News Summarization Using T5 transformer

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.

2. Description of your individual work.

Started with some preprocessing steps for the implementation of the LSTM model. Then I have

applied word2vec embedding on the validation and test data.

The seq2seq LSTM model was built but the

3. Describe the portion of the work that you did on the project in detail.

1. **Environment Setup:**

   - Checks if a GPU is available and sets the device accordingly (either GPU or CPU).

   - Loads the "multi_news" dataset using the Hugging Face datasets library.

2. **Model and Tokenizer Initialization:**

   - Initializes a T5 transformer model (`T5ForConditionalGeneration`) and its

corresponding tokenizer (`T5Tokenizer`).

- Moves the model to the specified device (GPU or CPU).

3. **Data Preprocessing:**

- Defines a preprocessing function (`preprocess_function`) to tokenize and preprocess input data for training.

- Converts inputs and labels to PyTorch tensors and moves them to the specified device.

4. **Tokenization and Dataset Mapping:**

- Tokenizes and preprocesses the entire dataset using the `map` function, applying the preprocessing function to each batch of the dataset.

5. **Training Configuration:**

- Sets up training arguments, specifying the output directory, learning rate, batch sizes, number of training epochs, and weight decay.

6. **Trainer Initialization:**

- Initializes a Trainer from Hugging Face's Transformers library, providing the model, training arguments, and training and evaluation datasets.

7. **Training:**

- Trains the model using the `train` method of the Trainer.

8. **Summarization Function:**

   - Defines a function (`summarize`) to generate summaries for input documents using the trained model.

9. **ROUGE Score Calculation:**

   - Defines a function (`calculate_rouge_score`) to calculate ROUGE scores using the Rouge library.

10. **User Interaction and Evaluation:**

    - Prompts the user to input a document and a reference summary for evaluation.
    - Generates a summary using the `summarize` function.
    - Calculates and prints the ROUGE score for the generated summary compared to the reference summary.

In summary, this code demonstrates the end-to-end process of training a T5 model on a summarization dataset, generating summaries, and evaluating the quality of the generated summaries using the ROUGE metric. The trained model is saved in the specified output directory.

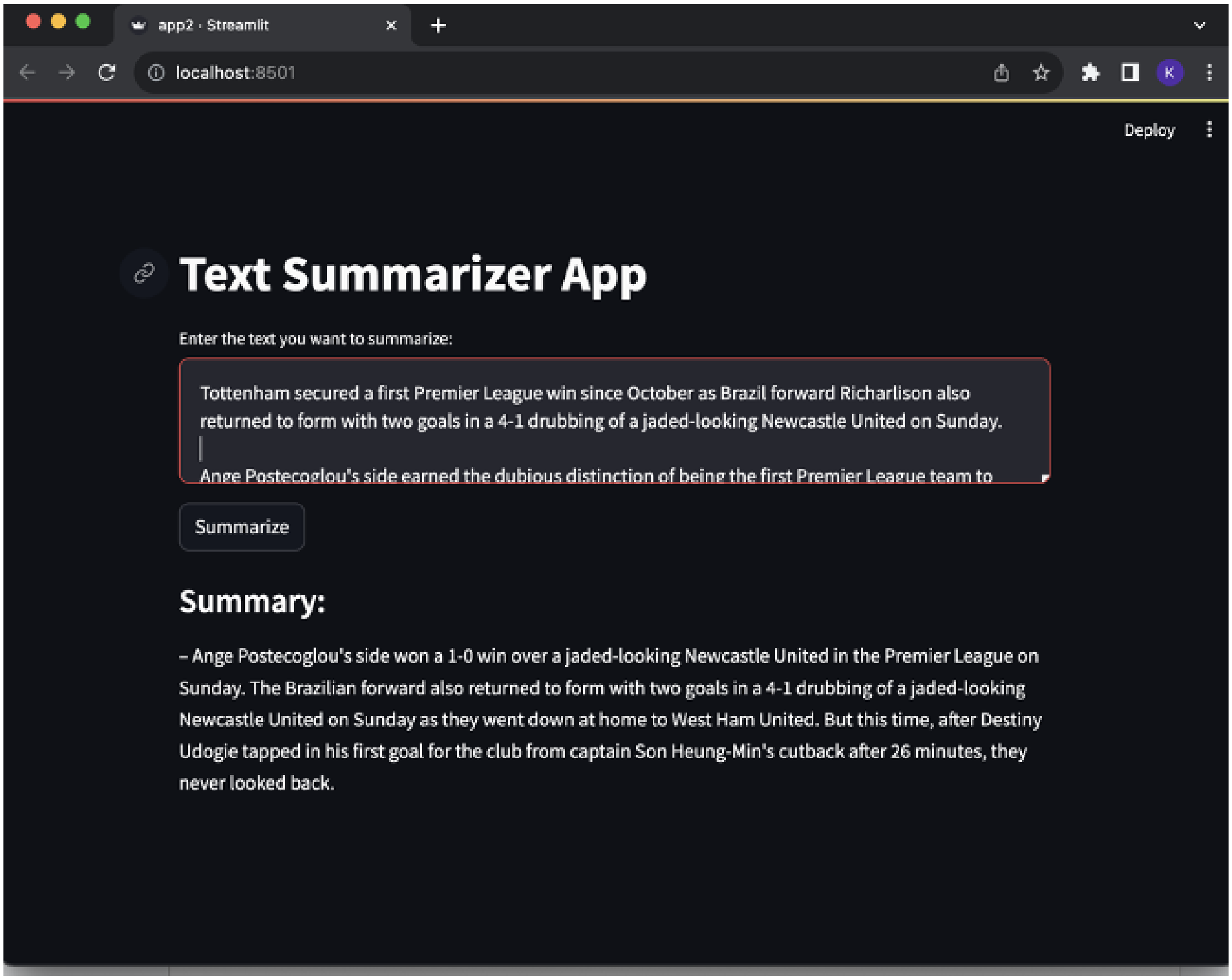
4. Results.

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:



- 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**.**

5. Summary and conclusions.

Learnings:

T5 Model for Summarization: Explored the application of the T5 model, originally designed for text-to-text tasks, in news summarization. Learned how to fine-tune the model for abstractive summarization tasks.

Data Preprocessing: Understood the importance of data preprocessing in preparing the input for a summarization model, including tokenization and conversion to PyTorch tensors.

ROUGE Metric: Learned about the ROUGE metric, a widely used evaluation measure for automatic summarization tasks. Understand how it assesses the quality of generated summaries by comparing them to reference summaries.

Model Evaluation: Explored the process of evaluating summarization models, including user interaction, summary generation, and quantitative assessment using ROUGE scores.

6. Calculate the percentage of the code that you found or copied from the internet.

83 - 10/80+5 = 85%

7. References

https://medium.com/artificialis/t5-for-text-summarization-in-7-lines-of-code-b665c9e40771