# Text Summarization

*Mentor: Mr. Narendra Kumar*

*Group:* 4

# Introduction

## Problem Statement & Planning

# Introduction

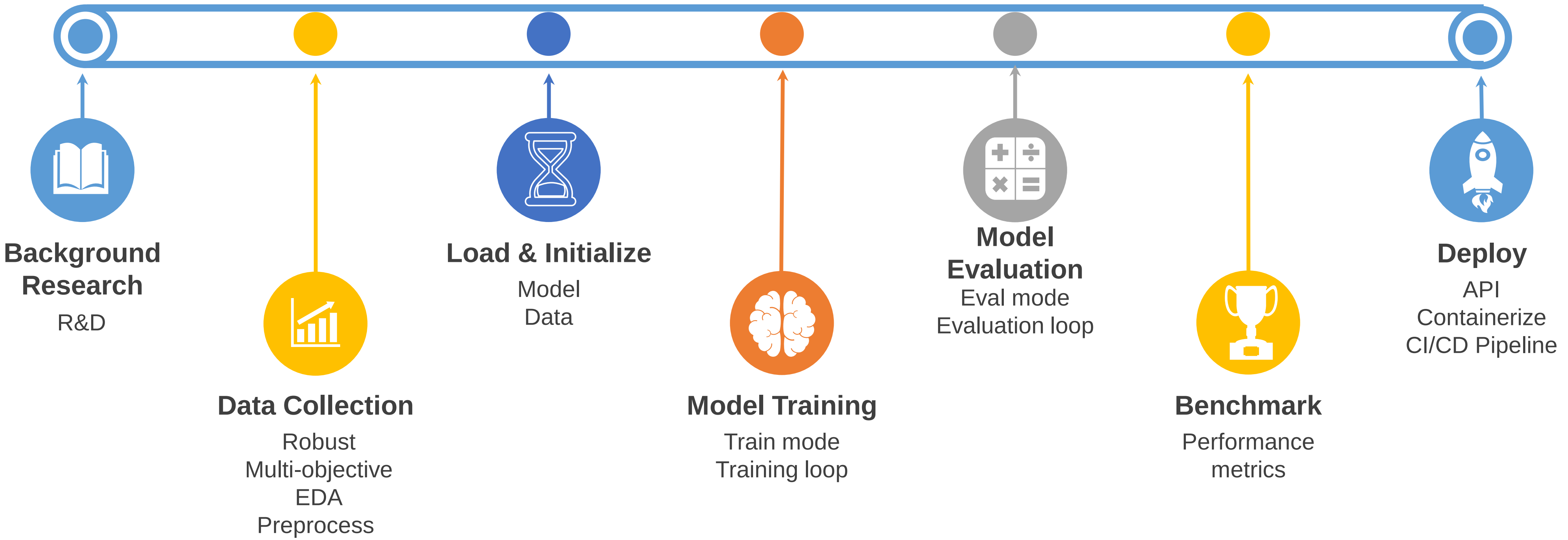
## Problem Statement

- Developing an automated text summarization system that can accurately and efficiently condense large bodies of text into concise summaries is essential for enhancing business operations.

- This project aims to deploy NLP techniques to create a robust text summarization tool capable of handling various types of documents across different domains.

- The system should deliver high-quality summaries that retain the core information and contextual meaning of the original text.

# *INTENDED PLAN*



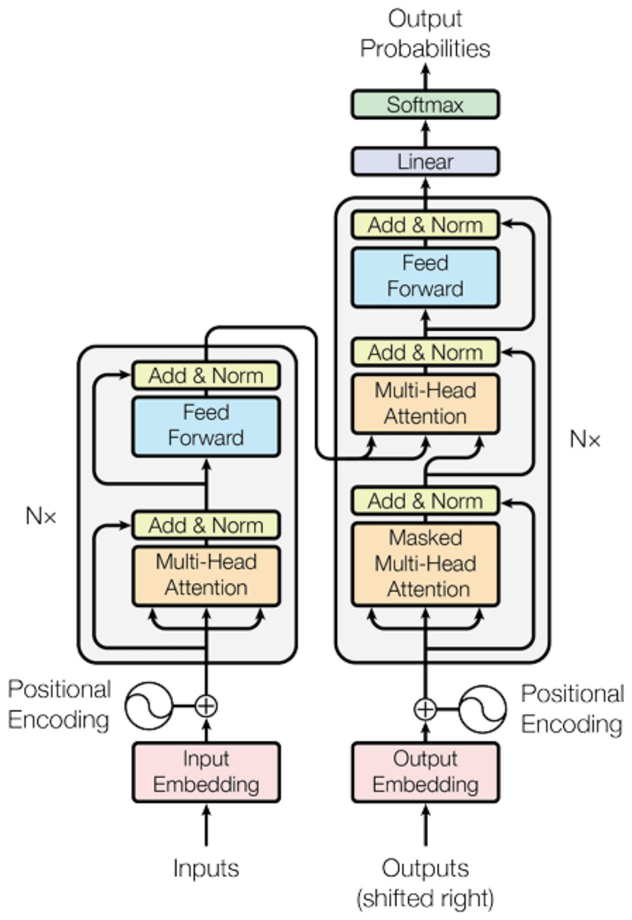
# Background Research

## Literature Review & Findings

# Background Research Literature Review

| S. No | Use-Case | Paper Title | Year | Method | Dataset | Results | Limitations |
|---|---|---|---|---|---|---|---|
| 1 | General text summarization | Text Summarization Using Deep Learning Techniques: A Review | 2023 | Deep Learning (Seq2Seq, Attention, Transformers) | CNN/Daily Mail, XSum | Improved performance in capturing semantic relationships, better coherence | Computationally expensive, requires large datasets |
| 2 | Implementation of the Transformer architecture | Attention is all you need | 2023 | Transformer | WMT 2014 English-German, WMT 2014 English-French | Introduced the Transformer architecture, significantly improving the performance of text summarization tasks. | Requires large datasets and computational resources for training. |
| 3 | Multi-document summarization | Surveying the Landscape of Text Summarization with Deep Learning | 2023 | Deep learning methods. Various techniques like RBMs and fuzzy logic employed for summarization. | CNN/DailyMail | Incorporating transfer learning enhances summary quality and reduces data demand. | Complex models, high computational resources |
| 4 | Abstractive summarization | Pegasus: Pre-training with gap-sentences for abstractive summarization | 2020 | Transformer (Pegasus) | XSum, CNN/DailyMail, and Reddit TIFU | Significant improvements in abstractive summarization quality | Resource-intensive |
| 5 | Extractive summarization | Text Summarization with Pretrained Encoders | 2019 | Intersentence Transformer layers for summarization | CNN/Daily Mail, NYT, Xsum, DailyMail | BERT-based models outperformed other approaches in abstractive summarization. | High computational resources required |

# Research
## Selected Architecture




[2] Fig. :Transformer architecture:

- **Implementation methods:**
  - *From Scratch*
    - Build Model
      - NN
    - Initialize normalized W&B
    - Train model with extensive data
    - Hence,
      - Computationally Intensive
      - Sub-Optimal usage of resources
      - Out-of-scope
  - *Using Pre-trained model*
    - Load Model & its parameters
    - Re-Train with specific dataset
    - Evaluate
    - Hence,
      - Innovation can be done at intended tasks
      - Optimal utilization of resources

# Proposals

# Proposal
## Workflow

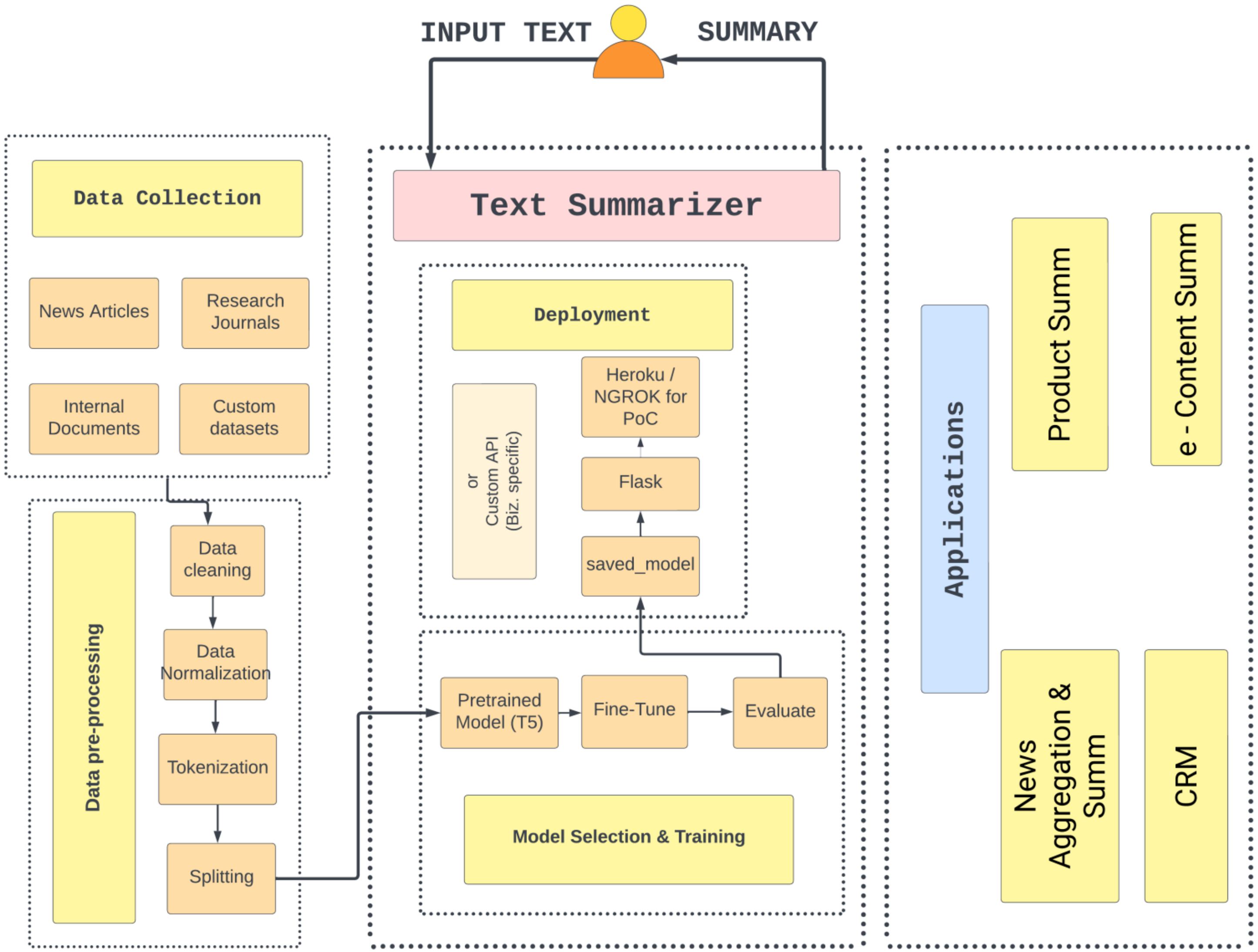



Fig. : Proposed Workflow for Abstractive Text Summarization

# Proposal
## Workflow

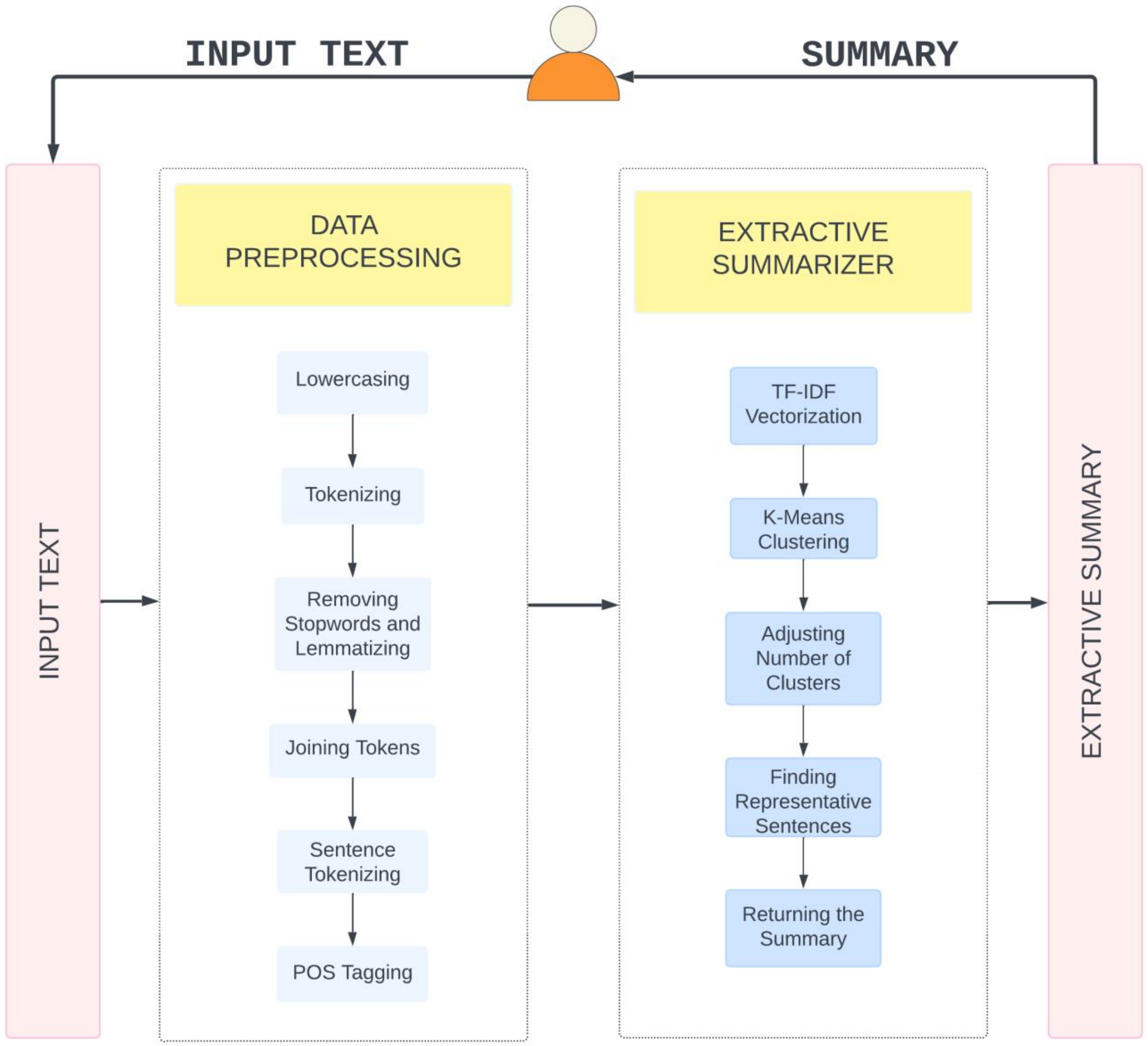



Fig. : Proposed Workflow for Extractive Text Summarization

# Proposal

## Dataset

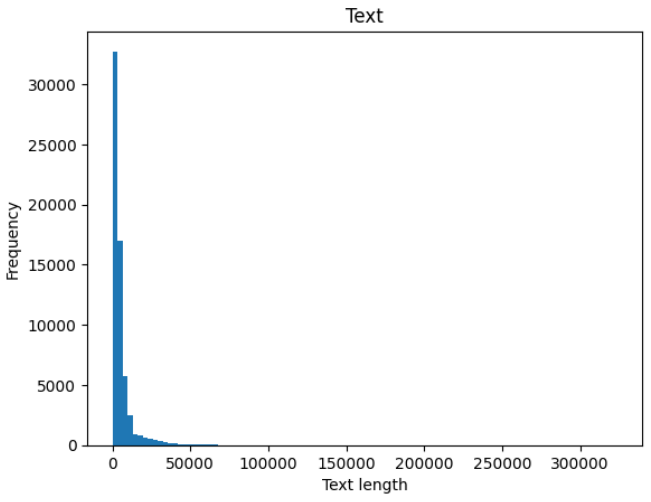
- Merged selective dataset from
  - CNN, Daily Mail : News,
  - BillSum: Legal,
  - ArXiv : Scientific
  - Dialoguesum : Conversations.

- Completed  - data preprocessing
  - Removed
    - NULL records, punctation, stop-words
  - Lowercasing, lemmatization.

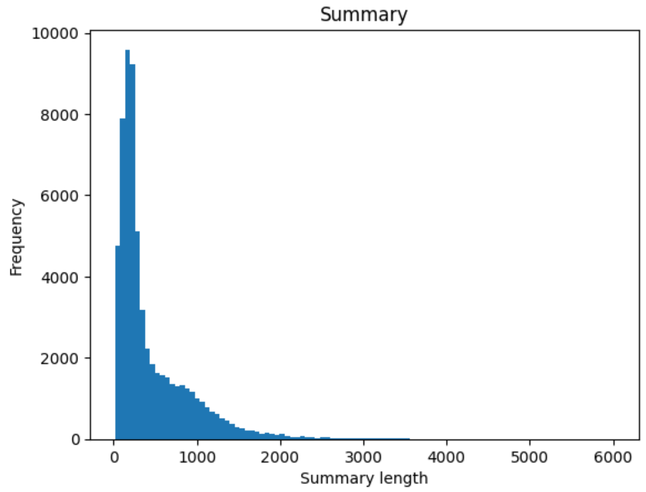
| | text | summary |
|---|---|---|
| 0 | section 1 liability business entity providing ... | shield business entity civil liability relatin... |
| 1 | section 1 short title act may cited human righ... | human right information act requires certain f... |
| 2 | section 1 short title act may cited jackie rob... | jackie robinson commemorative coin act directs... |
| 3 | section 1 nonrecognition gain rollover small b... | amends internal revenue code provide temporari... |
| 4 | section 1 short title act may cited native ame... | native american energy act sec 3 amends energy... |
| ... | ... | ... |
| 62702 | person1 excuse mr green manchester arent perso... | tan ling pick mr green easily recognized white... |
| 62703 | person1 mister ewing said show conference cent... | person1 person2 plan take underground together... |
| 62704 | person1 help today person2 would like rent car... | person2 rent small car 5 day help person1 |
| 62705 | person1 look bit unhappy today whats person2 w... | person2s mom lost job person2 hope mom wont fe... |
| 62706 | person1 mom im flying visit uncle lee family n... | person1 asks person2s idea packing bag visitin... |

62707 rows × 2 columns

```
count     62707.000000
mean       5211.270975
std        7794.860686
min          83.000000
25%        1275.000000
50%        3176.000000
75%        5684.500000
max      323742.000000
Name: text, dtype: float64
```




```
count    62707.000000
mean       448.081937
std        459.087443
min         16.000000
25%        154.000000
50%        255.000000
75%        618.000000
max       6014.000000
Name: summary, dtype: float64
```




* In characters.

https://drive.google.com/drive/folders/1yH89iZmARdc-R7QY6pwfE8tbOJI_n9K8?usp=sharing

Infosys_Springboard_Text-Summarization/src/data_preprocessing.ipynb at main · MohanKrishnaGR/Infosys_Springboard_Text-Summarization (github.com)

# Proposal
## Model Training

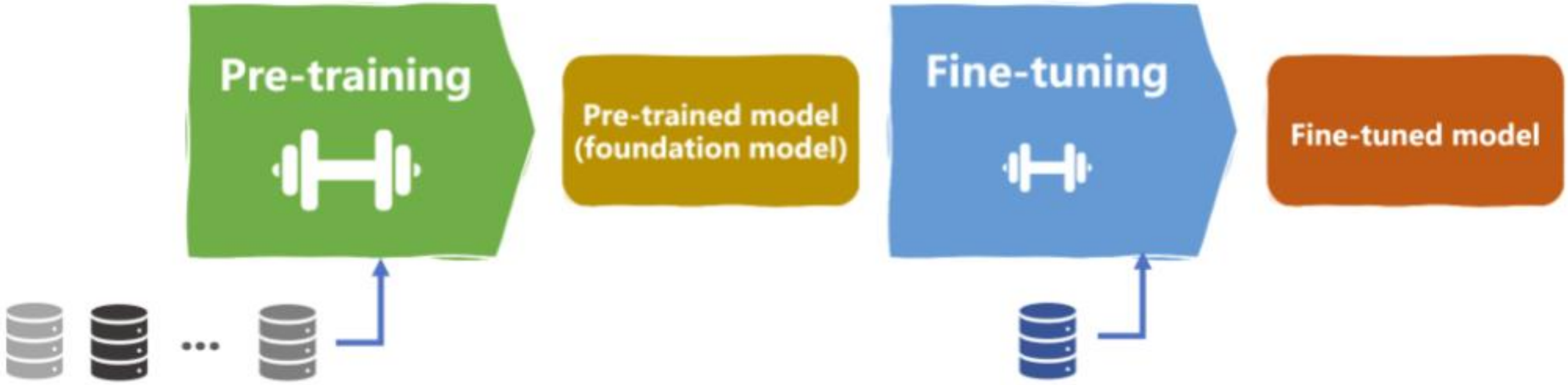



Fig. : Fine-Tunning Overview

- Proposed implementation – Two – 2 Methods
  - Method 1 – Native PyTorch Method
  - Method 2 – Trainer Class Method

# Proposal

## Model Training (Method 1)

- Load pre-trained transformer
  - Facebook's Bart Large

- OOP implementation of Dataset
  - Feature, Target
  - Tokenize
  - Padding, Truncate
  - Convert to Tensor
  - Pass to: DataLoader – with batch size

- Training Loop
  - Adam optimizer
  - Forward pass & compute loss
  - Backward pass
  - Update params – compute gradient
  - Update LR
  - Zero the gradients
  - Update total loss

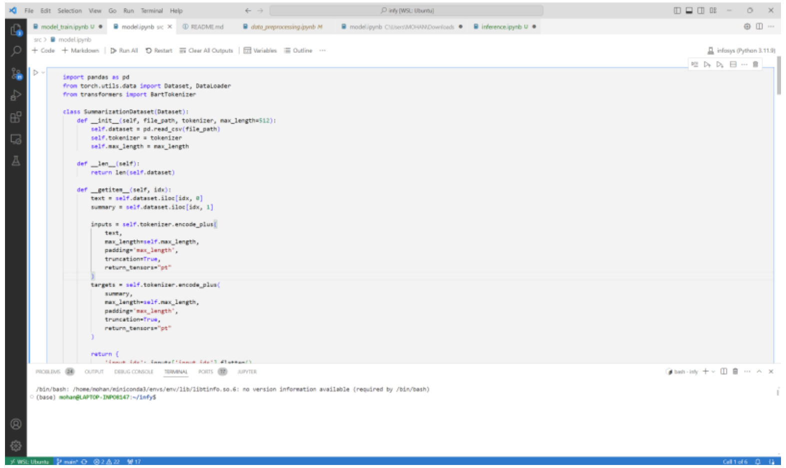



Fig. : Screenshot

- Only minimal train loss of 1.3280.
  - But, produced inconsistent results.
  - Cannot be pushed into production.

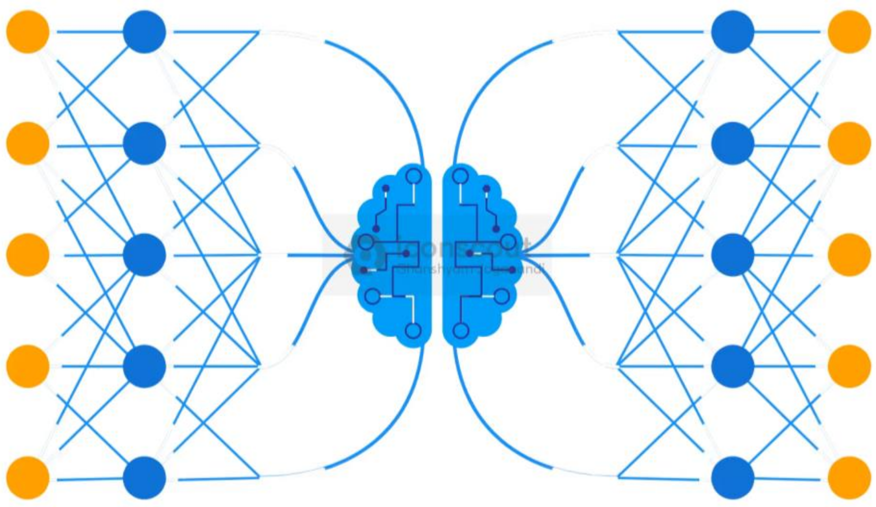
- Raises the need for optimized training and eval loop for Transformer.

[Infosys_Springboard_Text-Summarization/src/model.ipynb at main · MohanKrishnaGR/Infosys_Springboard_Text-Summarization (github.com)]

# Proposal

## Model Training (Method 2)

- Trainer Method
- Implemented in src/bart.ipynb.
- A function was implemented for the dataset, to convert text data into model inputs and targets.
- Trainer class from transformer package was utilized for training and evaluation. Tainer is a simple but feature-complete training and eval loop for PyTorch, optimized for transformers.
- The model was trained with whole dataset for 10 epochs for 26:24:22 (HH:MM:SS) in 125420 steps.
- Training Loss = 17.4700
- Considered the performance metrics of the models trained by the forementioned methods. After the due analysis, the model trained using 'Method 2' was selected.



[Infosys_Springboard_Text-Summarization/src/bart.ipynb at main · MohanKrishnaGR/Infosys_Springboard_Text-Summarization (github.com)](https://github.com)

https://drive.google.com/drive/folders/1tNdLI67UTc5es6VB_dml8b5gkRUWzupI?usp=drive_link

# Proposal
## Model Validation

- Performance metrics – **ROUGE** (Recall-Oriented Understudy for Gisting Evaluation)
  - Overlap between generated summary and reference summary.
  - Best suited : evaluating 'Text Summarization' tasks.
  - Other options : **BLEU**.

- ROUGE-N: Measures the overlap of n-grams (contiguous sequences of n items) between the candidate summary and the reference summaries.
  - **ROUGE-1**:
    - Overlap of unigrams (single words).
  - **ROUGE-2**:
    - Overlap of bigrams (two-word sequences).
  - **ROUGE-L**:
    - Measures the longest common subsequence (LCS) between the candidate and reference summaries.
  - **ROUGE-LSUM**
    - (LCS Summary) - variant of the ROUGE-L metric, specifically designed to evaluate the quality of summaries.

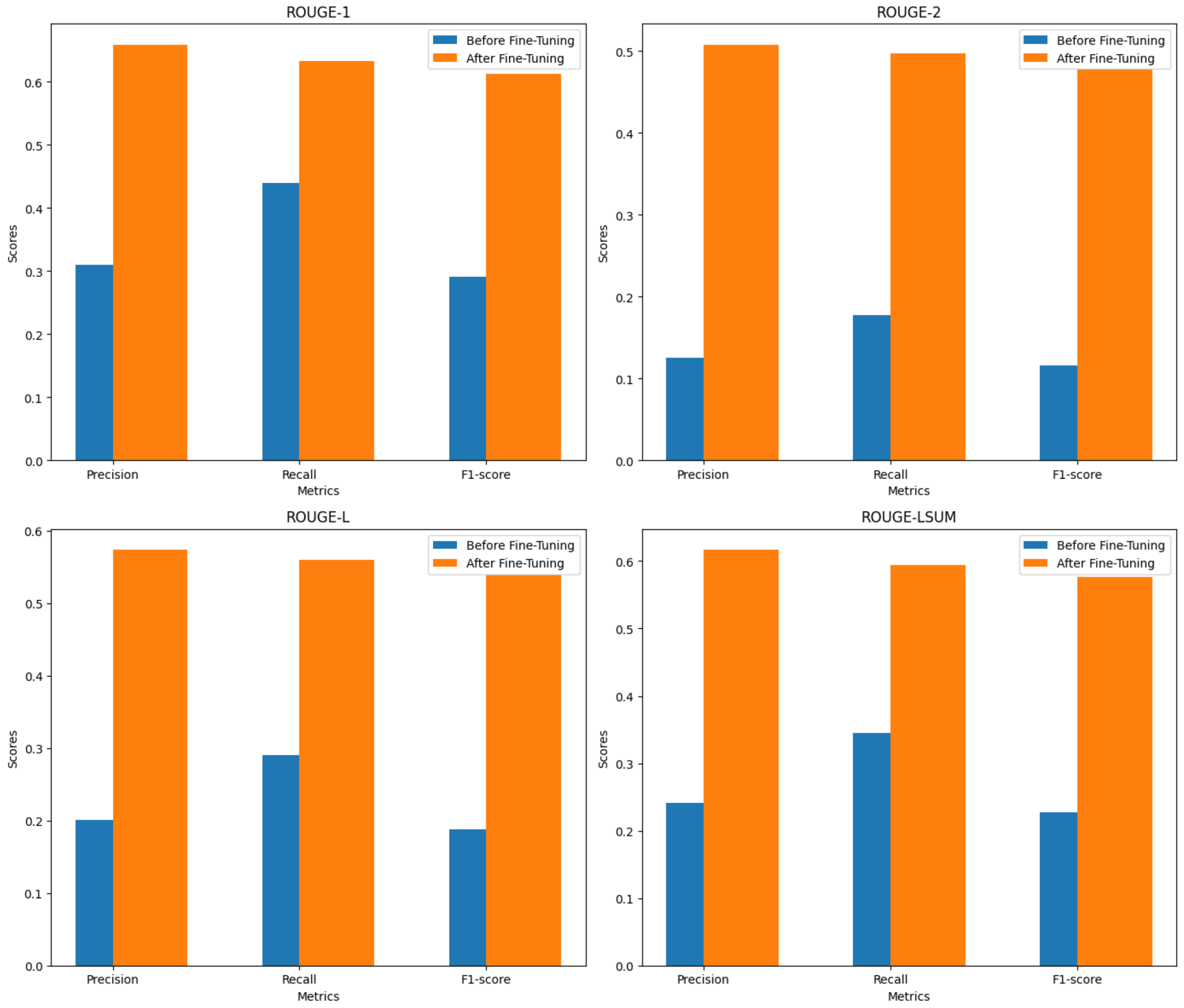

- Aimed to: implement custom evaluation function, using ROUGE based on model's inference.

Infosys_Springboard_Text-Summarization/src/evaluation.ipynb at main · MohanKrishnaGR/Infosys_Springboard_Text-Summarization (github.com)

Infosys_Springboard_Text-Summarization/src/rogue.ipynb at main · MohanKrishnaGR/Infosys_Springboard_Text-Summarization (github.com)

# Proposal
## Comparative Analysis

- Analysis of the transformer's performance metrics before and after Fine-Tuning.
- The transformer model shows significant improvements across all ROUGE metrics after fine-tuning.
- The most substantial gains observed in ROUGE-2 scores. (F1-score=61.32)
- This indicates that the fine-tuning process has notably enhanced the model's ability to generate more accurate and relevant summaries.
- The model is now more proficient at generating summaries that are precise, comprehensive, and contextually accurate.
- Will act as a powerful tool for a variety of Business applications that require efficient and effective text summarization.



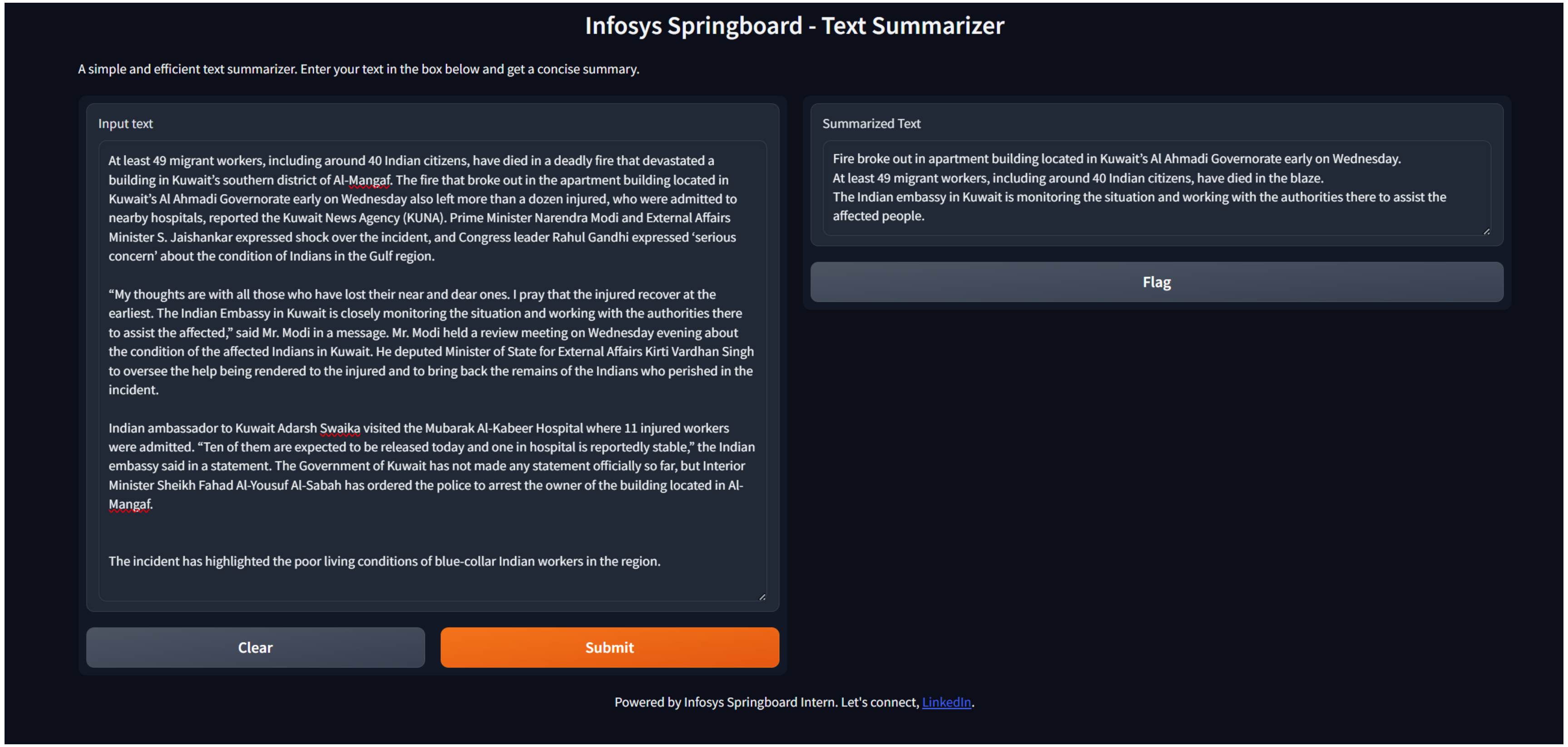



[Infosys_Springboard_Text-Summarization/src/compare.ipynb at main · MohanKrishnaGR/Infosys_Springboard_Text-Summarization (github.com)]

# Proposal

## Testing

gradio




- Simple interface for the Deep Learning model, developed using Gradio.
- Gradio is an open-source Python package that allows us to quickly build a demo - web-application for the trained models.
- Enables us to test and even deploy the trained model.

[Infosys_Springboard_Text-Summarization/src/interface.ipynb at main · MohanKrishnaGR/Infosys_Springboard_Text-Summarization (github.com)](https://github.com)

# Proposal

## Extractive Text Summarization

- Rather than choosing computationally intensive deep-learning models, utilizing a rule based approach will result in optimal solution. Utilized a new-and-novel approach of combining the matrix obtained from TF-IDF and KMeans Clustering methodology.
- It is the expanded topic modeling specifically to be applied to multiple lower-level specialized entities (i.e., groups) embedded in a single document. It operates at the individual document and cluster level.
- The sentence closest to the centroid (based on Euclidean distance) is selected as the representative sentence for that cluster.
- Implementation: Preprocess text, extract features using TF-IDF, and summarize by selecting representative sentences.
- Source code for implentation & evaluation: src/Extractive_Summarization.ipynb
- ROUGE1 (F-Measure) = 24.71

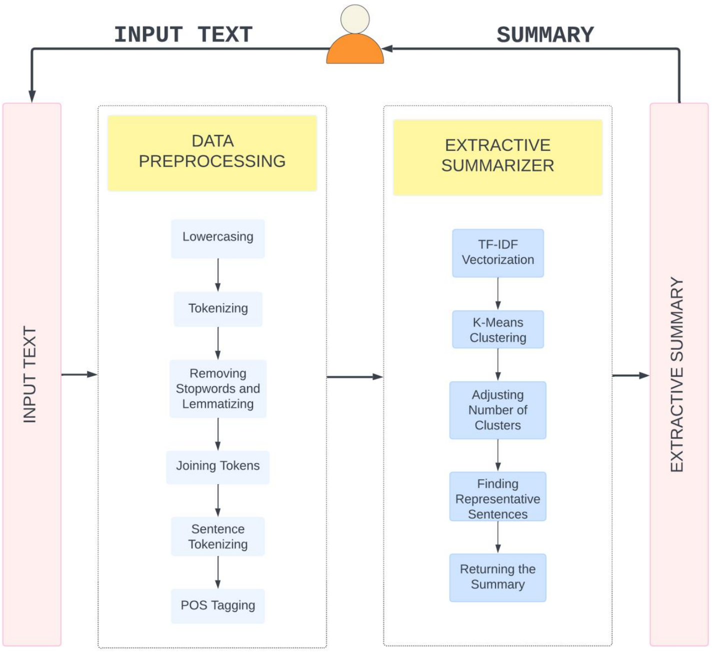
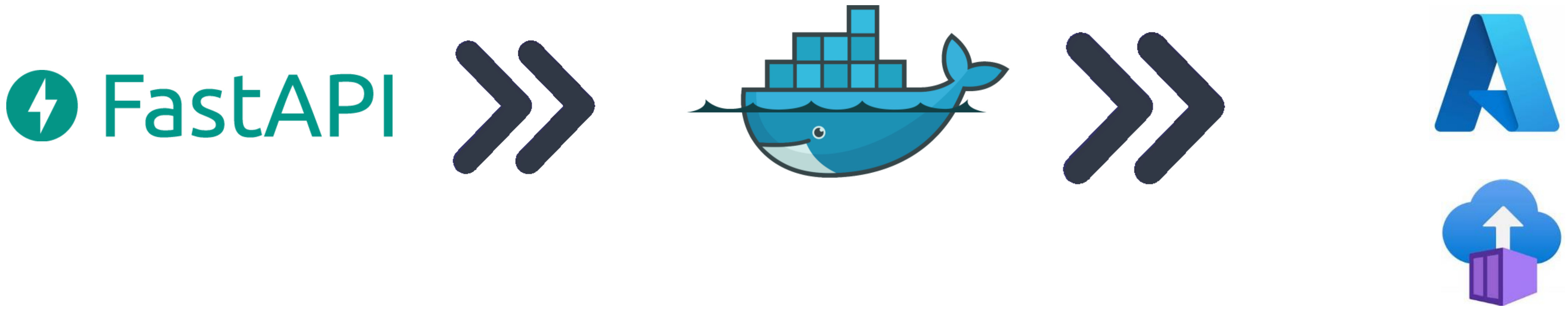



Fig. : Proposed Workflow for Extractive Text Summarization

[Infosys_Springboard_Text-Summarization/src/Extractive_Summarization.ipynb at main · MohanKrishnaGR/Infosys_Springboard_Text-Summarization (github.com)]

# Proposal

## Deployment




[Infosys_Springboard_Text-Summarization/summarizer at main · MohanKrishnaGR/Infosys_Springboard_Text-Summarization (github.com)](https://github.com/MohanKrishnaGR/Infosys_Springboard_Text-Summarization)

# Proposal

## Deployment

- Implemented extractor modules for text extraction from URL, PDF, docx.
- Defined the API endpoints. (FastAPI)
  - Accepts: Text, URL, Files (PDF, docx)
  - Returns:
    - Abstractive & Extractive Summary
- Utilized 'jQuery' for a dynamic webpage.

- Containerized the entire application along with the deep-learning models.
  - Built the image & Pushed into docker hub.

Amazon EC2

- Drawback = Less computation for free-tier plan (t2.micro)

- Deployed the docker image using Azure Container Instance
- Integrated with GitHub actions – CI/CD pipeline
- Advantage = 4 CPU cores for Free Trail

# Results

# Results

## GitHub Repository

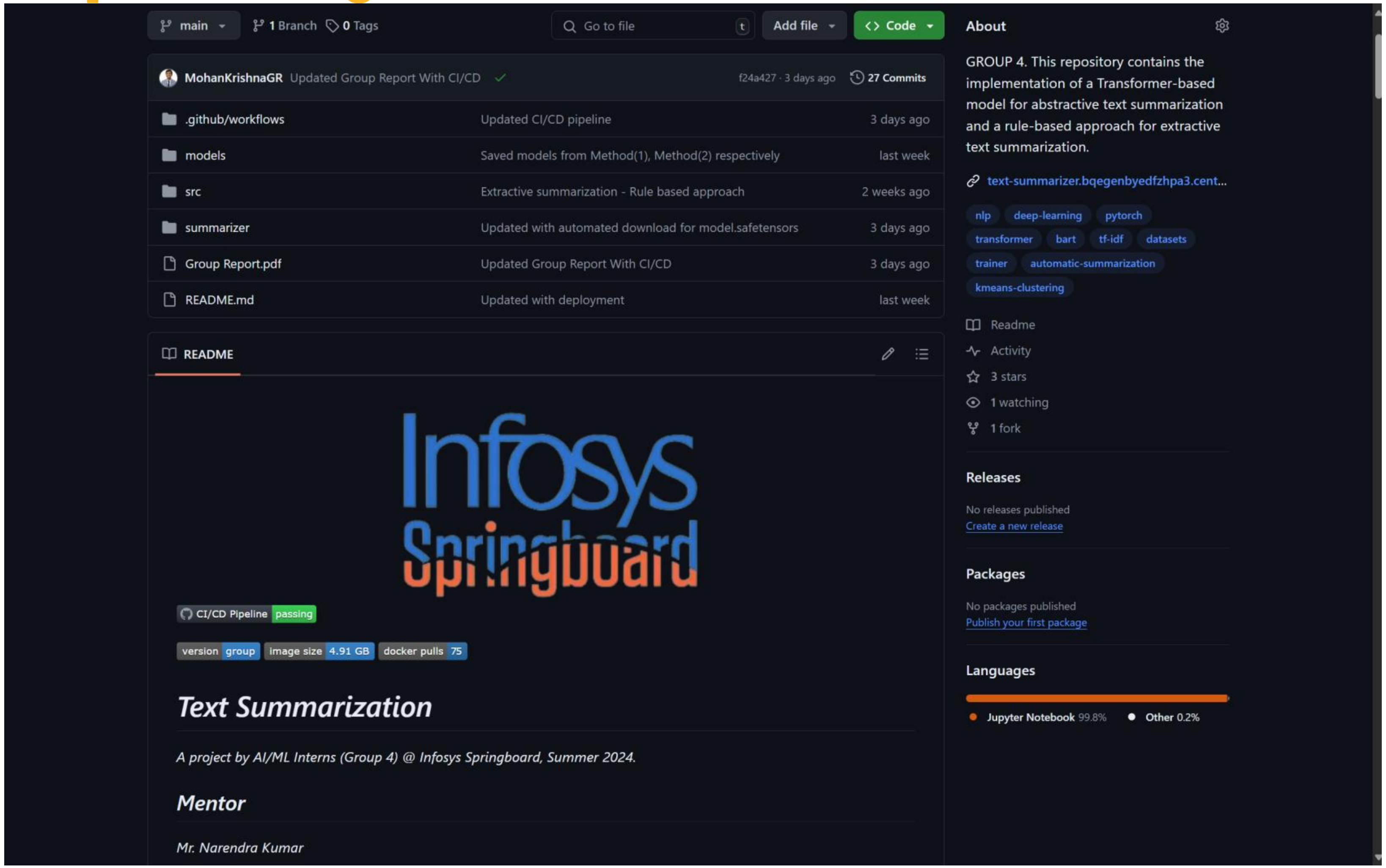



Fig. : Screenshot of the GitHub repository.

[Infosys_Springboard_Text-Summarization]

# Results

## Deployment - Ref. Links

- mohankrishnagr/infosys_text-summarization - Docker Image | Docker Hub

- Text-Summarizer
  - http://text-summarizer.bqegenbyedfzhpa3.centralindia.azurecontainer.io:8000/
  - http://20.235.235.107:8000/

# Results
## CI/CD pipeline

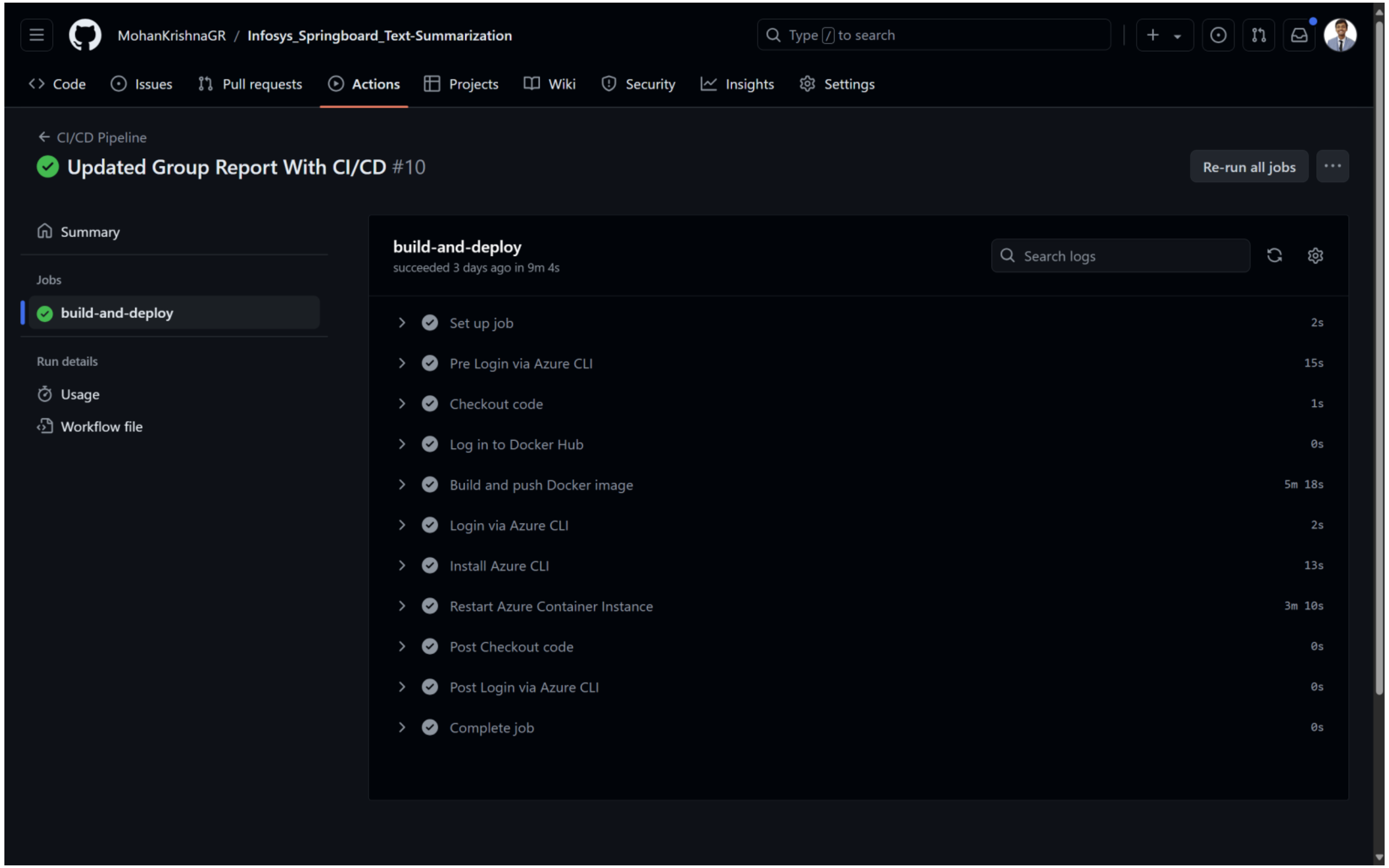



Fig. : Screenshot of the CI/CD pipeline.

# Results

## Deployed Application

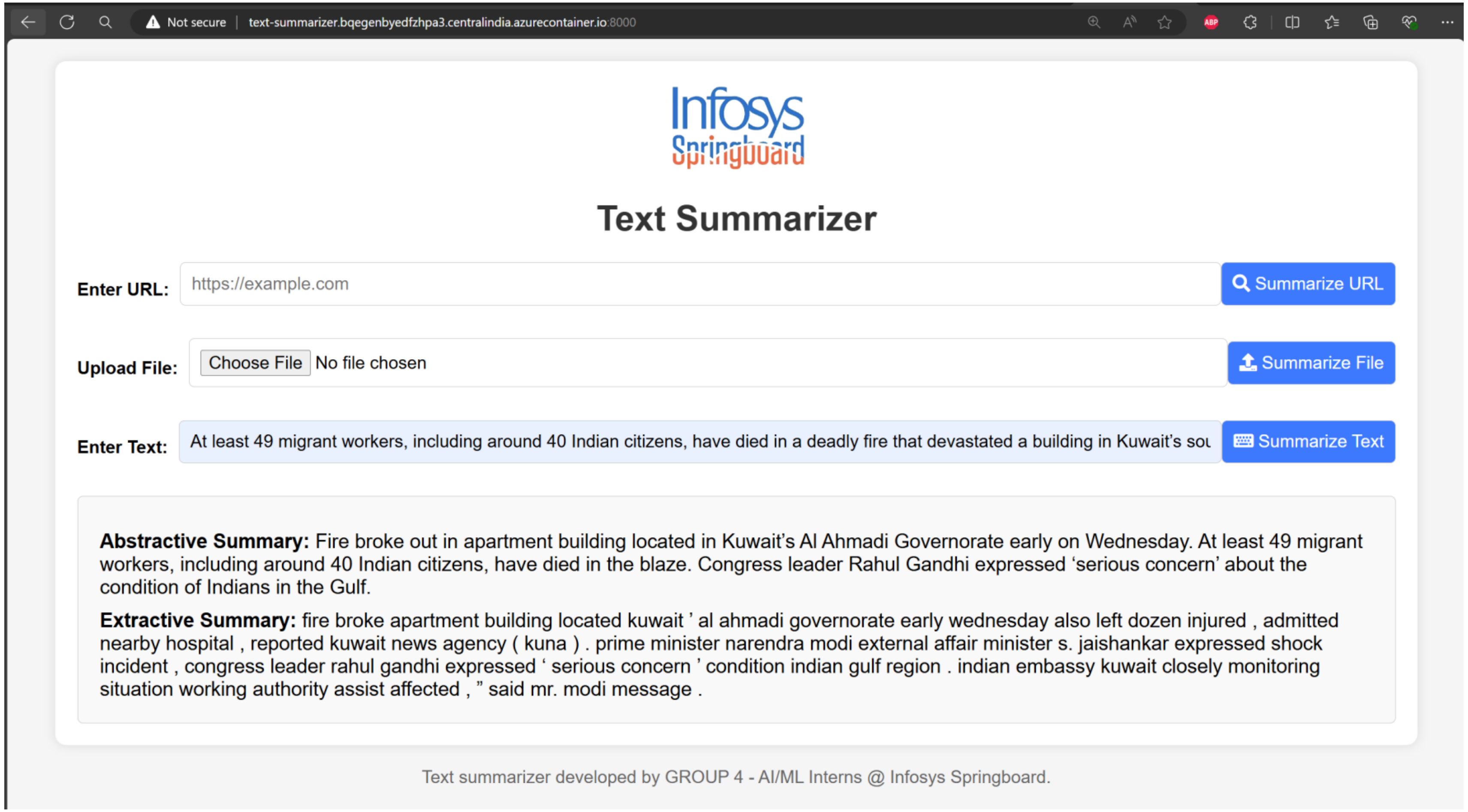



Fig. : Screenshot of the Deployed application – Text input.

# Results

## Deployed Application

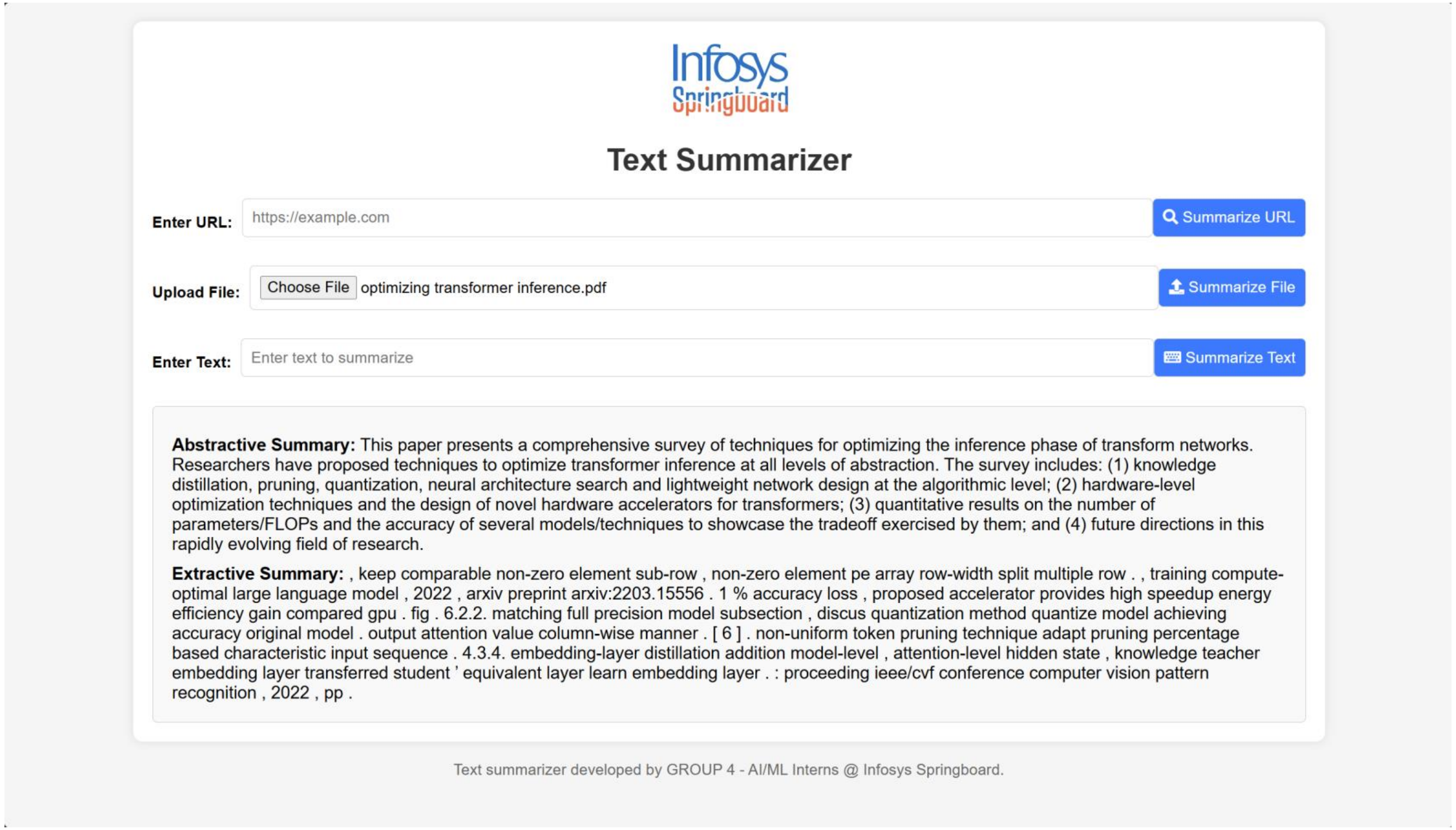



Fig. : Screenshot of the Deployed application – PDF input.

# Results

## Deployed Application

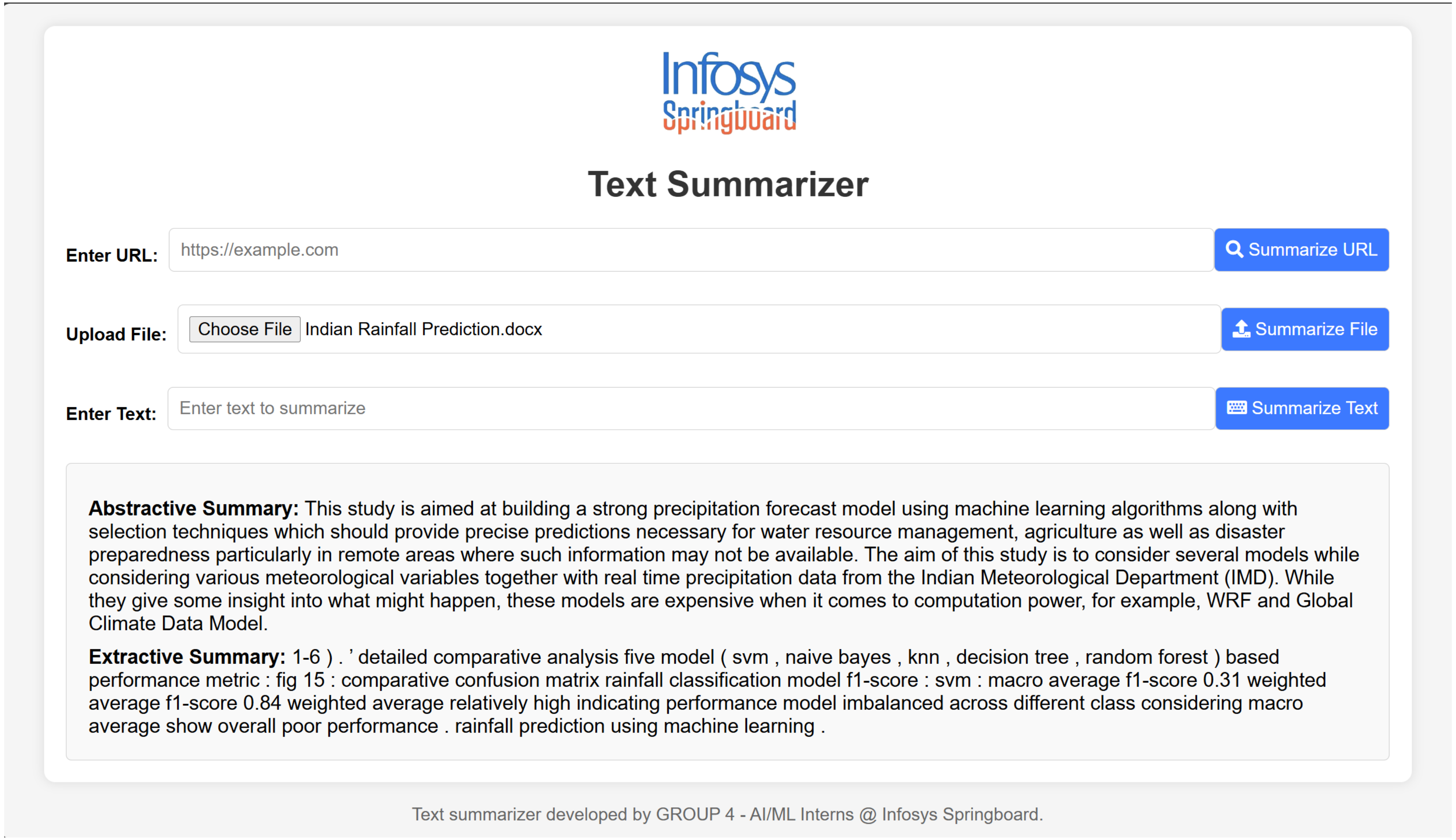



Fig. : Screenshot of the Deployed application – DOCX input.

# Results

## Deployed Application

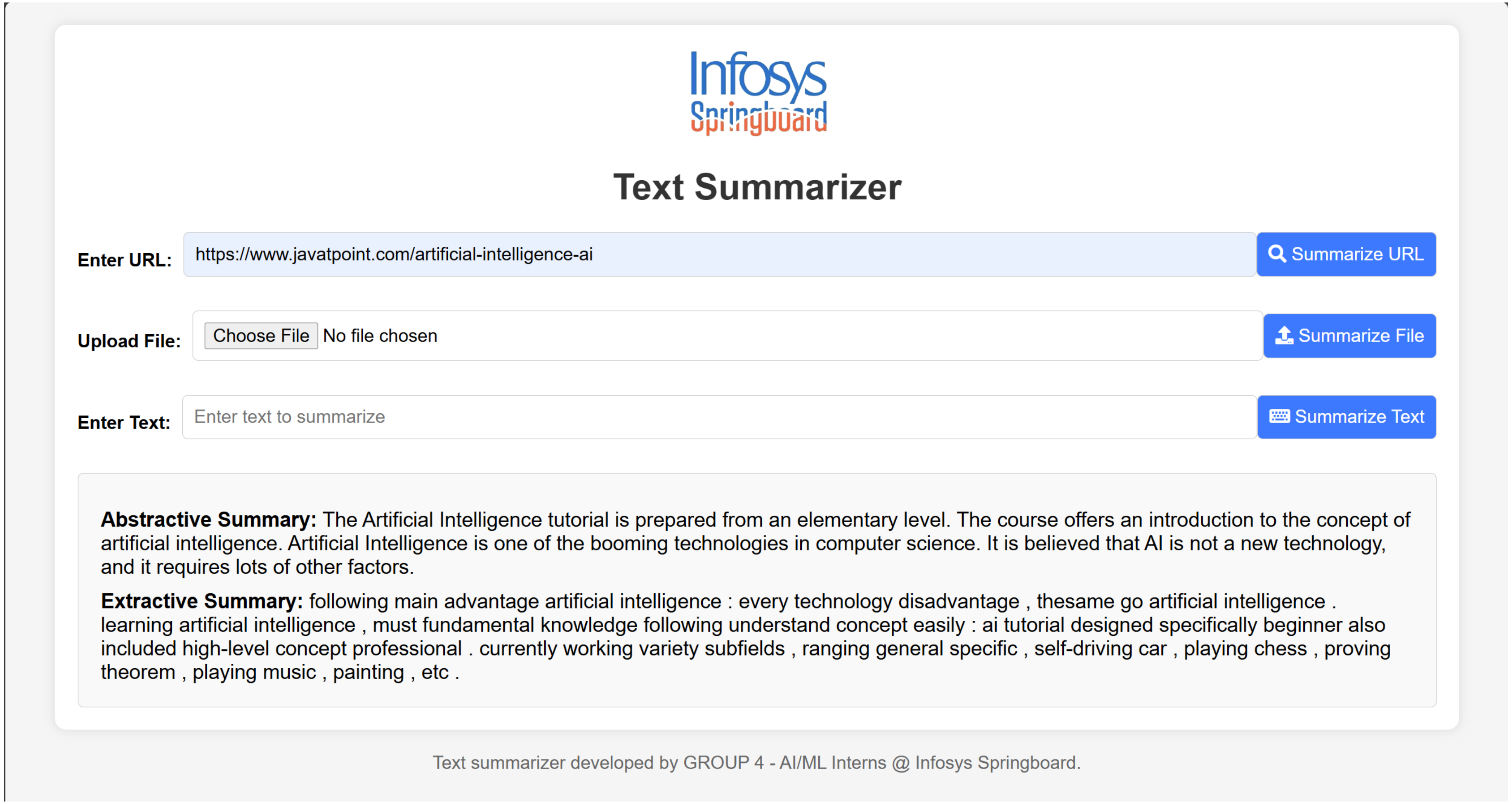



Fig. : Screenshot of the Deployed application – URL input.

# THANK YOU!!!