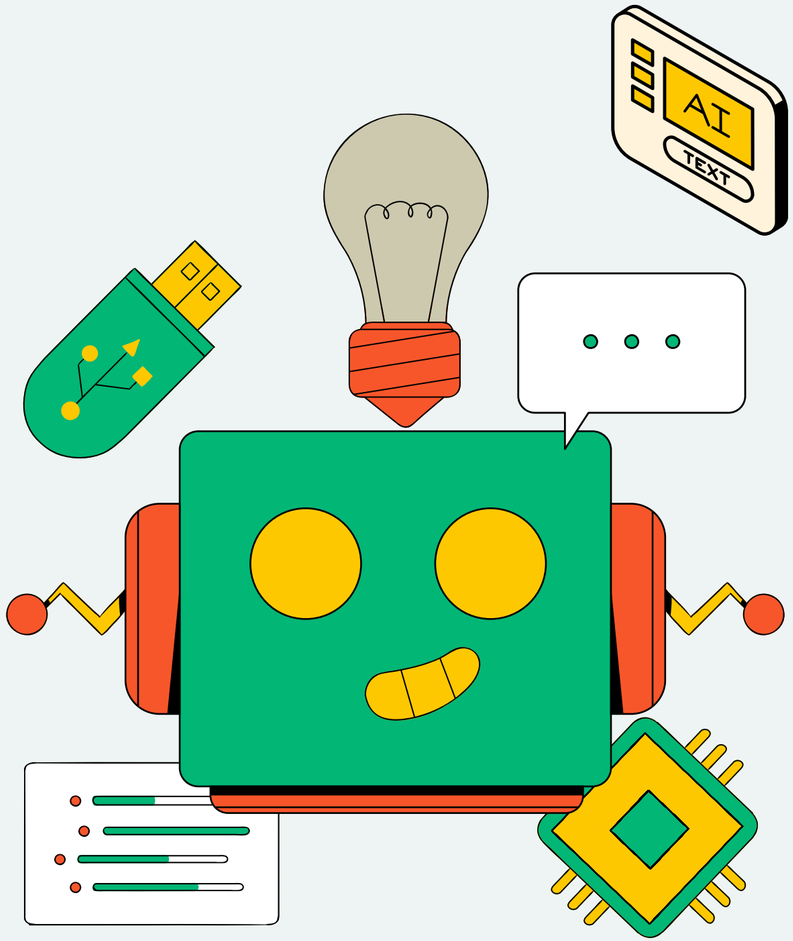
Infosys Springboard



TEXT SUMMARIZATION

GROUP – 4

PRESENTED BY:

SOUNAK PATRA

INTRODUCTION

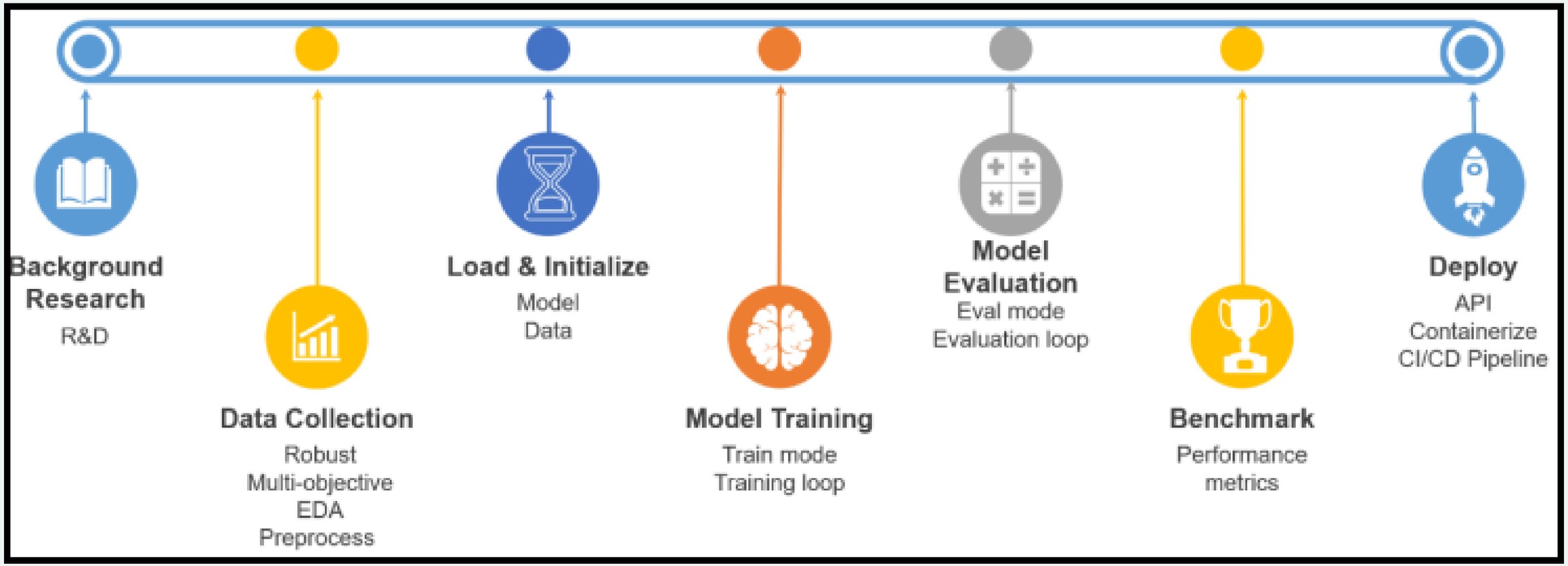
This project focuses on developing a comprehensive text summarization application that leverages both abstractive and extractive summarization techniques.

The goal is to provide a user-friendly interface where users can input text and receive concise summaries, improving information consumption and comprehension.

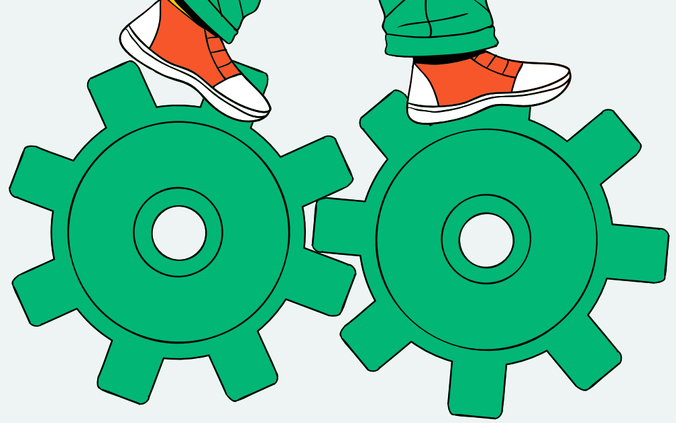
OBJECTIVES

- **Implement Dual Summarization Methods:** Integrate both abstractive and extractive summarization models to offer versatile summarization options.
- **Create a User-Friendly Interface:** Design and develop an intuitive and responsive UI using HTML, CSS, JavaScript, etc.
- **Evaluate and Optimize:** Assess the performance of the summarization models and optimize the application for accuracy and efficiency.
- **Deploy Using Modern Technologies:** Utilize FastAPI for backend processing and Docker for containerized deployment, ensuring scalability and ease of access.

INTENDED PLAN



RESEARCH



2019

EXTRACTIVE SUMMARIZATION :
Selects and extracts key sentences or phrases directly from the original text to form the summary.
Examples of tools: Sentence Transformers, NLTK.

2020

ABSTRACTIVE SUMMARIZATION :
Generates summaries by interpreting and rephrasing the original text.
Examples of models: Pegasus, BART.

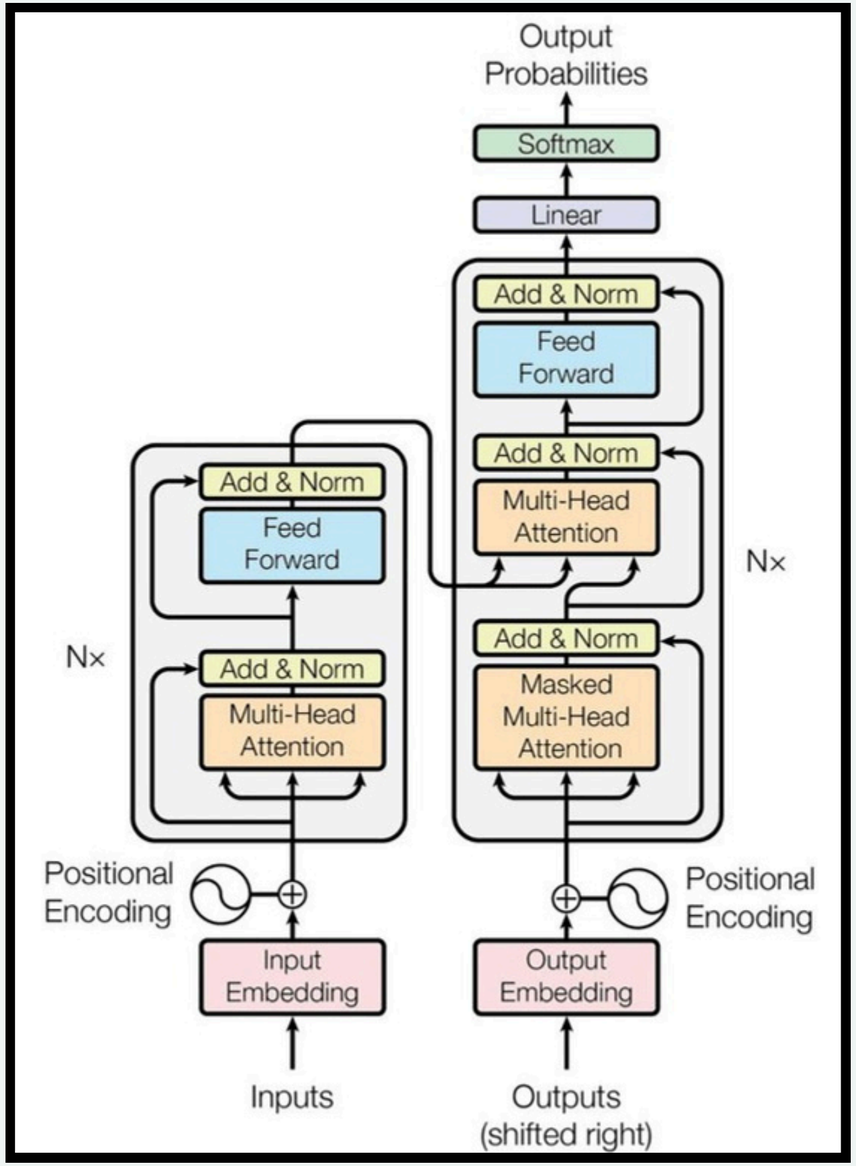
2023

MULTI-DOCUMENT SUMMARIZATION :
Combines and condenses information from multiple documents on the same topic into a single summary.
Examples of models: BERT, BART.

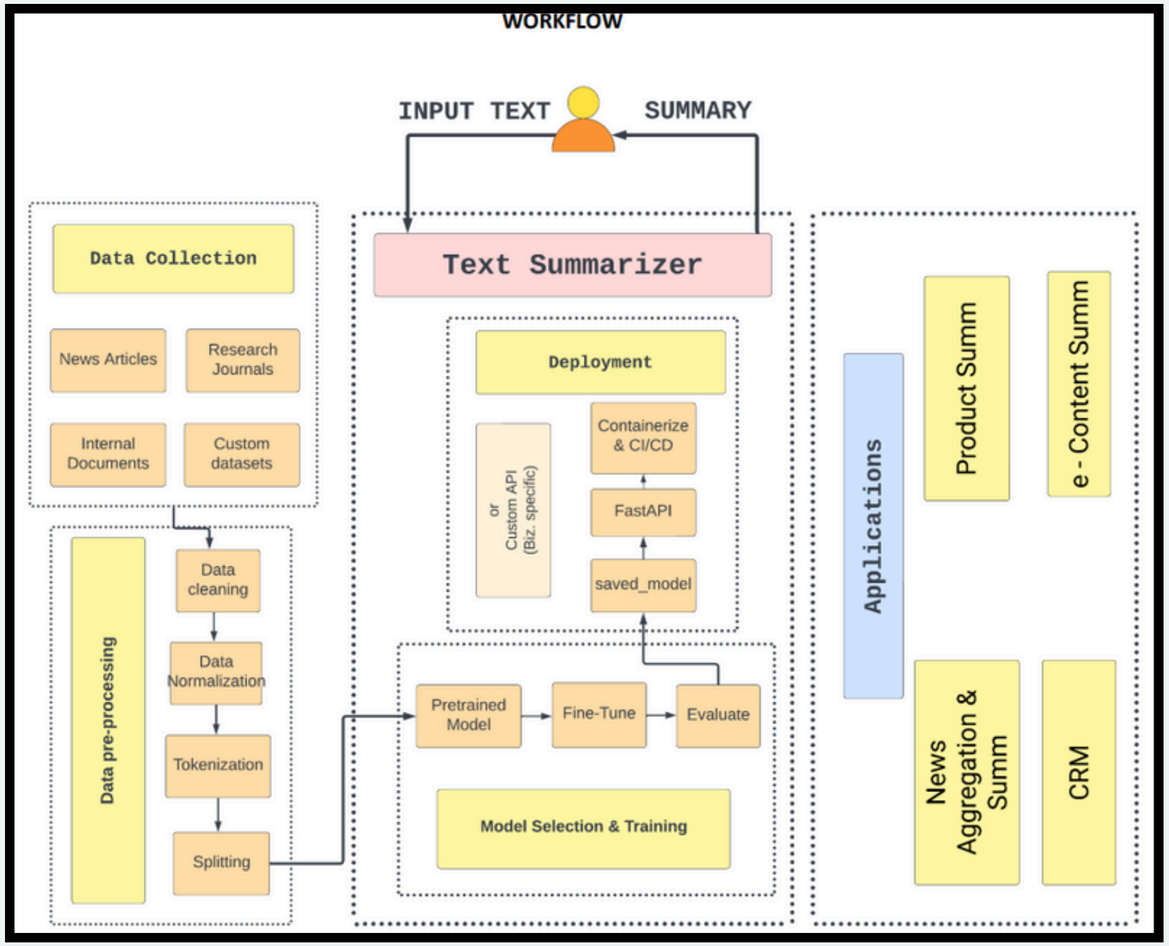
2023

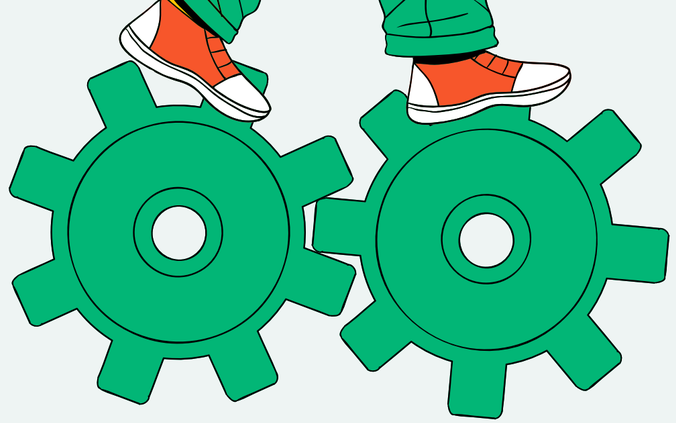
GENERAL TEXT SUMMARIZATION :
Text Summarization using Deep Learning Techniques.
Examples of Methods used: Deep Learning (Seq2Seq, Attention, Transformers).

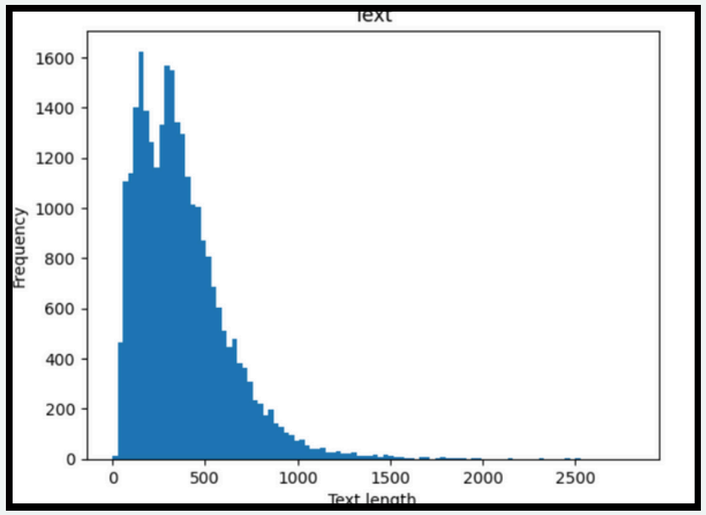
ARCHITECTURE SELECTED



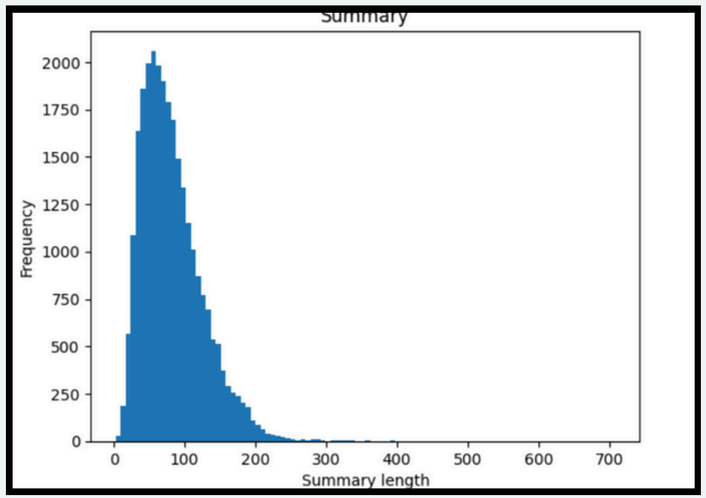
DESIGN WORKFLOW (ABSTRACTIVE)








```
count    27192.000000
mean       376.641586
std        251.860155
min          0.000000
25%        190.000000
50%        330.000000
75%        496.250000
max       2810.000000
Name: text, dtype: float64
```




```
count    27192.000000
mean        82.862570
std         43.924925
min          3.000000
25%         51.000000
50%         75.000000
75%        106.000000
max        708.000000
Name: summary, dtype: float64
```

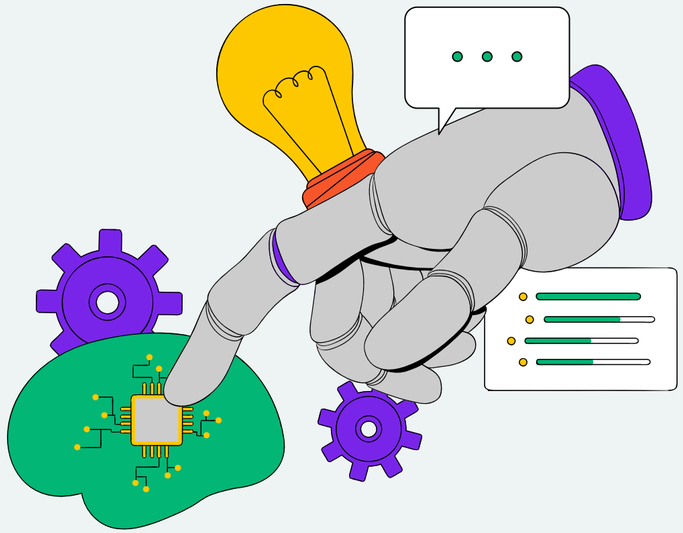
DATASET USED

- Datasets have been collected from the Huggingface datasets library.
- Data collected from different source :
  - **alexfabbri/multi_news :** Multi-News, consists of news articles and human-written summaries of these articles from the site newser.com.
  - **knkarthick/dialogsum :** DialogSum is a large-scale dialogue summarization dataset.
- Data Pre-processing : completed
  - Removed NULL records, punctuation, stop words, Lowercasing, etc.

MODEL TRAINING

- Load pre-trained transformer model
  - **Google /Pegasus-cnn_dailymail**
- 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. Trainer is a simple but feature – complete training and eval loop for PyTorch, optimized for transformers.

- The model was trained with whole dataset for 2 epochs for 1:15:11, (HH:MM:SS) in 1556 steps.
- Train loss = 1.32 (final)
- Trained model and obtained training history.
- Saved fine-tuned PEGASUS model and tokenizer.



```python
# Training arguments for Trainer
from transformers import TrainingArguments, Trainer

trainer_args = TrainingArguments(
    output_dir='/kaggle/working/pegasus-dialogsum', num_train_epochs=2, warmup_steps=500,
    per_device_train_batch_size=1, per_device_eval_batch_size=1,
    weight_decay=0.01, logging_steps=10,
    evaluation_strategy='steps', eval_steps=500, save_steps=1e6,
    gradient_accumulation_steps=16
)

# Trainer for Seq2Seq model training
trainer = Trainer(model=model_pegasus, args=trainer_args,
                  tokenizer=tokenizer, data_collator=seq2seq_data_collator,
                  train_dataset=dataset_dialogsum_pt["train"],
                  eval_dataset=dataset_dialogsum_pt["validation"])

# Train Seq2Seq model
trainer.train()
```

MODEL VALIDATION

We will use the performance metric to validate our model:

- **ROUGE (Recall-Oriented Understudy for Gisting Evaluation):** This metric helps in measuring the quality of summaries by comparing the overlap of n-grams, word sequences, and word pairs between the generated summary and a reference summary.
- **ROUGE-1:** Overlap of unigrams (single words)
- **ROUGE-2:** Overlap of bigrams (two word sequences)
- **ROUGE-L:** Measures the longest common subsequence (LCS) b/w candidate and reference summaries.
- **ROUGE-LSUM:** (LCS Summary) – variant of the ROUGE-L metric.

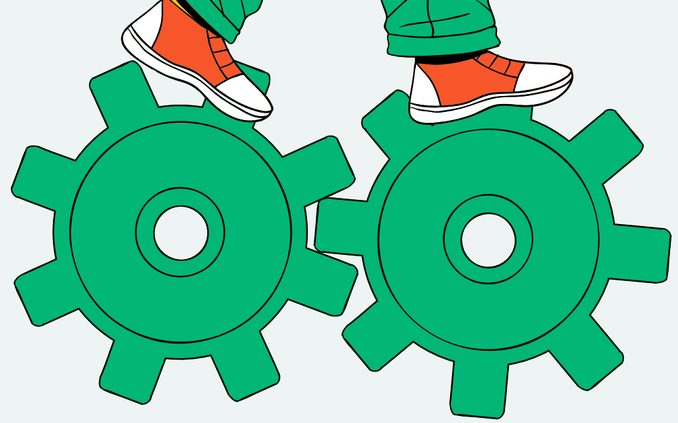
BEFORE FINE-TUNING

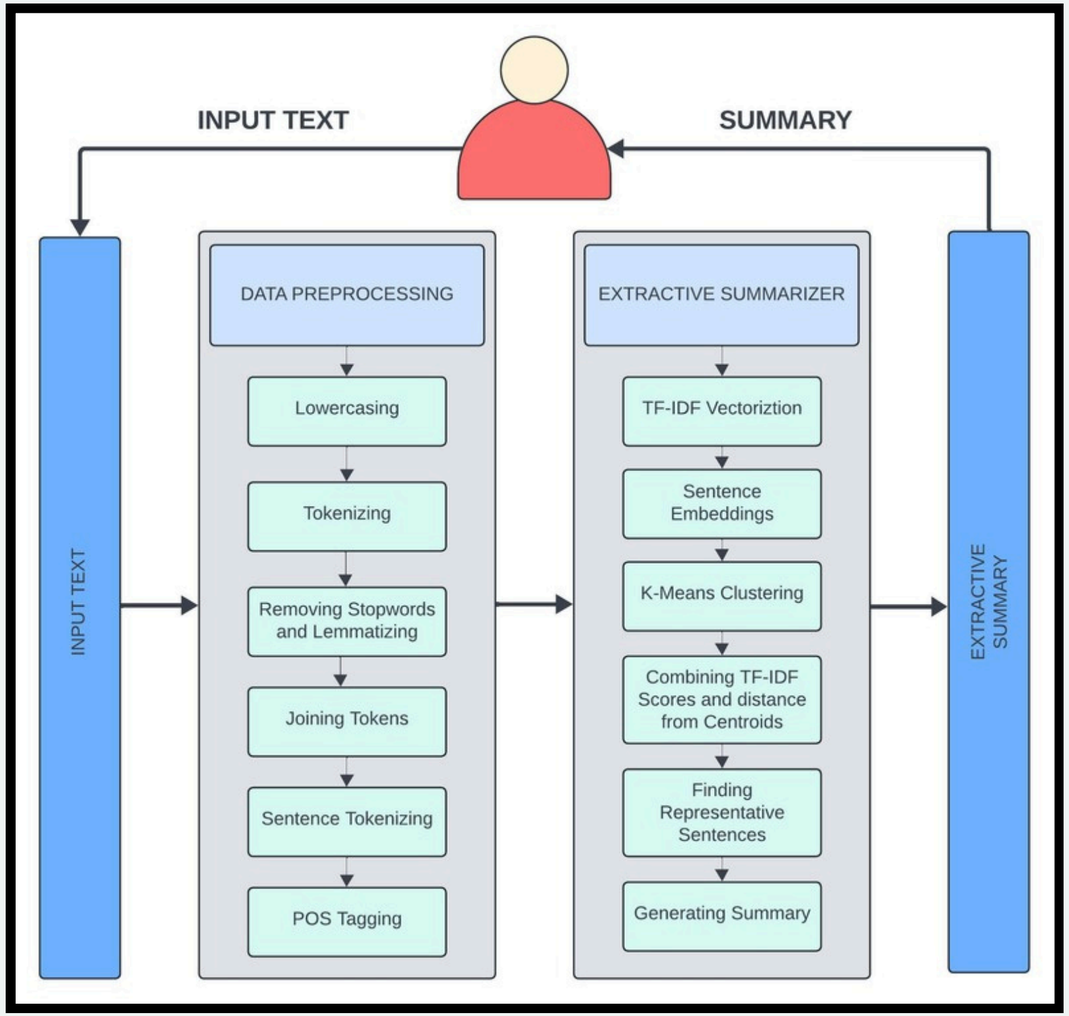
	rouge1	rouge2	rougeL	rougeLsum
pegasus	0.246641	0.06343	0.186421	0.186551

AFTER FINE-TUNING

	rouge1	rouge2	rougeL	rougeLsum
pegasus	0.404575	0.165404	0.332067	0.332024

DESIGN WORKFLOW (EXTRACTIVE)





EXTRACTIVE SUMMARIZATION

- The process of combining the matrix obtained from TF-IDF scores and KMeans Clustering methodology is used.
- Convert the articles/passages into a list of sentences using nltk's sentence tokenizer.
- For each sentence, extract contextual embeddings using Sentence transformer.
- Apply K-means clustering on the embeddings. The idea is to cluster the sentences that are contextually similar to each other & pick one sentence from each cluster that is closest to the mean(centroid).
- For each sentence embedding, calculate the distance from centroid. The distance would be zero if centroid itself is the actual sentence embedding.

MODEL VALIDATION

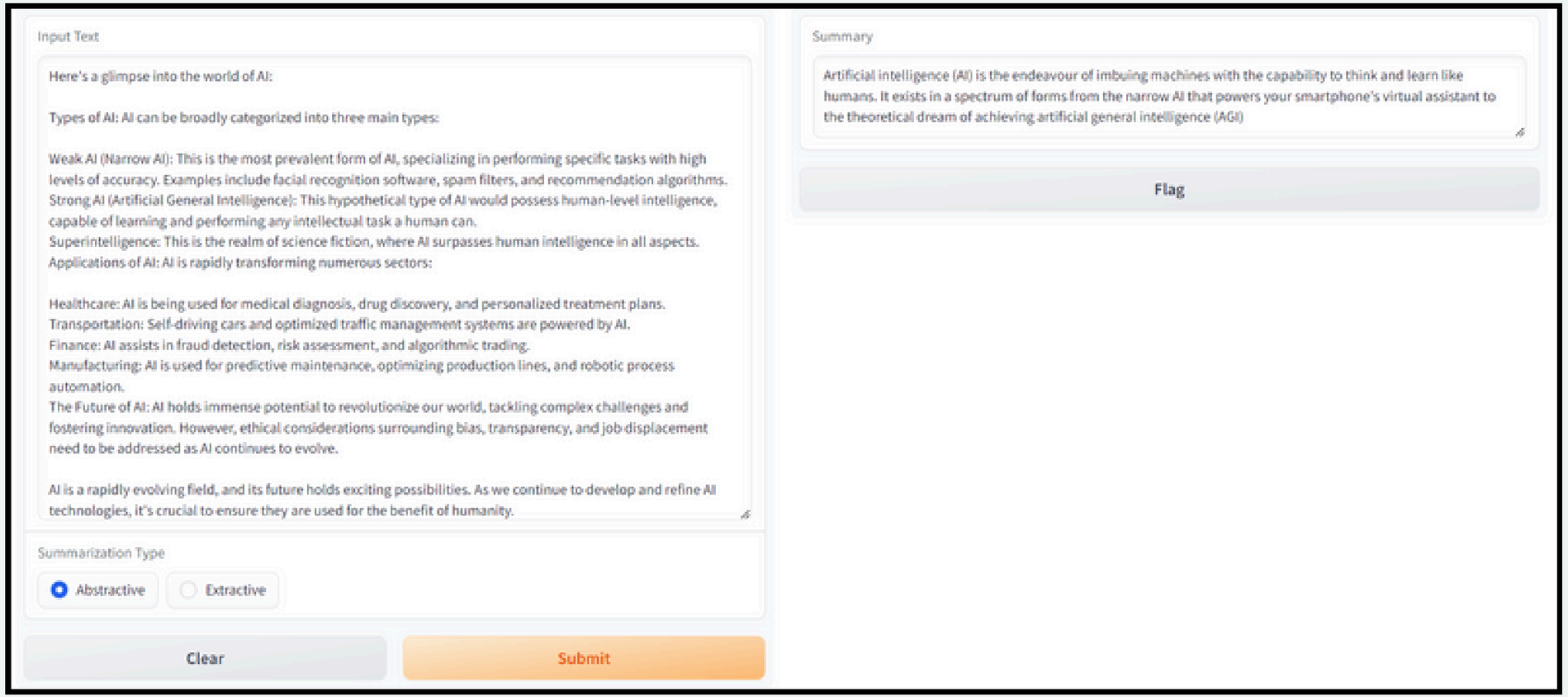
ROUGE (RECALL-ORIENTED UNDERSTUDY FOR GISTING EVALUATION)

This metric helps in measuring the quality of summaries by comparing the overlap of n-grams, word sequences, and word pairs between the generated summary and a reference summary.


```
ROUGE-1 Score:   0.4020618556701031
ROUGE-2 Score:   0.15625000000000003
ROUGE-L Score:   0.2268041237113402
```

TESTING

gradio



DEPLOYMENT

FastAPI

Docker

Azure Portal

Azure Container Instances

DEPLOYMENT

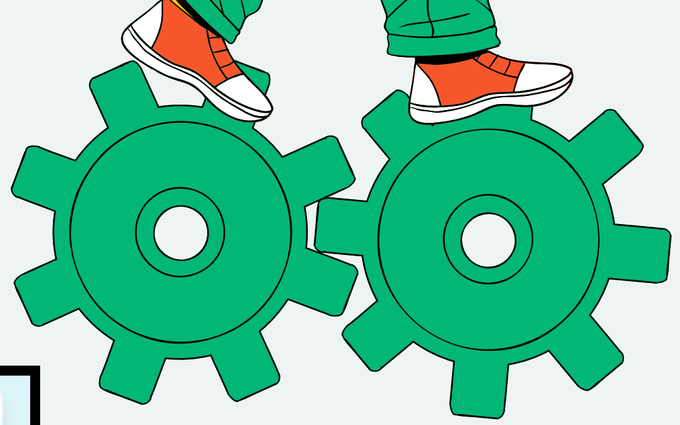
FastAPI

- Utilized the FastAPI framework to create a web application for text summarization.
- Defined API endpoints.
  - Accepts Text Input
  - Returns Abstractive and Extractive Summary based on choice.

- Containerized the entire Project along with the models used.
- Built the image and pushed to docker hub.

- **Deployed the docker image using docker Azure Container Instances.**
- **4 CPU cores for free Trail is a big advantage.**

DEPLOYED APPLICATION



Infosys Springboard

Text Summarizer

Artificial intelligence (AI) has become a ubiquitous term, appearing everywhere from news headlines to science fiction films. But what exactly is AI? In essence, it's the endeavor of imbuing machines with the capability to think and learn like humans. This involves simulating human intelligence processes such as reasoning, problem-solving, and learning from experience.

The concept of AI has been around for decades, but significant advancements in computing power and algorithms have propelled it into the forefront of technological development. Today, AI exists in a spectrum of forms, from the narrow AI that powers your smartphone's virtual assistant to the theoretical dream of achieving artificial general intelligence (AGI), a machine that surpasses human cognitive abilities.

Here's a glimpse into the world of AI:

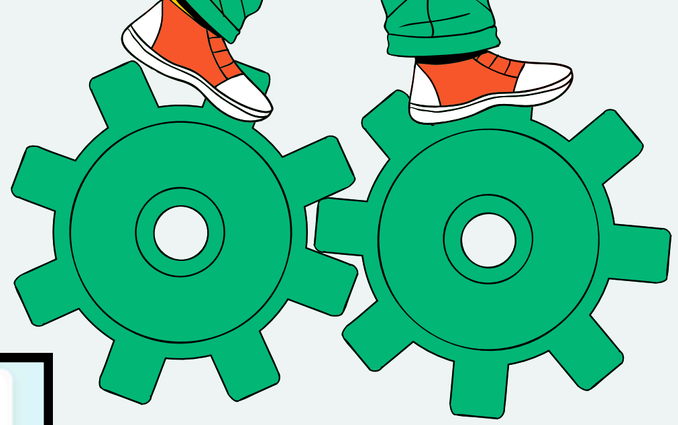
Choose summarization type: Abstractive

Summarize Clear

Summary:

Artificial intelligence (AI) is the endeavour of imbuing machines with the capability to think and learn like humans. Today, AI exists in a spectrum of forms, from the narrow AI that powers your smartphone's virtual assistant to the theoretical dream of achieving artificial general intelligence (AGI)

DEPLOYED APPLICATION



Infosys Springboard

Text Summarizer

Artificial intelligence (AI) has become a ubiquitous term, appearing everywhere from news headlines to science fiction films. But what exactly is AI? In essence, it's the endeavor of imbuing machines with the capability to think and learn like humans. This involves simulating human intelligence processes such as reasoning, problem-solving, and learning from experience.

The concept of AI has been around for decades, but significant advancements in computing power and algorithms have propelled it into the forefront of technological development. Today, AI exists in a spectrum of forms, from the narrow AI that powers your smartphone's virtual assistant to the theoretical dream of achieving artificial general intelligence (AGI), a machine that surpasses human cognitive abilities.

Here's a glimpse into the world of AI:

Choose summarization type: Extractive

Summarize Clear

Summary:

The concept of AI has been around for decades, but significant advancements in computing power and algorithms have propelled it into the forefront of technological development. But what exactly is AI? Superintelligence: This is the realm of science fiction, where AI surpasses human intelligence in all aspects. The Future of AI: AI holds immense potential to revolutionize our world, tackling complex challenges and fostering innovation. Finance: AI assists in fraud detection, risk assessment, and algorithmic trading. In essence, it's the endeavor of imbuing machines with the capability to think and learn like humans.

CONCLUSION

In this project, we developed a comprehensive text summarization application that supports both extractive and abstractive summarization techniques.

By leveraging state-of-the-art models like Pegasus for abstractive summarization, we ensured high-quality and coherent summaries. The application was implemented using a combination of Python libraries, including the Hugging Face Transformers library for model training and FastAPI for creating an interactive user interface

FUTURE SCOPE

- Multilingual Summarization:
  - Extend the application to support multiple languages, allowing for summarization of text in languages other than English. This would involve training and fine-tuning models on multilingual datasets.
- User Feedback & Continuous Improvement:
  - Implement mechanisms for users to provide feedback on the generated summaries, and use this feedback to iteratively improve the models and the application.

THINK UNLIMITED

WE LEARN FOR THE FUTURE

THANK YOU