

**TRIBHUVAN UNIVERSITY**

**INSTITUTE OF ENGINEERING**

**THAPATHALI CAMPUS**

**Proposal**

**On**

**Presentify: Presentation Slide Generation using NLP**

**& Deep Learning**

**Submitted By:**

Atul Shreewastav (THA077BCT013)

Bidhan Acharya (THA077BCT015)

Nischal Paudel (THA077BCT028)

Yugratna Humagain (THA077BCT047)

**Submitted To:**

Department of Electronics and Computer Engineering

Thapathali Campus

Kathmandu, Nepal

February, 2024



**TRIBHUVAN UNIVERSITY**

**INSTITUTE OF ENGINEERING**

**THAPATHALI CAMPUS**

**Proposal**

**On**

**Presentify: Presentation Slide Generation using NLP**

**& Deep Learning**

**Submitted By:**

Atul Shreewastav (THA077BCT013)

Bidhan Acharya (THA077BCT015)

Nischal Paudel (THA077BCT028)

Yugratna Humagain (THA077BCT047)

**Submitted To:**

Department of Electronics and Computer Engineering

Thapathali Campus

Kathmandu, Nepal

In partial fulfillment for the award of the Bachelor’s Degree in

Computer Engineering

**Under the Supervision of**

Er. Saroj Sakya

February, 2024

# ACKNOWLEDGEMENT

We would like to deeply thank Department of Electronics and Computer Engineering of Institute of Engineering, Thapathali Campus for allowing us to broaden our knowledge and concepts on programming with the accomplishment of this project. We are also thankful towards our teacher and supervisor Er. Saroj Sakya sir and our senior Rajendra Baskota for providing us the ideas and resources necessary for this project. This project would not be initiated without the guidance and supervision of our respected supervisor.

Atul Shreewastav (THA077BCT013)

Bidhan Acharya (THA077BCT015)

Nischal Paudel (THA077BCT028)

Yugratna Humagain (THA077BCT047)

# ABSTRACT

This project employs a fine-tuned T5 transformer model trained on custom data to automatically extract key points from computer science research articles and transform these key points into presentation slides. The goal is to streamline the extraction of key insights from diverse articles and transform them in the form of a presentation. The T5 model is proficient in text-to-text transfer tasks and is fine-tuned and evaluated using domain-specific metrics. The resulting system integrates this capability and transforms a research article into coherent presentation slides. This project holds potential for revolutionizing how technical information is summarized and presented, fostering clearer communication in the research community.

*Keywords: T5 Transformer, Transfer Learning, Gemini Pro Model*

TABLE OF CONTENT

[ACKNOWLEDGEMENT i](#_Toc158115205)

[ABSTRACT ii](#_Toc158115206)

[List of Figures vi](#_Toc158115207)

[List of Tables vii](#_Toc158115208)

[List of Abbreviations viii](#_Toc158115209)

[1. Introduction 1](#_Toc158115210)

[1.1 Background 1](#_Toc158115211)

[1.2 Motivation 1](#_Toc158115212)

[1.3 Problem Definition 1](#_Toc158115213)

[1.4 Objectives 2](#_Toc158115214)

[1.5 Scope and Application 2](#_Toc158115215)

[2. Literature Review 3](#_Toc158115216)

[2.1 Background and Related work 3](#_Toc158115218)

[3. Feasibility analysis 5](#_Toc158115219)

[3.1 Economic Feasibility 5](#_Toc158115221)

[3.2 Technical Feasibility 5](#_Toc158115222)

[3.3 Operational Feasibility 5](#_Toc158115223)

[3.4 Software Requirements 5](#_Toc158115224)

[3.4.1 Functional Requirements: 5](#_Toc158115225)

[3.4.2 Non-Functional Requirements: 6](#_Toc158115226)

[3.5 Hardware Requirements 7](#_Toc158115227)

[4. Dataset analysis 8](#_Toc158115228)

[4.1 Data Collection 8](#_Toc158115230)

[4.2 Data Extraction 8](#_Toc158115231)

[4.3 Dataset Characteristics 8](#_Toc158115232)

[4.4 Data Preprocessing 9](#_Toc158115233)

[5. Methodology 10](#_Toc158115234)

[5.1 Data Collection 10](#_Toc158115236)

[5.2 Data Preprocessing 10](#_Toc158115237)

[5.3 Model Selection 10](#_Toc158115238)

[5.4 Summarization and Slide Generation 10](#_Toc158115239)

[5.5 Block Diagram and Architecture 11](#_Toc158115240)

[5.6 Dataset Extraction 11](#_Toc158115241)

[5.7 Description of Working Principle 12](#_Toc158115242)

[5.7.1 System Architecture of Presentify 12](#_Toc158115243)

[5.8 Tools and Environment 12](#_Toc158115244)

[6. Implementation design 14](#_Toc158115245)

[6.1 Article Summarization 14](#_Toc158115247)

[6.1.1 T5 Model Integration 14](#_Toc158115248)

[6.1.2 Model Loading 14](#_Toc158115249)

[6.1.3 Summarization Process 14](#_Toc158115250)

[6.2 Slide Generation 15](#_Toc158115251)

[6.2.1 Structural Design 15](#_Toc158115252)

[6.2.2 Text-to-Slide Mapping 15](#_Toc158115253)

[6.2.3 User Interface 15](#_Toc158115254)

[6.2.4 Fine-Tuning 15](#_Toc158115255)

[6.2.5 Error Analysis 16](#_Toc158115256)

[7. Result and analysis 17](#_Toc158115257)

[8. Remaining task 18](#_Toc158115258)

[9. appendices 19](#_Toc158115259)

# List of Figures

[Figure 3‑1: RAM and Disk Usage 7](#_Toc158108940)

[Figure 5‑1: Dataset Extraction 10](#_Toc158108941)

[Figure 5‑2: System Architecture of Presentify 10](#_Toc158108942)

# List of Tables

# List of Abbreviations

|  |  |
| --- | --- |
| **Abbreviations** | **Definition** |
| NLP | Natural Language Processing |
| NLU | Natural Language Understanding |
| NLG | Natural Language Generation |
| PDF | Portable Document Format |
| LSTM | Long Short-Term Memory |
| RNN | Recurrent Neural Network |
| Seq2Seq | Sequence-to-Sequence |
| ROGUE | Recall-Oriented Understudy for Gisting Evaluation |
| BLEU | BiLingual Evaluation Understudy |
| BERT | Bidirectional Encoder Representations from Transformers |
| AI | Artificial Intelligence |
| NumPy | Numerical Python |
| HTML | Hyper Text Markup Language |
| XML | Extensible Markup Language |
| API | Application Programming Interface |
| RAM | Random Access Memory |
| CPU | Central Processing Unit |
| GPU | Graphics Processing Unit |
| PC | Personal Computer |

# Introduction

To deal with the overwhelming volume of research articles in both academic and industrial domain, this project aims to utilize the transformative capabilities of natural language processing (NLP), through the fine-tuning of a T5 transformer model on a custom dataset drawn from the computer science domain research articles. It aims to use the model's ability in transforming complex textual data to extract key points and core concepts from these articles for the generation of presentation slides. By adapting the pre-trained T5 model parameters to better align with the computer science literature, we aim to restructure the process of knowledge extraction and presentation preparation from research articles.

## Background

In contemporary professional and educational settings, the creation of presentation slides plays a pivotal role in conveying information effectively. However, the manual generation of these slides can be time-consuming and resource-intensive. The advent of Natural Language Processing (NLP) technologies presents an opportunity to streamline this process by automating the extraction and organization of relevant content. This project addresses the need for a more efficient and user-friendly approach to presentation slide creation through the integration of NLP.

## Motivation

The motivation behind this project stems from the recognition of the challenges individuals face in generating presentation slides efficiently. With the proliferation of information in diverse formats, including PDF documents, there is a growing demand for tools that can seamlessly convert this content into visually engaging slides. The motivation is grounded in the belief that leveraging NLP technologies can significantly enhance the speed and accuracy of slide creation, thereby empowering users to focus more on the content itself rather than the manual aspects of formatting and organization.

## Problem Definition

The core problem this project aims to address is the labor-intensive nature of creating presentation slides, particularly when dealing with diverse data sources such as PDF documents. Manual extraction and structuring of content are prone to errors and can impede productivity. The project seeks to mitigate these challenges by automating the process using NLP algorithms, thereby reducing the time and effort required for slide creation.

## Objectives

The main objectives of our project are listed below:

* To develop an interface to facilitate the automated generation of presentation slides from user-inputted PDF documents.

## Scope and Application

The objective of this project is to optimize and improve the process of developing presentation slides specifically tailored for academic research papers within the domain of computer science. The core functionality will revolve around users submitting research papers in PDF format, with the system automatically generating concise and informative presentation slides based on the extracted content from these documents. This application will leverage advanced natural language processing (NLP) techniques to carefully analyze and condense the essential information contained in the PDFs, identifying crucial points, thematic elements, and key concepts. Furthermore, the system will intelligently organize the extracted content into a coherent slide format, ensuring a systematic flow of information. The overarching aim is to automate the labor-intensive task of manually constructing presentations, offering users a more efficient and user-friendly alternative within the context of academic research in computer science.

The application's application lies in various professional and educational settings where creating presentations is a frequent requirement. Professionals, students, and educators who regularly deal with PDF documents can benefit from this tool by saving time and effort in the presentation creation process. It caters to individuals who may not have the expertise or time to manually sift through lengthy documents to distill the key points for a presentation. Additionally, the project has the potential to improve accessibility, making information more digestible for a broader audience. Overall, the project aims to be a valuable asset in enhancing productivity and facilitating effective communication through automated presentation slide generation from PDF content.

# Literature Review



## Background and Related work

The realm of natural language processing (NLP) and text summarization has witnessed a flourishing tapestry of advancements, with pivotal research shaping the contours of innovative solutions. Key contributions and terminologies that have significantly influenced this domain include

Abstractive Summarization Paradigms: Noteworthy strides have been made in the development of abstractive summarization models, characterized by their ability to distill essential information from the source text while introducing novel expressions. Seminal works, including, exemplify the evolution of abstractive summarization towards capturing deeper contextual nuances.

Sequence-to-Sequence Architectures with Attention Mechanisms: The adoption of sequence-to-sequence architectures, particularly those endowed with attention mechanisms, stands as a cornerstone in NLP tasks, including text summarization. Research by has showcased the prowess of these architectures in dynamically weighing input elements, thereby enhancing the model's understanding and generation of coherent summaries.

Extractive Summarization Techniques: Complementary to abstractive approaches, extractive summarization techniques have garnered attention. These methods involve identifying and selecting salient sentences or phrases directly from the source text. Works such as have explored the efficacy of extractive methods in crafting concise and information-rich summaries.

Evaluation Metrics for Summarization: The establishment of robust evaluation metrics is critical for assessing the efficacy of summarization models. Common metrics include ROUGE (Recall-Oriented Understudy for Gisting Evaluation) and BLEU (Bilingual Evaluation Understudy). These metrics, detailed in, provide quantitative measures to gauge the coherence and informativeness of generated summaries.

Learning Based Slide Generator: An automated slide generator will help to save time, effort and subsequently cost. At present, tools such as Microsoft PowerPoint and Open Office assist researchers in providing an outline and theme for the slides but do not help researchers to select contents of slides. An academic presentation is a sort of advertisement for the paper than an attempt to present all the information in the paper. In the PowerPoint presentation, it is an acceptable idea to find a picture that describes the aim of your research. Visuals are considered very effective tools for engaging the audience and maintaining their interest in conveying an important point or thought. This project proposes an automated system which generates presentation slides from research papers. The proposed system accepts research papers in PDF format as input and helps to generate the corresponding presentation slides. Papers and slides are learned and trained by Bidirectional Encoder Representations from Transformers (BERT) model. The research papers are summarized using the Google BERT algorithm which is a custom module that was released by Google. The sentence important scores are predicted by the pre-trained model of BERT. Text from the papers is extracted using Python’s unpdfer tool. The generated summary is used by BERT for making slides. As the presentation slides are of vital importance in a person's career, a significant amount of time and effort is spent on its preparation.

DocuBot: Generating ﬁnancial reports using natural language interactions:

The ﬁnancial services industry perpetually processes an over-

whelming amount of complex data. Digital reports are often

created based on tedious manual analysis as well as visualiza-

tion of the underlying trends and characteristics of data. Of-

ten, the accruing costs of human computation errors in creat-

ing these reports are very high. We present DocuBot, a novel

AI-powered virtual assistant for creating and modifying con-

tent in digital documents by modeling natural language in-

teractions as “skills” and using them to transform underly-

ing data. DocuBot has the ability to agglomerate saved skills

for reuse, enabling humans to automatically generate recur-

rent reports. DocuBot also has the capability to continuously

learn domain-speciﬁc and user-speciﬁc vocabulary by inter-

acting with the user. We present evidence that DocuBot adds

value to the ﬁnancial industry and demonstrate its impact with

experiments involving real and simulated users tasked with

creating PowerPoint presentations

The ﬁnancial services industry perpetually processes an over-

whelming amount of complex data. Digital reports are often

created based on tedious manual analysis as well as visualiza-

tion of the underlying trends and characteristics of data. Of-

ten, the accruing costs of human computation errors in creat-

ing these reports are very high. We present DocuBot, a novel

AI-powered virtual assistant for creating and modifying con-

tent in digital documents by modeling natural language in-

teractions as “skills” and using them to transform underly-

ing data. DocuBot has the ability to agglomerate saved skills

for reuse, enabling humans to automatically generate recur-

rent reports. DocuBot also has the capability to continuously

learn domain-speciﬁc and user-speciﬁc vocabulary by inter-

acting with the user. We present evidence that DocuBot adds

value to the ﬁnancial industry and demonstrate its impact with

experiments involving real and simulated users tasked with

creating PowerPoint presentations

The financial services industry perpetually processes an overwhelming amount of complex data. Digital reports are often created based on tedious manual analysis as well as visualization of the underlying trends and characteristics of data. Often, the accruing costs of human computation errors in creating these reports are very high. We present DocuBot, a novel AI-powered virtual assistant for creating and modifying content in digital documents by modeling natural language interactions as "skills" and using them to transform underlying data. DocuBot has the ability to agglomerate saved skills for reuse, enabling humans to automatically generate recurrent reports. DocuBot also has the capability to continuously learn domain-specific and user-specific vocabulary by interacting with the user. We present evidence that DocuBot adds value to the financial industry and demonstrate its impact with experiments involving real and simulated users tasked with creating PowerPoint presentations.

# Feasibility analysis



## Economic Feasibility

The project is cost-effective in the long run despite initial development expenses. Users save time and resources compared to manual slide creation, leading to a positive return on investment. Potential for widespread adoption in professional and educational sectors further supports economic viability.

## Technical Feasibility

The project is technically feasible with well-established technologies like Transformers that use deep learning frameworks and NLP algorithms. Integration of PDF parsing and content extraction tools can be done using existing python libraries and APIs.

## Operational Feasibility

The project seamlessly integrates into existing workflows with an intuitive user interface. Minimal training is required due to the automated nature of the system, making it accessible to a wide range of users. Scalability ensures the project can handle varying workloads and accommodate a growing user base. Regular updates and maintenance are crucial to address evolving user needs and ensure continued operational efficiency.

## Software Requirements

### Functional Requirements:

1. **User Input:**

* The system should allow users to input data in the form of PDF documents or arXiv links.

1. **Content Extraction:**

* Implement algorithms to intelligently extract relevant information from the provided input.

1. **Model Integration:**

* The system should integrate the fine-tuned T5 transformer model for key point extraction.
* Implement text extraction and preprocessing mechanisms compatible with T5 requirements.

1. **Summarization Process:**

* Automatically generate concise summaries of computer science research articles using the T5 model.

1. **Automated Slide Generation:**

* Develop mechanisms for automatically generating presentation slides based on the extracted content.

1. **Content Organization:**

* Organize the extracted content into coherent and visually appealing slides.

1. **Input Validation:**

* Enforce comprehensive input validation to address potential errors, particularly in the context of file uploads, ensuring the system can adeptly handle a diverse range of input data by overcoming limitations related to file types, sizes, and formats.

1. **Customization Options:**

* Provide options for users to customize the appearance and layout of generated slides.

### Non-Functional Requirements:

1. **Performance:**

* The system should perform content extraction and slide generation efficiently, even with large and complex input data.

1. **Scalability:**

* Design the system to be scalable, allowing it to handle an increasing number of users and larger datasets.

1. **Security:**

* Implement security measures to ensure the confidentiality and integrity of user data, especially when handling sensitive information.

1. **Reliability:**

* Ensure the system's reliability by minimizing downtime and errors during the content extraction and slide generation processes.

1. **Usability:**

* The user interface should be user-friendly, requiring minimal training for users to navigate and utilize the application effectively.

1. **Compatibility:**

* Ensure compatibility with popular web browsers to enhance the accessibility of the application.

1. **Maintainability:**

* Design the system with modular and well-documented code to facilitate ease of maintenance and future updates.

1. **Compliance:**

* Ensure compliance with relevant data protection regulations and standards to safeguard user privacy.

1. **Error Handling:**

* Implement robust error-handling mechanisms to address potential issues during content extraction and slide generation.

1. **Backup and Recovery:**

* Implement regular backup procedures and a robust recovery mechanism to prevent data loss and ensure system stability.

## Hardware Requirements

Any normal PC is preferable for training the model using Google Colab but for training the model on a local system:

* Multi-core CPU (Intel Xeon or AMD Ryzen) to handle concurrent requests and computations.
* RAM (16GB or more) is required.

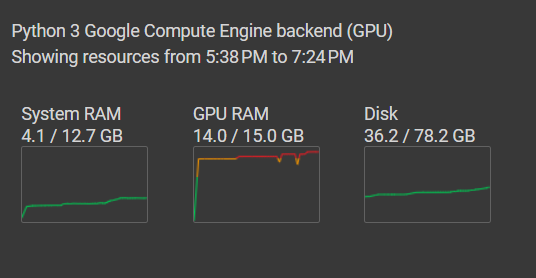
High Performance GPU to expedite model training and inference.

Figure 3‑1: RAM and Disk Usage

# Dataset analysis

The dataset utilized in this study consists solely of research articles sourced from the computer science domain via arXiv.org. This focused selection ensures relevance and alignment with the research topic, providing a targeted dataset for analysis and exploration.



## Data Collection

To compile the dataset, an automated script was developed to scrape PDF URL links specifically from the computer science section of arXiv.org. This approach ensured the inclusion of articles directly related to the study's research focus, omitting documents from other disciplines. The dataset encompasses a diverse range of computer science topics, including but not limited to artificial intelligence, machine learning, computer vision, and software engineering.

## Data Extraction

PDF data extraction was performed using the PyMuPDF library, enabling the systematic parsing and extraction of textual content, figures, and metadata from the selected research articles. The extracted data were organized into a structured format suitable for subsequent analysis, ensuring the preservation of key information while discarding irrelevant content. This extraction process facilitated the identification of trends, patterns, and insights within the dataset, contributing to the overall research objectives.

## Dataset Characteristics

The dataset comprises a comprehensive collection of scholarly articles exclusively focused on computer science research. Each document contains detailed information, including abstracts, introductions, methodologies, experimental results, and discussions, providing valuable insights into the respective research topics. Additionally, the dataset encompasses a variety of figures, tables, algorithms, and source code snippets, reflecting the diverse nature of computer science research and enhancing the richness of the dataset for analysis.

## Data Preprocessing

Prior to analysis, the extracted data underwent preprocessing steps tailored to the characteristics of computer science research articles. This involved techniques such as noise reduction, text normalization, and code snippet extraction to enhance the quality and usability of the dataset. Special attention was given to preserving the integrity of code samples, mathematical equations, and technical terminology commonly found in computer science literature.

# Methodology

This report outlines the progress and objectives of our project, focusing on the application of the T5 transformer model for summarizing computer science domain research articles and subsequently converting the summarized content into presentation slides.



## Data Collection

Curated a dataset of computer science research articles from diverse sub-domains. Ensured the inclusion of articles with varying levels of complexity to enhance the T5 model's adaptability.

## Data Preprocessing

Utilized T5-compatible methods to extract and preprocess text from research articles. Addressed formatting nuances and specific terminology present in computer science literature.

## Model Selection

Chose the T5 transformer model configured for summarization tasks. Customized the model input and output layers to accommodate the technical nature of computer science content.

## Summarization and Slide Generation

Trained the T5 model to generate informative summaries of research articles. Developed a mechanism to convert the summarized content into presentation slides, ensuring clarity and coherence.

## Block Diagram and Architecture

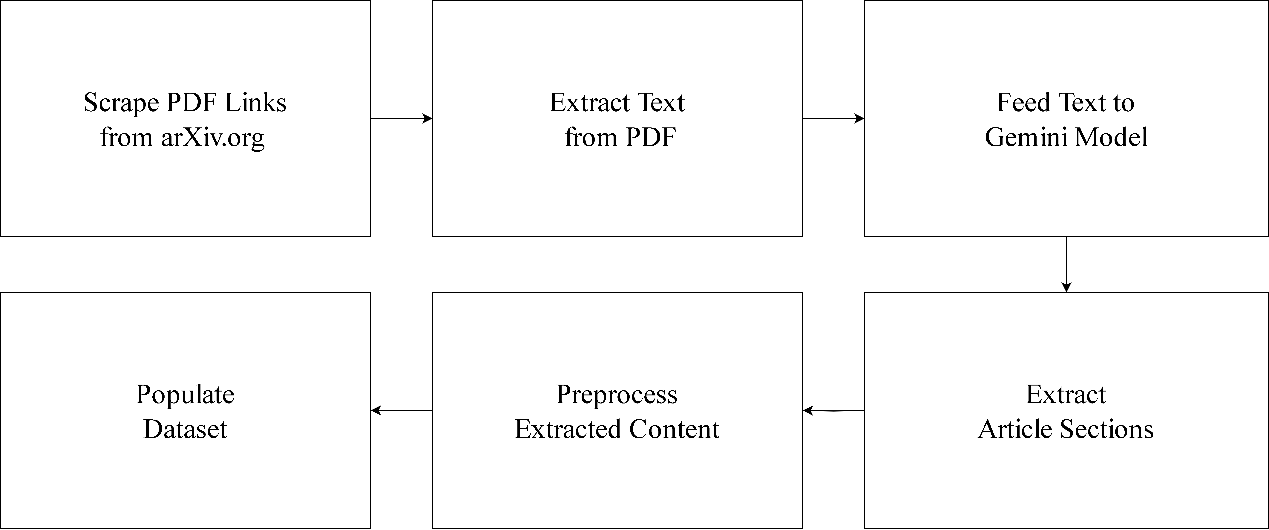


Figure 5‑1: Dataset Extraction

## Dataset Extraction

1. Extracting Link of PDF by web scraping.
2. Extracting Text through PDF.
3. Gemini with appropriate prompt.
4. Generating Required Fields.
5. Populating Dataset.

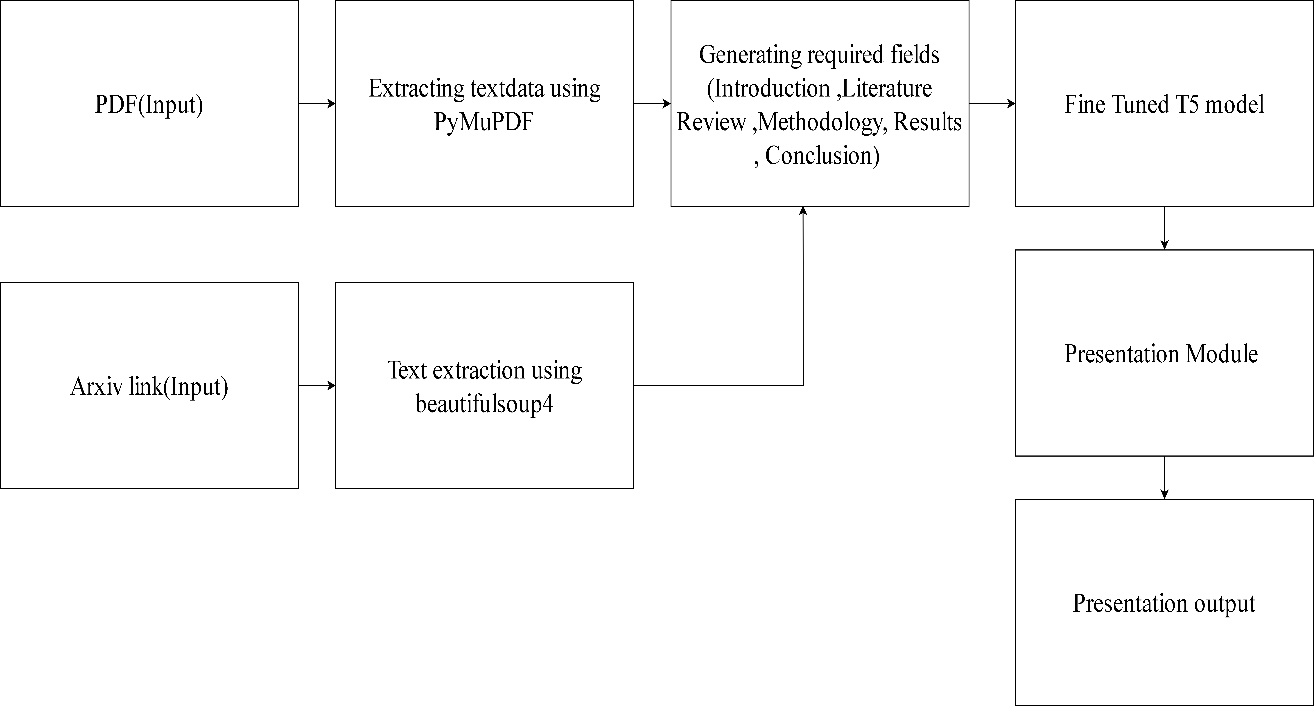


Figure 5‑2: System Architecture of Presentify

## Description of Working Principle

### System Architecture of Presentify

1. **Input:**

System takes input either in PDF or Arxiv Link.

1. **Extracting text data:**

Now the text is extracted from the given input, using PyMuPDF for PDF and BeautifulSoup4 for the Arxiv Link.

1. **Generating required fields:**

The text data extracted is then fed to Gemini with a prompt such that it gives us the following topics with its content: Introduction, Literature Review, Methodology, Results and Conclusion.

1. **Fined Tuned T5 Model:**

The Content of these topics is then fed serially to the Fined Tuned T5 model that summarizes the given content.

1. **Presentation Module:**

The summarized content is then given to the presentation module for converting it to a presentation format.

1. **Presentation Output:**

User is then provided with a standardly formatted Presentation Slide.

## Tools and Environment

1. **Python:** Python is a high level, general purpose, interpreted, dynamic programming language. Python supports multiple programming paradigms, including object-oriented, imperative and functional programming or procedural styles.
2. **NumPy:** NumPy is a Python library for numerical computing, offering powerful data structures and tools for working with arrays, matrices, and mathematical functions. Its efficient operations make it essential for scientific computing, data analysis, and machine learning in Python.
3. **Pandas:** A Python library specializing in data manipulation and analysis, providing high-performance, easy-to-use data structures and tools for working with structured data like tables and time series.
4. **JavaScript/HTML/CSS:** For frontend development, creating an intuitive user interface.
5. **BeautifulSoup:** Beautiful Soup is a Python package for parsing HTML and XML documents (including having malformed markup, i.e., non-closed tags, so named after tag soup). It creates a parse tree for parsed pages that can be used to extract data from HTML, which is useful for web scraping. Beautiful Soup.
6. **Google Colab:** Colab is a hosted Jupyter Notebook service that requires no setup to use and provides free access to computing resources, including GPUs and TPUs. Colab is especially well suited to machine learning, data science, and education.
7. **Jupyter Notebooks**: Preferred IDEs for Python development.
8. **FastAPI:** for backend development.

# Implementation design



## Article Summarization

### T5 Model Integration

1. Hugging Face Transformers

* Utilize the Hugging Face Transformers library to easily access and integrate the pre-trained T5 transformer model for natural language processing.

### Model Loading

* Load the T5 model with its pre-trained weights, ensuring compatibility with the summarization task.

### Summarization Process

#### Text Extraction

* Use PyMuPDF for efficient extraction of text from various research article formats, including PDFs and HTML.

#### Preprocessing

* Implement preprocessing steps to handle formatting nuances and prepare the extracted text for input to the T5 model.

#### Summarization Algorithm

* Develop an algorithm that leverages the T5 model to generate concise summaries of research articles.
* Fine-tune the model using Computer Science data to enhance summarization accuracy.

## Slide Generation

### Structural Design

#### Section Definition

* Design a system that identifies key sections within the summarized content, allowing for a logical structure in the presentation slides.

### Text-to-Slide Mapping

#### Content Alignment

* Develop a mapping mechanism that aligns T5-generated summaries with predefined slide structures.
* Ensure coherence between the content of the summaries and the designated sections within the slides.

### User Interface

#### Frontend Development

* Use HTML/CSS/JS for building a dynamic and responsive user interface.
* Implement user-friendly controls for inputting articles and accessing generated slides.

### Fine-Tuning

#### Model Adjustments

* Iteratively fine-tune the T5 model based on evaluation metrics.
* Refine hyperparameters to enhance summarization accuracy.

#### Slide Generation Improvements

* Enhance the slide generation process to improve coherence and clarity.

### Error Analysis

#### Identification

* Conduct detailed error analysis to identify and address any limitations or inaccuracies in the summarization and slide generation processes.

#### Iterative Fixes

* Implement iterative fixes and updates to eliminate identified errors and enhance system performance.

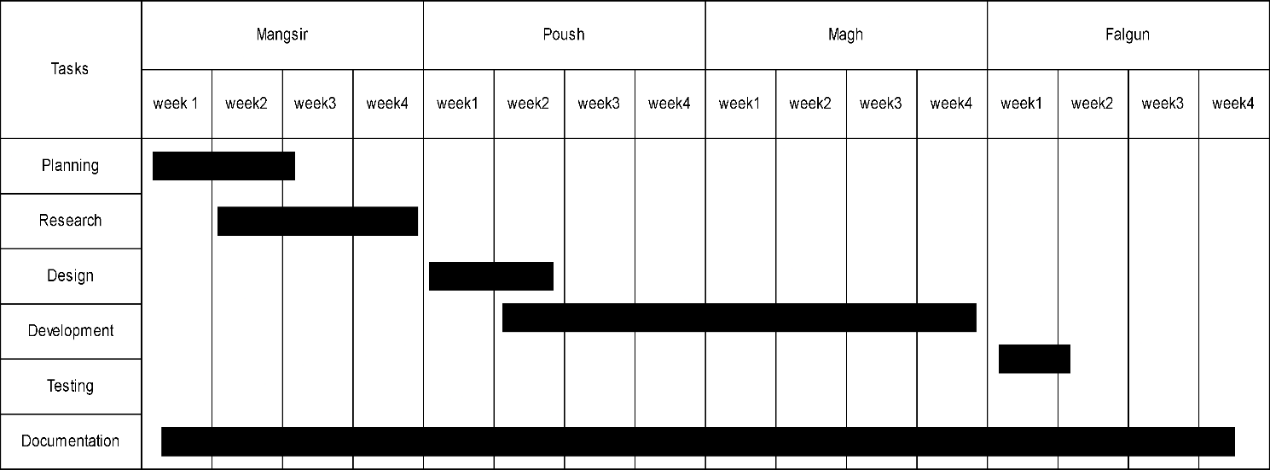
# Result and analysis

# Remaining task

1. Frontend Development.
2. Customization by user for pptx formatting
3. Increasing Accuracy

# appendices

1. **Appendix A: PROJECT SCHEDULE**

Table 1: Project Schedule

1. **Appendix B: PROJECT BUDGET**

|  |  |
| --- | --- |
| Task | Price |
| Tesla T4 GPU usage Cost | Rs.46.66/hour \*112.2hour = Rs. 5235 |
| Cloud Storage | Rs. 2.66 per GB per Month \*100GB\*2Months = Rs. 532 |
| Miscellaneous | Rs. 1000 |
| Total | Rs. 6767 |

Table 2: Project Budget

1. **Appendix C: Dataset Code**