AI Engineering LLM Assignment – Text Summarization

Introduction:

This assignment focuses on building and evaluating an LLM-based text summarization system using a dataset of real user reviews from the Infloso mobile application — a platform that connects influencers with brands. The objective is to explore the dataset, preprocess the text for model consumption, apply a pretrained transformer-based model for abstractive summarization, and evaluate the quality of generated summaries using standard metrics.

Through this project, we aim to understand practical implementation aspects of LLM-powered summarization, analyze model behavior on real-world noisy review data, and discuss potential improvements and future research directions in text summarization systems.

Data Exploration:

The dataset reviews.txt contains 100 user reviews collected for the mobile application Infloso, a platform connecting influencers with brands. Each line in the file represents one user review.

After inspection:

1.Total reviews: 100
2.Usable cleaned reviews: 89
3.Reviews vary in length from short feedback to long multi-sentence complaints.
4.Language includes English and mixed Hindi-English.
5.Reviews contain both positive and negative user experiences.

This dataset is suitable for text summarization using pretrained language models.

Data Preprocessing:

To clean and standardize the text data, the following preprocessing steps were applied:

1.Converted text to lowercase.
2.Removed URLs and HTML tags.
3.Removed emojis and unnecessary special characters.
4.Normalized extra whitespace.
5.Removed very short lines (<20 characters).

After preprocessing, 89 clean reviews remained for summarization.

Summarization Strategy:

Since the dataset does not include labeled summaries, a pretrained abstractive summarization model was used.

Model chosen: T5-small (Text-to-Text Transfer Transformer)

Reason for selection:
- Converts all NLP tasks into text-to-text format.
- Performs well on conversational review text.
- Lightweight and efficient for Colab execution.
- No domain specific fine-tuning required for baseline results.

Summarization was performed in a zero-shot setting using beam search decoding.

Model Architecture and Hyperparameters:

Model: T5-small
Encoder Layers: 6
Decoder Layers: 6
Hidden Size: 512
Max Input Tokens: 512
Max Output Tokens: 120
Beam Search Width: 4
Decoding Strategy: Beam Search
Fine-tuning: Not applied (pretrained weights used)

Top 5 Summarized reviews:

```
sorted_reviews = sorted(cleaned_reviews, key=len, reverse=True)
top5_reviews = sorted_reviews[:5]

top5_summaries = [summarize(r) for r in top5_reviews]

for i in range(5):
    print(f"\n--- Review {i+1} Summary ---")
    print(top5_summaries[i])
```

```
--- Review 1 Summary ---
i m a software developer and being professional this app has the worst ux i ve ever experienced. i m a software developer and being professional this a

--- Review 2 Summary ---
the infloso app is a game changer!firstly, the user interface is clean and intuitive. the task management features are top notch.

--- Review 3 Summary ---
worst app ever doesn t deserve a single star. i can contact no person through any means. will definitely not recommend anyone about this app wasted my

--- Review 4 Summary ---
infloso is a great application this app provide us branded products peoples see the video of product for better results the influencers of this applica

--- Review 5 Summary ---
infloso is an exceptional paid barter collaboration app that has revolutionized my content creation journey. the platform s innovative features and use
```

Evaluation Strategy:

ROUGE metrics were used to evaluate summarization quality.

ROUGE-1 measures unigram overlap,
ROUGE-2 measures bigram overlap,
ROUGE-L measures longest common subsequence.

ROUGE scores indicate that the model preserves key information from the original reviews while reducing text length significantly

```
[17]
import evaluate

rouge = evaluate.load("rouge")

scores = rouge.compute(
    predictions=top5_summaries,
    references=top5_reviews
)

print(scores)
```

```
{'rouge1': np.float64(0.5373137119348375), 'rouge2': np.float64(0.49404923914388227), 'rougeL': np.float64(0.5084679574816063), 'rougeLsum': np.float64
```

Future Improvements:

- Fine-tune T5 on manually created review-summary pairs
- Use larger models such as T5-base or PEGASUS
- Apply sentiment-aware or aspect-based summarization to separate pros and cons