It's important to document experiments, even those that lead to choosing an alternative model. This demonstrates a thorough approach in your capstone project.

Here is a comprehensive documentation section detailing the experimentation with the T5 model, comparing it with BART, and justifying the final model choice, based on the provided logs and our discussions.

Model Selection: Experimentation with T5

1. Rationale for Considering T5

While developing the email subject line generation model, we explored multiple state-of-the-art sequence-to-sequence architectures available through the Hugging Face library. Alongside the BART model (known for its effectiveness in generation tasks due to its denoising autoencoder pre-training), the **T5 (Text-to-Text Transfer Transformer)** model was identified as a strong candidate.

T5's appeal lies in its versatile **text-to-text framework**, which treats every NLP task as a process of converting an input text sequence into an output text sequence. This unified approach has shown impressive results across various benchmarks. We hypothesized that this framework, potentially with a task-specific prompt, could be effective for the abstractive summarization required for subject line generation.

To conduct a practical comparison within project constraints (time, compute resources), we opted to evaluate the smaller, faster **t5-small** checkpoint against the more resource-intensive facebook/bart-large model chosen based on initial analysis.

2. T5 Experiment Methodology

The T5 experiment followed the same overall pipeline as the BART experiment but with model-specific adjustments:

1. **Data Loading:** The *exact same cleaned datasets* (train_cleaned_*.csv, dev_cleaned_*.csv, test_cleaned_*.csv) generated by the cleaning script (2_...) were used as input. This ensured a fair comparison based on identical source data.
2. **Model & Tokenizer:**
    - The t5-small checkpoint was loaded using AutoModelForSeq2SeqLM.from_pretrained("t5-small").
    - The corresponding AutoTokenizer.from_pretrained("t5-small") was used. T5 utilizes the SentencePiece tokenizer.
3. **Preprocessing (T5 Specific):**

- **Prefix:** A crucial difference for T5 is the standard practice of using task-specific prefixes. For this summarization-like task, the prefix "summarize: " was added to the beginning of every input email body before tokenization.
- **Tokenization:** The preprocess_function_t5 (as implemented in 3_t5_a_subjectline_dataprocessing_v1.py) applied the T5 tokenizer to the prefixed bodies and the target subjects, handling truncation to MAX_INPUT_LENGTH (512) and MAX_TARGET_LENGTH (32), respectively.
- The processed data, containing input_ids, attention_mask, and labels, was saved to disk similar to the BART preprocessing step (likely in a directory like t5_processed_datasets).

4. **Training Setup (T5):**
   - **Environment:** Training was conducted in Google Colab using a GPU (as indicated by the "[T5 Colab]" tags in the logs).
   - **Trainer:** The Hugging Face Seq2SeqTrainer was used.
   - **Hyperparameters:** To maintain comparability where feasible, similar hyperparameters to the BART run were intended:
     - Epochs: 3
     - Optimizer: AdamW (default for Trainer)
     - Learning Rate: 3e-5 (a common value for T5 fine-tuning)
     - Batch Size: 16 per device (possible due to t5-small's smaller size compared to bart-large). Gradient accumulation was likely set to 1.
     - Evaluation/Saving/Logging: Performed each epoch.
     - Best Model Selection: Based on validation rougeL.
5. **Evaluation Metrics:** The same compute_metrics function using the evaluate library's ROUGE implementation was used, calculating ROUGE-1, ROUGE-2, ROUGE-L, ROUGE-Lsum, and median generated length (gen_len). NLTK sentence tokenization was used within compute_metrics for standard ROUGE-Lsum calculation.

3. T5 Experiment Results

The T5-small model was trained for 3 epochs. The logs indicate the following performance progression on the validation set and final results on the test set:

T5-small Fine-tuning Log Summary:

Epoch Training Loss Validation Loss Rouge1 Rouge2 Rougel Rougelsum Gen Len

1	3.4499	2.9305	0.2696	0.1302	0.2660	0.2661	16.0
2	3.1753	2.8765	0.2730	0.1340	0.2689	0.2692	17.0
3	3.1232	**2.8637**	**0.2744**	**0.1353**	**0.2698**	**0.2698**	**17.0**

(Best validation scores achieved at Epoch 3)

Final T5-small Test Set Evaluation (Using Epoch 3 Checkpoint):

- test_loss: 2.7351
- test_rouge1: 0.2909
- test_rouge2: 0.1549
- test_rougeL: 0.2859
- test_rougeLsum: 0.2855
- test_gen_len: 19.0

Observations from T5 Run:

- The model trained successfully, with both training and validation loss generally decreasing over epochs.
- ROUGE scores showed slight improvement over the epochs, peaking at Epoch 3 on the validation set.
- The median generated length (gen_len) on the validation set increased slightly from 16 to 17 tokens and was 19 on the test set. This is still significantly longer than the human average but shorter than BART-large's final test length.
- The training was significantly faster than BART-large (~15 minutes vs. ~70+ minutes).
- An error occurred during the saving of training metrics (save_metrics() got an unexpected keyword argument 'metrics_path'), although the model itself saved correctly, and test metrics were calculated and logged. This seems like a minor script argument issue specific to that run.

4. Comparison: T5-small vs. BART-large

Let's compare the best results achieved by each model on the **test set**:

Metric	T5-small (Epoch 3)	BART-large (Epoch 2)	Winner
Test ROUGE-1	0.2909	**0.3560**	BART-large
Test ROUGE-2	0.1549	**0.1929**	BART-large

Test ROUGE-L	0.2859	**0.3460**	BART-large
Test ROUGE-Lsum	0.2855	**0.3455**	BART-large
Test Gen Len	**19.0**	22.0	T5-small
Approx Train Time	~15 min	~70+ min	T5-small

Analysis of Comparison:

- **Performance (ROUGE):** BART-large significantly outperformed t5-small on all standard ROUGE metrics on the unseen test set. The difference is quite substantial (e.g., ~0.06 points absolute difference on ROUGE-L).
- **Generated Length:** t5-small produced slightly shorter subject lines on average (median 19 tokens) compared to bart-large (median 22 tokens) on the test set. However, both models generated subjects considerably longer than the human average (~4 tokens).
- **Training Efficiency:** t5-small trained dramatically faster due to its much smaller size.
- **Potential Reasons for Difference:**
    - **Model Size/Capacity:** bart-large (~400M parameters) has a much higher capacity to learn complex patterns compared to t5-small (~60M parameters). This is likely the dominant factor.
    - **Pre-training Objective:** BART's denoising autoencoder objective is often cited as being particularly well-suited for generation tasks like summarization, potentially giving it an edge over T5's masked span prediction objective for this specific application.
    - **Prompting:** While the standard "summarize: " prefix was used for T5, it's possible that further prompt engineering could improve its performance, but this requires additional experimentation. BART required no prefix.

5. Conclusion and Final Model Decision

The experiment comparing t5-small and facebook/bart-large provided valuable insights. While t5-small offered significantly faster training, facebook/bart-large demonstrated substantially better performance according to the standard ROUGE metrics on the held-out test set for this email subject line generation task.

Although bart-large produced slightly longer subject lines on average in the final test evaluation compared to t5-small (median 22 vs. 19 tokens), its superior performance in capturing lexical overlap (ROUGE scores) suggests it learned a better mapping from email body to relevant subject content. The generated length is a parameter that can potentially

be controlled further during inference or through more advanced decoding strategies if needed.

Given the capstone project's goal to develop an effective subject line generation model and the clear quantitative advantage shown by bart-large on this dataset, **the decision was made to proceed with the fine-tuned facebook/bart-large model** (specifically, the best checkpoint saved from its training run) for subsequent steps, including human evaluation and the final Streamlit demonstration app. The significantly longer training time for bart-large was deemed an acceptable trade-off for the better generation quality indicated by the automated metrics.
