Title : AI powered Text Summarizer

1. Project Overview

The AI Text Summarizer is a web-based system designed to automatically condense long texts into concise, meaningful summaries. It provides two types of summarization:

1. **Abstractive Summarization** – Generates a new text that conveys the key points of the original content, similar to how a human would summarize.
2. **Extractive Summarization** – Selects important sentences directly from the original text to form a summary.

The system supports input via **pasted text** or **file upload** (PDF and TXT formats), and can handle multiple files simultaneously. It also evaluates the quality of summaries using standard **ROUGE metrics**, which measure similarity between the generated summary and the original text.

Problem Statement

- **Input:** Large text files (e.g., articles, research papers) stored in the /input folder. Example: one research paper and sample.txt.
- **Output:** Concise summaries consisting of **3–5 sentences**.

Note:

- Ensure all input files are copied to local storage before running the project.
- Clear the contents of the /input and /output folders for demonstration purposes.
- After execution, output files will be saved in the /output folder in the backend.
- **Pretrained Model:** facebook/bart-large-cnn (1.6GB).
- **User Interaction:** Through the **frontend interface** or directly via **API endpoints** using Swagger UI, Postman, or other API clients.
- **Evaluation Metric:** ROUGE scores.

---

2. System Architecture

2.1 Backend

The backend is built using a web API framework **(FastAPI)** that exposes endpoints for uploading files, submitting text, and retrieving summaries. Its main responsibilities include:

- Accepting text input or uploaded files.
- Extracting text from PDF or TXT files.

- Performing both abstractive and extractive summarization.
- Calculating ROUGE scores to evaluate summary quality.
- Saving summaries and ROUGE results in an organized manner.

Key Features:

- Handles single and multiple file uploads.
- Supports combining multiple files into a single merged summary.
- Performs text chunking to manage long documents efficiently.

2.1.1 Using Backend Only

- Open Swagger UI at: http://127.0.0.1:8000/docs
- **Single File Demonstration:**
    1. Navigate to **/api/files/extract** → Click **Try it out** → Upload sample.txt → Click **Execute**.
    2. To get the summary, go to **/api/summaries** → Enter the file name sample.txt → Click **Execute** → JSON output will be displayed.
- **Multiple Files Demonstration:**
    1. Navigate to **/api/files/summarize** → Click **Try it out** → Upload multiple files.
    2. Set the **Merge** option to true or false depending on whether you want a merged summary.
    3. Click **Execute** → Summaries and ROUGE scores will be returned.

This approach allows testing the backend independently of the frontend interface.

2.2 Summarization Engine

The system uses a dual-method summarization approach:

1. **Abstractive Summarization**
    - Uses a state-of-the-art transformer model designed for text summarization.
    - Capable of generating natural language summaries that paraphrase the original content.
    - Automatically handles long texts by breaking them into smaller manageable chunks.
2. **Extractive Summarization**
    - Uses an algorithm that identifies and selects the most important sentences from the original text.

- Ensures that key information is retained exactly as written.

3. **Quality Evaluation**
    - ROUGE metrics are calculated for both summary types.
    - Measures include ROUGE-1, ROUGE-2, and ROUGE-L, which quantify overlap in terms of words, phrases, and sentence structure.

2.3 Frontend

The frontend is a user-friendly interface that allows users to interact with the summarization system. It provides:

- **Text Input Area**: Users can paste text directly.
- **File Upload**: Users can upload PDF or TXT files for summarization.
- **Single and Multi-File Handling**: Users can summarize one or multiple files simultaneously.
- **Merged Summaries**: Users have the option to combine multiple files into a single summary.
- **Summary Display**: Both abstractive and extractive summaries are displayed with clear formatting.
- **ROUGE Metrics Display**: Users can view quality scores alongside the summaries.
- **Navigation Bar** : Contains Navigation for single file and separate for quick display of summaries of multiple files with cta to multiple files

The interface is designed for usability, with clear labeling, responsive layout, and visual separation between inputs and outputs.

---

3. Workflow

1. **Input Submission**
    - The user either pastes text or uploads one or more files.
    - For files, the system extracts textual content from PDF or TXT formats.
2. **Text Processing**
    - For long documents, the text is split into smaller sections to ensure optimal summarization performance. Note the provided input file **sample.txt** takes **5 mins**
    - On laptop and **research pdf** provided takes **9 -11 mins** to process
3. **Summary Generation**
    - The abstractive summarizer generates paraphrased summaries.
    - The extractive summarizer selects key sentences from the text.
4. **Quality Evaluation**

- ROUGE metrics are calculated for each summary type.
- Scores indicate how well the summary represents the original text.

5. **Output Delivery**
    - Summaries and scores are returned to the frontend for display.
    - Summaries are also saved in an organized output folder for future reference.

---

4. Features and Capabilities

- **Multiple Input Types**: Supports both direct text and file uploads.
- **Single and Multi-File Summarization**: Can handle one or several files simultaneously.
- **Merged Summaries**: Optional feature to combine multiple files into a single summary.
- **Dual Summarization Approach**: Provides both abstractive and extractive summaries for comprehensive coverage.
- **Quality Assessment**: ROUGE metrics give users insight into summary accuracy.
- **User-Friendly Interface**: Clean layout with separate areas for input, output, and quality metrics.

---

5. Benefits

- **Time Efficiency**: Quickly condenses large documents into short, readable summaries.
- **Enhanced Understanding**: Extractive summaries retain key sentences, while abstractive summaries provide paraphrased insights.
- **Quality Feedback**: ROUGE metrics allow users to assess the reliability of summaries.
- **Flexible Input Handling**: Works with both raw text and document uploads, including multiple files at once.

---

6. Use Cases

1. **Students and Academics**: Summarize long articles, research papers, or lecture notes.
2. **Business Professionals**: Condense reports, meeting notes, or project documentation.
3. **Content Creators**: Generate summaries for blogs, news articles, or research content.
4. **General Users**: Quickly understand large texts without reading the full content.

---

7. System Requirements

- **Backend**:

    - Python environment with machine learning libraries.
    - Access to sufficient memory and CPU/GPU resources for model inference.
- **Frontend**:
    - Modern web browser.
    - JavaScript-enabled interface for interactive summarization.
- **Optional**:
    - GPU acceleration for faster processing of large documents.

---

8. Limitations and Considerations

- Extremely large PDFs may take longer to process.
- ROUGE scores are useful for comparison but do not always reflect human-like understanding.
- Abstractive summaries may occasionally miss minor details from the original text.
- File types are limited to PDF and TXT formats.
- **Summaries are strickly restricted to 5 lines as per problem statement**

---

9. Future Enhancements

- Support for additional file formats (DOCX, HTML, etc.).
- Batch summarization for large-scale document processing.
- Integration with cloud storage for automatic file input/output.
- User customization of summary length and style.

10. API Routes

The backend exposes several API routes that can be accessed via frontend or directly using tools like **Swagger UI** or **Postman**.

1. Root Route

- **Path:** /
- **Method:** GET
- **Purpose:** Health check to confirm the backend is running.
- **Usage:** Returns a simple message indicating the service is operational.

---

2. Single File or Text Upload

- **Path:** /api/files/extract
- **Method:** POST
- **Purpose:** Upload a single file (PDF/TXT) or submit raw text for extraction.
- **Details:**
    - Accepts either a text input or a single file.
    - Extracts the text from the file if provided.
    - Returns the extracted text in JSON format.
- **Notes:** This route is specifically for **single file or text input**. It can be used directly without a frontend through Swagger UI or similar API testing tools.

---

3. Multiple File Upload and Summarization

- **Path:** /api/files/summarize
- **Method:** POST
- **Purpose:** Upload multiple files for summarization.
- **Details:**
    - Accepts multiple files (PDF/TXT).
    - Optionally merges all files into a single summary if the merge option is enabled.
    - Returns both **abstractive** and **extractive summaries** for each file or the merged summary, along with ROUGE scores.
- **Notes:** Supports summarization of multiple files simultaneously. Can also be accessed directly without a frontend for automated batch processing.

---

4. Summarization by Text or Existing File

- **Path:** /api/summaries
- **Method:** POST
- **Purpose:** Generate summaries from either raw text or a previously uploaded file.
- **Details:**
    - Accepts a text input or the name of a file already stored on the backend.
    - Returns both **abstractive** and **extractive summaries**, along with ROUGE evaluation scores.
- **Notes:** Useful for summarization without uploading new files. Can also be used directly via Swagger UI or API clients.

11. Explanation of Separate Routes

In this project, I implemented separate FastAPI routes for file/text extraction and summarization to make the API modular, maintainable, and flexible. Each route has a clear purpose, allowing for easier testing, reusability, and extension.

1. /api/files/extract – File/Text Extraction
This route handles raw text input or single file uploads (PDF/TXT). It is responsible for extracting the textual content from the file or using provided text. The route returns a standardized JSON structure in the format files: [{name, text}], making it easy for other endpoints to consume the data. Separating extraction allows files to be pre-processed independently without triggering summarization.

2. /api/files/summarize – Multiple Files Summarization
This route handles batch uploads of multiple files and optionally merges them into a single summary. The summarization logic is called after extraction. Keeping this route separate allows the extraction logic to be reused for multiple files and provides flexibility in handling merged or individual summaries.

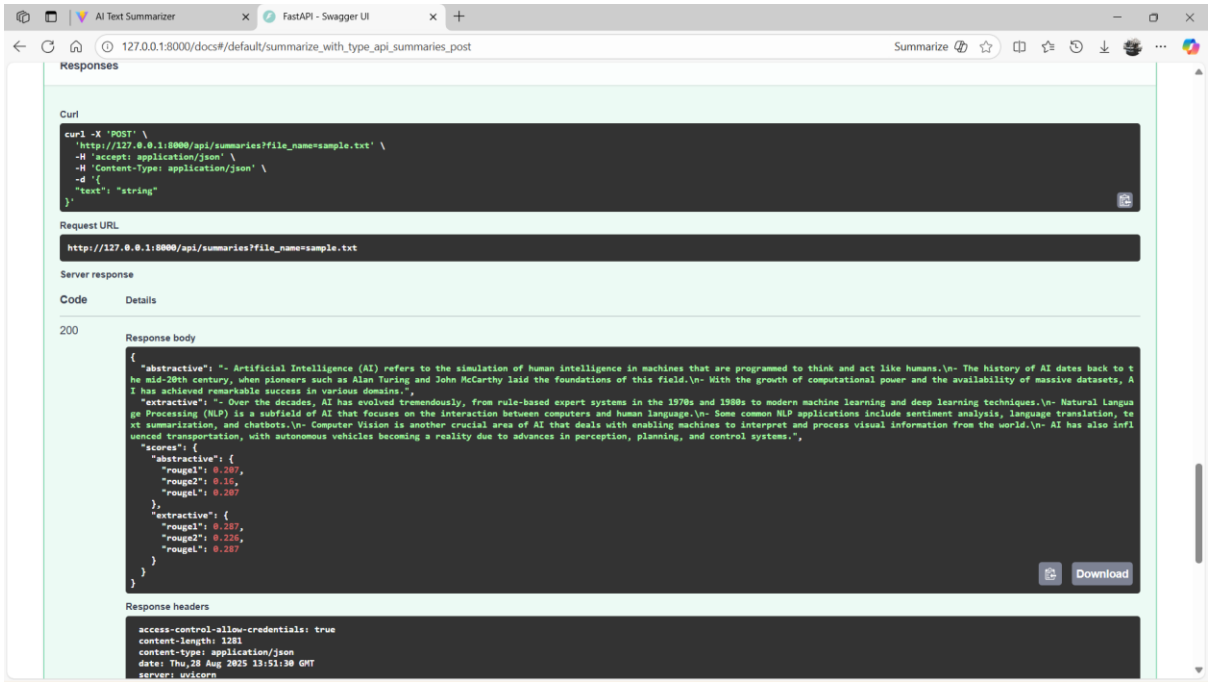
3. /api/summaries – Single Text or File Summarization
This route summarizes already extracted text or a single uploaded file. It enables users to directly summarize text without uploading files, supporting both file-based and text-based workflows. The endpoint returns both abstractive and extractive summaries along with ROUGE evaluation scores.

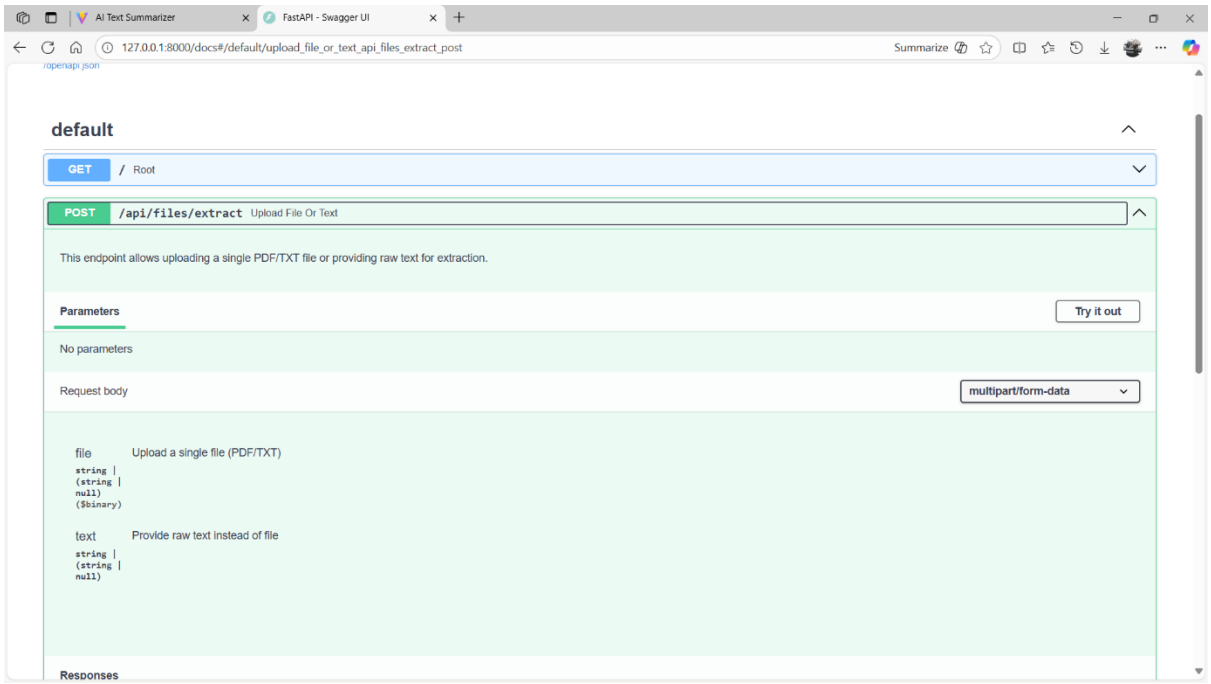
Benefits of This Separation:

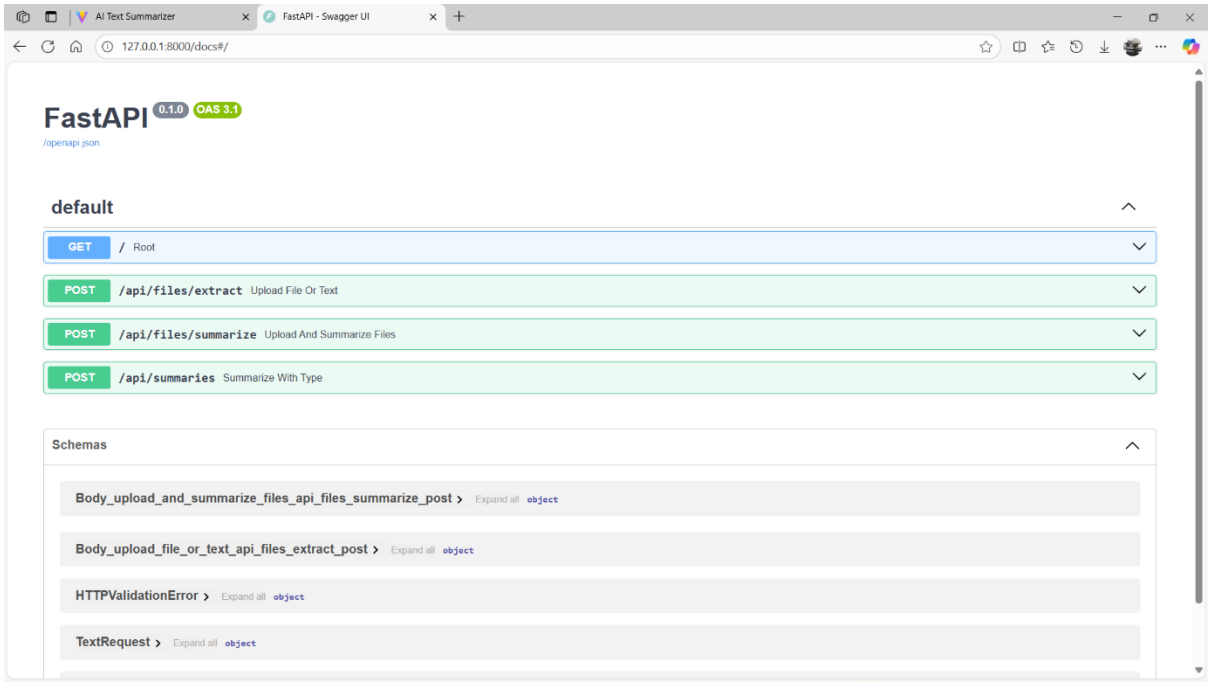
- **Modularity:** Each route has a single responsibility (extract vs summarize).
- **Reusability:** Extraction logic can be used independently by other endpoints or clients.
- **Scalability:** Easier to extend, e.g., adding new summarization models or supporting additional file types.
- **Swagger Clarity:** Each endpoint has a clear purpose and schema, making the API easier to document, test, and consume.

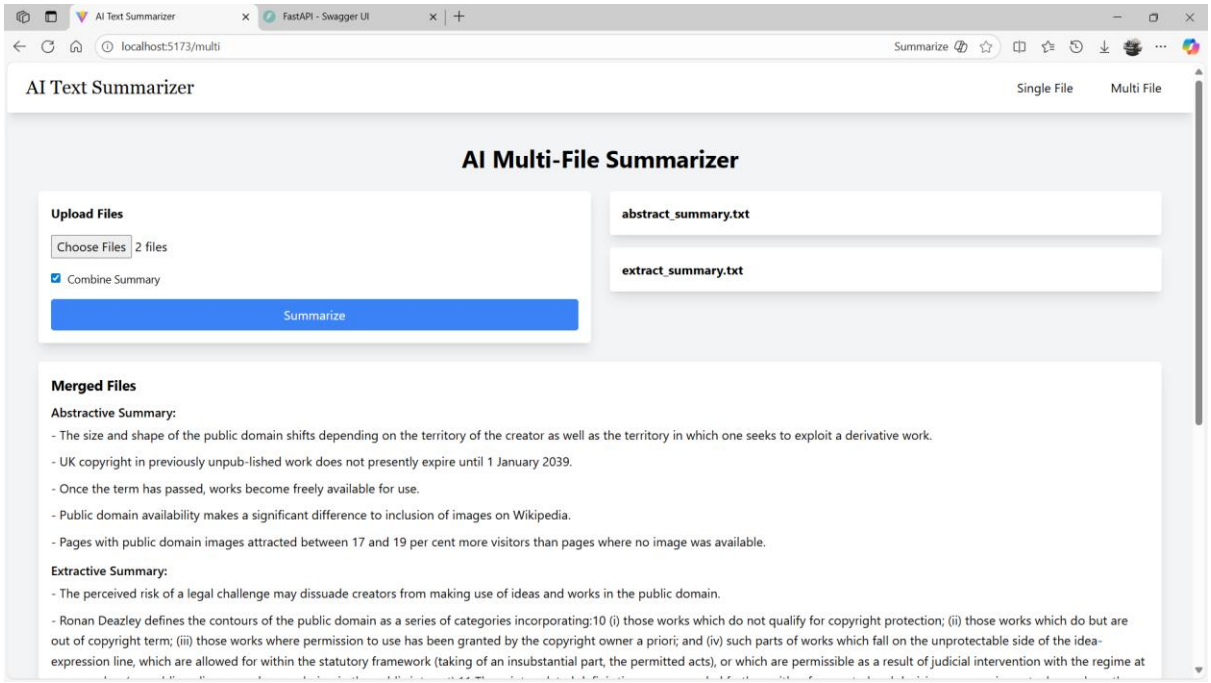
12. Snapshots

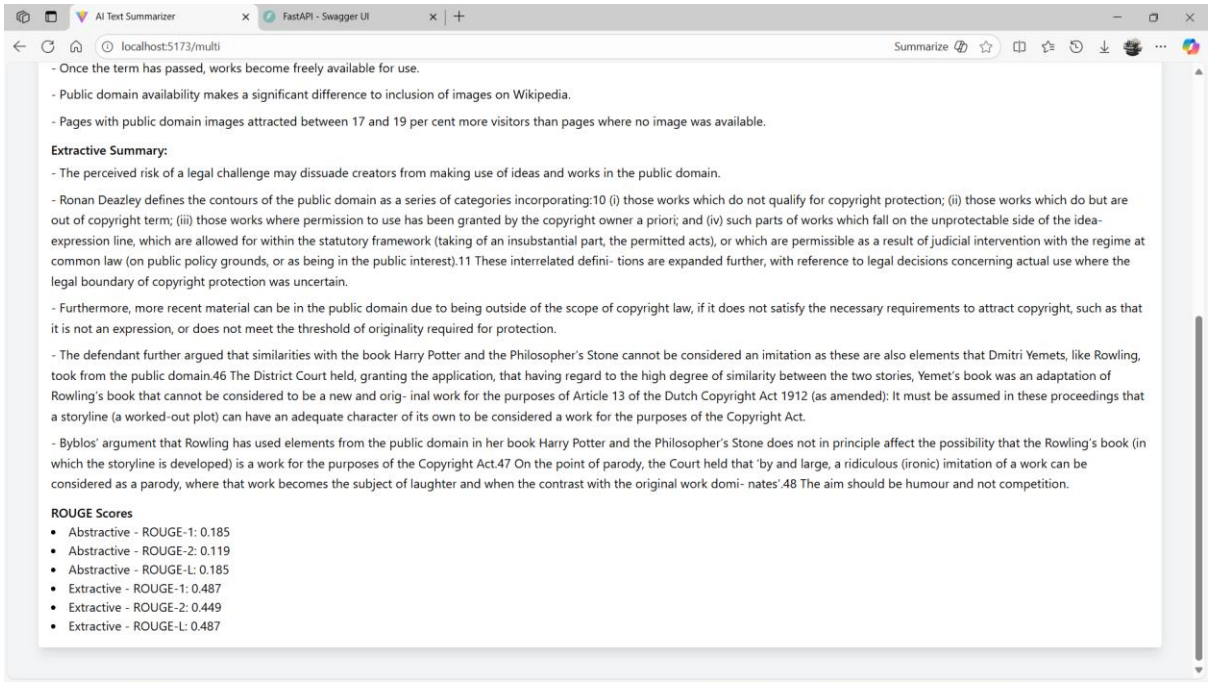
Swagger UI of FASTAPI

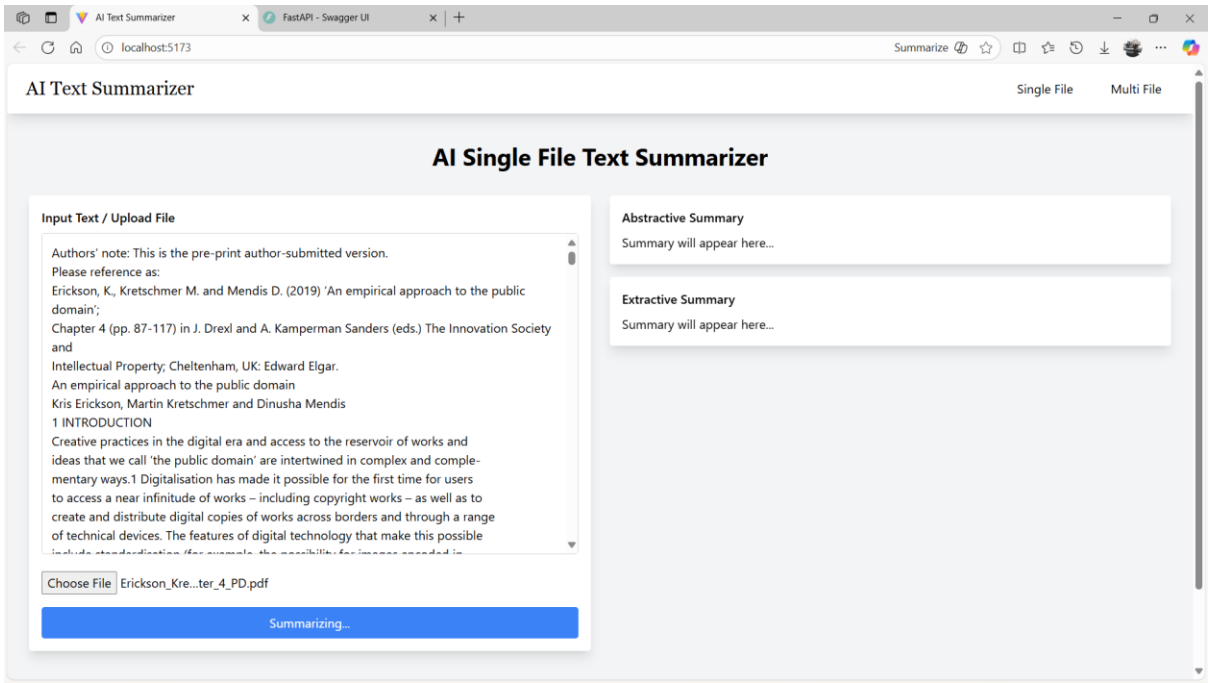

AI Text Summarizer
FastAPI - Swagger UI
127.0.0.1:8000/docs#/default/summarize_with_type_api_summaries_post
Summarize
Responses
Curl
curl -X 'POST' \
'http://127.0.0.1:8000/api/summaries?file_name=sample.txt' \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{
"text": "string"
}'
Request URL
http://127.0.0.1:8000/api/summaries?file_name=sample.txt
Server response
Code
Details
200
Response body
{
"abstractive": "- Artificial Intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think and act like humans.\n- The history of AI dates back to the mid-20th century, when pioneers such as Alan Turing and John McCarthy laid the foundations of this field.\n- With the growth of computational power and the availability of massive datasets, AI has achieved remarkable success in various domains.",
"extractive": "- Over the decades, AI has evolved tremendously, from rule-based expert systems in the 1970s and 1980s to modern machine learning and deep learning techniques.\n- Natural Language Processing (NLP) is a subfield of AI that focuses on the interaction between computers and human language.\n- Some common NLP applications include sentiment analysis, language translation, text summarization, and chatbots.\n- Computer Vision is another crucial area of AI that deals with enabling machines to interpret and process visual information from the world.\n- AI has also influenced transportation, with autonomous vehicles becoming a reality due to advances in perception, planning, and control systems.",
"scores": {
"abstractive": {
"rouge1": 0.207,
"rouge2": 0.16,
"rougeL": 0.207
},
"extractive": {
"rouge1": 0.287,
"rouge2": 0.226,
"rougeL": 0.287
}
}
}
Download
Response headers
access-control-allow-credentials: true
content-length: 1281
content-type: application/json
date: Thu,28 Aug 2025 13:51:30 GMT
server: uvicorn

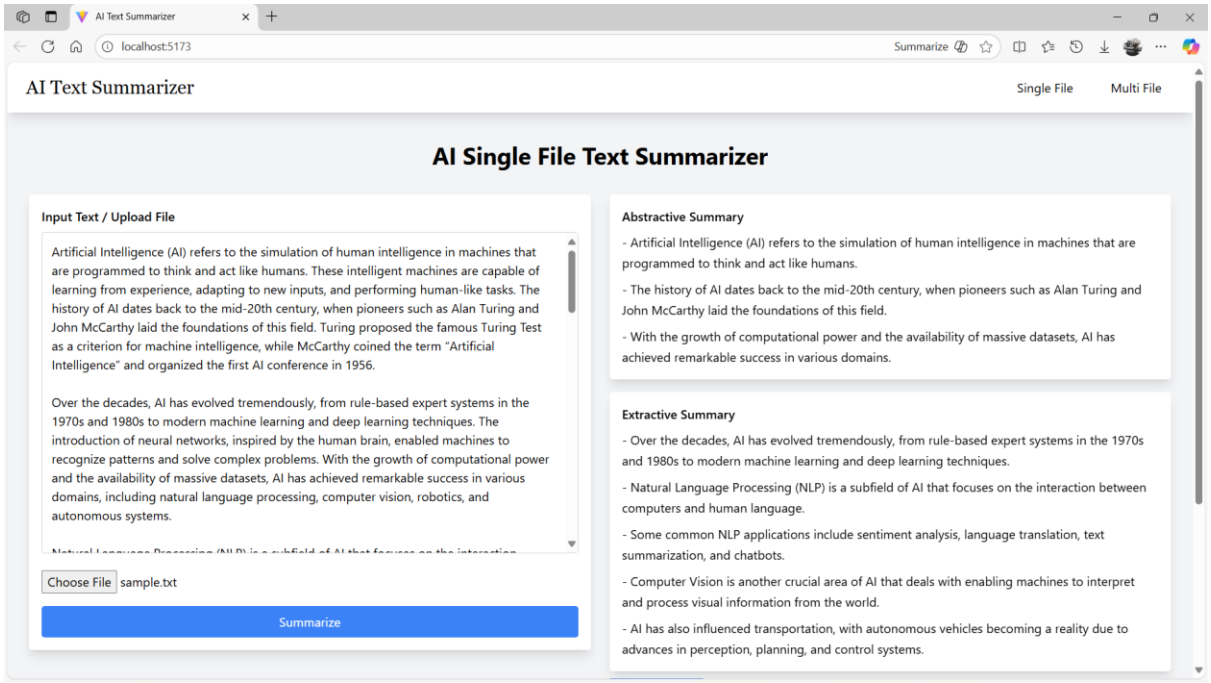

AI Text Summarizer
FastAPI - Swagger UI
127.0.0.1:8000/docs#/default/upload_file_or_text_api_files_extract_post
Summarize
/openapi.json
default
GET / Root
POST /api/files/extract Upload File Or Text
This endpoint allows uploading a single PDF/TXT file or providing raw text for extraction.
Parameters
Try it out
No parameters
Request body
multipart/form-data
file
Upload a single file (PDF/TXT)
string | (string | null) ($binary)
text
Provide raw text instead of file
string | (string | null)
Responses

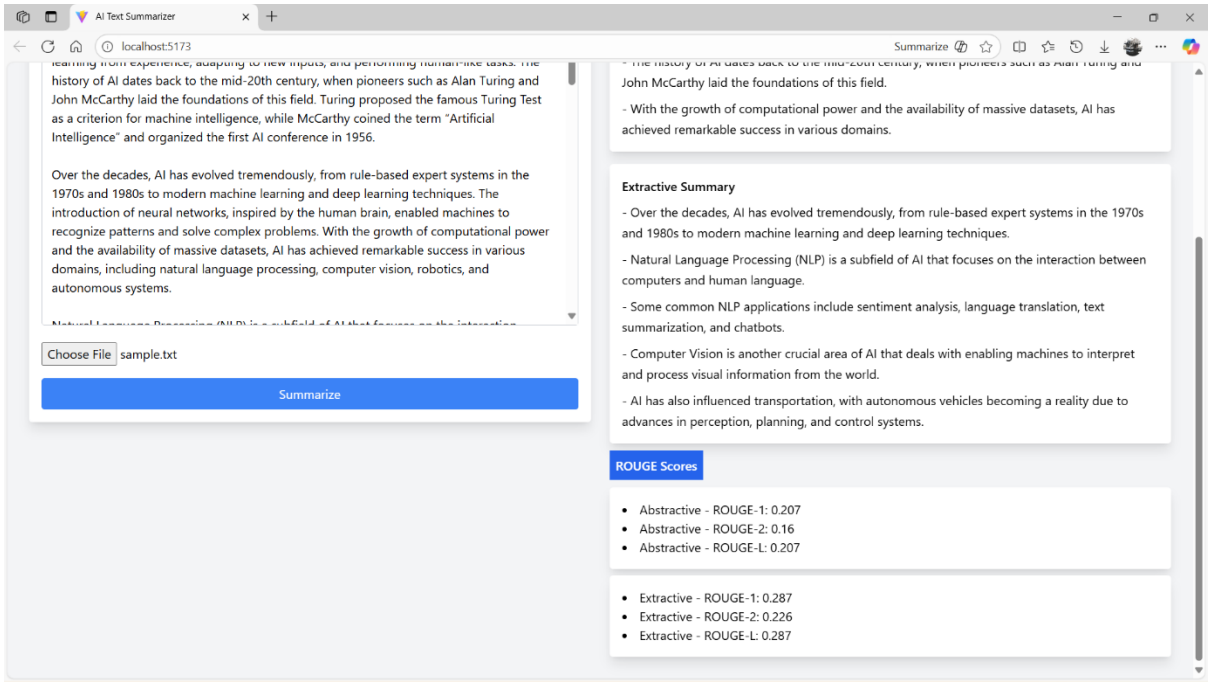

AI Text Summarizer
FastAPI - Swagger UI
127.0.0.1:8000/docs#/
FastAPI
0.1.0
OAS 3.1
/openapi.json
default
GET / Root
POST /api/files/extract Upload File Or Text
POST /api/files/summarize Upload And Summarize Files
POST /api/summaries Summarize With Type
Schemas
Body_upload_and_summarize_files_api_files_summarize_post
Body_upload_file_or_text_api_files_extract_post
HTTPValidationError
TextRequest
Expand all
object


AI Text Summarizer
FastAPI - Swagger UI
localhost:5173/multi
Summarize
AI Text Summarizer
Single File
Multi File
AI Multi-File Summarizer
Upload Files
Choose Files
2 files
Combine Summary
Summarize
abstract_summary.txt
extract_summary.txt
Merged Files
Abstractive Summary:
- The size and shape of the public domain shifts depending on the territory of the creator as well as the territory in which one seeks to exploit a derivative work.
- UK copyright in previously unpub-lished work does not presently expire until 1 January 2039.
- Once the term has passed, works become freely available for use.
- Public domain availability makes a significant difference to inclusion of images on Wikipedia.
- Pages with public domain images attracted between 17 and 19 per cent more visitors than pages where no image was available.
Extractive Summary:
- The perceived risk of a legal challenge may dissuade creators from making use of ideas and works in the public domain.
- Ronan Deazley defines the contours of the public domain as a series of categories incorporating:10 (i) those works which do not qualify for copyright protection; (ii) those works which do but are out of copyright term; (iii) those works where permission to use has been granted by the copyright owner a priori; and (iv) such parts of works which fall on the unprotectable side of the idea-expression line, which are allowed for within the statutory framework (taking of an insubstantial part, the permitted acts), or which are permissible as a result of judicial intervention with the regime at


AI Text Summarizer
FastAPI - Swagger UI
localhost:5173/multi
Summarize
- Once the term has passed, works become freely available for use.
- Public domain availability makes a significant difference to inclusion of images on Wikipedia.
- Pages with public domain images attracted between 17 and 19 per cent more visitors than pages where no image was available.
Extractive Summary:
- The perceived risk of a legal challenge may dissuade creators from making use of ideas and works in the public domain.
- Ronan Deazley defines the contours of the public domain as a series of categories incorporating:10 (i) those works which do not qualify for copyright protection; (ii) those works which do but are out of copyright term; (iii) those works where permission to use has been granted by the copyright owner a priori; and (iv) such parts of works which fall on the unprotectable side of the idea-expression line, which are allowed for within the statutory framework (taking of an insubstantial part, the permitted acts), or which are permissible as a result of judicial intervention with the regime at common law (on public policy grounds, or as being in the public interest).11 These interrelated defini- tions are expanded further, with reference to legal decisions concerning actual use where the legal boundary of copyright protection was uncertain.
- Furthermore, more recent material can be in the public domain due to being outside of the scope of copyright law, if it does not satisfy the necessary requirements to attract copyright, such as that it is not an expression, or does not meet the threshold of originality required for protection.
- The defendant further argued that similarities with the book Harry Potter and the Philosopher's Stone cannot be considered an imitation as these are also elements that Dmitri Yemets, like Rowling, took from the public domain.46 The District Court held, granting the application, that having regard to the high degree of similarity between the two stories, Yemet's book was an adaptation of Rowling's book that cannot be considered to be a new and orig- inal work for the purposes of Article 13 of the Dutch Copyright Act 1912 (as amended): It must be assumed in these proceedings that a storyline (a worked-out plot) can have an adequate character of its own to be considered a work for the purposes of the Copyright Act.
- Byblos' argument that Rowling has used elements from the public domain in her book Harry Potter and the Philosopher's Stone does not in principle affect the possibility that the Rowling's book (in which the storyline is developed) is a work for the purposes of the Copyright Act.47 On the point of parody, the Court held that 'by and large, a ridiculous (ironic) imitation of a work can be considered as a parody, where that work becomes the subject of laughter and when the contrast with the original work domi- nates'.48 The aim should be humour and not competition.
ROUGE Scores
Abstractive - ROUGE-1: 0.185
Abstractive - ROUGE-2: 0.119
Abstractive - ROUGE-L: 0.185
Extractive - ROUGE-1: 0.487
Extractive - ROUGE-2: 0.449
Extractive - ROUGE-L: 0.487


AI Text Summarizer
FastAPI - Swagger UI
localhost:5173
Summarize
AI Text Summarizer
Single File
Multi File
AI Single File Text Summarizer
Input Text / Upload File
Authors' note: This is the pre-print author-submitted version.
Please reference as:
Erickson, K., Kretschmer M. and Mendis D. (2019) 'An empirical approach to the public domain';
Chapter 4 (pp. 87-117) in J. Drexl and A. Kamperman Sanders (eds.) The Innovation Society and
Intellectual Property; Cheltenham, UK: Edward Elgar.
An empirical approach to the public domain
Kris Erickson, Martin Kretschmer and Dinusha Mendis
1 INTRODUCTION
Creative practices in the digital era and access to the reservoir of works and
ideas that we call 'the public domain' are intertwined in complex and comple-
mentary ways.1 Digitalisation has made it possible for the first time for users
to access a near infinitude of works – including copyright works – as well as to
create and distribute digital copies of works across borders and through a range
of technical devices. The features of digital technology that make this possible
Choose File
Erickson_Kre...ter_4_PD.pdf
Summarizing...
Abstractive Summary
Summary will appear here...
Extractive Summary
Summary will appear here...


AI Text Summarizer
localhost:5173
Summarize
AI Text Summarizer
Single File
Multi File
AI Single File Text Summarizer
Input Text / Upload File
Artificial Intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think and act like humans. These intelligent machines are capable of learning from experience, adapting to new inputs, and performing human-like tasks. The history of AI dates back to the mid-20th century, when pioneers such as Alan Turing and John McCarthy laid the foundations of this field. Turing proposed the famous Turing Test as a criterion for machine intelligence, while McCarthy coined the term "Artificial Intelligence" and organized the first AI conference in 1956.
Over the decades, AI has evolved tremendously, from rule-based expert systems in the 1970s and 1980s to modern machine learning and deep learning techniques. The introduction of neural networks, inspired by the human brain, enabled machines to recognize patterns and solve complex problems. With the growth of computational power and the availability of massive datasets, AI has achieved remarkable success in various domains, including natural language processing, computer vision, robotics, and autonomous systems.
Choose File
sample.txt
Summarize
Abstractive Summary
- Artificial Intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think and act like humans.
- The history of AI dates back to the mid-20th century, when pioneers such as Alan Turing and John McCarthy laid the foundations of this field.
- With the growth of computational power and the availability of massive datasets, AI has achieved remarkable success in various domains.
Extractive Summary
- Over the decades, AI has evolved tremendously, from rule-based expert systems in the 1970s and 1980s to modern machine learning and deep learning techniques.
- Natural Language Processing (NLP) is a subfield of AI that focuses on the interaction between computers and human language.
- Some common NLP applications include sentiment analysis, language translation, text summarization, and chatbots.
- Computer Vision is another crucial area of AI that deals with enabling machines to interpret and process visual information from the world.
- AI has also influenced transportation, with autonomous vehicles becoming a reality due to advances in perception, planning, and control systems.


AI Text Summarizer
localhost:5173
Summarize
history of AI dates back to the mid-20th century, when pioneers such as Alan Turing and John McCarthy laid the foundations of this field. Turing proposed the famous Turing Test as a criterion for machine intelligence, while McCarthy coined the term "Artificial Intelligence" and organized the first AI conference in 1956.
Over the decades, AI has evolved tremendously, from rule-based expert systems in the 1970s and 1980s to modern machine learning and deep learning techniques. The introduction of neural networks, inspired by the human brain, enabled machines to recognize patterns and solve complex problems. With the growth of computational power and the availability of massive datasets, AI has achieved remarkable success in various domains, including natural language processing, computer vision, robotics, and autonomous systems.
Choose File
sample.txt
Summarize
John McCarthy laid the foundations of this field.
- With the growth of computational power and the availability of massive datasets, AI has achieved remarkable success in various domains.
Extractive Summary
- Over the decades, AI has evolved tremendously, from rule-based expert systems in the 1970s and 1980s to modern machine learning and deep learning techniques.
- Natural Language Processing (NLP) is a subfield of AI that focuses on the interaction between computers and human language.
- Some common NLP applications include sentiment analysis, language translation, text summarization, and chatbots.
- Computer Vision is another crucial area of AI that deals with enabling machines to interpret and process visual information from the world.
- AI has also influenced transportation, with autonomous vehicles becoming a reality due to advances in perception, planning, and control systems.
ROUGE Scores
- Abstractive - ROUGE-1: 0.207
- Abstractive - ROUGE-2: 0.16
- Abstractive - ROUGE-L: 0.207
- Extractive - ROUGE-1: 0.287
- Extractive - ROUGE-2: 0.226
- Extractive - ROUGE-L: 0.287