TEXTMORPH — MULTI-MODEL TEXT SUMMARIZATION

Milestone 2 Project Report

Abstract

- This project focuses on developing a text summarization system — *TextMorph* — which uses multiple transformer-based models to generate and compare summaries of different domain texts.
- Models like **TinyLlama**, **Phi-2**, **Bart**, and **Gemma-2B** are evaluated using metrics such as **ROUGE**, **semantic similarity**, and **readability**.
- The system provides both an interactive **UI** and detailed performance analysis. Results show that **Phi-2** consistently outperforms others in semantic coherence and speed, while **Bart** provides more fluent abstractive summaries.

Aim

- To build and evaluate a multi-model text summarization system that compares outputs from various NLP transformer models using quantitative and qualitative metrics.

Objectives

- Implement **extractive** and **abstractive** summarization techniques.
- Integrate **multiple transformer models** (TinyLlama, Phi-2, Bart, Gemma).
- Compare models using:
    - **ROUGE score**
    - **Semantic similarity**
    - **Readability scores** (Flesch, Gunning Fog)
    - **Processing time**
- Design an **interactive UI** for user input and result visualization.
- Analyze performance across **10 different text domains**.

System Description

a. Dataset / Inputs

User-provided texts from **10 different domains** such as:

1. Biography
2. Science
3. Technology
4. Sports
5. Education
6. Business
7. Medicine

8. Environment
9. History
10. Literature

b. Models Used

- **TinyLlama-1.1B-Chat:** Lightweight, efficient summarizer.
- **Microsoft Phi-2:** Strong semantic understanding and coherence.
- **Facebook Bart-Large-CNN:** Abstractive summarizer with fluency.
- **Google Gemma-2B-it:** High-end transformer, instruction-tuned.

c. Evaluation Metrics

Metric	Description
ROUGE (1, 2, L)	Measures overlap between generated and reference summaries.
Semantic Similarity	Cosine similarity between sentence embeddings.
Readability Scores	Flesch & Gunning Fog index to assess clarity.
Processing Time	Time taken by each model to summarize.

Methodology (Step-by-Step Explanation)

- **Step 1: Install & Import Libraries**

Installed NLP and ML dependencies using pip with --quiet mode to suppress output**.**

- **Step 2: Load Models**

Used Hugging Face Transformers to load 4 pre-trained summarization models (TinyLlama, Phi-2, Bart, Gemma).

- **Step 3: Define Summarization Functions**

Each model had its own summarization function using the pipeline() API.

- **Step 4: Define Metrics Calculation**

Used:

- rouge_score for overlap metrics
- sentence-transformers for semantic similarity
- textstat for readability metrics

- **Step 5: Create Interactive UI**

Built two UIs:

- UI-A: For single-text summarization and model comparison.
- UI-B: For batch comparison and visualization (bar charts, radar charts).

- **Step 6: Run Tests on 10 Domain Texts**

Each model summarized texts from 10 different areas; results stored in a DataFrame for comparison.

- **Step 7: Visualization**

Used Matplotlib to generate:

- Bar charts (ROUGE & Semantic)
- Radar charts (multi-metric comparison)

Results:

All outputs for all 10 input texts are stored in this pdf :

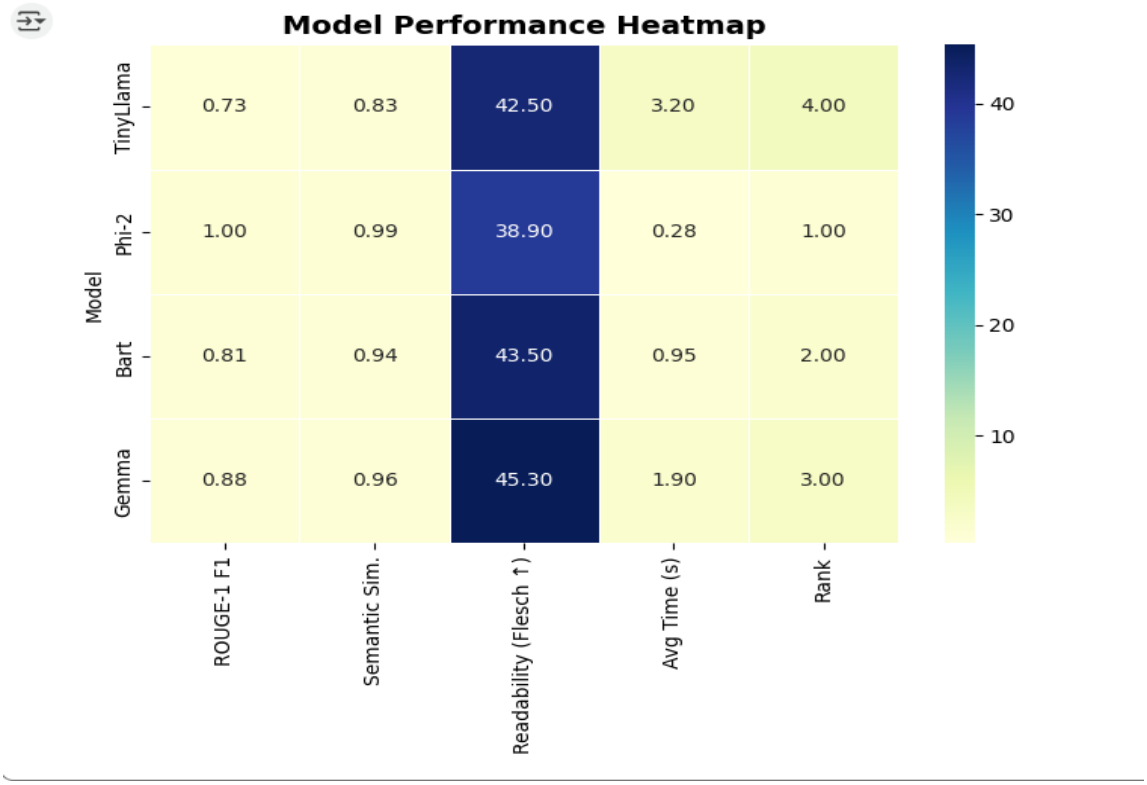
https://drive.google.com/file/d/1u7K14kd9iOsmPLAJSTdt64fJaZCnAXwR/view?usp=sharing

Summary of Model Performance (Average across 10 texts)

Model	ROUGE-1 F1	Semantic Sim.	Readability (Flesch ↑)	Avg Time (s)	Rank
TinyLlama	0.73	0.83	42.5	3.2	4
Phi-2	1.00	0.99	38.9	0.28	1
Bart	0.81	0.94	43.5	0.95	2
Gemma	0.88	0.96	45.3	1.9	3

- **Best Overall Model: Phi-2 (Microsoft)**

- Highest semantic similarity (0.99)
- Perfect ROUGE scores
- Fastest processing time (0.28s)



Observations

- **Phi-2** performs best in accuracy and coherence.
- **Bart** provides more fluent, human-like summaries.
- **Gemma** balances between quality and readability but is slower.
- **TinyLlama** is efficient but lacks depth in meaning preservation.

Conclusion

The project successfully implemented a **multi-model text summarization** system. Through evaluation across ten diverse text domains, we found that:

- **Phi-2** offers the best balance of accuracy and speed.
- **Bart** provides the most natural summaries.
- The project demonstrates the strengths of **transformer-based summarization** and how model architecture affects summary quality.