TEXT SUMMARIZATION . LARGE LANGUAGE MODELS
GENERATIVE AI

MINI PROJECT

- **Title:-**

  Text Summarization using LLM

- **Domain:-**

  Machine Learning (Generative AI)

Project Guide:
Dr. DVSS Subrahamanyam Sir
HoD, CSE

By:
Manchala Sai Venkata Krishna Bhargav
245521733-303

Problem Statement

- Develop a comprehensive application that utilizes open-source large language models (LLMs) to process and analyze various types of data inputs, including PDF files, images containing text, and plain text files. The application will provide two main functionalities:

- 1. Text Summarization
- 2. Question Answering (QA) Chatbot

Abstract

- Text summarization is a critical task in natural language processing (NLP) that seeks to condense large volumes of text into concise, informative summaries, preserving essential information and key insights. This project, "Text Summarization and Question Answering Using Open-Source Large Language Models (LLMs)," leverages advanced machine learning techniques and state-of-the-art LLMs to develop a powerful, versatile tool for both text summarization and interactive question answering (QA).

- The application is designed to process various types of text input, including PDFs, images containing text, and plain text files, and provide summaries in English. It utilizes open-source LLMs, ensuring efficiency, accuracy, and flexibility in handling text summarization tasks. The tool also features an interactive QA chatbot that allows users to ask questions related to the uploaded content and receive precise answers, enhancing user engagement and providing quick access to specific information.

- Key Features :
    1. Text, PDF and Image Summarization
    2. Interactive Q&A.

- This project aims to simplify the extraction of meaningful summaries from extensive documents and images, providing valuable tools for researchers, students, and professionals to manage information overload and enhance productivity.

Introduction

- **Overview:**

 - Text summarization is a crucial task in natural language processing (NLP).

 - The project aims to condense large volumes of text into concise summaries, preserving key information and insights.

- **Technologies Used:** State-of-the-art open-source Large Language Models (LLMs).
- **Key Features:**

  Summarization of text, PDFs, and images.

  Interactive Q&A.

- **Target Audience:** Researchers, students, and professionals seeking to manage information overload and enhance productivity.
- **Project Goals:**

 - Simplify the extraction of meaningful summaries from extensive documents and images.

 - Provide valuable tools to manage large volumes of information efficiently.

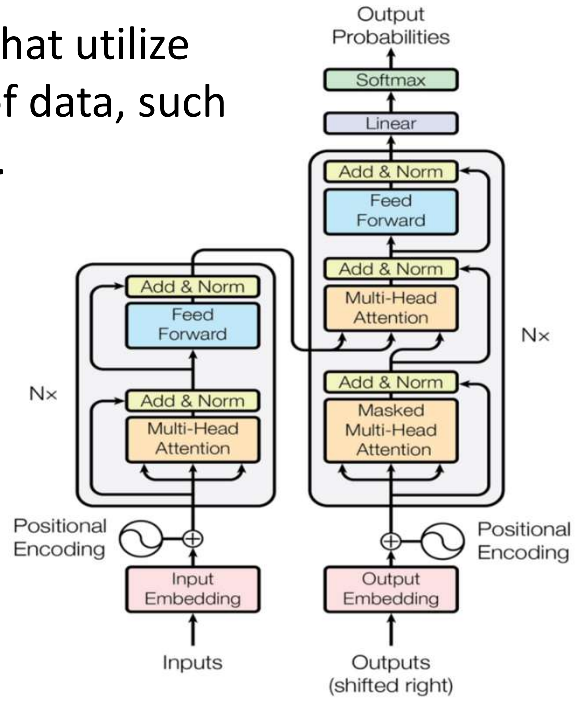
Existing System and its disadvantages:

- Other LLM’s (Chat-GPT , Claud ai , Gemini, Bing copilot etc) , But they all are **online** models

- Provide only limited access for pdf/image processing (in free version)

What is a Transformer?

Transformers in generative AI are neural network architectures that utilize self-attention mechanisms to process and generate sequences of data, such as text, enables translation, summarization, and text generation.



Transformer Architecture:

- Paper: Attention is All You Need
- Encoder-Decoder Structure
- Self-Attention Mechanism
- Multi-Head Attention
- Positional Encoding

- This application uses **T5 Transformer model** from Hugging Face, a leading provider of pre-trained language models.
- The T5 model is employed for **text summarization**, transforming input text into concise summaries.
- T5, with its 220 million parameters in the T5-Base configuration, is capable of handling diverse **NLP tasks**, including translation, summarization, and question answering.

What is Langchain?

Langchain is a framework designed for building applications that integrate with Large Language Models (LLMs), allowing developers to create sophisticated AI-driven tools and workflows.

Why is Langchain Used?

LangChain is used to simplify the development of applications that require interaction with LLMs. It provides modular components and utilities for managing prompts, chaining tasks, integrating with data sources, and handling complex workflows involving LLMs.

How is Langchain Used?

- Define the use case
- Select and configure the LLM
- Integrate modular components (e.g., prompt templates, chains, memory)
- Integrate external data sources if needed
- Manage and create tailored prompts
- Develop chains for linking multiple tasks
- Test and iterate based on performance
- Deploy the application
- Monitor and maintain the application

🦜🔗 LangChain

Applications:

- Text Summarization, Question Answering, Automated Writing, Data Integration, Workflow Automation

What are chunks?

- **Chunks**:

Smaller segments of a large text document, created to fit within the token limit of a Large Language Model (LLM).

- **How Chunks are Loaded to LLM:**

- Splitting: The large text is split into chunks that are within the LLM's token limit.

- Sequential Loading: Chunks are loaded one at a time into the LLM for processing.

- Processing: Each chunk is independently processed by the LLM, either for summarization or other tasks.

- **How Chunks are Used in Text Summarization:**

- Chunk Summarization: The LLM generates summaries for each individual chunk.

- Aggregation: The summaries of all chunks are combined to form a comprehensive summary of the entire text.

- Final Summarization: Optionally, the aggregated summaries can be further summarized into a more concise final output.

What is LLM?

- An LLM (Large Language Model) is a neural network trained on extensive text data to understand and generate human-like language.

- **How does an LLM work?**

It processes text input through layers of neurons, using attention mechanisms to focus on relevant parts of the text. (Transformers). Hugging Face community consists of few thousands of LLM’s.

HUGGING FACE

- **What does an LLM use?**

It uses large datasets, complex algorithms, and significant computational power to learn language patterns.

- **Why is an LLM used?**

It is used for tasks like text generation, translation, summarization, and question answering.

- **How is an LLM different from other models?**

It differs by its scale, ability to generate context-aware responses, and its versatility in handling a wide range of language tasks.

- **Popular LLMs: GPT-4** and **GPT-3.5** (OpenAI), **BERT** and **T5** (Google), **RoBERTa** (Facebook AI), **XLNet** (Google/CMU), **ERNIE** (Baidu), **LaMDA** (Google), **Claude** (Anthropic), **LLaMA** (Meta), **Mistra**l (Mistral AI), and **Bloom** (BigScience).

LLM Comparison

Model	**Developer**	**Focus**	**Parameters**
Llama 3	Meta	Natural Language Processing	8 Billion
LA-Mini Flant T5 248M	Hugging Face	Text Generation	248 Million
GPT-4	OpenAI	General-Purpose Language Model	1.75 Trillion
PaLM	Google	General-Purpose Language Model	540 Billion

LaMini-LM Flan T5 248M

Model Description: LaMini-Flan-T5-248M

- T5 is a tokenizer from Google
- LaMini-Flan-T5-248M is a fine-tuned version of the Flan-T5 model, optimized for summarization tasks.
- It has 248M parameters and was developed as part of the La-Mini-LM model series.
- The model is fine-tuned with a dataset of 2.58M samples for instruction-based tasks, ensuring high-quality output while being more resource-efficient compared to larger models like GPT-3.5-turbo.
- Specially used for Text Summarization and text-to-text generation tasks

Key Features:

- Instruction Fine-Tuning: The model has been fine-tuned for instruction-based tasks, making it particularly effective in generating summaries.

- Model Distillation: Derived from larger models to retain generative capabilities in a smaller, more efficient model.

Parameters for Inference:

- Learning Rate: 0.0005

- Train Batch Size: 128

- Eval Batch Size: 64

- Seed: 42

-- Total Train Batch Size: 512

- Optimizer: Adam with betas=(0.9,0.999) and epsilon=1e-08

- LR Scheduler Type: Linear

- Number of Epochs: 5

Proposed Logic

1. Upload Text
2. Extract text and split into chunks
3. Load LLM model
4. Create chain to connect loaded model
5. Prompt the input (Max Summary Length)
6. Summary generated
7. Prompt the question
8. Answer is obtained

Technology Stack

1. Platform:

Streamlit

2. Technologies:

Python

PyTorch

Hugging Face Transformers

Langchain

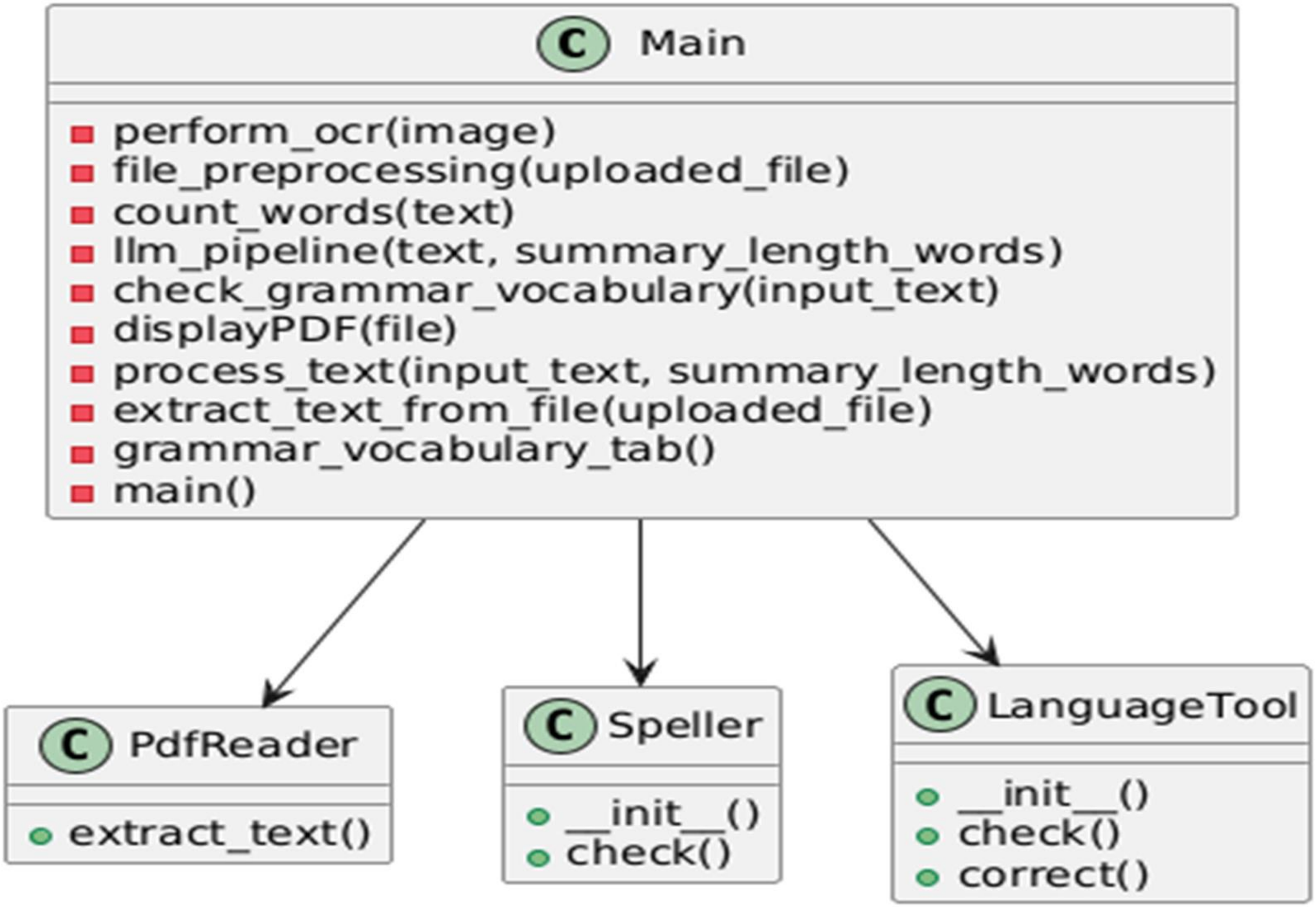
Tesseract OCR

3.Tools:

LaMini-Flan-T5-248M

UML Diagrams

Class Diagram



C Main

- perform_ocr(image)
- file_preprocessing(uploaded_file)
- count_words(text)
- llm_pipeline(text, summary_length_words)
- check_grammar_vocabulary(input_text)
- displayPDF(file)
- process_text(input_text, summary_length_words)
- extract_text_from_file(uploaded_file)
- grammar_vocabulary_tab()
- main()

C PdfReader

- extract_text()

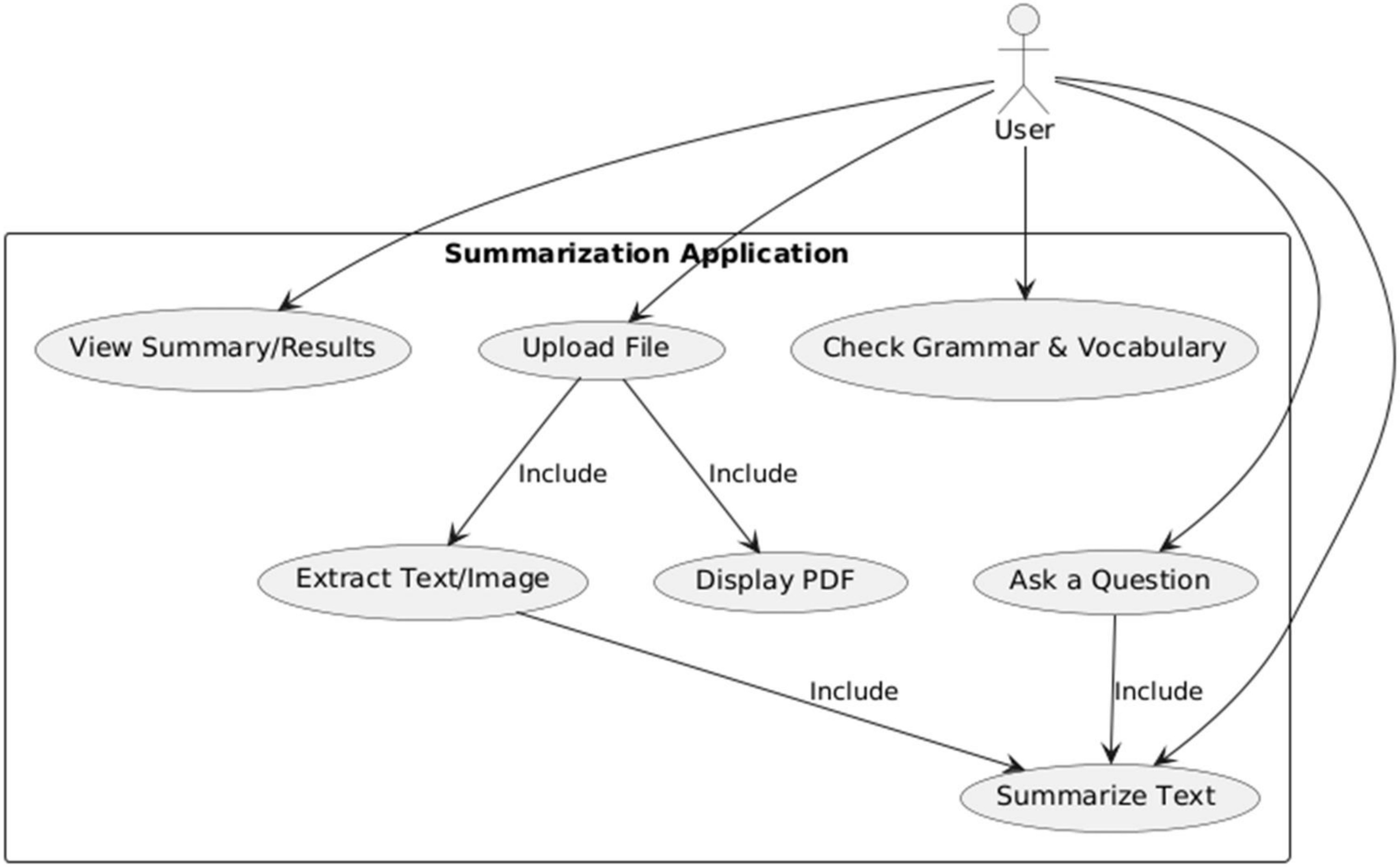
C Speller

- __init__()
- check()

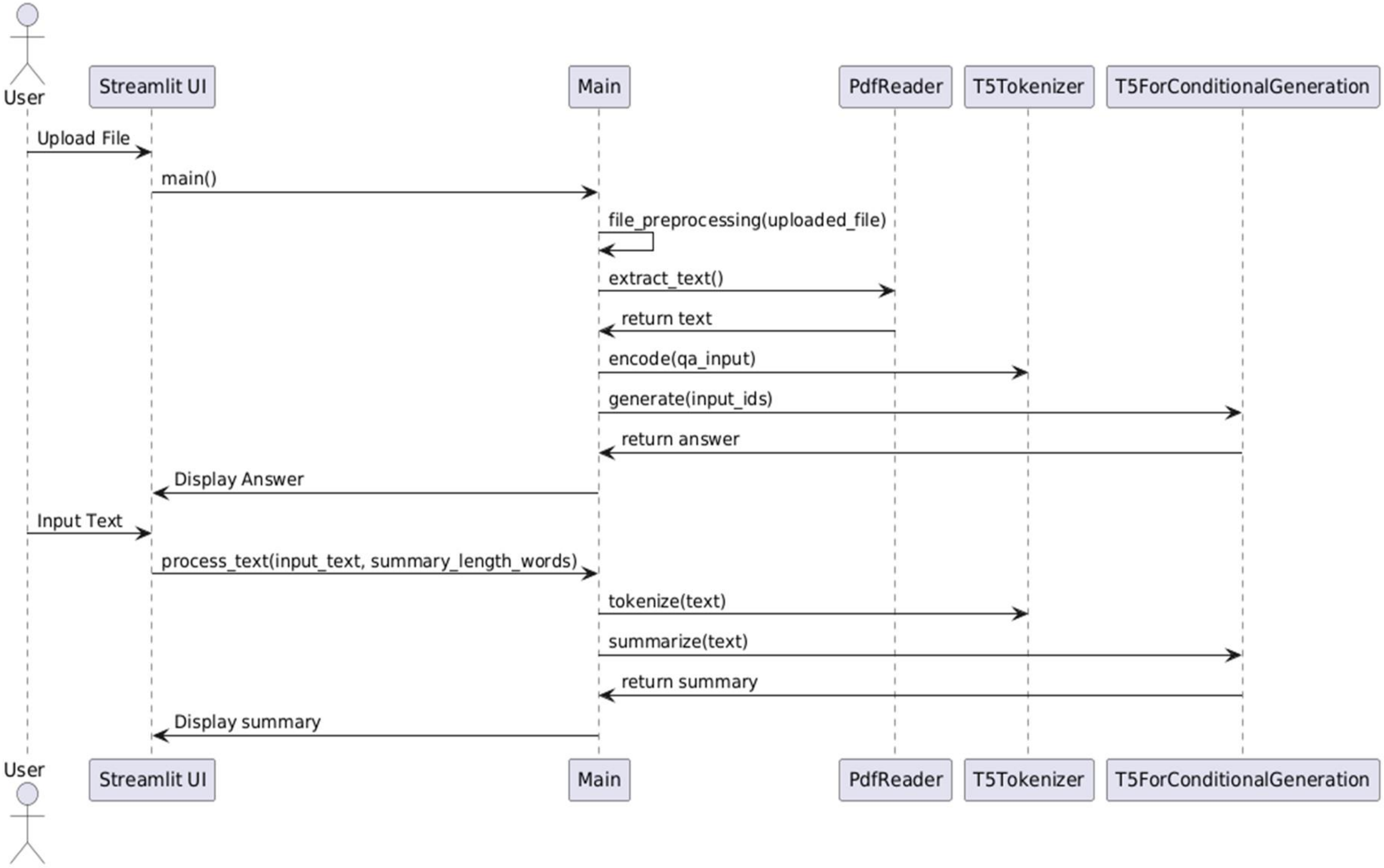
C LanguageTool

- __init__()
- check()
- correct()

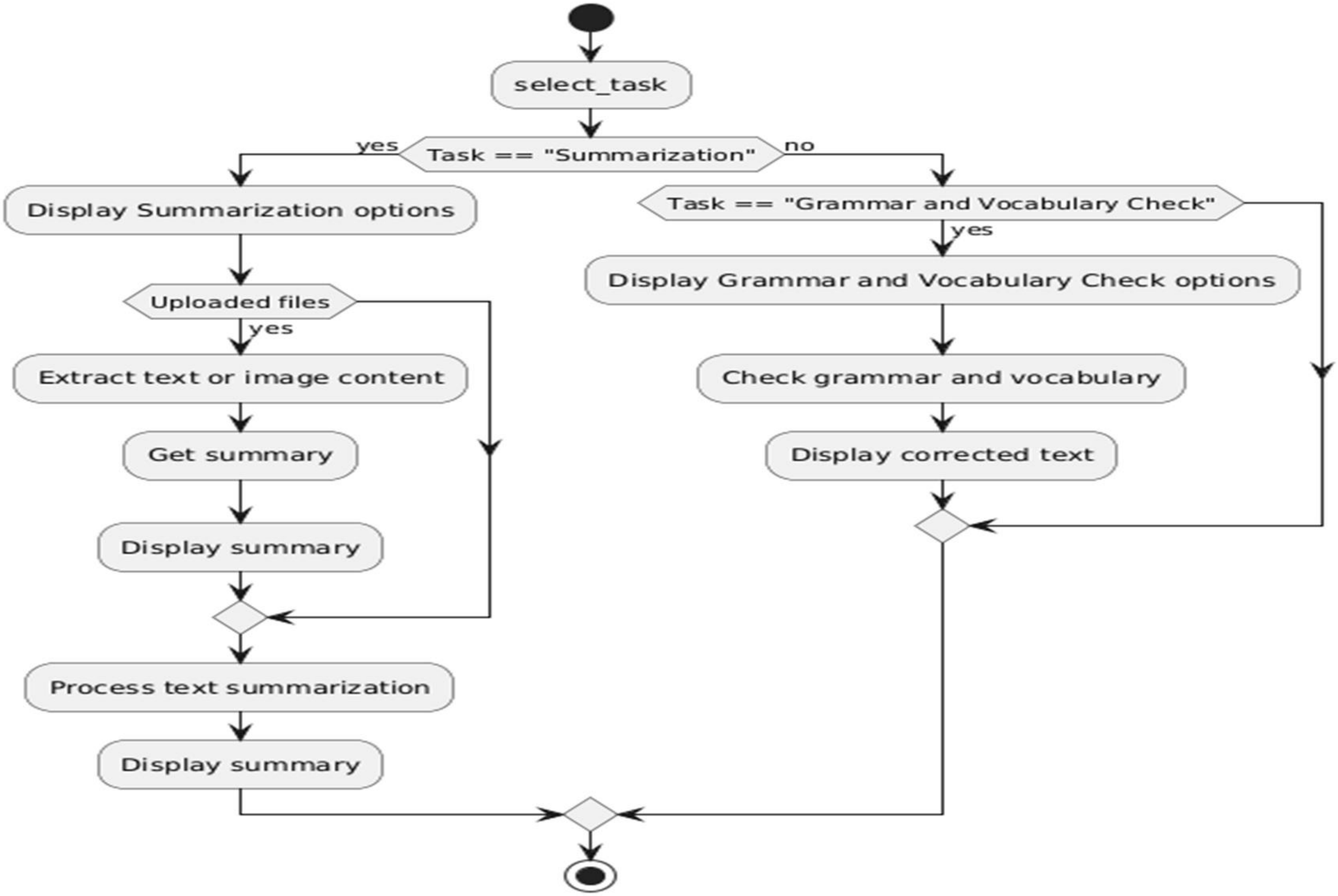
Use case Diagram



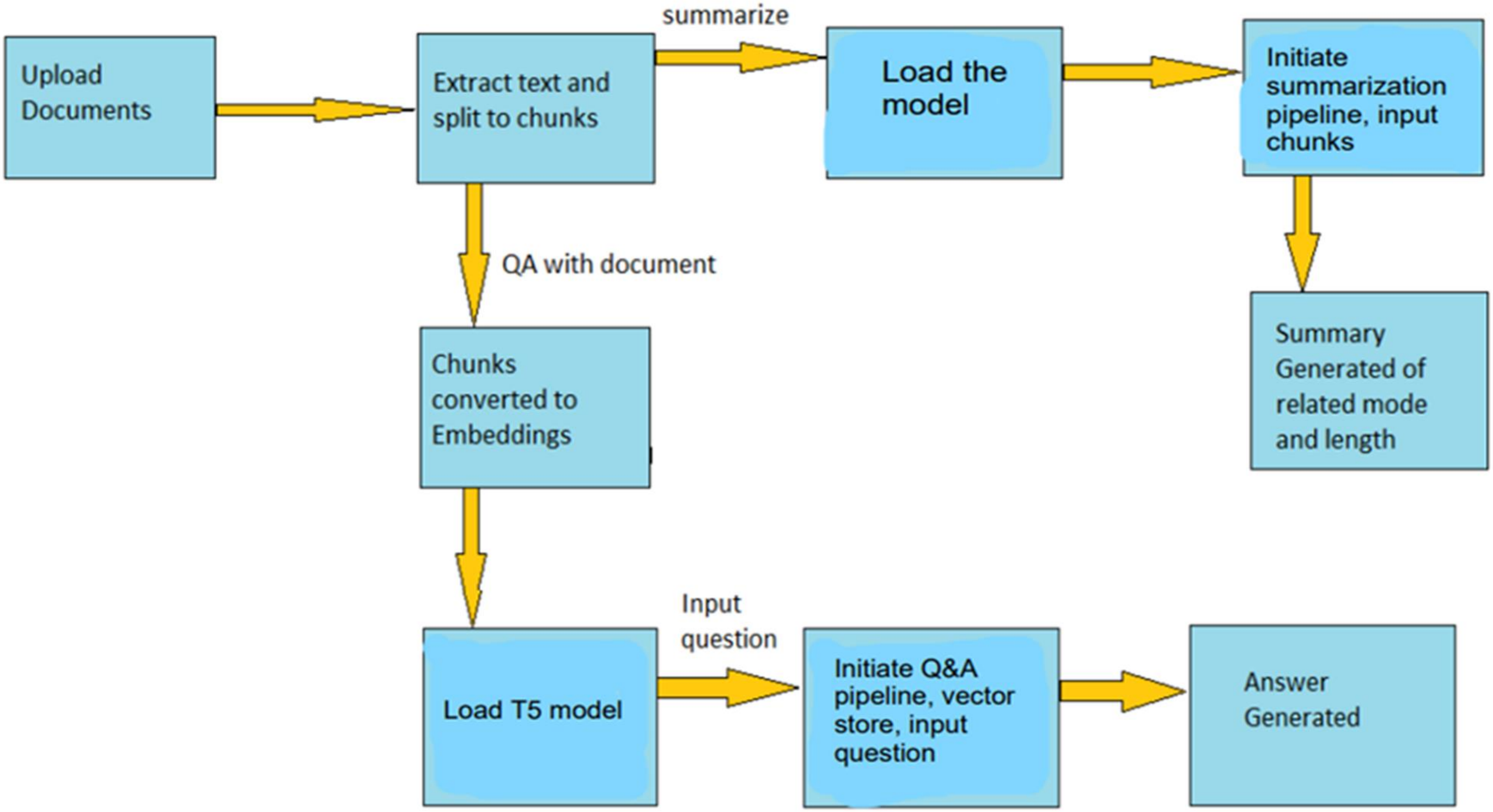
Sequence Diagram



Activity Diagram

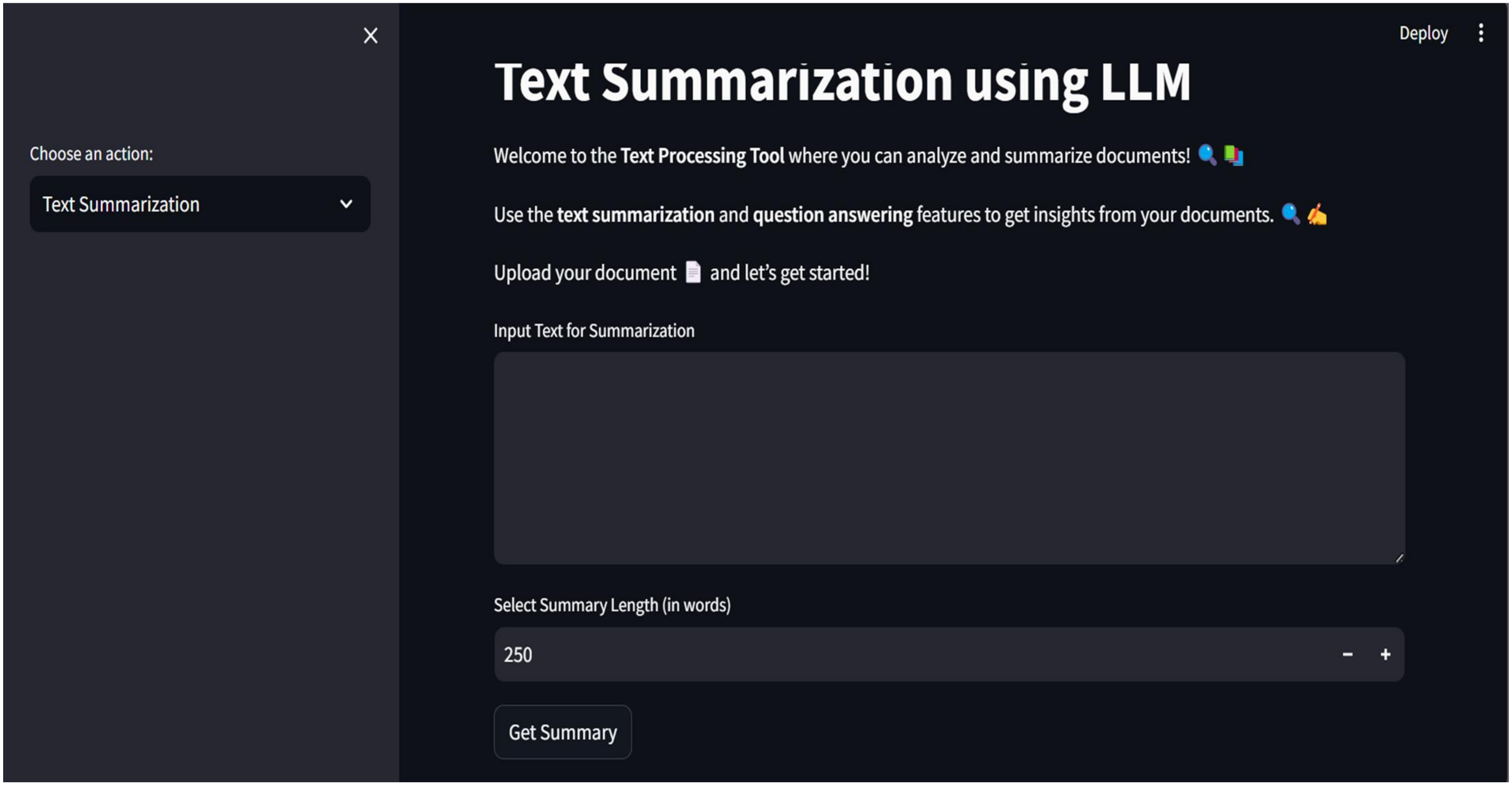

select_task
Task == "Summarization"
yes
no
Display Summarization options
Uploaded files
yes
Extract text or image content
Get summary
Display summary
Process text summarization
Display summary
Task == "Grammar and Vocabulary Check"
yes
Display Grammar and Vocabulary Check options
Check grammar and vocabulary
Display corrected text

Implementation


Upload Documents
Extract text and split to chunks
summarize
Load the model
Initiate summarization pipeline, input chunks
Summary Generated of related mode and length
QA with document
Chunks converted to Embeddings
Load T5 model
Input question
Initiate Q&A pipeline, vector store, input question
Answer Generated

Output Screens

Text Summarization Page



Choose an action:

Text Summarization ⌄

Text Summarization using LLM

Welcome to the **Text Processing Tool** where you can analyze and summarize documents! 🔍 📚

Use the **text summarization** and **question answering** features to get insights from your documents. 🔍 ✍️

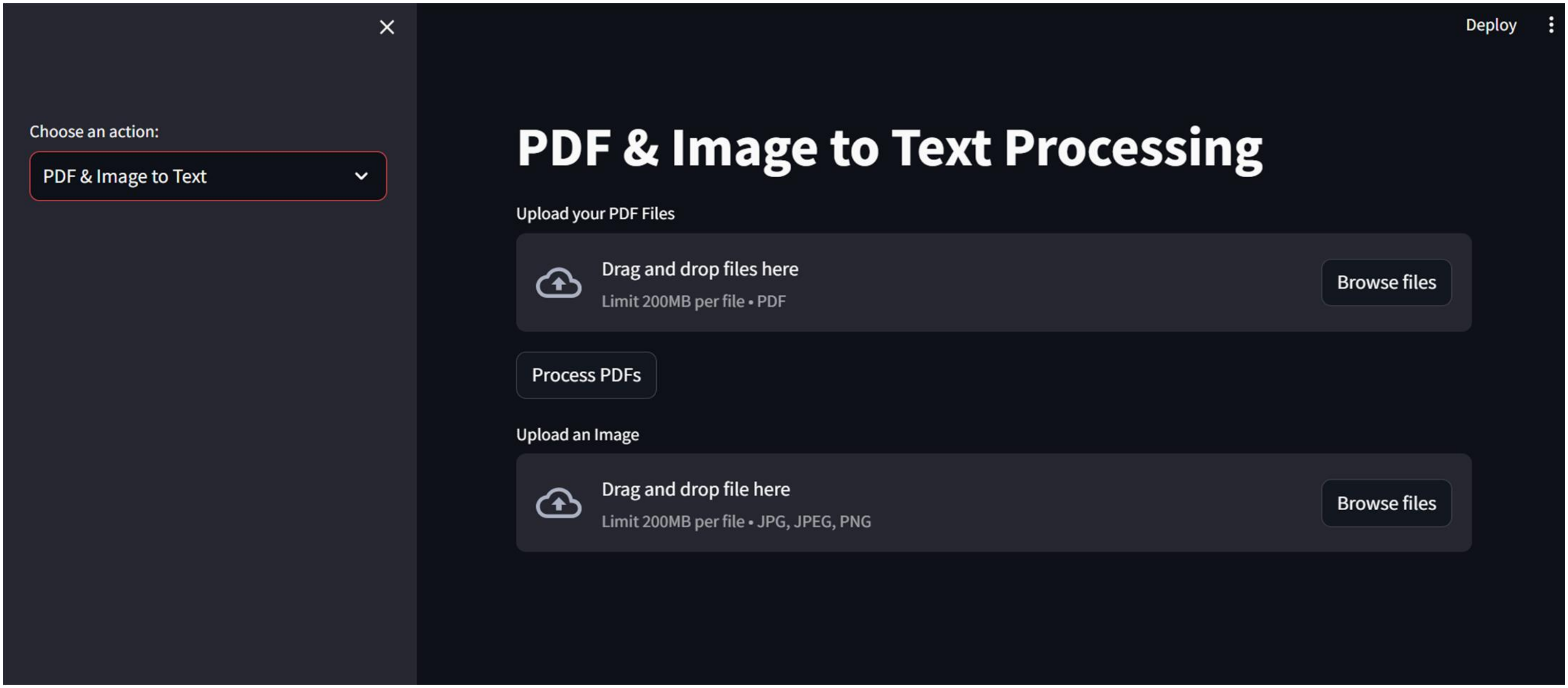
Upload your document 📄 and let's get started!

Input Text for Summarization

Select Summary Length (in words)

250

PDF & Image to Text Processing Page



Choose an action: PDF & Image to Text

PDF & Image to Text Processing

Upload your PDF Files

Drag and drop files here

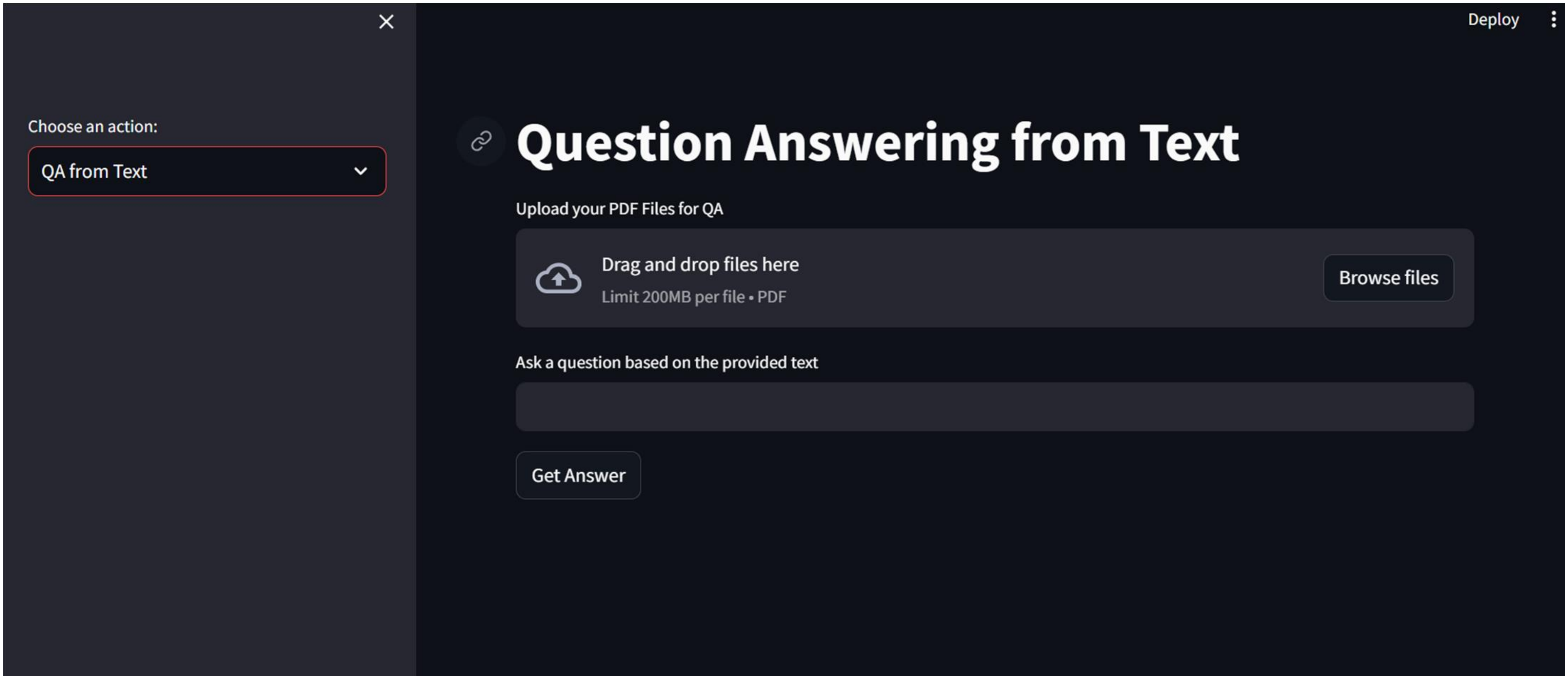
Limit 200MB per file • PDF

Upload an Image

Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG

QA Page



Choose an action:

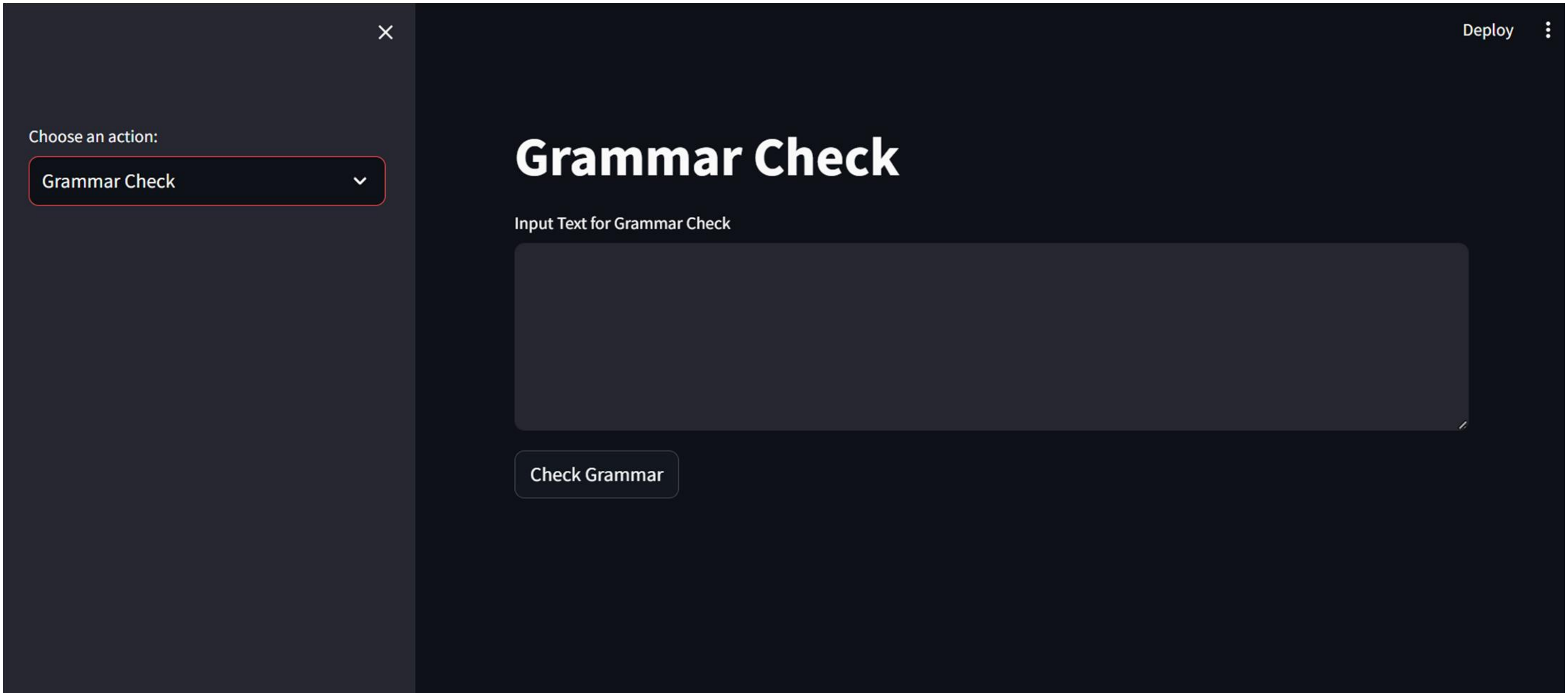
QA from Text

Question Answering from Text

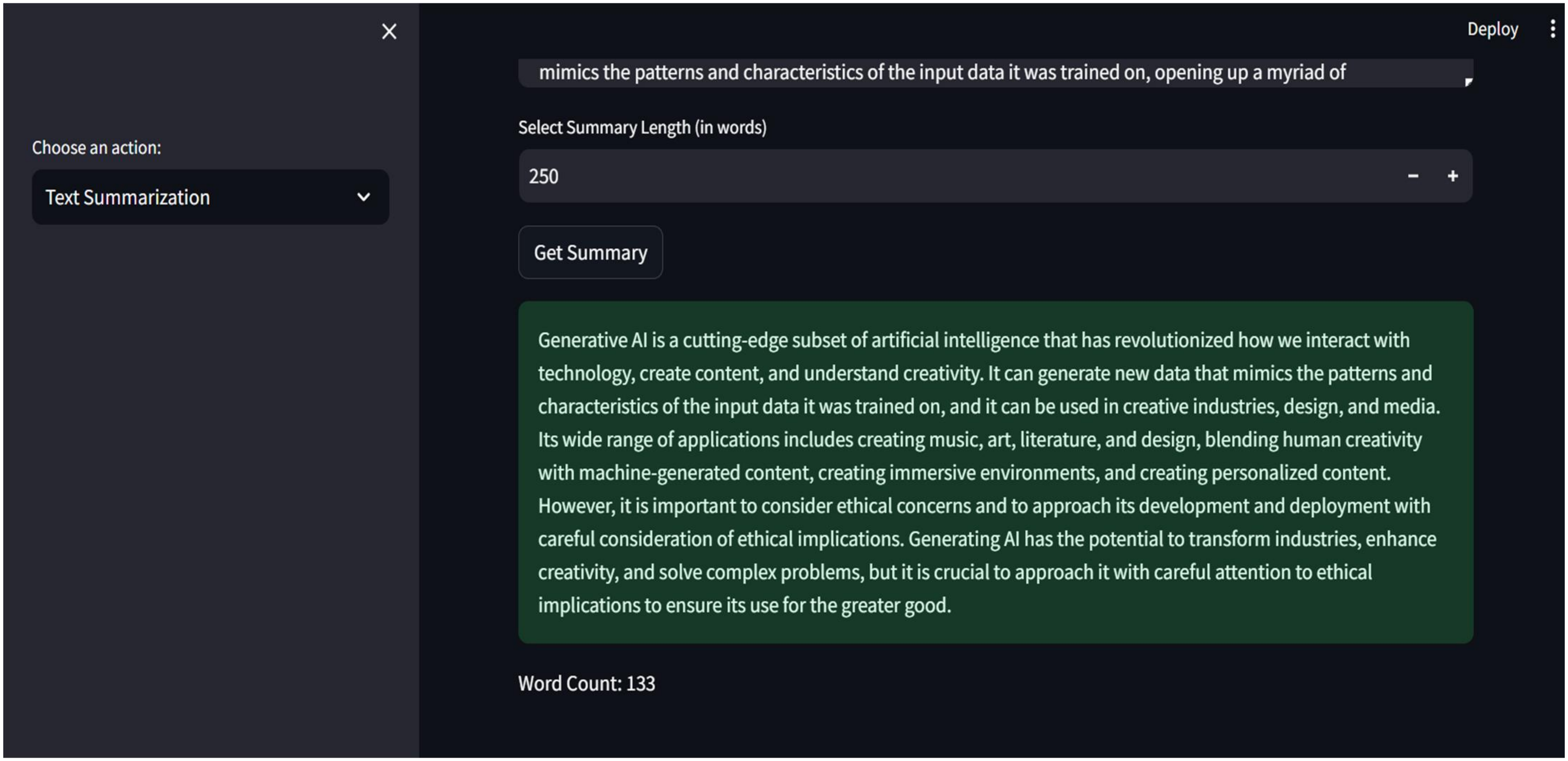
Upload your PDF Files for QA

Ask a question based on the provided text

Grammar Check Page



Summary of input Text



Choose an action:

Text Summarization

mimics the patterns and characteristics of the input data it was trained on, opening up a myriad of

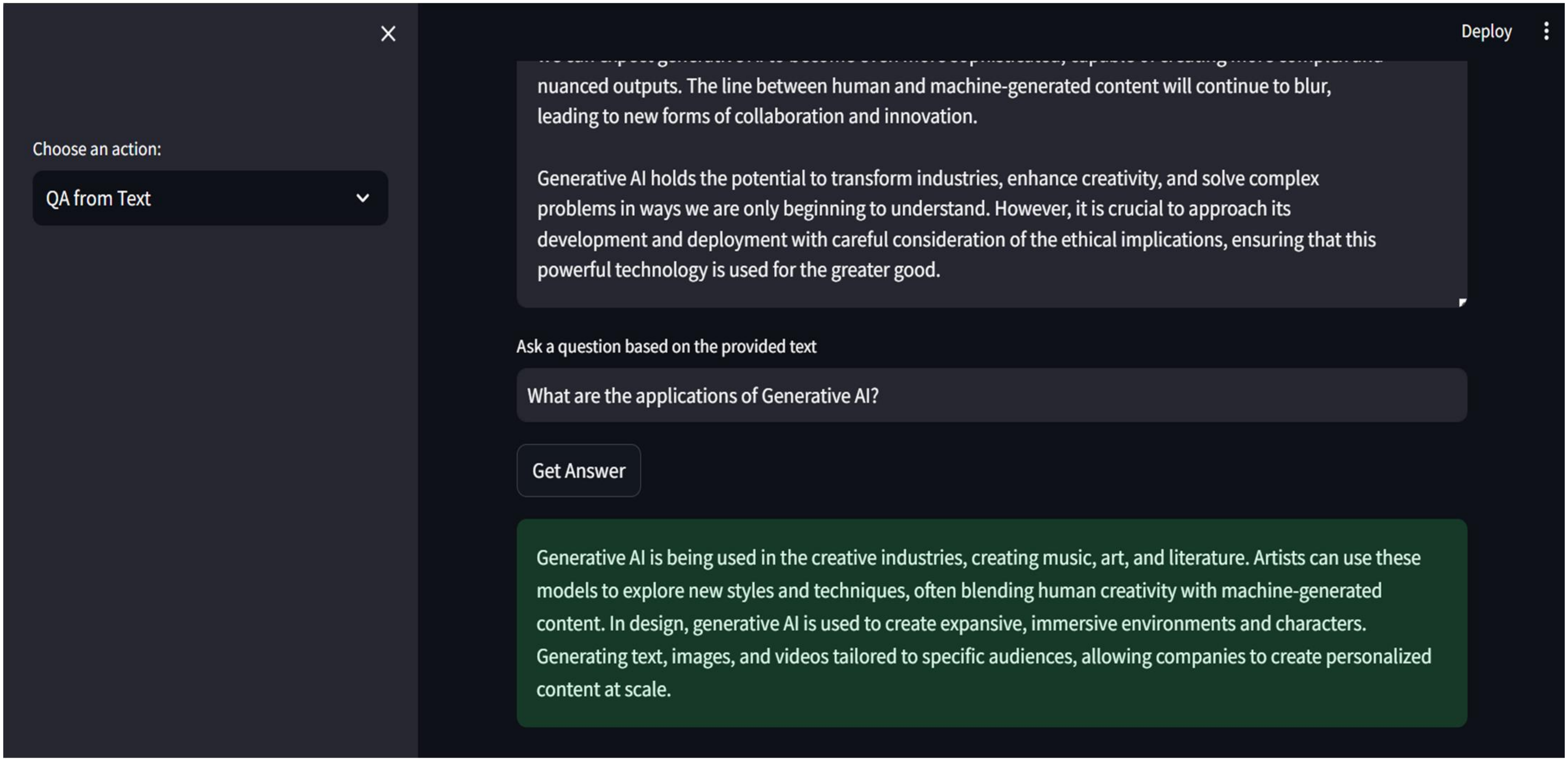
Select Summary Length (in words)

250

Generative AI is a cutting-edge subset of artificial intelligence that has revolutionized how we interact with technology, create content, and understand creativity. It can generate new data that mimics the patterns and characteristics of the input data it was trained on, and it can be used in creative industries, design, and media. Its wide range of applications includes creating music, art, literature, and design, blending human creativity with machine-generated content, creating immersive environments, and creating personalized content. However, it is important to consider ethical concerns and to approach its development and deployment with careful consideration of ethical implications. Generating AI has the potential to transform industries, enhance creativity, and solve complex problems, but it is crucial to approach it with careful attention to ethical implications to ensure its use for the greater good.

Word Count: 133

Question Answering



Choose an action:

QA from Text

nuanced outputs. The line between human and machine-generated content will continue to blur, leading to new forms of collaboration and innovation.

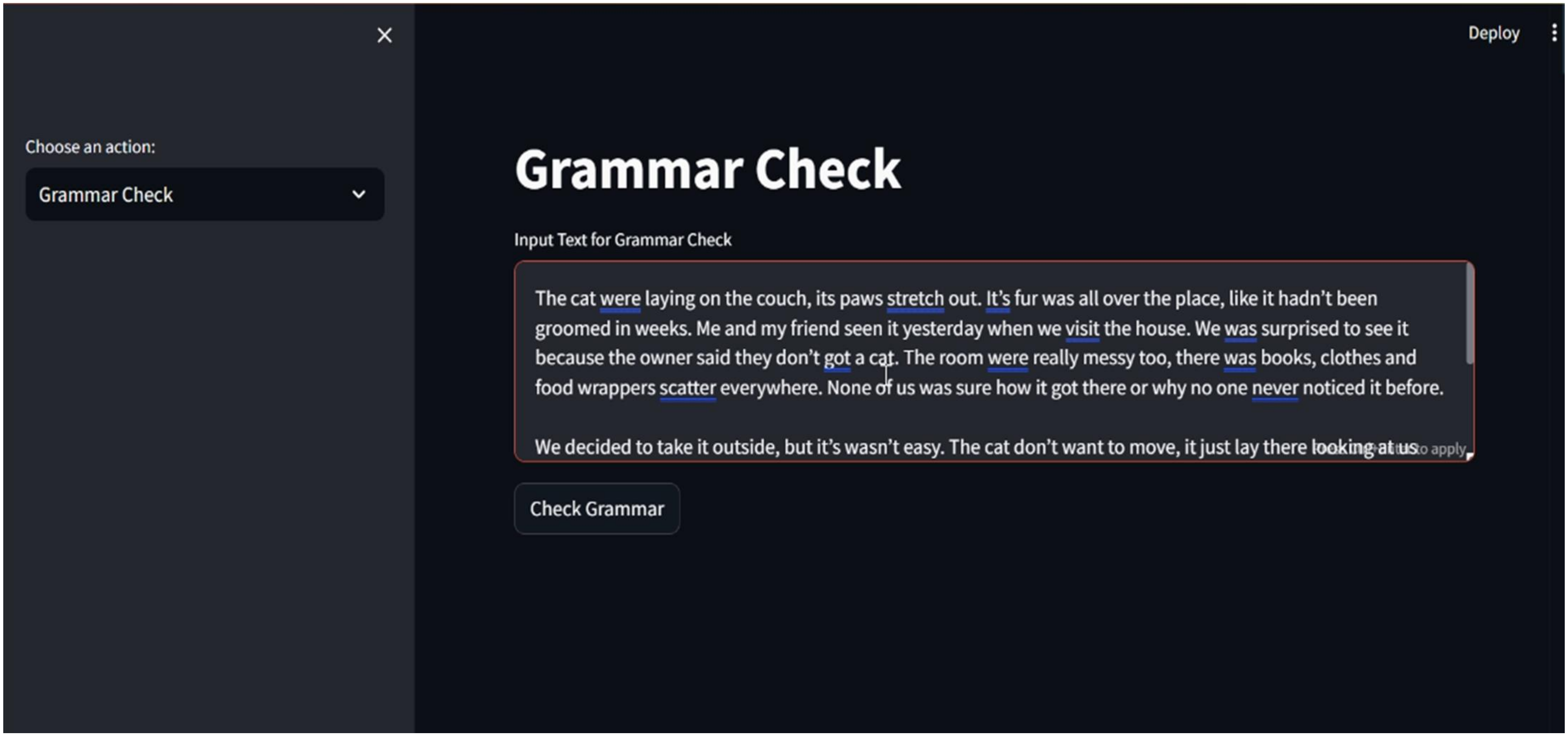
Generative AI holds the potential to transform industries, enhance creativity, and solve complex problems in ways we are only beginning to understand. However, it is crucial to approach its development and deployment with careful consideration of the ethical implications, ensuring that this powerful technology is used for the greater good.

Ask a question based on the provided text

What are the applications of Generative AI?

Generative AI is being used in the creative industries, creating music, art, and literature. Artists can use these models to explore new styles and techniques, often blending human creativity with machine-generated content. In design, generative AI is used to create expansive, immersive environments and characters. Generating text, images, and videos tailored to specific audiences, allowing companies to create personalized content at scale.

Grammar Check output



Grammar Check

Input Text for Grammar Check

The cat were laying on the couch, its paws stretch out. It's fur was all over the place, like it hadn't been groomed in weeks. Me and my friend seen it yesterday when we visit the house. We was surprised to see it because the owner said they don't got a cat. The room were really messy too, there was books, clothes and food wrappers scatter everywhere. None of us was sure how it got there or why no one never noticed it before.

We decided to take it outside, but it's wasn't easy. The cat don't want to move, it just lay there looking at us

Check Grammar

Conclusion and Future Scope

1.Conclusion

The application successfully integrates text extraction, summarization, and grammar correction into a single platform, improving accessibility and usability for users handling diverse text inputs.

2.Future Scope

Future enhancements could include:

- Adding support for additional file formats (e.g., DOCX).
- Improving model accuracy with larger data models.
- Expanding the system to support multilingual text processing.
- Summarizing YouTube videos using audio-to-text models (e.g., Whisper).
- Summarizing from video transcripts.
- Addressing real-time problems that can be solved using text summarization.

3.Github Link

www.https//github.com/bhargavmanchala/text-summarization-using-llm-miniproject

References

Papers:

1. LaMini-LM documentation

LaMini-LM: A Diverse Herd of Distilled Models from Large-Scale Instructions

2. Google-T5/ T5-base Model documentation

Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

Web Resources:

1. Hugging Face Transformers Documentation
2. LangChain Documentation
3. GitHub - AIAnytime
4. GitHub - Jalammar

THANK YOU