**Text Summarization**

Submitted by

[Abdalrahman Ashraf 202202066](mailto:s-abdalrahman.alasil@zewailcity.edu.eg)

[Omar Yasser 202201589](mailto:s-omar_yasser@zewailcity.edu.eg)

Date

May 10, 2025



School of Computational Sciences and Artificial Intelligence

**University of Science and Technology**

**Zewail City of Science, Technology, and Innovation**

[**1.Introduction 2**](#_hxax45o1zvq2)

[**2.Related works 2**](#_f3ea4u229s5d)

[**3. Methodology 2**](#_20jkqqpqo62u)

[3.1 Data Preprocessing 2](#_6jh3bmek0c7a)

[3.2 Model Training 2](#_7mmksqqrono)

[3.3 Evaluation 3](#_shho5inco334)

[3.4 API Deployment 3](#_ok8jtq3kr1oa)

[**4. Results 3**](#_acvcnhils742)

[**5. Comparison to Previous Work 3**](#_abjoj28k7a5w)

[**6. Successes 4**](#_bybmfcvqvvuh)

[**7. Challenges 4**](#_qfgaguj5hnw7)

[References 5](#_muyv0nk77bzc)

# 1.Introduction

Abstractive dialogue summarization generates concise, paraphrased summaries of conversational text, a key NLP task for applications like chat logs and meeting notes. This project fine-tuned a BART model on the SAMSum Corpus [1], implemented in PyTorch, and deployed it via a Flask API. The goals were to preprocess data, train and evaluate the model, and provide an accessible interface. This report outlines the methodology, results, comparisons to prior work, main ideas, successes, and challenges, addressing a discrepancy between validation and reported test set scores.

# 2.Related works

Recent efforts in abstractive dialogue summarization have led to the development of datasets tailored for this task. Gliwa *et al.* [1] introduced the **SAMSum Corpus**, a human-annotated dataset containing messenger-like dialogues paired with summaries. This dataset has become a standard benchmark for evaluating summarization models in conversational contexts.

# 3. Methodology

The project followed a pipeline of data preprocessing, model training, evaluation, and API deployment.

## 3.1 Data Preprocessing

The SAMSum Corpus, with 14,264 training, 818 validation, and 819 test dialogue-summary pairs, was used. Dialogues were tokenized using the BART tokenizer (BartTokenizer) with PyTorch tensors (return\_tensors="pt") .The dataset was saved to disk for efficiency.

## 3.2 Model Training

A BART-base model (facebook/bart-base), with 12 transformer layers and approximately 140 million parameters, was fine-tuned using the Hugging Face Trainer API in PyTorch. Training occurred on a Kaggle T4 GPU over 3 epochs, with a batch size of 1, 4-beam search, and early stopping. Training loss decreased from 0.78 to 0.60, and validation loss from 0.70 to 0.67. The model was saved to be used in api testing .

## 

## 3.3 Evaluation

Performance was evaluated on the validation set using ROUGE scores (ROUGE-1, ROUGE-2, ROUGE-L, ROUGE-Lsum) via the evaluate library. The evaluate\_in\_chunks function processed batches of 1 to manage GPU memory, using torch.no\_grad().

## 3.4 API Deployment

A Flask API with a /summarize endpoint accepted POST requests, tokenized dialogue, generated summaries with BART, and returned JSON output. Public access via ngrok failed due to authentication issues, limiting use to local testing.

# 4. Results

The fine-tuned BART model achieved the following ROUGE scores on the validation set:

**Table 1: ROUGE Scores on the SAMSum Validation Set**

| **Metric** | **ROUGE-1** | **ROUGE-2** | **ROUGE-L** | **ROUGE-Lsum** |
| --- | --- | --- | --- | --- |
| Score (%) | **50.20** | **26.14** | **41.80** | **41.79** |

Training and validation losses over 3 epochs were:

| **Epoch** | **Training Loss** | **Validation Loss** |
| --- | --- | --- |
| **1** | **0.78** | **0.70** |
| **2** | **0.73** | **0.68** |
| **3** | **0.60** | **0.67** |

# 5. Comparison to Previous Work

The SAMSum Corpus benchmarks dialogue summarization, and our results are compared to prior work:

* **Baseline Models [1]** : They reported ROUGE-1 scores of 38-41 for extractive models. Our testing score of 50.20 exceeds this, validating BART’s abstractive strength.
* **Deployment Focus**: Unlike most research, our Flask API targets practical use .

The test scores proves that our model has overbecome the baseline models

# 6. Successes

Key successes included:

* **Competitive Validation Scores**: ROUGE-1 of 50.2 matches state-of-the-art, reflecting effective training.
* **Loss Improvement**: Training and validation loss decreased over 3 epochs, indicating model learning.
* **API Functionality**: The Flask API processed dialogues locally, demonstrating deployment potential.

# 7. Challenges

Challenges included:

* **Limited Epochs**: Only 3 epochs may have constrained performance, as loss was still decreasing.
* **GPU Memory Constraints**: Batch size of 1 slowed training and evaluation..

# 

# References

[1] B. Gliwa *et al*., “SAMSum Corpus: A human-annotated dialogue dataset for abstractive summarization,” in *Proc. 2nd Workshop on New Frontiers in Summarization*, 2019.