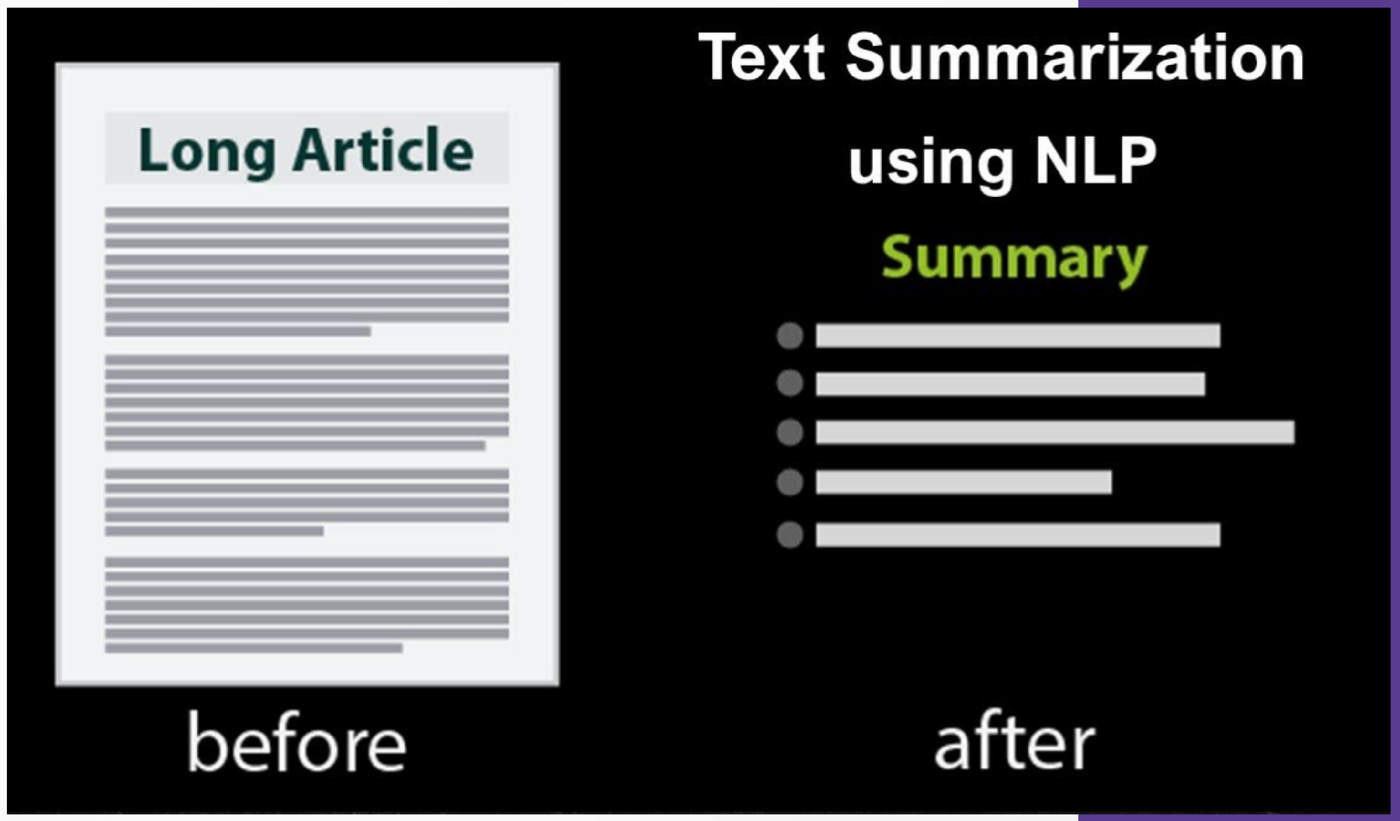
NEWS ARTICLE TEXT SUMMARIZATION

PROBLEM STATEMENT

DEVELOP AN AUTOMATED NEWS ARTICLE SUMMARIZATION SYSTEM USING NLP TECHNIQUES TO HELP USERS EFFICIENTLY EXTRACT KEY INFORMATION FROM LENGTHY NEWS ARTICLES. THE SYSTEM SHOULD PROVIDE CONCISE AND COHERENT SUMMARIES WHILE MAINTAINING THE ORIGINAL ARTICLE'S CONTEXT AND ESSENTIAL DETAILS



Dataset Descriptions

• CNN Dailymail

- The CNN / DailyMail Dataset is an English-language dataset containing just over 300k unique news articles as written by journalists at CNN and the Daily Mail. The current version supports both extractive and abstractive summarization, though the original version was created for machine reading and comprehension and abstractive question answering.

• Samsum Dataset

- The SamSum dataset contains about messenger-like conversations with summaries. Conversations were created and written down by linguists fluent in English. Linguists were asked to create conversations similar to those they write on a daily basis, reflecting the proportion of topics of their real-life messenger convesations. The style and register are diversified - conversations could be informal, semi-formal or formal, they may contain slang words, emoticons and typos. Then, the conversations were annotated with summaries. It was assumed that summaries should be a concise brief of what people talked about in the conversation in third person. The SamSum dataset was prepared by Samsung R&D Institute Poland and is distributed for research purposes

Models

BART

- BART is a denoising autoencoder for pretraining sequence-to-sequence models. It is trained by (1) corrupting text with an arbitrary noising function, and (2) learning a model to reconstruct the original text. It uses a standard Transformer-based neural machine translation architecture.
- This type of model is relevant for machine translation, question-answering , text summarization, or sequence classification (categorizing input text sentences or tokens).

T5

- T5 (Text-To-Text Transfer Transformer) is a transformer model that is trained in an end-to-end manner with text as input and modified text as output, in contrast to BERT-style models that can only output either a class label or a span of the input. This text-to-text formatting makes the T5 model fit for multiple NLP tasks like Summarization, Question-Answering, Machine Translation, and Classification problems.

Working Principle BART Model

1. Data Preparation:

- We have a suitable dataset for training and testing our BART model. Summarization datasets like CNN/Daily Mail or SamSum.
- Preprocess and clean our data, removing any unnecessary information and formatting it appropriately for our model.

2. Model Selection:

- We choose from various pre-trained BART models. The choice depends on the computational resources available and the specific requirements of our project.

3. Fine-Tuning:

- Fine-tune the BART model on our summarization dataset. We can use popular deep learning frameworks like PyTorch or TensorFlow for this purpose.
- Implement the appropriate loss function for the summarization task, such as token-level cross-entropy loss or a combination of loss functions like reconstruction loss and language modeling loss.

4. Evaluation Metrics:

- Used suitable evaluation metrics like ROUGE (Recall-Oriented Understudy for Gisting Evaluation) to measure the quality of your model's summaries. ROUGE measures the overlap between the generated summary and the reference summary.

5. Hyperparameter Tuning:

- Experiment with different hyperparameters, such as learning rates, batch sizes, and the number of training epochs, to optimize your model's performance.

6. Text Preprocessing:

- Preprocess our input text by tokenizing and encoding it appropriately for the BART model. BART typically uses subword tokenization, such as Byte Pair Encoding (BPE).

7. Inference:

- Once our model is trained and fine-tuned, we can use it for inference. Provide a document as input, and your model will generate a summary.

8. Post-processing:

- We may need to perform post-processing on the generated summaries, such as detokenization and removing extra symbols or artifacts.

Working Principle T5 Small Model

1. Data Preparation:

- We gather a suitable dataset for your text summarization task. Common summarization datasets like CNN/Daily Mail or samsum used.
- Clean and preprocess our data, ensuring it's formatted appropriately for the T5 model.

2. Model Selection:

- We choose the T5-Small model for our project. T5-Small is a smaller version of the T5 model and may be more manageable if we have limited computational resources.

3. Fine-Tuning:

- Fine-tune the T5-Small model on your summarization dataset using a deep learning framework like Hugging Face's Transformers library, which provides pre-trained T5 models and tools for fine-tuning.
- Define the loss function for your summarization task, such as token-level cross-entropy loss.

4. Evaluation Metrics:

- Utilize appropriate evaluation metrics like ROUGE. ROUGE measures the overlap between the generated summary and the reference summary.

5. Hyperparameter Tuning:

- Experiment with different hyperparameters, such as learning rates, batch sizes, and the number of training epochs, to optimize your model's performance.

6. Text Preprocessing:

- Preprocess our input text by tokenizing and encoding it according to the T5 model's requirements. T5 typically uses subword tokenization methods, such as SentencePiece or Byte Pair Encoding (BPE).

7. Inference:

- After fine-tuning, you can use your T5-Small model for summarization. Provide the input text, and the model will generate a summary.

8. Post-processing:

- Perform any necessary post-processing on the generated summaries, such as detokenization and removal of extra symbols or artifacts.

Performance Metric of two models :

```
rouge_dict  {'rouge1': 0.365079365079365, 'rouge2': 0.14516129032258066, 'rougeL': 0.20634920634920634, 'rougeLsum': 0.2857142857142857}
rouge_dict  {'rouge1': 0.1758241758241758, 'rouge2': 0.0, 'rougeL': 0.13186813186813187, 'rougeLsum': 0.15384615384615383}
rouge_dict  {'rouge1': 0.3655913978494624, 'rouge2': 0.13186813186813184, 'rougeL': 0.2150537634408602, 'rougeLsum': 0.3225806451612903}
```

	rouge1	rouge2	rougeL	rougeLsum
baseline	0.365079	0.145161	0.206349	0.285714
t5	0.175824	0.000000	0.131868	0.153846
bart	0.365591	0.131868	0.215054	0.322581

Output :

For Customized input

```
[23] pipe = pipeline("summarization", model="facebook/bart-large-cnn")
     sampleex="Editor's note: In our Behind the Scenes series, CNN correspondents share their experiences in covering news and analyze the stories behind the events.
     pipe_out = pipe(sampleex)
```

28s

```
[24] pipe_out
```

0s

```
[{'summary_text': 'Judge Steven Leifman is an advocate for justice and the mentally ill. About one-third of all people in Miami-Dade county jails are mentally
ill, he says. He says the sheer volume is overwhelming the system. Starting in 2008, many inmates will be sent to a new mental health facility.'}]
```

THANK YOU