Text Summarizer

1st Ram Kumar Ippili
CECS Department
University of Michigan - Dearborn
Michigan, USA
rippili@umich.edu

2nd Sai Revanth Iddum
CECS Department
University of Michigan - Dearborn
Michigan, USA
idvsr@umich.edu

***Abstract*—This project is a demonstration of the development of an abstractive text summarizer system using BART (Bidirectional and Auto Regressive Transformer) Model. The application leverages the CNN/DailyMail dataset to produce concise summaries of long articles. Key aspects include data preprocessing, model fine tuning, and evaluation using ROUGE metrics. The document also contains the challenges encountered during development, like resource constraints, dataset size reduction, model configuration optimization and training strategies. This application is able to support custom input text summarization providing an adaptable summarization tool for practical use. The full implementation of this project is available in Github and Since it might take a lot of time for the project to run this Google Drive has the model but trained with 3% of dataset.**

***Index Terms*—dataset, optimization, model, metrics, input, transformer, data, constraints, tool, evaluation**

I. Introduction

In today's fast-paced world, where huge amounts of textual data is generated daily, the ability to skim through the main content and catch on the potentially new issues is very precious. Text summarization is a key activity in NLP (Natural Language Processing) that caters for such needs. By condensing the large documents into brief synopses, summarization not only saves time for the users but also retains the most vital information. Techniques of summarization that involve exactitude and brevity fall under the first category, while it is the second type that entails rewriting or rewriting the information in question. However, although more complex, abstractive summarization further takes the role of human summarization in restructuring and rewriting of information closer.

This program is based on BART (Bidirectional and Auto-Regressive Transformer), an excellent transformer model used to solve text generation tasks. BART fine-tuned on the CNN/DailyMail dataset can produce comprehensive summary documents that are coherent and story-like. This project tackles both academic and practical applications, such as news article summaries, corporate documents, or research papers.

II. Types of Text Summarization Methods

To understand the scope of this project, it is essential to understand about the techniques used in text summarization:

A. Extractive Summarization

Extractive Summarization involves selecting a few sentences or phrases directly from the source based on their relevance. This method relies on ranking the algorithms and feature-based analysis, such as importance of the sentence, word frequency, and positional information. Although it may retain the factual content, it often lacks the logical flow.

1) Advantages: It is Simple to implement and Retains the factual accurance as it copies sentences from the source itself.

2) Limitations: The output may feel abrupt due to the lack of paraphrasing and cannot be interpreted or restructured for better readability.

3) Example Models:

- TextRank: A graph based ranking algorithm inspired by Google's PageRank.
- BERTSUM: A transformer-based extractive summarizer that classifies sentences for inclusion in summaries.

B. Abstractive Summarization

Abstractive Summarization generates entirely new text by paraphrasing and restructuring the content. This approach requires much deeper semantic understanding and advanced language modeling. Models like BART, Pegasus, and T5 fall under this category.

1) Advantages: This produces fluent, coherent summaries that resemble human writing and is capable of condensing and rephrasing information for clarity.

2) Limitations: This requires significant resource for training and sometimes generate hallucinations i.e., content that is plausible but factually inaccurate.

3) Example Models:

- BART: A transformer model pre-trained as a denoising autoencoder.
- Pegasus: Trailored for summarization with specialized pretraining objectives.
- T5: A versatile text-to-text transformer capable of summarizing text in multiple languages.

This project adopts abstractive summarization because of its potential to generate high-quality, human-like summaries, making it suitable for real-world applications.

III. Related Work

The first attempts to summarize were based on mathematical methods such as Term Frequency-Inverse Document Frequency (TF-IDF) and Singular Value Decomposition (SVD). These methods, though they are basic, did not fully capture the complicated relationship of the sentences and words. The

development of neural networks brought with it huge progress, especially when considering models such as Recurral Neural Networks (RNNs) and Long Short-Term Memory's (LSTMs). But, these models were not able to sustain long sequences and were unable to cover the global context.

Since transformer-based architectures were introduced, great changes have occurred in this area. Models such as BERT and GPT provided unique contextual knowledge, even if they were primarily made for extractive tasks or generative tasks like the dialogue. BART's sequence-to-sequence framework bridged this gap by combining a bidirectional encoder with an autoregressive decoder, making it ideal for summarization tasks.

Pegasus and T5 then targeted the part of abstractive summarization superbly. Pegasus, during pre-training, generates a gap sentence to concentrate on the summarization, whereas T5, redefines NLP tasks as text-to-text transformations. Despite these, BART stands out as a general choice for summarization not only by virtue of its strong pre-training objectives but also its general application.

IV. Dataset and Preprocessing

The CNN/DailyMail dataset is often widely regarded as a benchmark for summarization research. It consists of news articles paired with professionally written highlights. The dataset contains approximately 287k training samples, 13k validation samples and 11k test samples. This diversity ensures that models trained on this can generalize across various domains, such as politics, technology and entertainment.

A. Preprocessing Steps

1) Tokenization:

- Articles are truncated to 1024 tokens to meet BART's input constraints.
- Highlights are also truncated to 128 tokens for conciseness.
- To address computational limitations, only 3% of the training, validation and test datasets were used.
- Preprocessed datasets were saved in a foramt compatible with hugging face's library, enabling seamless integration during training.

V. Model Fine-Tuning

Fine Tuning is very essential to adapt the pre-trained BART Model for this specific task of abstractive summarization. Using Hugging Face's transformer library, the model's trained on preprocessed dataset, allowing it to generate summaries tailored to the CNN/DailyMail dataset.

A. Training Configuration

- Model: facebook/bart-base (a smaller version of bart-large)
- Batch Size: 2 per Device, suitable for Google Colab's GPU Environment.
- Gradient Accumulation: 8 steps, stimulating a larger effective batch size.
- Learning Rate: 5e-5.
- Epochs: 1 (For speed)

B. Challenges

1) Slow Training: Reduced the dataset size and leveraged GPU acceleration on Google Colab.

2) Memory Constraints: Truncated sequences and used gradient accumulation to optimize memory usage.

3) Overfitting: Regularization techniques such as dropout were applied, and training was limited to one epoch.

VI. Results

The model's performance was evaluated using ROUGE metrics. ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a set of metrics widely used in summarization tasks to compare human-written reference with machine generated summaries.

A. Evaluation Scores & Interpretation of Results

- ROUGE - 1: 32.69% →Shows that the model captures a good proportion of important words from the reference summaries.
- ROUGE - 2: 13.79% →Indicates reasonable capture of two-word sequences, suggesting some fluency in generated summaries.
- ROUGE - L: 23.93% →Reflects that the generated summaries maintain logical sentence structure.
- ROUGE - LSUM: 30.36% →Shows the model's strength in aligning with human summarization patterns.

These results demonstrate that the model effectively captures key information while maintaining linguistic fluency. The ROUGE-2 score, which reflects bigram overlap, suggests room for improvement in generating coherent multi-word phrases.

B. Limitations

Scores are slightly lower than benchmarks due to constraints like reduced training data (3%) and limited computational resources.

VII. Discussion & Future Work

While the results are promising, several areas for improvement remain:

1) Dataset Size: Fine-tuning on the full dataset could enhance generalization and performance.
2) Decoding Strategies: Beam search and nucleus sampling could improve summary quality and diversity.
3) Alternative Architecture: Models like Pegasus or T5 could be explored for comparative analysis.
4) Real-World Applications: A web-based tool or API could make the summarization system accessible to end users.

VIII. Custom Text Summarization

The application supports summarizing custom inputs, allowing user to modify the summary length and style through adjustable parameters.

A. Example Input

Artificial intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think like humans and mimic their actions. The term may also be applied to any machine that exhibits traits associated with a human mind such as learning and problem-solving.

B. Example Output

In general, AI is a simulation of the human mind . It may be used to simulate the behavior of a machine that is programmed to act like humans .

IX. Conclusion

This project demonstrates the feasibility of fine-tuning pre-trained transformers for abstractive summarization. Despite computational constraints, the system achieved competitive results and provides a basis for further research and development in text summarization.

References

[1] M. Lewis, Y. Liu, N. Goyal, et al., "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension," in Proceedings of ACL 2020, pp. 7871-7880, 2020.

[2] A. See, P. J. Liu, and C. D. Manning, "Get To The Point: Summarization with Pointer-Generator Networks," in Proceedings of ACL 2017, pp. 1073-1083, 2017.

[3] Hugging Face Transformers Library. Available: https://huggingface.co/transformers

[4] CNN/DailyMail Dataset. Available: https://huggingface.co/datasets/cnn_dailymail