BBC NEWS SUMMARIZATION USING TRANSFORMERS

Introduction : The goal of this news summarization project is to condense a lengthy document or news story so that no important information is lost.

Summarization Techniques:
Extractive Summarization - There are essentially two methods for summarising text.
By choosing a portion of the entire sentence base, extractive summarization creates a summary from the given text. Based on a score that is calculated based on the words in that sentence, the most crucial phrases or sentences from the text are determined and chosen.

Abstractive Summarization - The text document is first analysed in the abstractive summary approach in order to provide an interpretation. The computer generates a summary based on this interpretation. By paraphrasing portions of the original text, it changes the meaning of the text.

Since abstractive summarization is more rigorous and simulates human vision when producing summaries, it will be the main focus of our project.

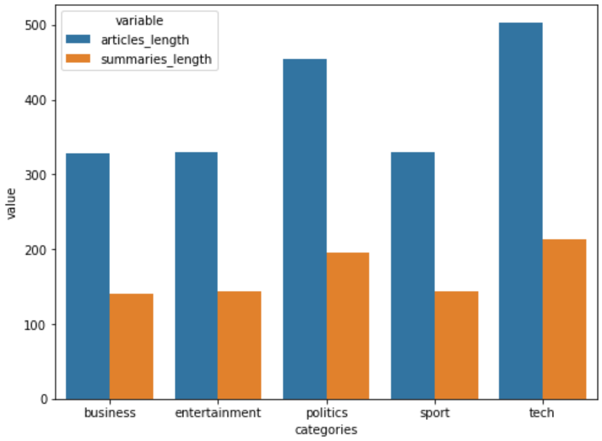
Methodology

Dataset

BBC News dataset, consisting of 2225 documents from the BBC news website matching to articles in five thematic categories from 2004–2005, was utilised to build this dataset for text summarisation.

EDA

The average length of summaries of the 167 words and articles length is 389



Model Explanation

All the model are based on Transformers

We will be using multiple Pre Trained Model from Hugging Face

Many helpful NLP packages and datasets are available through the open source project hugging face. The Transformer library is the one that is most well-known. The transformer library includes many pre-trained models that can be adjusted for any dataset to predict text summaries. We will talk about various pre-trained models that were optimised and put into use for the BBC news dataset to provide pretty accurate summaries.

BART BASE : Bidirectional and Auto Regressive Transformers, or BART. It was constructed using a seq2seq model that has denoising as a pre-training objective. It employs the common seq2seq model design, which combines a decoder that resembles GPT with an encoder that is comparable to BERT. The pre-training work entails randomly switching the original phrase orders as well as a new method that switches text ranges with a single mask token. It is somewhat similar to the BERT model, however BART has around 10% more characteristics than the equivalent-sized BERT model. The autoregressive BART decoder is regulated to produce sequential NLP tasks like text summarization. The denoising pre-training aim is strongly tied to the fact that the data is taken from the input but altered. As a result, the encoder's input is the input sequence embedding, and the decoder's output is produced autoregressively. The pre-trained model "`BART_Finetuned_CNN_dailymail`" and the Bart tokenizer, which is made from the GPT-2 tokenizer, were both utilised. Because of this, words are encoded differently based on where they are in a phrase.

T5 : Text-to-Text Transfer Transformer is referred to by the acronym T5, or T5. Transfer learning is the principle underpinning the T5 model. The model was originally trained using Transfer Learning on a task with a lot of text before being fine-tuned on a downstream task so that the model acquires general-purpose abilities and knowledge to be used to tasks like summarization. T5 employs a sequence-to-sequence generation technique that feeds the decoder's output, which is autoregressive, the encoded input through cross-attention layers. We have improved a T5 model in which the encoder receives as input a list of tokens that are then translated into a list of embeddings. The encoder block has a block with two subcomponents, a selfattention layer and a feed forward network.The only structural difference between the encoder and decoder is that every selfattention layer is followed by a generalised attention mechanism. This enables the model to only work with the earlier results. An output from the last decoder block is passed into the following layer. The activation function in this last layer, which is dense and uses softmax, is. The input embedding matrix receives the weights from this layer's output.

PEGASUS : Pre-training with Extracted Gap-sentences for Abstractive Summarization Sequence-to-Sequence Models is known as PEGASUS.\Significant lines are removed from the input text and compiled as distinct outputs in this model.
Additionally, selecting only pertinent sentences works better than selecting sentences at random. As it is analogous to the work of reading the complete material and producing a summary, this style is chosen for abstractive summarising. The PEGASUS model is created by training a Transformer model using text input. The CNN/DailyMail summary datasets are used to pretrain the algorithm.It was discovered that masking whole sentences from the text and creating gap sentences from the remaining text works well as a previous training objective for text summarising.

Results

The inference is done with `min_length` = 20 and `max_length` = 90
This is due the business constraint of summary being no longer than a 30 second read and should cover all the points in the article that gives us summary to be of range 75-90 words

1) Qualitative Analysis

We Fine Tuned the following transformer-based pre-trained language models from the Hugging face library to provide summaries on BBC News Dataset that has human generated summmaries.

We will use one of these summaries from the test data and perform comprehensive analysis of summaries generated by different models.
We will compare summaries output from different models with the given reference summary

“Google shares have risen since but fell $12.33 on Tuesday to close at $172.55.There were fears that the potential increase of shares in circulation from Tuesday would ease demand for stock.In one of the most closely-watched initial public offerings in stock market history, the US-based company sold 19.6 million shares at $85 each.Selling the shares could turn many of Google's workers into millionaires.However, analysts say they expected most shareholders would be holding back from selling all their shares immediately, as Google's good performance and future growth potential means demand will hold.”

Summary Generated by BART:
“Shares in Google have fallen 6.7% after employees and early investors in the web search took advantage of the first chance to sell their holdings.Google's main service - its internet search - is free to users, so the firm makes much of its money from selling advertising space linked to the words for which its users search.However, analysts say they expected most shareholders would be holding back from selling all their shares immediately, as Google's”

Summary Generated by T5:
"There were fears that the potential increase of shares in circulation from Tuesday would ease demand for stock.Shares in Google have fallen 6.7% after employees and early investors in the web search took advantage of the first chance to sell their holdings.However, analysts say they expected most shareholders would be holding back from selling all their shares immediately, as Google's good performance and future growth potential means demand will hold.Google shares have risen since"

Summary Generated by Pegasus:
“There were fears that the potential increase of shares in circulation from Tuesday would ease demand for stock.Shares in Google have fallen 6.7% after employees and early investors in the web search took advantage of the first chance to sell their holdings.In its first earnings report since floating on the stock market, Google said it made a net profit of $52m in the three months ending 30 September.Google shares have risen since but fell $12.33 on Tuesday”

Findings after analysis summaries from all three transformers qualitatively

BART : The summaries produced were fluid, precise, and included corroborating information from the source paper. As a result, the summaries produced by the BART pretrained model show that the BART model is effective for text comprehension

T5: The T5 model displays successful outcomes. Coherent and precise summaries are produced. These summaries adequately matched the original synopsis and kept the text's intended meaning.

Pegasus : We obtained fluid and cogent summaries after optimising and applying our model to our dataset. These summaries closely matched the format of ground truth summaries and had excellent linguistic quality.

2) Quantitative analysis

Models	**ROUGE-1**	**ROUGE-2**	**ROUGE-3**	**ROUGE-Lsum**
BART BASE	57.8	53.4	46.4	46.5
T5	54.7	49.9	43.5	43.5
Pegasus	**60.8**	**57.2**	**49.1**	**49.1**

From the above table we clearly can see that Pegasus model outpeforms other transformers models

Conclusion

For the purpose of summarization, we constructed pre-trained language models based on the transformer architecture. Our study led us to the conclusion that carefully calibrated transformers built on top of language models that had already been trained produced excellent results and produced a sound and fluid summary of a given text content. For comparison studies, we calculated ROUGE scores for each model's predictions, and we found that the Pegasus model performed better than the other models.

What else could have done :

Preprocessing: Remove stopwords and punctuation, for example, as part of the correct preprocessing of the data.

Examine several models: Although we utilised the standard model for summarization in our lesson, there are many more models that you may use to complete this work. One of those may be superior.

Perform hyperparameter adjustment: We utilised a certain set of hyperparameters while training the model (learning rate, number of epochs, and so on).Use varied parameters for generating text — We have previously created summaries using various settings to make use of beam search and sampling. Test out various settings and parameters.

Experiments : All the experiment are done on single gpu g4dnx.large
And the deployment of endpoint is done using the same

List of ways to decrease the model latency

1. Compress to ONNX format (with fastT5 for example) the finetuned T5 base model.
2. Upload the ONNX T5 base model to S3 in AWS
3. Use the ONNX T5 base model in AWS SageMaker DLC in order to make inferences
4. Qunatize the pytorch using https://pytorch.org/docs/stable/quantization.html.PyTorch supports INT8 quantization compared to typical FP32 models allowing for a 4x reduction in the model size and a 4x reduction in memory bandwidth requirements

Example https://gist.github.com/patil-suraj/09244978af5f7598dd30fb9b4f54fe29 for onnx t5 based model inference