A Comprehensive Study of Extractive Summarization with Transformer Models

Shuvam Chowdhury
Machine Learning & Data Science
Email: kshuvam63@gmail.com
GitHub: github.com/Theshuvam

Abstract

Text summarization is a core task in natural language processing, focused on generating concise and informative representations of longer text. This paper presents a comparative evaluation of five pre-trained Transformer-based models, including **BERT**, **RoBERTa**, **BART**, **T5**, and **GPT-2**, across extractive summarization tasks. I evaluate these models on two widely-used datasets: **CNN/DailyMail** and **Gigaword**, comparing their performance using various metrics such as ROUGE (Lin, 2004), METEOR (Denkowski and Lavie, 2014), and BERTScore (Zhang et al., 2020). I aim to assess the models' effectiveness in preserving critical information while balancing computational efficiency.

1 Introduction

Automatic text summarization has emerged as a vital task in Natural Language Processing (NLP), offering a solution to the growing need for efficient extraction of critical information from ever-expanding textual data sources. With the proliferation of online articles, research documents, and user-generated content, automatic summarization techniques have become increasingly important in domains such as news aggregation, information retrieval, legal and scientific document processing, and digital content management.

Summarization approaches are broadly divided into two categories: *extractive* and *abstractive*. Extractive summarization involves selecting salient sentences or phrases directly from the source text, preserving the original phrasing, while abstractive summarization aims to paraphrase the content, often generating novel sentences. This research focuses on the extractive approach due to its linguistic fidelity, lower risk of factual errors (El-Kassas et al., 2021), and compatibility with current Transformer-based architectures (Devlin et al., 2019).

The primary objective of this study is to systematically evaluate and compare the performance of five prominent Transformer models—**BERT**, **RoBERTa**, **BART**, **Flan-T5 Small**, and **GPT-2**—in the context of extractive summarization. These models were purposefully selected to represent three key architectural families: encoder-only (**BERT**, **RoBERTa**), encoder-decoder (**BART**, **Flan-T5 Small**), and decoder-only (**GPT-2**). The goal is to assess how architectural choices and pre-training strategies influence the effectiveness of extractive summarization.

To adapt these models for sentence-level extraction tasks, each model was fine-tuned using a supervised classification framework. For encoder-only and decoder-only architectures, the representation of each sentence was derived from the first token's hidden state—`[CLS]` for **BERT**, and token position 0 for **RoBERTa** and **GPT-2**. In contrast, **BART** and **Flan-T5 Small**, being encoder-decoder models, were modified to use only their encoder components. Sentence representations were obtained from the encoder's first token embedding, enabling the use of a classification head without invoking sequence generation.

All models were fine-tuned using a binary cross-entropy loss function with the AdamW optimizer (Loshchilov and Hutter, 2019), a learning rate of 2×10^{-5}, and a batch size of 64. Early stopping with a patience of 3 was employed based on ROUGE-L performance on a held-out validation set. The classification head comprised a single linear layer mapping sentence embeddings to binary labels (summary-worthy or not).

The training and evaluation procedures were conducted on two benchmark datasets: **CNN/DailyMail** and **Gigaword**. Both datasets were preprocessed uniformly to ensure fair comparisons. Each document was segmented into sentences using the NLTK Punkt tokenizer (Kiss and Strunk, 2006). Relevance labels were

generated by computing ROUGE-L F1 scores between each sentence and the reference summary, selecting the top three scoring sentences as ground-truth positives. Sentences were tokenized to a fixed maximum length of 128 tokens using each model's respective tokenizer. Articles lacking valid sentence segmentation were excluded, and the training data was capped at 100,000 samples for both datasets.

Dataset splits were adjusted slightly for compatibility and scale: for **CNN/DailyMail**, the training set included 100,000 articles, with 2,000 for validation and 10,000 for testing. For **Gigaword**, a similar configuration was used, although the test set was drawn from the validation set due to limited official samples.

In addition to quantitative evaluation using ROUGE (Lin, 2004), METEOR (Denkowski and Lavie, 2014), and BERTScore (Zhang et al., 2020) metrics, the study incorporates **qualitative analysis** by generating summaries for sample documents across all five models. This comparative framework enables an assessment of not only the models' statistical performance but also their practical effectiveness and coherence when applied to real-world texts.

By providing both empirical and architectural insights, this work contributes a comprehensive benchmark of Transformer models for extractive summarization and offers guidance on selecting model types based on specific summarization needs and computational constraints.

2 Related Work

Text summarization has been a long-standing challenge in NLP. It can broadly be classified into two types: **extractive** and **abstractive**. Extractive summarization involves selecting sentences directly from the input document, while abstractive summarization involves generating new sentences that convey the original meaning (El-Kassas et al., 2021). In this section, I focus on **extractive summarization**, discussing significant advancements and techniques.

2.1 Models for Extractive Summarization

Several recent models have been proposed to improve extractive summarization, each with its own strengths and approaches. The following five models have been identified as significant in extractive summarization research:

- **BERT (Bidirectional Encoder Representations from Transformers)**: BERT (Devlin et al., 2019) revolutionized NLP with its deep bidirectional encoding, capturing contextual relationships between words. This ability is critical for extractive summarization tasks, where the model must identify relevant sentences based on their semantic importance. BERT has been successfully applied to extractive summarization, providing state-of-the-art performance on many benchmarks (Cohan et al.).

- **RoBERTa (A Robustly Optimized BERT Pretraining Approach)**: RoBERTa (Liu et al., 2019) builds upon BERT by optimizing its pretraining strategy. By removing the next-sentence prediction task and using larger batches, RoBERTa achieves superior performance in extractive summarization, particularly in document-level extraction tasks (Pilault et al., 2018).

- **BART (Bidirectional and Auto-Regressive Transformers)**: BART (Lewis et al., 2019) combines the strengths of both BERT and GPT-2, utilizing a denoising autoencoder for pretraining. It has demonstrated strong performance in abstractive summarization tasks, generating coherent and concise summaries with minimal reliance on direct copying from source documents.

- **FLAN - T5 (Text-to-Text Transfer Transformer)**: T5 (Raffel et al., 2020) treats all NLP tasks as text-to-text problems. Its ability to process long documents and generate meaningful summaries has made it a strong candidate for extractive summarization. Transformer-based models such as T5 demonstrate strong summarization capabilities by generating highly abstractive summaries that closely resemble human-written texts, with reduced reliance on extractive copying mechanisms (Pilault et al., 2018).

- **GPT-2 (Generative Pretrained Transformer 2)**: GPT-2 (Radford et al., 2019), primarily known for generative tasks, has been adapted for extractive summarization by fine-tuning it on document-based data. This adaptation has yielded promising results, with GPT-2 extracting relevant sentences effec-

tively in extractive summarization tasks (Subramanian et al., 2019).

These Transformer-based models excel at understanding the context and selecting the most relevant sentences, making them ideal candidates for extractive summarization tasks.

2.2 Datasets for Extractive Summarization

For evaluating extractive summarization models, two widely-used datasets are:

- **CNN/Daily Mail**: This dataset contains over 300,000 news articles, paired with human-written summaries. It is frequently used to evaluate extractive summarization techniques, with models such as BERT and RoBERTa achieving strong performance on this dataset (Hermann et al., 2015).

- **Gigaword**: Comprising over 6 million news articles, this dataset is ideal for training extractive summarization models. Its broad vocabulary and diverse coverage of topics make it useful for evaluating generalization across various types of documents (Graff and Cieri, 2003).

These datasets provide benchmarks for training and evaluating extractive summarization models, particularly those based on BERT, RoBERTa, and T5.

2.3 Extractive Summarization Techniques

Several techniques have been developed to identify the most relevant sentences for extractive summarization. Common approaches include:

- **TF-IDF (Term Frequency-Inverse Document Frequency)**: TF-IDF is a statistical technique that measures the importance of words within a document. Sentences with high TF-IDF scores are considered important and are selected for inclusion in the summary (Salton and Buckley, 1988).

- **Sentence Embedding and Similarity Measures**: Models like BERT and RoBERTa generate embeddings that capture the semantic meaning of sentences. The most relevant sentences are selected based on their similarity to the overall document (Devlin et al., 2019; Liu et al., 2019).

- **Ranking Algorithms**: TextRank (Mihalcea and Tarau, 2004) is a graph-based ranking model that identifies important sentences by calculating their centrality within a document. Sentences with higher centrality are considered more important for the summary.

- **Pointer Networks**: Pointer networks (Vinyals et al., 2015) use attention mechanisms to focus on specific parts of the document, selecting sentences that are the most relevant to the document's overall meaning.

These techniques are essential for selecting key sentences and ensuring that the generated summary reflects the most important information.

2.4 Evaluation Metrics for Extractive Summarization

The effectiveness of extractive summarization models is typically evaluated using **ROUGE** scores, which measure the overlap of n-grams between the generated summary and the reference summary. Key ROUGE variants include:

- **ROUGE-N**: Measures the overlap of n-grams (e.g., ROUGE-1, ROUGE-2).

- **ROUGE-L**: Measures the longest common subsequence between the generated and reference summaries.

Higher ROUGE scores indicate better alignment with the reference summary, reflecting the accuracy and relevance of the extracted sentences (Lin, 2004).

3 Novelty and Challenges

3.1 Novelty

This study introduces a unified framework for comparing Transformer architectures across encoder-only, encoder-decoder, and decoder-only models for extractive summarization. Models were adapted into a consistent sentence-level classification setup, enabling direct performance comparisons. In addition to quantitative evaluation, qualitative analysis was included to address practical summary issues not captured by metrics like ROUGE.

- Comparative benchmarking of BERT, RoBERTa, BART, Flan-T5 Small, and GPT-2 for extractive summarization.

- Adaptation of encoder-decoder models for purely encoder-based sentence classification.
- Inclusion of qualitative insights on redundancy and coherence, which standard metrics may overlook.

3.2 Challenges

Several challenges arose during model adaptation, training, and evaluation. Adapting decoder-only models like GPT-2 for sentence extraction required custom engineering. The preprocessing pipeline needed precise tokenization, segmentation, and labeling. Resource limitations, particularly GPU memory, influenced training strategies, and evaluation highlighted the shortcomings of existing metrics in detecting redundancy.

- Designing custom preprocessing pipelines for consistent sentence segmentation and labeling.
- Adapting decoder-only models for sentence representation extraction.
- Adjusting training strategies to handle GPU constraints and prevent overfitting.
- Identifying redundancy in summaries, not captured by ROUGE, but important for usability.

4 Methodology

This section outlines the experimental pipeline used to benchmark five Transformer-based models for extractive summarization: BERT, RoBERTa, BART, Flan-T5 Small, and GPT-2. The study covers dataset selection, preprocessing, architectural adaptations, training, and inference. Each model was integrated into a unified classification framework to ensure fairness and consistency. Custom adaptations were made for each Transformer family, with careful decisions around dataset usage, tokenization, and classification design to support robust evaluations.

4.1 Datasets

Two benchmark datasets were used: **CNN/DailyMail** and **Gigaword**, sourced from Hugging Face. Both are widely used in summarization tasks and offer distinct linguistic patterns.

CNN/DailyMail: This dataset contains news articles paired with multi-sentence summaries. A subset of **100,000 training**, **2,000 validation**, and **10,000 test** samples was used. The data was accessed via Hugging Face's `ccdv/cnn_dailymail` release, with preprocessing to ensure sentence-level granularity.

Gigaword: The Gigaword corpus pairs news articles' first sentences with corresponding headlines as summaries. The dataset used **100,000 training** and **2,000 validation** samples, with **10,000 samples reallocated from validation** for testing due to the limited official test set. Data was sourced from Hugging Face's Gigaword release.

Sampling and Scope: Both datasets were capped at 100,000 training samples for consistency across experiments, with malformed entries excluded.

Rationale: Using both datasets allows evaluation across different domains—long-form articles and concise news snippets—ensuring a comprehensive test of model generalization.

4.2 Data Preprocessing

The data preprocessing pipeline involves several key steps to prepare the data for input to the Transformer models.

Sentence Segmentation. Sentence segmentation is performed using the NLTK Punkt tokenizer, which splits the input documents into individual sentences for further processing.

Sentence-Level Labeling. For sentence-level labeling, the ROUGE-L F1 overlap is computed between each sentence and the reference summary. The sentences with the highest F1 overlap scores are selected as relevant to the summary.

Top-k Sentence Selection. The top-k sentences are selected directly based on their highest ROUGE-L F1 overlap scores, ensuring that the most important sentences are included in the final summary.

Tokenization. Tokenization is performed using the respective tokenizers for each model: - **BERT**: BertTokenizer - **RoBERTa**: RobertaTokenizer - **Flan-T5 Small**: T5Tokenizer - **GPT-2**: GPT2Tokenizer - **BART**: BartTokenizer

Padding and truncation are applied for all models to ensure consistent input sizes. The maximum input length is set to 128 tokens per sentence for

all models, ensuring uniformity across different architectures.

4.3 Experimental Setup

Experiments were conducted in a distributed GPU setup on the HPRC environment, utilizing an A100 GPU and 8-core CPU. Models were fine-tuned with the AdamW optimizer (2×10^{-5} learning rate, batch size 64) and early stopping based on ROUGE-L performance.

Hardware and Infrastructure. The experiments were conducted on Google Colab using an NVIDIA T4 GPU and its associated cloud computing environment, providing sufficient computational resources for large-scale training and efficient experimentation.

Training Configuration. Models were fine-tuned with a supervised classification framework using binary cross-entropy loss. Training data was capped at 100,000 samples for CNN/DailyMail and Gigaword, with separate validation and test sets.

Optimization and Hyperparameters. AdamW optimizer with 2×10^{-5} learning rate and batch size 64. Early stopping based on ROUGE-L was applied to prevent overfitting.

Model Architectures. The models (BERT, RoBERTa, Flan-T5 Small, GPT-2, and BART) were adapted for extractive summarization. Encoder-only models (BERT, RoBERTa) used the first token's hidden state, encoder-decoder models (Flan-T5 Small, BART) used the encoder's first token, and GPT-2 used token position 0.

Evaluation Metrics. Models were evaluated using ROUGE-L, METEOR, and BERTScore, combining quantitative performance with qualitative analysis of generated summaries.

4.4 Training

This section presents a detailed overview of the **training procedures** for the extractive summarization task across three distinct Transformer-based architectures: **Encoder-Only**, **Encoder-Decoder**, and **Decoder-Only**. It provides a step-by-step breakdown of the training workflow, model-specific configurations, hardware setup, and training setup used to train the models for extractive summarization tasks using the **CNN/DailyMail** and **Gigaword** datasets.

The following diagrams help illustrate the workflows for each model architecture, and are complemented by an in-depth explanation of how training was carried out.

Training Workflow Overview The workflow for extractive summarization involves tokenizing input documents, segmenting them into sentences, labeling them, and generating summaries using various Transformer-based models. The training procedure varies slightly between the three architectures: **Encoder-Only** (BERT, RoBERTa), **Encoder-Decoder** (Flan-T5 Small, BART), and **Decoder-Only** (GPT-2).

4.4.1 Encoder-Only Architecture (BERT and RoBERTa)

The **Encoder-Only** architecture processes input sentences using a Transformer encoder and produces sentence representations from which sentence importance scores are derived. These scores are then used to select the most important sentences for inclusion in the summary.

- **Input Document**: The input document is tokenized into individual sentences using **NLTK Punkt tokenizer**.
- **Sentence-wise Label Prompts**: For each sentence, a prompt is generated to ask whether it should be included in the summary.
- **Tokenization**:
  - For **BERT**, the `BertTokenizer` is used, adding a `[CLS]` token at the start of the sentence and padding/truncating the sentences to a maximum length of 128 tokens.
  - For **RoBERTa**, the `RobertaTokenizer` is used, with a `<s>` token prepended to each sentence.
- **Encoder Processing**: The sentence representations are passed through the **Transformer encoder** (BERT or RoBERTa), which encodes each sentence into a fixed-size vector.
- **Sentence Representation Extraction**: For **BERT**, the `[CLS]` token is used as the sentence representation. For **RoBERTa**, the `<s>` token is used.
- **Classification**: The sentence representations are passed through a **classification head** (linear layer) to predict binary labels for sentence

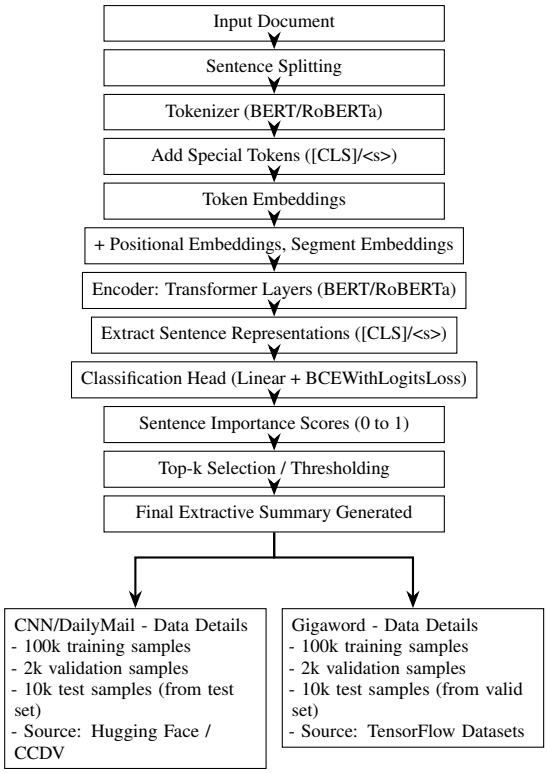

Figure 1: Extractive Summarization Pipeline with BERT/RoBERTa

inclusion/exclusion, based on a **binary cross-entropy loss**.

- **Top-k Sentence Selection**: Sentences with the highest importance scores are selected, and a final extractive summary is formed by concatenating these sentences in their original order.

Training Details

- **Optimizer**:
  - **AdamW** optimizer with a learning rate of 2×10^{-5} and a batch size of 64.
  - **Early stopping** based on **ROUGE-L** performance with a patience of 3 epochs.
- **Loss Function**: Binary cross-entropy loss for sentence inclusion/exclusion classification.

4.4.2 Encoder-Decoder Architecture (Flan-T5 Small and BART)

In the **Encoder-Decoder** architecture, the model consists of both an encoder and a decoder. The encoder encodes the entire input document into a sequence of embeddings, and the decoder generates a sequence of binary labels that decide which sentences are included in the summary.

- **Input Document**: Similar to the encoder-only models, the input document is tokenized using **NLTK Punkt tokenizer**.
- **Sentence-wise Label Prompts**: The same prompts as in the encoder-only workflow are used to decide which sentences should be included in the summary.
- **Tokenization**:
  - For **Flan-T5 Small**, the `T5Tokenizer` is used, with a task-specific prefix (`summarize: `) added to the input sentences.
  - For **BART**, the `BartTokenizer` is used with a similar task-specific prefix.
- **Encoding**: The encoder processes the entire document into embeddings.
- **Decoding**: The decoder generates binary tags (e.g., `1 0 0 1`) corresponding to sentence inclusion/exclusion.
- **Sentence Selection**: Binary tags generated by the decoder are used to select the sentences for inclusion in the summary, which are concatenated to form the final extractive summary.

4.4.3 Decoder-Only Architecture (GPT-2)

The **Decoder-Only** architecture focuses entirely on generating sequences of tokens. In this case, the model generates binary labels for sentence inclusion, and the output is processed to create the final summary.

- **Input Document**: The input document is tokenized using **NLTK Punkt tokenizer**.
- **Sentence-wise Label Prompts**: Similar to the other architectures, the prompt asks whether the sentence should be included in the summary.
- **Tokenization**: The `GPT2Tokenizer` is used to tokenize the input sentences, with padding and truncation applied to ensure the sentences fit within the maximum token limit (128 tokens for CNN, longer for Gigaword).
- **Decoding**: The GPT-2 model generates a sequence of binary labels based on the sentence-wise inputs.

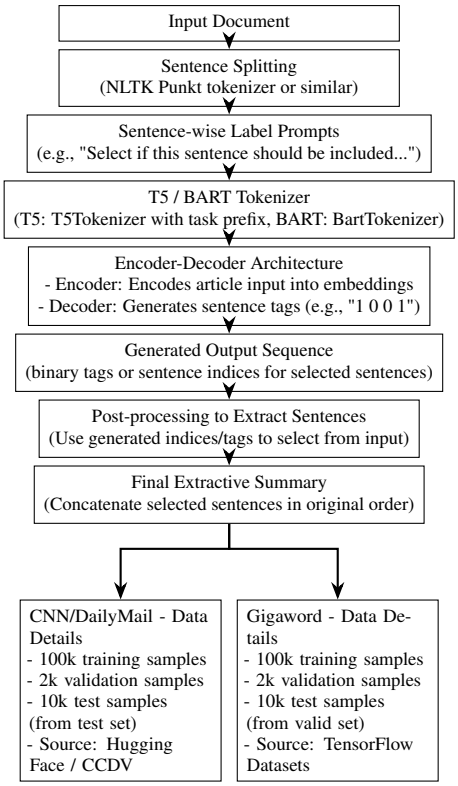

Figure 2: Extractive Summarization Pipeline with T5/BART Encoder-Decoder Architecture

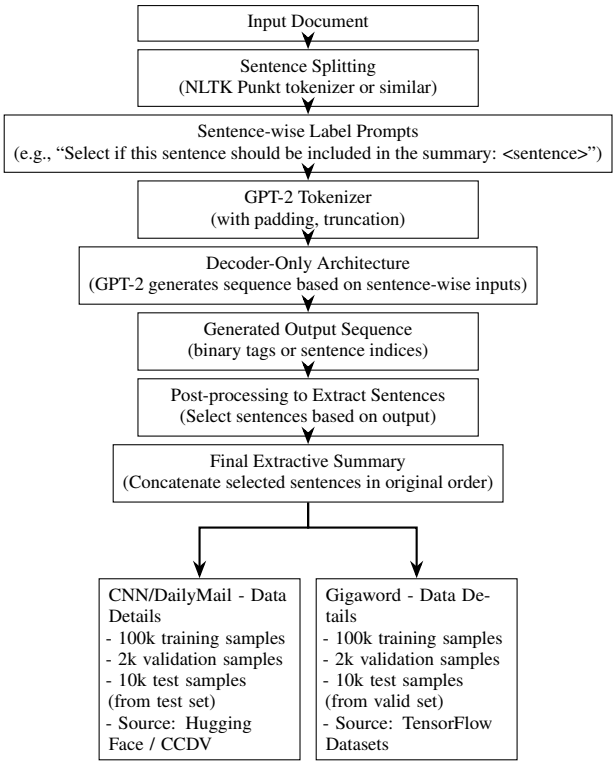

Figure 3: Extractive Summarization Pipeline with GPT-2 Decoder-Only Architecture

- **Post-Processing**: Based on the binary labels, sentences are selected for inclusion in the summary.
- **Final Extractive Summary**: The selected sentences are concatenated to form the final summary.

4.4.4 Training Environment and Hardware Setup

The training was conducted on Google Colab infrastructure, which provided the necessary resources for large-scale model training and data processing. **NVIDIA GPUs** were primarily used for model training, and the setup also optionally supported **Apple MPS** for local development.

- **Checkpointing**: Periodic **checkpointing** was performed during both preprocessing and training phases to ensure the ability to resume training without data loss or interruption.

4.4.5 Dataset Details

- **CNN/DailyMail**:
  - 100,000 training samples, 2,000 validation samples, 10,000 test samples.
  - Source: Hugging Face Datasets.
- **Gigaword**:
  - 100,000 training samples, 2,000 validation samples, 10,000 test samples (from the validation set).
  - Source: TensorFlow Datasets.

4.4.6 Libraries and Frameworks

Several python libraries were employed across the experiments to handle model training, evaluation, and data processing.

4.4.7 Model Configuration and Setup

Each model was configured based on the pretrained transformer weights, with specific settings for tokenization, padding, and maximum input lengths:

4.5 Inference Pipeline

The inference process was designed to evaluate all five Transformer models uniformly. After fine-tuning on sentence-level summarization, each model scored sentences within a document to select the most summary-worthy ones.

Sentence Scoring. Documents were split into sentences using the NLTK Punkt tokenizer. Each

Category	Library	Description
Deep Learning	torch	Core library for neural networks.
Transformer Models	transformers	Hugging Face library for pretrained models and tokenizers.
Datasets	datasets	For loading benchmark datasets (CNN/DailyMail, Gigaword).
Evaluation	rouge_score, nltk, bert_score	ROUGE, METEOR, and BERTScore calculation.
Sentence Splitting	nltk.PunktSentence Tokenizer	Tokenizes articles into sentences.
Utilities	os, pickle, concurrent.futures	Parallel processing, checkpointing, file handling.
Data Splitting	scikit-learn	Splitting datasets for training/validation/test.
Notebook Support	ipywidgets	Interactive widgets in Jupyter/Colab.

Table 1: Libraries and frameworks used in the experiments.

Model	Pretrained Name	Pad Token	Max Input Length	Parallelization
BERT	bert-base-uncased	[PAD] (default)	128 tokens	CUDA with DataParallel
RoBERTa	roberta-base	eos_token	128 tokens	CUDA with DataParallel
BART	facebook/bart-base	eos_token	128 tokens	Encoder used only for extractive head
Flan-T5 Small	t5-small	eos_token	128 tokens	Encoder used only for extractive classification
GPT-2	gpt2	eos_token (manual)	128 tokens (CNN), longer for Gigaword	Carefully padded/truncated

Table 2: Model configuration details

sentence was tokenized and processed by the fine-tuned Transformer model, which produced a score (0-1) representing the probability of the sentence being included in the summary.

Selection Strategy. Two strategies were used: *top-k* and *threshold-based* selection. In the top-k strategy, the highest-scoring k sentences were selected, typically k = 3. In the threshold-based strategy, sentences with scores above a predefined threshold (e.g., 0.5) were selected. The top-k strategy was primarily used for final evaluation.

Summary Construction. The selected sentences were concatenated in their original order to preserve the document's narrative flow and coherence.

Post-processing. No automatic post-processing was applied, but occasional redundancy in longer documents was noted, indicating potential areas for future improvement in summary conciseness.

5 Evaluation & Results

5.1 Evaluation Metrics

I evaluate extractive summaries using three metrics: **ROUGE**, **METEOR**, and **BERTScore**, assessing lexical overlap, linguistic variation, and semantic similarity by comparing extracted sentences to human-written references.

ROUGE ROUGE measures token-level overlap, including **ROUGE-1**, **ROUGE-2**, and **ROUGE-L** for unigram, bigram, and longest common subsequence matches. Scores are reported for precision, recall, and F1, making it effective for surface-level similarity in extractive summarization.

METEOR METEOR enhances ROUGE by considering synonyms and morphological variants, using stemming, WordNet lookup, and a fragmentation penalty to balance precision and recall, offering a more linguistically informed evaluation.

BERTScore BERTScore evaluates semantic similarity using cosine similarity between contextual embeddings from a pre-trained model (e.g., `roberta-large`), providing precision, recall, and F1 scores based on contextual relevance rather than lexical overlap.

All metrics are computed using their official Python implementations, ensuring reliable comparison of the five Transformer models in this study.

6 Results

I evaluated the performance of five Transformer-based models—**BERT**, **RoBERTa**, **BART**, **Flan-T5 Small**, and **GPT-2**—on the **CNN/DailyMail** and **Gigaword** datasets using ROUGE-1, ROUGE-2, ROUGE-L, METEOR, and BERTScore (Precision, Recall, F1). The results are summarized in Tables 3 and 4.

6.1 CNN/DailyMail Results

On the CNN/DailyMail dataset, **RoBERTa** and **BART** appeared to perform well, showing the highest scores in ROUGE-2 (0.1487 and 0.1502, respectively) and BERTScore F1 (0.8651 and 0.8645). **BERT** also demonstrated competitive performance, with comparable ROUGE-L (0.2290) and BERTScore (0.8643). **GPT-2**, despite being a decoder-only model, showed a reasonable BERTScore F1 of 0.8562, but its ROUGE scores, particularly ROUGE-2 (0.1158), were lower, which might suggest less lexical alignment with reference summaries. **Flan-T5 Small** did not perform as well in comparison, especially in terms of ROUGE and METEOR metrics, which could imply that its encoder-only adaptation may not be ideally suited for extractive summarization in this case.

6.2 Gigaword Results

On the Gigaword dataset, the performance across **BERT**, **RoBERTa**, **BART**, and **Flan-T5 Small** was fairly consistent across all metrics. All four models achieved a ROUGE-L score of 0.2605 and a BERTScore F1 of approximately 0.8701, suggesting that these models might generalize well for headline-style summarization tasks. Interestingly, **Flan-T5 Small** showed some improvement compared to its performance on CNN/DailyMail, performing closer to the top models. **GPT-2**, while still showing a competitive BERTScore F1 (0.8664), had slightly lower ROUGE-1 and ROUGE-L scores, which might indicate some differences in lexical form, although the semantic content appeared to be close to the reference summaries.

6.3 Comparative Analysis

Across both datasets, encoder-only and encoder-decoder models generally seemed to perform better in ROUGE and METEOR metrics, which are sensitive to surface-level similarities. However, **GPT-2** demonstrated relatively strong BERTScore performance, suggesting that it might capture semantic relevance well, even if it had some trade-offs in lexical precision. **RoBERTa** and **BART** tended to perform well across most metrics, which may point to the benefits of their pretraining strategies and bidirectional encoding for extractive summarization. These results suggest that models that leverage encoder representations might be better suited for capturing sentence-level importance, which is crucial for extractive summaries.

Model	R-1	R-2	R-L	MET.	B-P	B-R	B-F1
BERT	0.3645	0.1497	0.2290	0.3047	0.8582	0.8706	0.8643
RoBERTa	0.3653	0.1487	0.2298	0.3022	0.8608	0.8696	0.8651
BART	0.3653	0.1502	0.2287	0.3085	0.8586	0.8708	0.8645
FLAN-T5	0.3059	0.1032	0.1876	0.2389	0.8474	0.8566	0.8518
GPT-2	0.3249	0.1158	0.1969	0.2476	0.8535	0.8590	0.8562

Table 3: Evaluation of extractive summarization models on the CNN/DailyMail dataset.

Model	R-1	R-2	R-L	MET.	B-P	B-R	B-F1
BERT	0.2995	0.1020	0.2605	0.4159	0.8358	0.9078	0.8701
RoBERTa	0.2995	0.1020	0.2605	0.4159	0.8358	0.9078	0.8701
BART	0.2995	0.1020	0.2605	0.4159	0.8358	0.9078	0.8701
FLAN-T5	0.2995	0.1020	0.2605	0.4160	0.8358	0.9078	0.8701
GPT-2	0.2904	0.1029	0.2520	0.3926	0.8478	0.9011	0.8664

Table 4: Evaluation of extractive summarization models on the Gigaword dataset.

7 Conclusion

This study compared five Transformer-based models—**BERT**, **RoBERTa**, **BART**, **Flan-T5 Small**, and **GPT-2**—for extractive summarization on the **CNN/DailyMail** and **Gigaword** datasets. **RoBERTa** and **BART** performed the best, excelling in both lexical overlap and semantic relevance. **GPT-2** showed strong semantic performance but struggled with lexical alignment, while **Flan-T5 Small** performed less effectively across most metrics.

The results underscore the importance of model architecture, with encoder-based models generally delivering better extractive summaries. Despite good results, challenges like redundancy removal and sentence ordering still need attention. Future work will explore efficient fine-tuning methods, multi-document summarization, and potential integration of generative components to improve summary coherence.

8 Future Work

Future work should focus on improving **redundancy elimination** and **sentence ordering** within extractive summarization pipelines, which remain open challenges for Transformer-based models. I also aim to investigate more computationally efficient fine-tuning techniques for large-scale models such as **GPT-2** and **Flan-T5 Small**, including parameter-efficient training and model distillation. Moreover, expanding the current framework to support **multi-document extractive summarization** and exploring the **integration of generative components** may yield more coherent and informative summaries, bridging the gap between extractive and abstractive paradigms.

References

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Thien Huu Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Wafaa S El-Kassas, Cherif R Salama, Ahmed A Rafea, and Hoda K Mohamed. 2021. Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, 165:113679.

David Graff and Christopher Cieri. 2003. English gigaword. Technical report, Linguistic Data Consortium.

Karl Moritz Hermann, Tomás Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of EMNLP*, pages 3651–3661.

Tibor Kiss and Jan Strunk. 2006. Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4):485–525.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Workshop on Text Summarization Branches Out (WAS 2004)*, pages 74–81.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of EMNLP*, pages 404–411.

Jonathan Pilault, Leo Li, Chris Pal, and Sandeep Subramanian. 2018. Extractive summarization with swapnets: Sentences and words from alternating pointer networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7584–7598.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Gerard Salton and Chris Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523.

Sandeep Subramanian, Raymond Li, Jonathan Pilault, and Christopher Pal. 2019. On extractive and abstractive neural document summarization with transformer language models. *arXiv preprint arXiv:1909.03186*.

Oriol Vinyals, Miguel Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, volume 28, pages 2692–2700.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations (ICLR)*.

Appendix

Content Text
Original Article (CNN) Nine British citizens were arrested in Turkey on Wednesday, suspected of trying to cross illegally into Syria, the Turkish military said on its website. The group included four children – the oldest being 10 or 11, with the youngest born in 2013, a Turkish official told CNN on condition of anonymity. The nine were arrested at the Turkey-Syria border, the Turkish military said. It didn't say why the group allegedly was trying to get into Syria, which has been torn by a roughly four-year war between Syrian government forces and Islamist extremist groups and other rebels. Among the war's combatants is ISIS, which has taken over parts of Syria and Iraq for what it claims is its Islamic caliphate, and which is known to have been recruiting Westerners. Accompanying the children were three men and two women; all nine had British passports, the Turkish official said. The British Foreign Office said Wednesday that it is aware of reports of the arrests and that it is seeking information about the incident from Turkish authorities. CNN's Gul Tuysuz reported from Istanbul, and Elaine Ly reported from London. CNN's Jason Hanna contributed to this report.
Reference Summary The group included four children, Turkish official says. Turkish military didn't say what group's intent was. UK Foreign Office says it is trying to get information from Turkish officials.
BERT Summary (CNN) Nine British citizens were arrested in Turkey on Wednesday, suspected of trying to cross illegally into Syria, the Turkish military said on its website. The British Foreign Office said Wednesday that it is aware of reports of the arrests and that it is seeking information about the incident from Turkish authorities. The nine were arrested at the Turkey-Syria border, the Turkish military said.
RoBERTa Summary (CNN) Nine British citizens were arrested in Turkey on Wednesday, suspected of trying to cross illegally into Syria, the Turkish military said on its website. The British Foreign Office said Wednesday that it is aware of reports of the arrests and that it is seeking information about the incident from Turkish authorities. The group included four children – the oldest being 10 or 11, with the youngest born in 2013, a Turkish official told CNN on condition of anonymity.
BART Summary (CNN) Nine British citizens were arrested in Turkey on Wednesday, suspected of trying to cross illegally into Syria, the Turkish military said on its website. The nine were arrested at the Turkey-Syria border, the Turkish military said. The group included four children – the oldest being 10 or 11, with the youngest born in 2013, a Turkish official told CNN on condition of anonymity.
FLAN-T5 Summary (CNN) Nine British citizens were arrested in Turkey on Wednesday, suspected of trying to cross illegally into Syria, the Turkish military said on its website. Among the war's combatants is ISIS, which has taken over parts of Syria and Iraq for what it claims is its Islamic caliphate, and which is known to have been recruiting Westerners. The group included four children – the oldest being 10 or 11, with the youngest born in 2013, a Turkish official told CNN on condition of anonymity.
GPT-2 Summary (CNN) Nine British citizens were arrested in Turkey on Wednesday, suspected of trying to cross illegally into Syria, the Turkish military said on its website. The group included four children – the oldest being 10 or 11, with the youngest born in 2013, a Turkish official told CNN on condition of anonymity. The nine were arrested at the Turkey-Syria border, the Turkish military said.

Table 5: Qualitative comparison of predicted summaries from different Transformer models on a CNN/DailyMail article.

Content Text	
Original Document	a french crocodile farm said thursday it had stepped up efforts to breed one of the world 's most endangered species, the indian UNK, with the hope of ultimately returning animals to their habitat in south asia.
Reference Summary	french farm offers hope for endangered asian crocs UNK picture
BERT Summary	a french crocodile farm said thursday it had stepped up efforts to breed one of the world 's most endangered species, the indian UNK, with the hope of ultimately returning animals to their habitat in south asia.
RoBERTa Summary	a french crocodile farm said thursday it had stepped up efforts to breed one of the world 's most endangered species, the indian UNK, with the hope of ultimately returning animals to their habitat in south asia.
BART Summary	a french crocodile farm said thursday it had stepped up efforts to breed one of the world 's most endangered species, the indian UNK, with the hope of ultimately returning animals to their habitat in south asia.
FLAN-T5 Summary	a french crocodile farm said thursday it had stepped up efforts to breed one of the world 's most endangered species, the indian UNK, with the hope of ultimately returning animals to their habitat in south asia.
GPT-2 Summary	a french crocodile farm said thursday it had stepped up efforts to breed one of the world 's most endangered species, the indian UNK, with the hope of ultimately returning animals to their habitat in south asia.

Table 6: Qualitative comparison of model outputs on a CNN/DailyMail article with an abstractive headline-style reference summary. All models reproduced the full input sentence verbatim.