BENCHMARKING BART FOR MULTI-DOCUMENT ABSTRACTIVE SUMMARIZATION ON THE XSUM DATASET

by

RAVNEET KAUR

Artificial Intelligence Intern
Batch-2

Abstract

This paper presents the implementation and evaluation of the BART transformer model for the task of multi-document abstractive summarization on the XSum dataset. The project automates data processing, summary generation, ROUGE evaluation, and visualization, forming the baseline for broader benchmarking of multiple models and datasets.

A Project Report

Submitted to Suvidha Foundation

In Fulfilment of the

Suvidha Foundation- Summer Internship

July 16, 2025

1 Introduction

In an age dominated by the exponential growth of digital content, the demand for intelligent text summarization tools has become more critical than ever. With the continuous proliferation of online articles, journals, and reports, particularly from news sources, it has become virtually impossible for users to consume, understand, and retain all available information manually. This challenge has propelled the development of automatic text summarization systems- a subfield of Natural Language Processing (NLP) focused on generating concise and meaningful summaries from larger bodies of text without losing key information or context.

Traditional summarization techniques have historically relied on extractive methods, which involve selecting and rearranging sentences from the source text to produce a summary. While extractive models can generate grammatically correct and information-rich outputs, they are inherently limited in flexibility, often failing to paraphrase or reword content naturally. To overcome these shortcomings, abstractive summarization approaches have emerged as a promising alternative. Unlike extractive methods, abstractive models aim to comprehend the semantic meaning of a document and generate entirely new sentences that convey the same information more succinctly and naturally.

With the rise of deep learning, especially transformer-based architectures like BERT, GPT, and T5, the field of abstractive summarization has witnessed significant advancements. One such model, BART (Bidirectional and Auto-Regressive Transformers), developed by Facebook AI, has shown notable performance across a range of summarization benchmarks. It combines the capabilities of BERT for encoding and GPT-like models for decoding, making it highly suitable for generative tasks like summarization.

This project focuses on benchmarking the BART model in the specific context of multi-document abstractive summarization. Our primary goal is to evaluate its performance on the XSum dataset- a widely used corpus for extreme summarization- where each summary is a single sentence capturing the essence of a full news article. The project forms a crucial first step toward a larger benchmarking study involving multiple models (PEGASUS, T5, etc.) and datasets (CNN/DailyMail, Newsroom).

By developing automated pipelines for data loading, model inference, ROUGE-based evaluation, and results visualization, this study contributes a reproducible, modular, and extensible framework for summarization research. The findings and resources from this work are not only intended to support academic inquiry but also to inform practical deployments of summarization systems in media, journalism, and enterprise communication.

2 Literature Review

Text summarization has been a long-standing problem in computational linguistics, evolving from rudimentary heuristic-based models to state-of-the-art deep learning architectures. Old approaches in summarization relied on statistical and rule-based systems such as sentence ranking using term frequency-inverse document frequency (TF-IDF), Latent Semantic Analysis (LSA), and graph-based algorithms like TextRank. These methods typically implemented extractive strategies, producing summaries by identifying and assembling the most informative sentences from the input document.

The emergence of deep learning has significantly transformed the summarization landscape. Sequence-to-sequence (Seq2Seq) models introduced a new paradigm for abstractive summarization. The encoder-decoder structure, initially implemented using recurrent neural networks (RNNs) and later enhanced by attention mechanisms, laid the groundwork for truly generative summarization models. However, RNN-based models suffered from vanishing gradients and difficulty capturing long-range dependencies.

The introduction of the Transformer architecture by Vaswani et al. in 2017 revolutionized the field by replacing recurrence with self-attention, leading to parallelized training and superior performance in language understanding tasks. Transformer-based models such as BERT (Devlin et al., 2018) excelled at masked language modeling and bidirectional contextualization but were not inherently generative. Subsequently, encoder-decoder models like T5 (Text-to-Text Transfer Transformer) and BART integrated the strengths of both architectures, enabling effective text generation.

BART (Lewis et al., 2020) stands out as a denoising autoencoder that combines a bidirectional encoder with an autoregressive decoder. Pretrained on a range of noising objectives, BART can be fine-tuned for various downstream tasks, including summarization. It has consistently demonstrated superior performance on benchmarks such as CNN/DailyMail, Gigaword and XSum. The XSum dataset itself represents a shift from traditional summarization corpora. Introduced by Narayan et al. (2018), it contains over 200,000 BBC news articles paired with single-sentence summaries. Unlike CNN/DailyMail, which often requires extracting multiple sentences, XSum demands models to generate highly abstractive and semantically rich summaries. This makes it an ideal testbed for evaluating abstractive summarization capabilities.

While models like PEGASUS leverage pretraining objectives tailored for summarization (e.g., Gap Sentence Generation), and Longformer or BigBird explore document-length scaling, BART provides a strong baseline for comparison. Numerous research papers have explored fine-tuning BART on XSum and analyzed challenges such as factual consistency, hallucination, and coverage. This study contributes to this ongoing literature by creating a hands-on benchmarking setup and implementing automated pipelines to simplify the evaluation process for future researchers.

3 Methodology

The methodology adopted in this study involves several systematic stages, starting from dataset acquisition and preprocessing, through model deployment and inference, to evaluation and result visualization. Each stage has been designed to ensure reproducibility, modularity, and ease of extension for future models and datasets.

3.1 Dataset Selection and Preparation

We selected the XSum dataset for this benchmark. It consists of BBC news articles paired with single-sentence summaries that provide an "extreme summarization" challenge. A subset of 10–20 articles was used, formatted into a CSV with two columns: `document` and `summary`, and placed within a standard `data/xsum/` directory.

3.2 Model Selection – BART

The model used is BART, a pretrained transformer-based encoder-decoder model implemented using HuggingFace Transformers. The `facebook/bart-large-cnn` variant was used. It processes sequences up to 1024 tokens and is well-suited for summarization tasks.

3.3 Inference Pipeline

A Python script ("bart_summarize.py") was written to automate document summarization. Articles are tokenized, processed by the model, and output summaries are saved to a text file. Batch processing and GPU support are optionally included for scalability.

3.4 Evaluation

Evaluation was conducted using the ROUGE metric — a standard for summarization tasks. The ROUGE-1, ROUGE-2, and ROUGE-L scores were calculated by comparing the generated summaries with the gold-standard reference summaries from the dataset. A separate evaluation script reads both outputs and computes the scores, storing the results in an Excel file: ("results_summary.xlsx").

3.5 Visualization

To enhance interpretability, a script ("plot_results.py") was developed to generate a bar chart of the ROUGE scores for each model-dataset pair. The chart is saved in (.png) format and can be embedded in reports and presentations. ("bart_xsum_rouge_scores.png").

3.6 Reproducibility and Structure

All code files were stored under a code/ folder, datasets in data/, and results in results/. A file: README.md explains usage, and requirements.txt lists dependencies. This ensures the entire setup is shareable, reproducible, and extendable by other researchers.

4 Results

The results of the benchmarking experiment focused on summarizing XSum articles using the BART model. A total of 10 articles were used for inference, and their respective summaries were generated and saved for comparison.

The key performance metrics used were ROUGE-1, ROUGE-2, and ROUGE-L, each capturing different levels of n-gram overlap and sequence similarity between the generated and reference summaries:

Metric	**Score**
ROUGE-1	0.1987
ROUGE-2	0.0681
ROUGE-L	0.1582

Table 1: ROUGE scores for BART on XSum

These results are consistent with expectations from prior literature. ROUGE-1 reflects decent unigram overlap, while ROUGE-2 is lower due to the abstractive nature of XSum and the model's attempt to rephrase content rather than reuse exact phrases. ROUGE-L, which measures longest common subsequence, also demonstrates moderate overlap.

The generated summaries were evaluated qualitatively as well. While BART often maintained factual relevance and fluency, some instances of minor hallucination or overgeneralization were observed. However, in most cases, the summaries were contextually appropriate and linguistically sound.

A bar chart was also plotted using matplotlib to visualize the comparative scores. This chart is saved as ("bart_xsum_rouge_scores.png") and provides an at-a-glance view of BART's strengths and weaknesses in this summarization setup.

Qualitative analysis showed that the model often retained the key idea of the article. Despite some minor hallucinations or generalizations, summaries were mostly accurate and fluent.

5 Discussion

The results of this project offer several insights into the performance and behavior of transformer-based models in extreme summarization tasks. BART, while not specifically pretrained for XSum-like objectives, shows promising results on the dataset. The ROUGE-1 and ROUGE-L scores indicate that BART captures the gist of the article effectively, although ROUGE-2 shows that higher-order phrase overlap is limited, which is expected given the one-sentence constraint in XSum.

An important observation is the role of dataset characteristics in influencing model performance. Unlike CNN/DailyMail, which features more extractive summaries, XSum demands true abstraction- i.e., the model must synthesize and rephrase rather than extract. This often leads to challenges such as hallucinations (generating false but plausible facts), under-specification, and lack of coverage. Nevertheless, the BART model handles this reasonably well without fine-tuning.

The choice of model size (facebook/bart-large-cnn) and limited input size (1024 tokens) also impacted performance. Longer articles may get truncated, losing context. For future experiments, models like Longformer or PEGASUS with longer context windows and task-specific pretraining might yield better results.

Moreover, the use of only a small dataset subset limits statistical generalizability. However, the aim of this phase was to validate the full pipeline- from input loading to summary generation to evaluation and reporting. This was successfully achieved, demonstrating the feasibility and reproducibility of the benchmarking framework.

BART, despite not being trained specifically on XSum, performs competitively. The dataset's extreme summarization nature challenges models to synthesize rather than extract. BART's architecture handles this well, but issues like hallucination and coverage gaps remain.

Longer articles getting truncated due to the 1024-token limit affected performance. Future improvements could involve models like PEGASUS or Longformer that support longer contexts or have summarization-specific pretraining.

While the evaluation was on a small subset, the goal was to validate the benchmarking framework, which was successfully accomplished.

6 Conclusion

This project presents a practical, modular benchmarking setup for evaluating abstractive summarization models, specifically demonstrating the application of the BART model on the XSum dataset. From setting up data pipelines to generating summaries, evaluating using ROUGE, and visualizing results, the entire cycle was automated and documented. The findings indicate that BART provides competent summarization performance on a challenging dataset like XSum, even without extensive fine-tuning.

The research lays a solid foundation for extending this work to more datasets (e.g., CNN / DailyMail, Newsroom) and additional models (PEGASUS, T5, Longformer, etc.). Furthermore, the modular codebase and standardized outputs allow for easy plug-and-play testing of new models in the future. All artifacts, including summaries, Excel results, and plots, are saved and organized to facilitate reproducibility and presentation.

As the field of NLP continues to evolve, the importance of benchmarking cannot be overstated. This work not only helps understand current model limitations but also provides infrastructure for tracking improvements and experimenting with novel architectures. In future iterations, incorporating human evaluation, fine-tuning, and model ensembles will further enrich the quality and impact of this research.

This project successfully establishes a working benchmark setup for summarization tasks. It automates data loading, model inference, evaluation, and visualization using BART on the XSum dataset.

The framework is clean, reproducible, and extensible. It allows future researchers to test new models and datasets easily. While BART performed reasonably well, further work with models like T5, PEGASUS, or Longformer can enhance results.

This project represents a foundational step in developing robust, scalable NLP pipelines for real-world applications.
