

---

# QA CHATBOT FOR RESEARCH PAPERS

TEAM MEMBERS:

NIRANJANI T(21Z326)

SHERIN CHRISTIANA(21Z347)

SREENITHI(21Z356)

GANGASRI(21I316)

---

# TABLE OF CONTENTS

01

PROBLEM  
STATEMENT

02

ABSTRACT

03

OVERVIEW

04

TECHSTACK

---

# TABLE OF CONTENTS

05

ARCHITECTURE &  
WORKFLOW

06

IMPLEMENTATION

07

CHALLENGES  
FACED

08

FUTURE  
ENHANCEMENTS

---

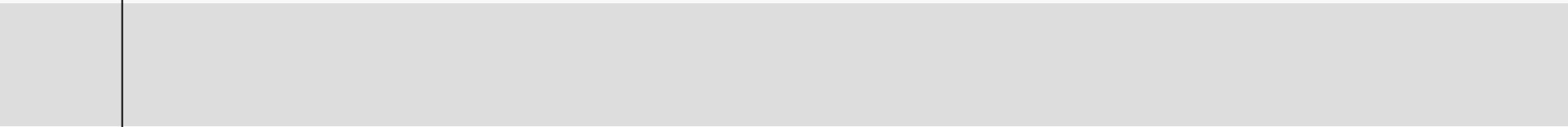
01

# PROBLEM STATEMENT

---

---

In today's digital age, vast amounts of information are stored in PDF documents, making it challenging for users to extract and understand relevant content quickly. Traditional methods of reading and summarizing text are often time-consuming and inefficient. Additionally, users frequently have specific questions about the content of these documents, requiring effective tools to retrieve precise answers from unstructured text. Therefore, there is a need for an automated solution that can:

- Extract text from PDF files effectively.
  - Summarize lengthy documents for quick comprehension.
  - Provide accurate answers to user queries based on the extracted content.
- 
- A solid gray horizontal bar spanning the width of the slide at the bottom.

02

# ABSTRACT

---

---

This project presents a web interface that leverages natural language processing (NLP) techniques to enhance the interaction with research papers. The application allows users to upload PDF files, from which it extracts and cleans the text for further processing. Utilizing state-of-the-art models, including T5 for summarization and RoBERTa and Llama for question answering, the application offers a user-friendly interface for obtaining concise summaries and accurate responses to user inquiries. This automated solution significantly improves the efficiency of information retrieval, empowering users to make better-informed decisions based on the content of their documents.

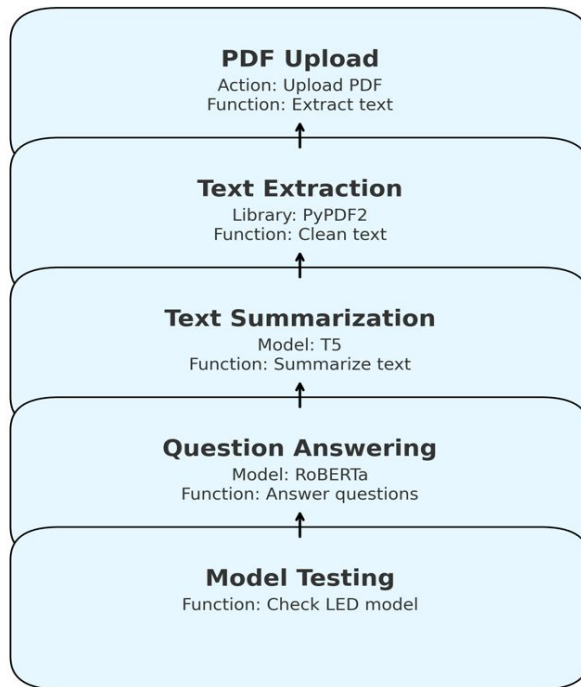
03

# PROJECT OVERVIEW

---



## Workflow Summary of the Project

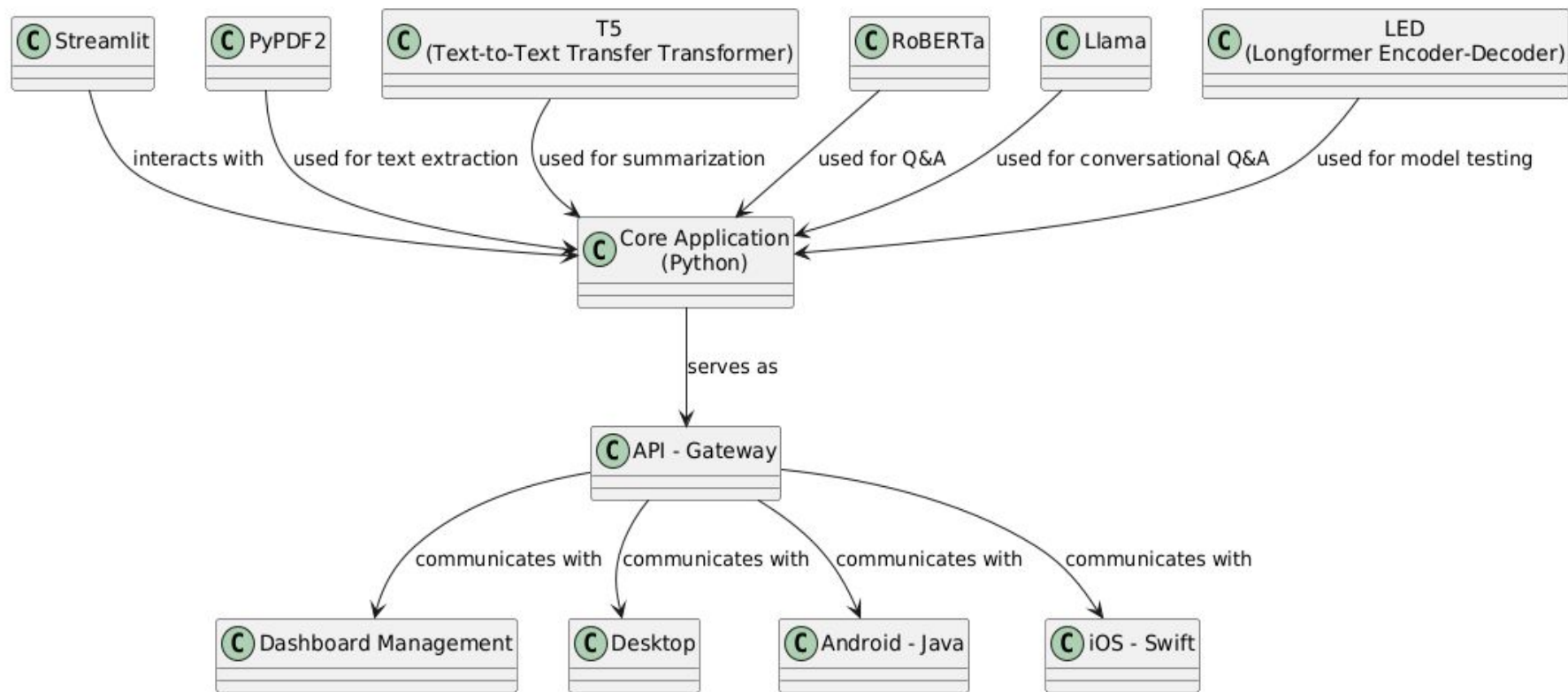


04

# TECH STACK

---

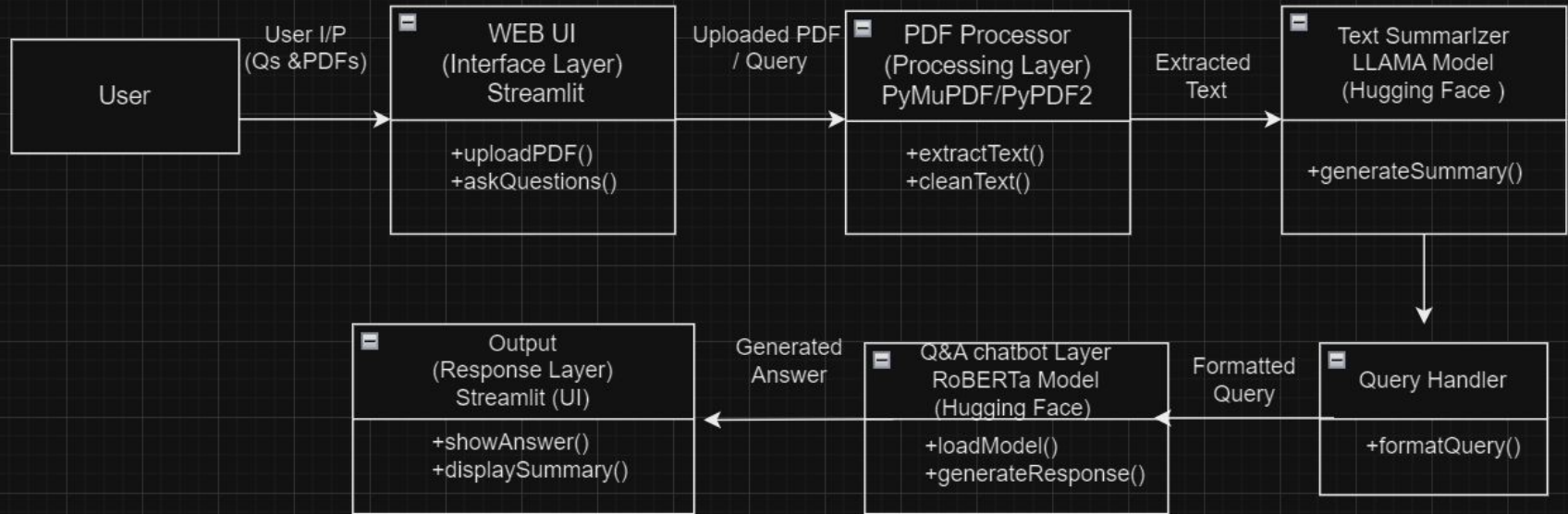
## Technology Stack



05

# ARCHITECTURE & WORKFLOW

---



# WORKFLOW

- **User Interaction**

**Action:** User uploads PDF documents and enters questions.

**Component:** Streamlit UI.

- **PDF Upload**

**Action:** The system receives the uploaded PDF files.

**Component:** PDF Processor Layer.

- **Text Extraction**

**Action:** Extract text from the uploaded PDFs.

**Methods:** Using PyPDF2 or PyMuPDF.

**Output:** Raw text data.

---

- **Text Cleaning**

**Action:** Clean the extracted text to remove unnecessary characters and formatting.

**Output:** Cleaned text ready for summarization.

- **Text Summarization**

**Action:** Generate concise summaries of the cleaned text.

**Model:** LLaMA (via Hugging Face Transformers).

**Output:** Summarized text.

- **Query Submission**

**Action:** User submits questions related to the PDF content.

**Component:** Query Handler Layer.

**Output:** Formatted query for processing.

---

- **Question Answering**

**Action:** Process the formatted query using the Q&A model.

**Model:** RoBERTa (via Hugging Face Transformers).

**Output:** Generated answer based on the PDF content.

- **Display Results**

**Action:** Display the summarized text and answers in the Streamlit UI.

**Output:** User receives relevant summaries and answers.

---

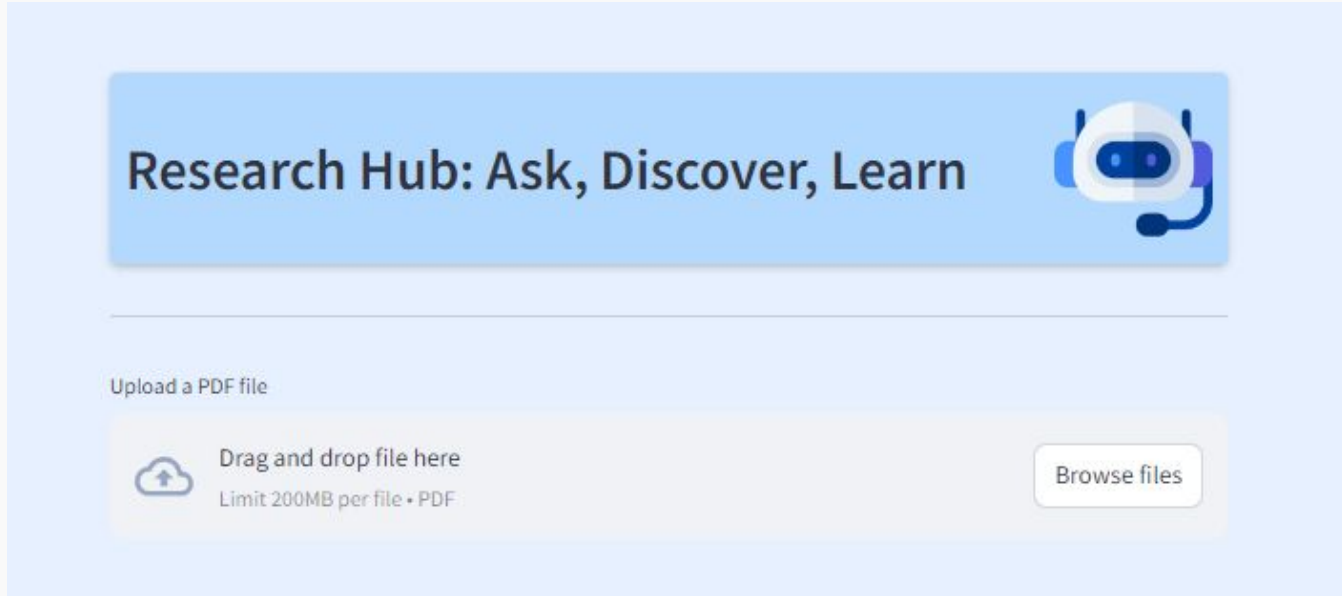


06

# Implementation

---

# Output Screenshot



Upload a PDF file



Drag and drop file here

Limit 200MB per file • PDF

Browse files



1705.07565.pdf 0.6MB



## Extracted Text

Extracted Text

Learning to Prune Deep Neural Networks via Layerwise Optimal Brain Surgeon Xin Dong Nanyang Technological University Singapore naentuedusg Shangyu Chen Nanyang Technological University Singapore schenentuedusg Sinno Jialin Pan Nanyang Technological University Singapore sinnopanntuedusg Abstract How to develop slim and accurate deep neural networks has become crucial for real world applications especially for those employed in embedded systems Though previous work along this research line has shown some promising results most existing methods either fail to significantly compress a welltrained deep network or require a heavy retraining process for the pruned deep network to reboost its prediction performance In this paper we propose a new layerwise pruning method for deep neural networks In our proposed method parameters of each individual layer are pruned independently based on second order derivatives of a layerwise error function with respect to the corresponding parameters We prove that the nal prediction performance drop after pruning is bounded by a linear combination of the reconstructed errors caused at each layer By controlling layerwise errors properly one only needs to perform a light retraining process on the pruned network to resume its original prediction performance We conduct extensive experiments on benchmark datasets to demonstrate the effectiveness of our pruning method compared with several stateoftheart baseline methods Codes of our work are released at <https://github.com/csyhhu/LOBS> Introduction Intuitively deep neural networks can approximate predictive functions of arbitrary complexity well when they are of a huge amount of parameters ie a lot of layers and neurons In practice the size of deep neural networks has been being tremendously increased from LeNet with less than M parameters to VGG with M parameters Such a large number of parameters not only make deep models memory intensive and computationally expensive but also urge researchers to dip into redundancy of deep neural networks On one hand in neuroscience

Summarize

Summarize

## Summary

in this paper we propose a new layerwise pruning method for deep neural networks. parameters of each individual layer are pruned independently based on second order derivatives of an error function with respect to the corresponding parameters We prove that nal prediction performance drop after pruning is bounded by the linear combination of reconstructed errors caused at each layer by each successive layer.

## Ask your Queries

Enter your question:

what is important topic it spoke about

## Answer

computer vision and pattern recognition

# Challenges Faced



## **Model Accuracy and Hallucination:**

- Open-source models used for summarization and Q&A might produce incorrect or "hallucinated" information.

## **Model Deployment and Integration:**

- Integrating large pre-trained models from Hugging Face into a web application requires careful management of dependencies, versions, and hardware limitations(some require GPU machine)

## **Scalability:**

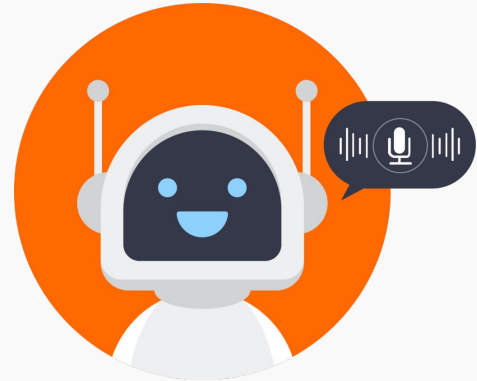
- Scaling the system to handle multiple concurrent users while maintaining quick response times could be challenging

## **Model Selection:**

- Choosing the right model for both summarization and question-answering is difficult. There are numerous available models, each varying in accuracy, speed, and resource requirements. The balance between model complexity and available hardware (CPU only) posed a significant challenge.

# Future Enhancements

1. **Model Optimization for CPU**
2. **Multi-document Summarization and Q&A**
3. **Multi-language Support:**
4. **Collaboration and Sharing**
5. **Voice-Enabled Q&A**



THANK YOU

---