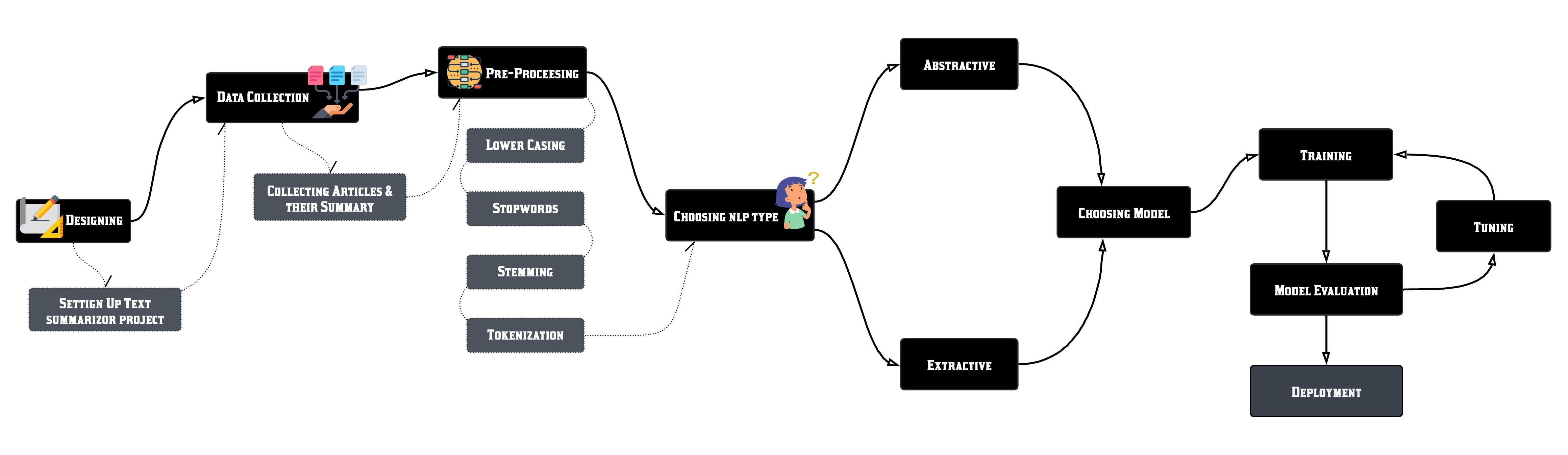
**Initial Design**



# Report on Text Summarizer Model Building

## Introduction

In this project, we aimed to build both extractive and abstractive text summarization models. The goal was to create models that can generate concise summaries of long text documents, significantly enhancing information consumption by reducing the time needed to understand large texts. Text summarization is a crucial task in natural language processing (NLP), with applications ranging from creating news digests to summarizing academic papers and reports.

## Types of Summarization

### Extractive Summarization

Extractive summarization involves selecting significant sentences, phrases, or sections directly from the source text and concatenating them to form a summary. This method retains the original wording and meaning of the text, making it simpler to implement as it does not require generating new text. The primary challenge in extractive summarization is identifying the most relevant parts of the text that represent the overall content effectively.

### Abstractive Summarization

Abstractive summarization, on the other hand, generates new sentences that convey the most critical information from the source text. This method involves understanding the text contextually and rewriting it concisely. Abstractive summarization is more complex as it requires the model to comprehend and paraphrase the content, often resulting in more coherent and human-like summaries compared to extractive methods.

## Extractive Summarization

Model Selection and Dataset

Initially, we chose the bert-base-uncased model from Hugging Face's Transformers library for extractive summarization. BERT (Bidirectional Encoder Representations from Transformers) is known for its robust language understanding capabilities, making it suitable for tasks requiring deep comprehension of the text. (Source: [Hugging Face](https://huggingface.co/docs/transformers/en/tasks/summarization))

For training and fine-tuning this model, we selected the CNN/DailyMail dataset, a widely recognized dataset for summarization tasks. This dataset consists of news articles and their corresponding summaries, providing a substantial amount of data for training summarization models.

* Here is the link to the Dataset: [CNN/DailyMail Dataset](https://www.kaggle.com/datasets/endofnight17j03/cnn-dailymail)

### Preprocessing

We used the BertTokenizer for tokenizing the input text. Tokenization involves splitting text into tokens, which are smaller units like words or subwords. This step is crucial for transforming the text data into a format that the BERT model can process. Proper tokenization ensures that the model can accurately interpret and learn from the input data.

### Challenges

Despite careful preprocessing, the model encountered errors during training. These errors persisted even after several attempts to troubleshoot and resolve them. After spending 2-3 days troubleshooting without success, we concluded that continuing with this model might not be the best approach. The errors could have been due to various factors such as Much Time Taking, model configuration problems, or hardware limitations.

### Model Transition

Based the challenges with the bert-base-uncased model, we decided to switch to a different model in order to save time and to improve result. We chose the t5-small model, also from Hugging Face, which is known for its efficiency and performance even on smaller hardware setups. T5 (Text-To-Text Transfer Transformer) is a versatile model designed to handle various text-to-text tasks, including summarization.

Using the t5-small tokenizer, we preprocessed the data again and trained the model. This time, the training process completed successfully, demonstrating the model's compatibility and robustness.

### Performance Evaluation

We evaluated the model using the ROUGE (Recall-Oriented Understudy for Gisting Evaluation) metric, which measures the quality of the summaries by comparing them to reference summaries. The ROUGE metric considers different aspects of the generated summary, such as the overlap of unigrams, bigrams, and longest common subsequences with the reference summary.

* **ROUGE-1**: 0.289743647211089
* **ROUGE-2**: 0.076129137694790
* **ROUGE-L**: 0.220193186663391

The scores indicated room for improvement, likely due to a limited number of training epochs and the nature of the CNN/DailyMail dataset, which often requires an abstractive summarization approach. The extractive summarization model performed adequately but could benefit from further fine-tuning and more extensive training.

## Abstractive Summarization

**Model and Dataset**

For the abstractive summarization task, we again used the t5-small model, given its suitability and our familiarity with it. Abstractive summarization involves generating new sentences, making it more complex and computationally demanding. We selected the XSum dataset for this task, which is tailored for abstractive summarization. The XSum dataset consists of single-sentence summaries of a news articles, providing a challenging and relevant dataset for our model.

* Here is the link: [XSum Dataset](https://huggingface.co/datasets/shalinik/xsum)

### Preprocessing and Training

We preprocessed the data using the t5-small tokenizer and trained the model with the tokenized data. The preprocessing step involved converting the text into a format that the T5 model can understand and process. This includes tokenizing the input text and creating attention masks to handle the varying lengths of the input sequences.

During training, we faced several challenges related to model performance. The initial fine-tuning attempts resulted in low ROUGE scores, indicating that the model's summaries were not accurately capturing the essence of the original text. To address this, we sought guidance from our mentor, Narendra.

### Fine-Tuning Iterations

As advised by Narendra, we fine-tuned the model multiple times to improve its performance. Fine-tuning is a critical step in training NLP models, where the pre-trained model is further trained on a specific dataset to adapt it to the desired task. This process involved adjusting the learning rate, batch size, and number of epochs to optimize the model's performance.

### Performance Evaluation

After several iterations of fine-tuning, the model's performance improved significantly, as measured by ROUGE scores:

* **ROUGE-1**: 0.289743647211089
* **ROUGE-2**: 0.076129137694790
* **ROUGE-L**: 0.220193186663391
* **ROUGE-Lsum**: 0.4918 (measures the longest common subsequence for summaries)

These scores were deemed acceptable for the abstractive summarization task, given the variability and complexity of generating new sentences. The model demonstrated a good balance between precision and recall, indicating its ability to generate coherent and relevant summaries.

## Conclusion

This project successfully explored and implemented both extractive and abstractive summarization models. Despite initial challenges with the bert-base-uncased model, transitioning to the t5-small model enabled us to build effective summarization systems. The development of a user-friendly interface further enhances the usability of these models, making text summarization accessible to a broader audience. The deployment phase will bring this tool into practical use, providing a valuable resource for users needing efficient text summarization.

By completing this project, we have gained valuable insights into the complexities of text summarization and the practical aspects of model training and deployment. This experience will undoubtedly contribute to future endeavors in the field of natural language processing and machine learning.