
  + Extracting specific data from HTML elements
* Popular Python libraries for HTML parsing include:
  + BeautifulSoup: Easy to use, great for beginners
  + lxml: Faster but more complex, good for larger projects

## API interactions (if your DMS has an API)

* APIs (Application Programming Interfaces) provide a structured way to access data or services.
* Key concepts include:
  + RESTful API principles
  + Authentication methods (API keys, OAuth, etc.)
  + JSON and XML data formats
  + Handling rate limits and pagination
* If your DMS has an API, you'll need to:
  + Read the API documentation
  + Understand the available endpoints and their parameters
  + Learn how to authenticate your requests
  + Parse and process the API responses

Here's a simple example of how these concepts might come together in Python code:

# Document Processing:

Text extraction from various file formats (PDF, DOC, etc.)

Optical Character Recognition (OCR) for scanned documents

# Natural Language Processing (NLP):

Text preprocessing techniques

Tokenization

Named Entity Recognition (NER)

Text vectorization methods

# Information Retrieval:

Search algorithms

Indexing techniques

Vector databases (e.g., Faiss, Pinecone)

# Machine Learning:

Basics of neural networks

Transfer learning

Fine-tuning pre-trained models

# Large Language Models (LLMs):

Architecture of transformer-based models

Prompt engineering

Few-shot and zero-shot learning

# Deep Learning Frameworks:

PyTorch or TensorFlow

# Data Storage and Databases:

SQL and NoSQL databases

Document stores (e.g., MongoDB)

# Cloud Computing:

Basic cloud concepts (e.g., AWS, Google Cloud, or Azure)

Containerization (e.g., Docker)

# Programming Languages:

Python (most common for ML/NLP tasks)

SQL for database queries

# Software Development Practices:

Version control (e.g., Git)

API development

Basic software architecture

# Ethics and Legal Considerations:

Data privacy laws

Ethical AI principles