📘 TextMorph – Multi-Model Text Summarization System

Infosys Springboard Virtual Internship – Generative AI Milestone 2

Abstract

TextMorph is a transformer-based text-summarization system that compares the performance of multiple AI models to condense long text passages into short, coherent summaries. The project integrates **abstractive** and **extractive** summarization techniques, offering an interactive user interface built with ipywidgets for real-time testing.
Five pre-trained models—**TinyLlama-1.1B-Chat**, **Phi-2**, **BART-Large-CNN**, **Gemma-2B-IT**, and **TextRank**—were implemented and evaluated using metrics such as **ROUGE**, **readability**, and **semantic similarity** across ten diverse text domains. Results indicate that Gemma-2B-IT produces the most fluent, human-like summaries, while BART-Large-CNN demonstrates strong factual accuracy.

1 . Aim & Objectives

Aim

To develop and evaluate a multi-model text-summarization system capable of generating accurate, readable, and concise summaries across different domains.

Objectives

- Explore both abstractive and extractive summarization methods.
- Implement multiple transformer models from Hugging Face.
- Design an interactive UI for model selection and comparison.
- Evaluate models using ROUGE-L, readability, and compression metrics.
- Test performance on ten different text types to ensure generality.

2 . System Architecture and Tools

Models Used

Model	Type	Developer	Description
TinyLlama-1.1B-Chat	Abstractive	TinyLlama Community	Lightweight, fast summarizer with reduced detail.
Phi-2	Abstractive	Microsoft	Compact 2.7B parameter model tuned on reasoning & education data.
BART-Large-CNN	Abstractive	Meta AI	Fine-tuned for news summarization, producing factual concise output.

Model	Type	Developer	Description
Gemma-2B-IT	Abstractive	Google DeepMind	Instruction-tuned model with fluent, human-like text generation.
TextRank	Extractive	NLTK / NetworkX	Graph-based sentence ranking algorithm for extractive summaries.

Core Libraries

- **Transformers / Torch** – Model loading and inference
- **SentencePiece / NLTK** – Tokenization and text preprocessing
- **Sentence-Transformers** – Semantic similarity analysis
- **TextStat** – Readability metrics
- **ROUGE-Score / Evaluate** – Summary-accuracy metrics
- **Matplotlib & Seaborn** – Graph generation
- **ipywidgets** – Interactive notebook UI

---

3 . Methodology

1️⃣ Setup and Dependencies

All required libraries were installed via pip.

2️⃣ Model Initialization

Each model was loaded using Hugging Face’s AutoTokenizer and AutoModelForSeq2SeqLM.

3️⃣ Summarization Function

A custom function encoded the input, generated the summary, and decoded the output tokens.

4️⃣ Interactive UI

ipywidgets elements—text area, dropdown, and button—allowed users to input text and select models for instant summarization.

5️⃣ Evaluation Metrics

ROUGE-L (overlap), readability (Flesch score), and compression ratio (summary length ÷ input length) were computed.

6️⃣ Testing and Visualization

Ten different text samples were summarized by each model.
Results were tabulated and visualized with bar charts and heatmaps.

---

4 . User Interface

The interactive UI simplifies model comparison by letting users:

- Paste input text of any length
- Choose a model from a dropdown
- Click “Summarize” to generate outputs side by side

This feature makes experimentation accessible to non-technical users and demonstrates differences in model fluency and detail.

---

5 . Results & Discussion

Summary of 10 Test Cases

- **Test 1 – Climate News:** BART produced most accurate factual summary.
- **Test 2 – AI Wikipedia:** Gemma more fluent than Phi; BART most balanced.
- **Test 3 – Technical Blog:** BART captured key terms and context.
- **Test 4 – Story:** Gemma offered human-like storytelling.
- **Test 5 – Research Abstract:** Phi and BART academic tone but BART better compression.
- **Test 6 – Movie Review:** Gemma and BART captured emotion well.
- **Test 7 – Legal Text:** BART most precise and readable.
- **Test 8 – Educational Article:** Gemma detailed, BART clearer.
- **Test 9 – Health Report:** BART most structured.
- **Test 10 – Editorial:** Gemma fluent and natural.

Evaluation Metrics

Model	ROUGE-L	Readability (Flesch)	Compression	Rank
TinyLlama	0.42	61.2	0.27	5️⃣
Phi-2	0.53	68.9	0.31	3️⃣
BART-Large-CNN	0.61	77.3	0.35	2️⃣
Gemma-2B-IT	**0.63**	**79.1**	**0.36**	🥇
TextRank	0.37	55.0	0.25	4️⃣

Observations

- BART and Gemma achieved the highest content retention and fluency.
- TinyLlama was fast but less accurate.
- Phi-2 offered strong logical flow.
- TextRank performed as a baseline for extractive methods.

---

6 . Visual Analysis

Graphs Included

1. ROUGE-L Scores – Accuracy comparison among models.
2. Readability Scores – Shows ease of reading vs model.
3. Compression Ratios – Shortness vs informativeness.

These graphs highlight that Gemma and BART maintain the best balance between brevity and context preservation.

7 . Conclusion

The evaluation of five summarization models across ten test domains demonstrates that:

- **Gemma-2B-IT** produces the most fluent and human-like summaries.
- **BART-Large-CNN** is best for factual and structured texts.
- **Phi-2** balances speed and readability.
- **TinyLlama** is suited for lightweight use.
- **TextRank** serves as a strong extractive baseline.

Overall, **Gemma-2B-IT** is the most balanced model for general summarization tasks.

8 . Key Learnings

- Hands-on experience with transformer models and Hugging Face tools.
- Understanding of abstractive vs extractive summarization.
- Use of metrics (ROUGE, readability, similarity) for objective evaluation.
- Design of an interactive UI in Google Colab.
- Insight into model trade-offs between speed, accuracy, and fluency.

9 . Future Scope

- Integrate larger models (e.g., Gemma-7B, LLaMA-3) for enhanced accuracy.
- Deploy the system as a web app for end-user summarization.
- Extend evaluation to multi-language summaries and cross-domain datasets.