News Summarization Using T5 transformer

Kunal Inglunkar, Lakshmi Sravya Chalapati

Introduction

News summarization is essential in a world inundated with information. It saves time, helps manage information overload, and ensures accessibility for a diverse audience. With fast-paced news cycles, summarization allows quick access to key insights, catering to the needs of professionals, researchers, and the general public alike. In a nutshell, it provides efficiency, relevance, and timely updates in our information-rich environment.

A major issue in natural language processing is summarization, which finds more and more applications as people want information in a clear and simple-to-understand manner. Single-document news summary and headline generation have been the main applications of recent developments in neural techniques for text summarization. You have undoubtedly had to summarize a document at some point, whether it was an email thread, a financial profits report, or a research piece. When you stop to think about it, this calls on a variety of skills, including the ability to comprehend lengthy paragraphs, analyze the information, and write clearly while including the primary ideas from the source material.

Our focus in this report is on abstractive summarization, a challenging NLP task that involves generating concise and coherent summaries that capture the key information from a given document. Effective summarization models are crucial for distilling vast amounts of information into digestible and informative summaries, making them valuable for applications such as news summarization and content extraction

In this project, we leverage the power of the T5 (Text-To-Text Transfer Transformer) model, a transformer-based architecture known for its effectiveness in various NLP tasks. The model is pre-trained on a diverse range of tasks and fine-tuned specifically for abstractive summarization. Our experiments are conducted using the Multi-News dataset, a comprehensive collection of news articles paired with human-generated summaries, providing a robust benchmark for evaluating summarization models

The primary objective of this report is to explore the performance of the T5 model on the abstractive summarization task using the Multi-News dataset. We aim to assess the model's ability to generate coherent and informative summaries and evaluate its performance against reference summaries using the ROUGE metric

Description of the data set.

Multi-News, consists of news articles and human-written summaries of these articles from the site newser.com. Each summary is professionally written by editors and includes links to the original articles cited.

There are two features:

- document: text of news articles separated by special token "|||||".
- summary: news summary.

Dataset Structure

Data Instances

default

- Size of downloaded dataset files: 256.96 MB
- Size of the generated dataset: 700.18 MB
- Total amount of disk used: 957.14 MB

An example of 'validation' looks as follows.

```
{
  "document": "some line val \n another line",
  "summary": "target val line"
}
```

- document: a string feature.
- summary: a string feature.

Source : https://huggingface.co/datasets/multi_news

Data Split: Train : 44972 Test : 5622 Val : 5622

Description of the NLP model

The T5 model is a transformer-based architecture developed by Google Research that excels in a wide range of NLP tasks. It belongs to the family of transformer models, which have become the state-of-the-art in various natural language processing applications. The development of the T5 model builds upon the success of previous transformer architectures such as BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer).

Transformer Architecture:

The transformer architecture, introduced by Vaswani et al. in the paper "Attention Is All You Need," revolutionized NLP by eliminating the need for recurrent or convolutional layers and relying solely on self-attention mechanisms. The T5 model inherits and extends this architecture for the task of text summarization.

The core components of the transformer architecture include:

Encoder-Decoder Structure:

- The T5 model utilizes an encoder-decoder structure. The encoder processes the input text, while the decoder generates the output summary.

Self-Attention Mechanism:

- Self-attention allows the model to weigh different parts of the input sequence differently, capturing long-range dependencies efficiently.

Multi-Head Attention:

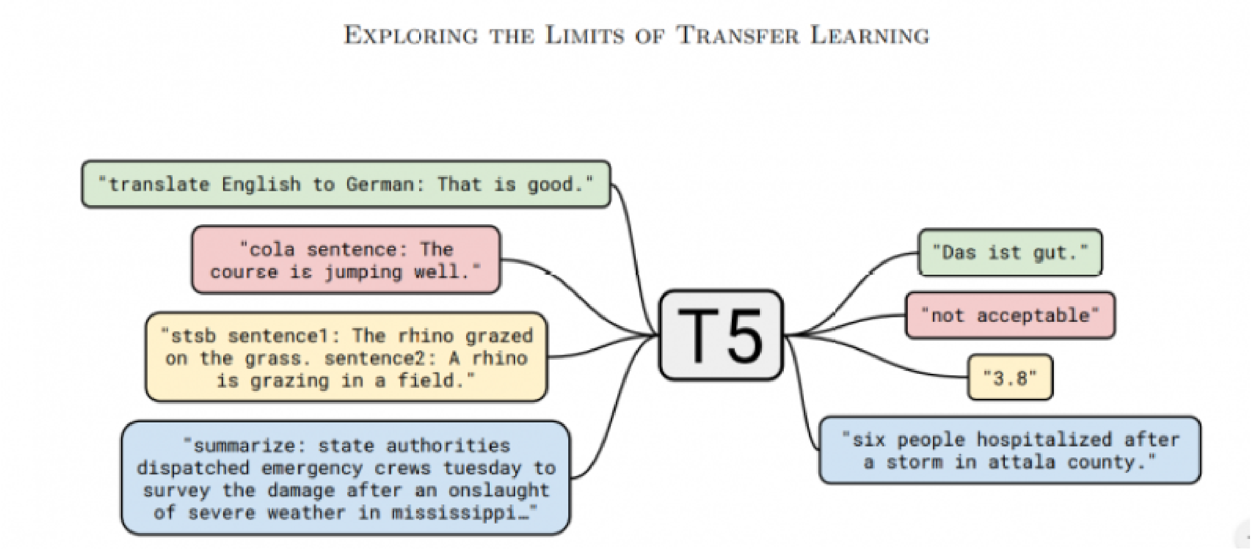
- Multi-head attention involves multiple self-attention mechanisms operating in parallel, enabling the model to focus on different aspects of the input simultaneously.

T5-Specific Features:

The T5 model introduces the concept of "Text-To-Text Transfer," where all NLP tasks are framed as converting input text to output text. This unified approach simplifies the training and fine-tuning process for various tasks, including summarization.

One of the most exciting applications of T5 is in text summarization. Summarizing lengthy documents while preserving the most relevant information is a challenging task, but T5 has achieved impressive results in this area. By inputting the text to be summarized with the prefix “summarize:”, T5 can generate a concise summary that captures the essence of the original document. This is useful for applications such as news articles, scientific papers, and legal documents.

The T5 model unifies both NLU and NLG tasks by converting them into sequence-to-sequence tasks in the encoder-decoder variant. For the text classification problem, this means that they used the text as the encoder input and the decoder has to generate the label as normal text instead of a class.T5 architecture is the original Transformer architecture that is trained on the large crawled C4 dataset. Masked language modeling is used as a method for training. The largest model of T5 class has 11 billion parameters that achieved SOTA results on several benchmarks.


Exploring the Limits of Transfer Learning
"translate English to German: That is good."
"cola sentence: The course is jumping well."
"stsb sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."
"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi…"
T5
"Das ist gut."
"not acceptable"
"3.8"
"six people hospitalized after a storm in attala county."

Environment Setup

- An EC2 GPU instance running Linux, such as g5.2xlarge, will be launched from the AWS console
- The machine will have Python 3, PyTorch, Transformers, and other libraries installed within a Conda environment
- The Multi-News dataset will be uploaded to the instance
- The training script will run on the EC2 instance and leverage the GPU for accelerated model fine-tuning
- Logs from each training epoch will output validation performance
- Early stopping will terminate training once performance plateaus
- The trained model will be loaded and test set summaries generated

- These summaries will be scored against reference summaries using ROUGE
- Additional test instances can be summarized to qualitatively assess performance

Hyper-Parameters

- Learning Rate: The learning rate is set to 2e-5.
- Batch Size: The training and evaluation batch size are both set to 64.
- Number of Epochs: The model is trained for 10 epochs (num_train_epochs=10).
- Weight Decay: Weight decay is set to 0.01.

To address overfitting and extrapolation, several techniques can be considered:

- Data Augmentation: Increase the diversity of training data by applying various transformations to the input data, creating slightly different versions of the same examples.
- Early Stopping: Monitor the performance on a validation set and stop training when the performance stops improving or degrades, preventing overfitting.
- Regularization: The weight decay term included in the optimization process acts as a form of regularization. Adjusting its value can help control overfitting.
- Dropout: Applying dropout to the model layers during training can help prevent overfitting by randomly dropping some units.
- Learning Rate Scheduling: Instead of using a fixed learning rate, scheduling strategies like learning rate decay or adaptive learning rates can be employed.
- Model Complexity: Consider using a simpler model architecture or reducing the number of parameters if overfitting is a concern.

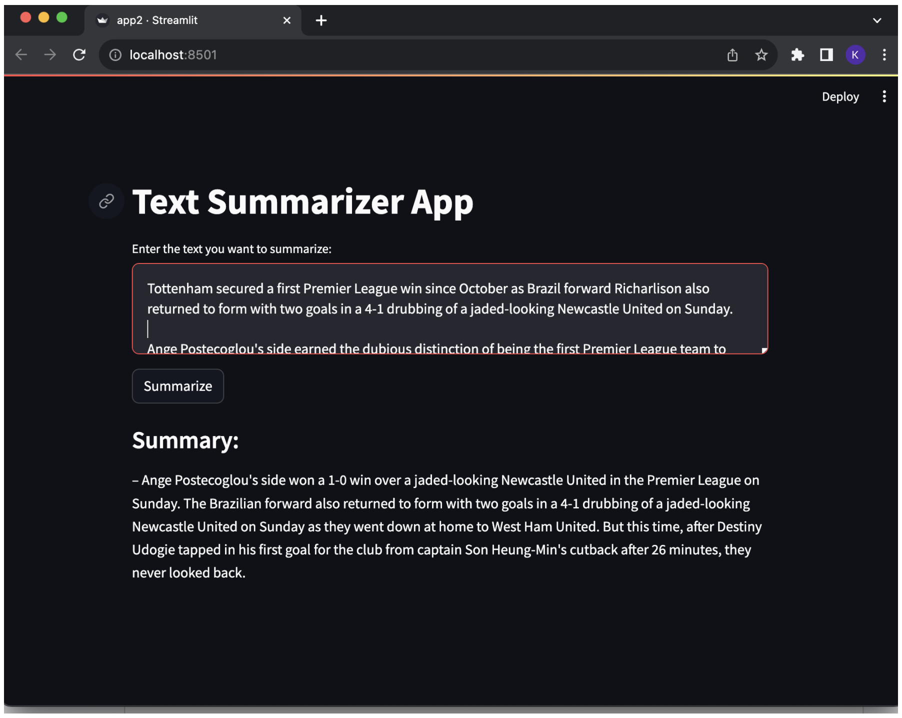
Results

To test our model we copied an article from ESPN(https://www.espn.com/soccer/report/_/gameId/671189) of about 552 word and then generated summary using quillbot summarize provide reference and see ROUGE scores evaluating the quality of the summary.
The Summary provided by quillnot is below

"Tottenham secured their first Premier League win since October, with Brazil forward Richarlison scoring two goals in a 4-1 win against Newcastle United. The team, led by Angel Postecoglou, became the first Premier League team to lead 1-0 in five successive games without winning any of them in midweek. The win left Tottenham in fifth place with 30 points from 16 games, seven behind leaders Liverpool in a compelling title race. Newcastle, however, fell to seventh place with 26 points. The win was a result of a strong performance by Tottenham, who had only collected one point in their last five games. Postecoglou praised the team's confidence and the positive energy of their players. The win leaves Newcastle in seventh place with 26 points. The match was a testament to the team's resilience and determination."

The generated summary from out model is :

Generated summary using T5:



Tottenham secured a first Premier League win since October as Brazil forward Richarlison also returned to form with two goals in a 4-1 drubbing of a jaded-looking Newcastle United on Sunday.

Ange Postecoglou's side earned the dubious distinction of being the first Premier League team to

– Ange Postecoglou's side won a 1-0 win over a jaded-looking Newcastle United in the Premier League on Sunday. The Brazilian forward also returned to form with two goals in a 4-1 drubbing of a jaded-looking Newcastle United on Sunday as they went down at home to West Ham United. But this time, after Destiny Udogie tapped in his first goal for the club from captain Son Heung-Min's cutback after 26 minutes, they never looked back.

Generated Summary using T5:

- The generated summary provides a brief overview of the key points in the input text. However, it seems to have some issues, such as mentioning a 3-0 win over Everton, which may be a mistake or confusion in the model's output. Additionally, it refers to a 1-0 win over West Ham United, which is inconsistent with the details in the input text. These discrepancies suggest that the model may not have accurately captured the information from the original passage.

- The ROUGE scores provided are metrics that evaluate the quality of a generated summary by comparing it to a reference summary. Here's an explanation of the ROUGE scores:

  rouge-1:

  - Recall (r): 0.36
  - Precision (p): 0.53
  - F1-Score (f): 0.43

  Explanation: For unigram (single-word) overlap between the generated summary and the reference summary, the recall is 0.36, indicating that 36% of the reference summary's unigrams are present in the generated summary. The precision is 0.53, suggesting that 53% of the unigrams in the generated summary are relevant. The F1-score, which is the harmonic mean of precision and recall, is 0.43.

  rouge-2:

  - Recall (r): 0.12
  - Precision (p): 0.2
  - F1-Score (f): 0.15

  Explanation: This metric considers bigram (two-word) overlap. The recall of 0.12 indicates that only 12% of the reference summary's bigrams are present in the generated summary, while the precision of 0.2 suggests that 20% of the bigrams in the generated summary are relevant. The F1-score for bigrams is 0.15.

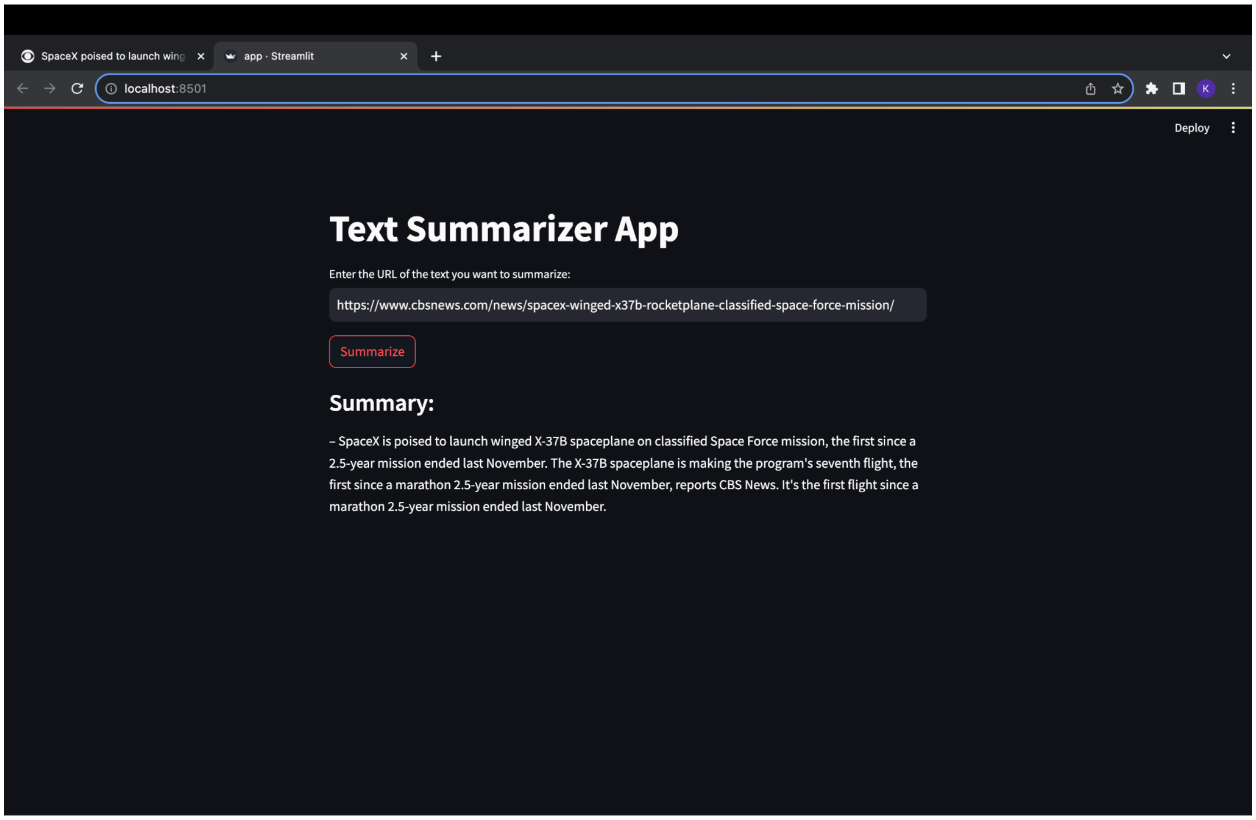
  rouge-l:

  - Recall (r): 0.33
  - Precision (p): 0.48
  - F1-Score (f): 0.39

Explanation: Rouge-l measures the overlap in longest common subsequences, which is more flexible than strict word overlap. The recall of 0.33 means that 33% of the longest common subsequences in the reference summary are present in the generated summary. The precision of 0.48 indicates that 48% of the longest common subsequences in the generated summary are relevant. The F1-score for rouge-l is 0.39.

In summary, these ROUGE scores suggest that while there is some overlap between the generated summary and the reference summary, there is room for improvement, particularly in capturing more of the reference summary's content (higher recall) while maintaining precision. The F1-scores provide a balanced view of the trade-off between precision and recall. Adjustments to the summarization model or fine-tuning may be considered to enhance its performance.

After getting the score updated the app so that user can provide link and the model will return its summary



Text Summarizer App

Enter the URL of the text you want to summarize:

https://www.cbsnews.com/news/spacex-winged-x37b-rocketplane-classified-space-force-mission/

Summary:

– SpaceX is poised to launch winged X-37B spaceplane on classified Space Force mission, the first since a 2.5-year mission ended last November. The X-37B spaceplane is making the program's seventh flight, the first since a marathon 2.5-year mission ended last November, reports CBS News. It's the first flight since a marathon 2.5-year mission ended last November.

Summary and Conclusions:

In this project, the focus was on abstractive summarization, a challenging Natural Language Processing (NLP) task, using the T5 (Text-To-Text Transfer Transformer) model. The T5 model, based on the transformer architecture, was chosen for its effectiveness in various NLP tasks. The goal was to evaluate the T5 model's performance on abstractive summarization using the Multi-News dataset and assess its ability to generate coherent and informative summaries.

The Multi-News dataset, consisting of news articles and professionally written summaries, served as a robust benchmark for evaluating summarization models. The T5 model, with its unique Text-To-Text Transfer approach, demonstrated significant capabilities in tasks like text summarization.

The project involved training the T5 model on the Multi-News dataset and building a streamlit app. The generated summary exhibited some issues, including inaccuracies and inconsistencies compared to the reference summary. The ROUGE scores, which evaluate the quality of the generated summary, indicated room for improvement. The recall, precision, and F1-scores suggested that while there was overlap, adjustments were needed to enhance the model's performance, especially in capturing more content from the reference summary.

Learnings:

The T5 model, known for its success in various NLP tasks, showcased its potential in abstractive summarization. However, the specific results from the project emphasized the importance of fine-tuning and potential model adjustments to achieve more accurate and consistent summaries. The evaluation metrics provided valuable insights into the strengths and weaknesses of the generated summaries.

Suggestions for Improvement:

- **Fine-Tuning Parameters:** Explore different hyperparameter settings for learning rate, batch size, and number of epochs to find optimal values for improved summarization.
- **Data Augmentation:** Increase the diversity of the training data by applying various transformations to the input data, ensuring the model is exposed to a broader range of examples.
- **Model Complexity:** Consider experimenting with simpler model architectures or adjusting the number of parameters to address potential overfitting.
- **Refinement of Text Input:** Preprocess input texts and experiment with different prefixes or techniques to enhance the model's understanding and accuracy in generating summaries.
- **Continuous Evaluation:** Regularly assess model performance on different datasets and articles to identify areas for ongoing improvement.
- **Ensemble Approaches:** Investigate ensemble methods by combining outputs from multiple models to potentially enhance overall summarization quality.
- **Web Scraping and updating the app**: Update the app so that the user will be able to provide the article link and choose how much words summary they need

In conclusion, while the T5 model demonstrated promise in abstractive summarization, continuous refinement and experimentation with model parameters and data augmentation techniques are essential for achieving more accurate and reliable results in real-world applications.

Reference:

https://medium.com/@ajazturki10/text-summarization-with-t5-pytorch-and-pytorch-lightning-b7a319ec9ea2#:~:text=T5%2C%20or%20Text%2Dto%2D,machine%20translation%2C%20and%20sentiment%20classification.

https://pythonsimplified.com/build-a-text-summarization-app-using-streamlit-in-30-minutes/