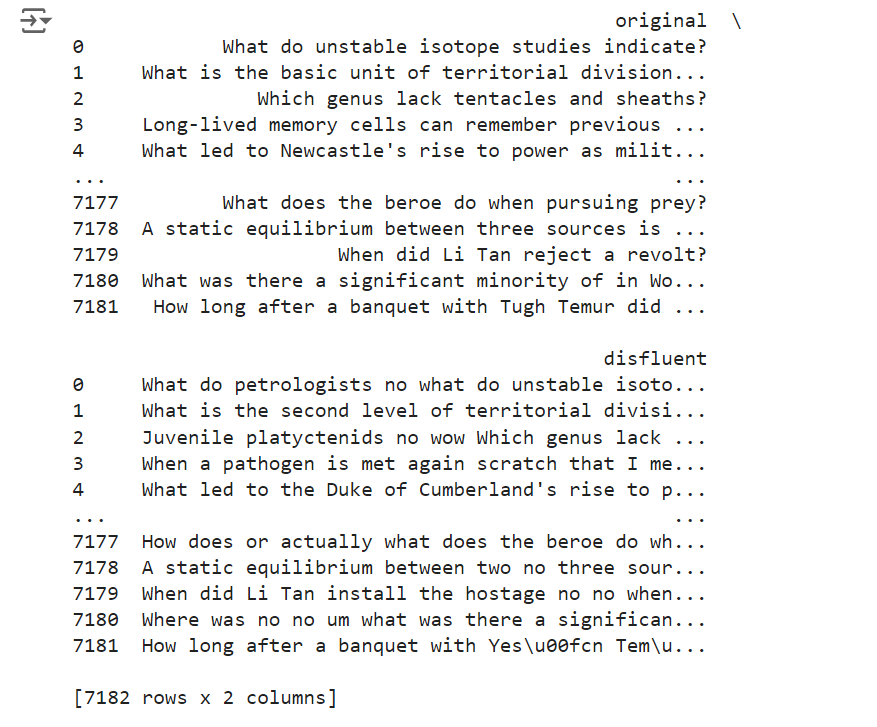
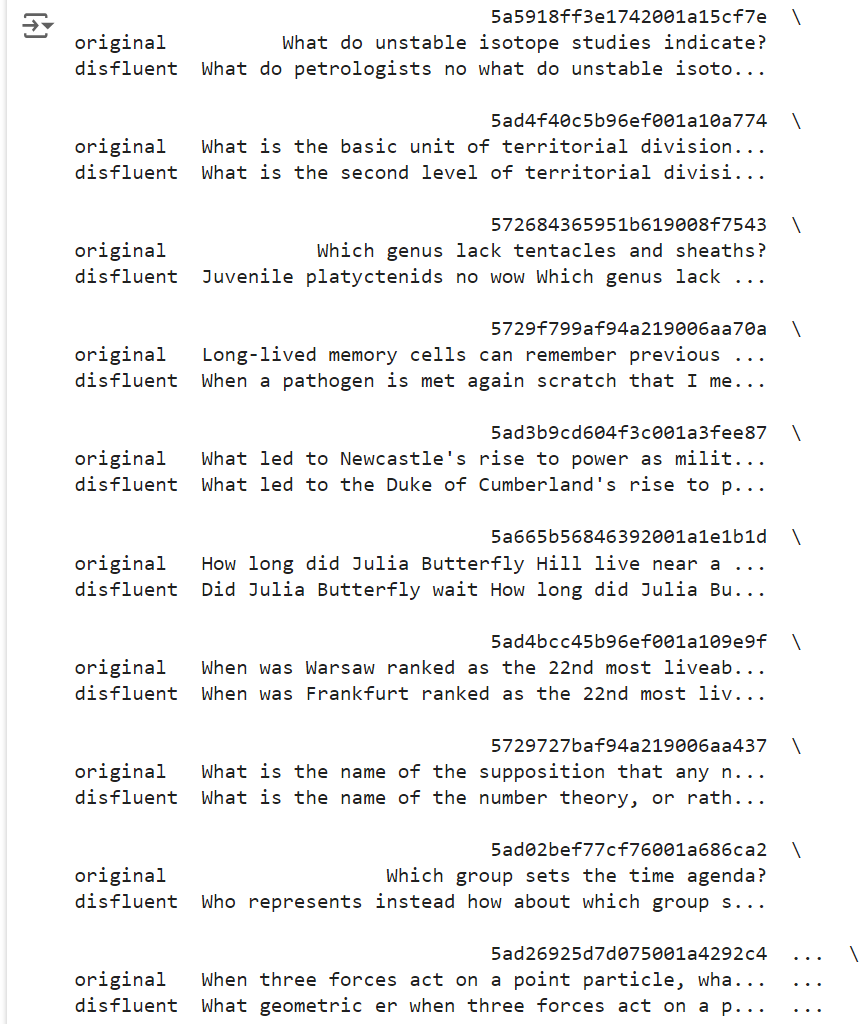
**Data Pre-processing**

The first step involved writing code to properly read the train.json, test.json, and dev.json files. These JSON files required transformation, and their indices needed to be reset to make them suitable for training, testing, and validation in subsequent steps. The images below illustrate the dataset before and after transformation and index resetting:



**Model Selection**

In the second phase, the selection of the appropriate NLP approach was crucial. Although several options were considered, a sequence-to-sequence (seq2seq) pre-trained model, fine-tuned on the training data, emerged as the most suitable approach, especially given the requirement to evaluate the model on unseen data with potentially unknown rules.

Two models were shortlisted for this task: BART (Bidirectional and Auto-Regressive Transformers) and T5 (Text-To-Text Transfer Transformer). Both models are well-suited for text generation tasks and are available on Hugging Face for fine-tuning.

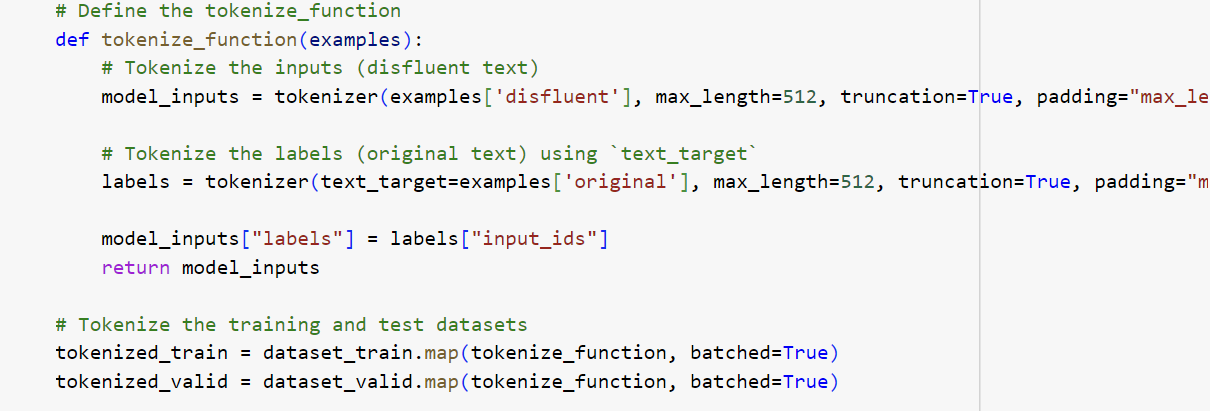
* **BART Base Model**: With approximately 139 million parameters, BART is particularly effective for tasks involving text correction, as it reconstructs corrupted text into its original form.
* **T5 Base Model**: T5, with around 220 million parameters, is a more general model that treats all NLP tasks as text-to-text problems. This generalization makes it versatile, though it may not be as specialized as BART for text correction tasks.

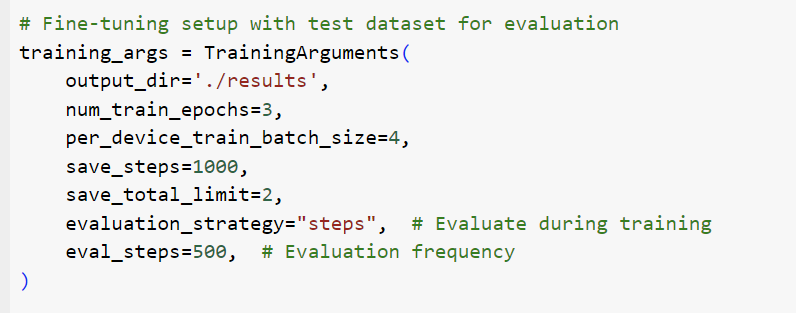
Given these considerations, both models were selected for further experimentation to evaluate their performance on the given task.

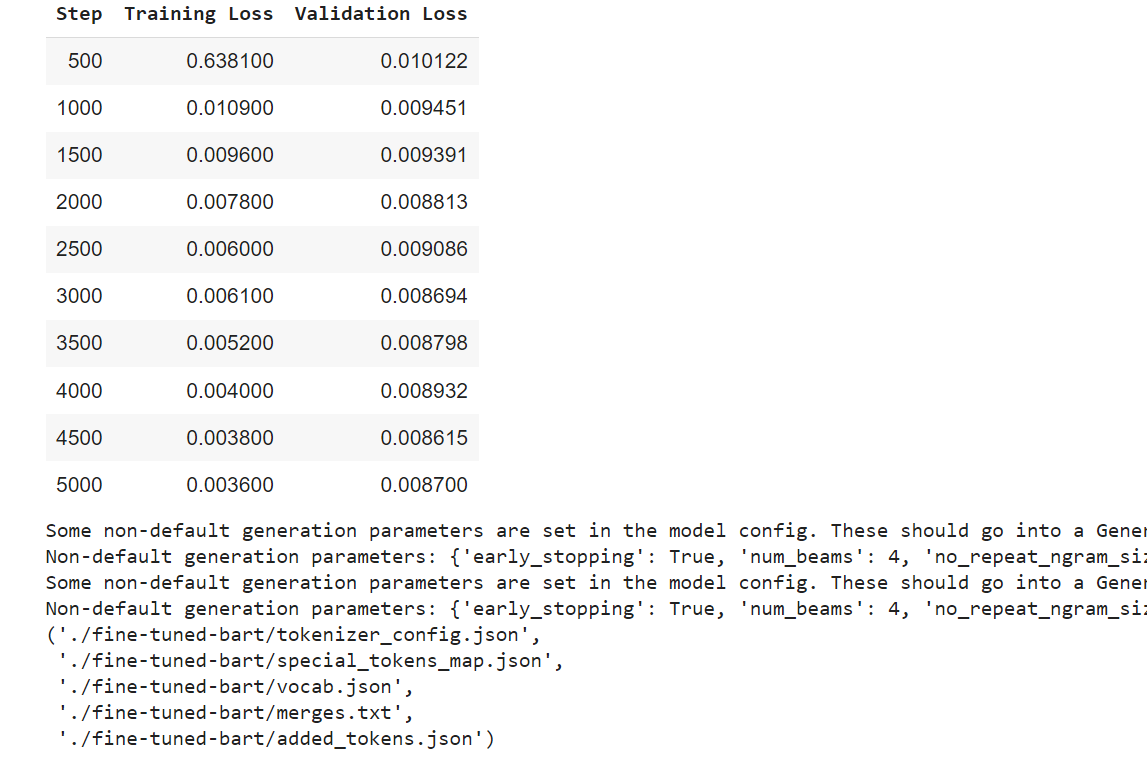
**Experiment Setup and Results**

For this project, the base versions of BART and T5 were used instead of their larger counterparts due to limitations in available GPU resources and task complexity. The experiments were conducted using a Google Colab A100 GPU.

**Tokenization**: Both the training and validation datasets were tokenized using the respective tokenizers for BART and T5. The tokenization process ensured that each sentence had a token size of 512. If a sentence exceeded this limit, it was truncated; if it was shorter, padding was added to maintain the 512-token size.



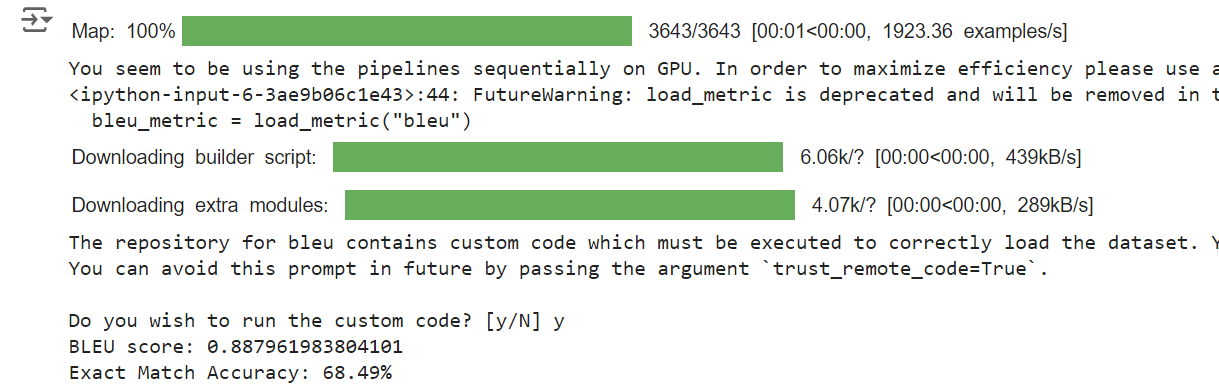
**Training Parameters**: The initial experiment was conducted using the BART base model. The training was carried out for three epochs, with a per-device batch size of four. The evaluation strategy was set to 'steps,' and evaluation steps were set to 500. While the parameters were conservative, minimizing the risk of overfitting, no specific measures were initially implemented to prevent overfitting. The image below shows the training parameters and the training-validation loss for each step:



The validation loss remained relatively stable towards the end, with only a slight increase, indicating no clear signs of overfitting. These insights guided the adjustment of training parameters for subsequent experiments with both the T5 base and BART models.

**Performance Evaluation**: The model's performance was evaluated using both BLEU (Bilingual Evaluation Understudy) and exact match accuracy metrics. BLEU scores range from 0 to 1 and assess how closely the generated sequence matches the original sentence using n-grams, with penalties for dissimilar sequences. Exact match accuracy, being stricter, requires the generated text to precisely match the original text.

Although both BLEU and exact match accuracy are high benchmarks for text-to-text tasks, they are not context-aware. Therefore, manual evaluation was also conducted on the test dataset by comparing the generated sentences with the originals. The following results were obtained for the initial experiment:



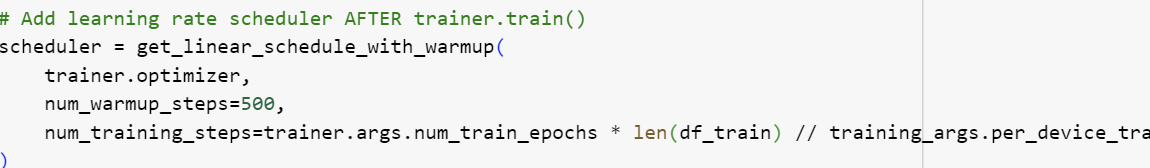
* **BLEU Score**: 88.79
* **Exact Match Accuracy**: 68.49%

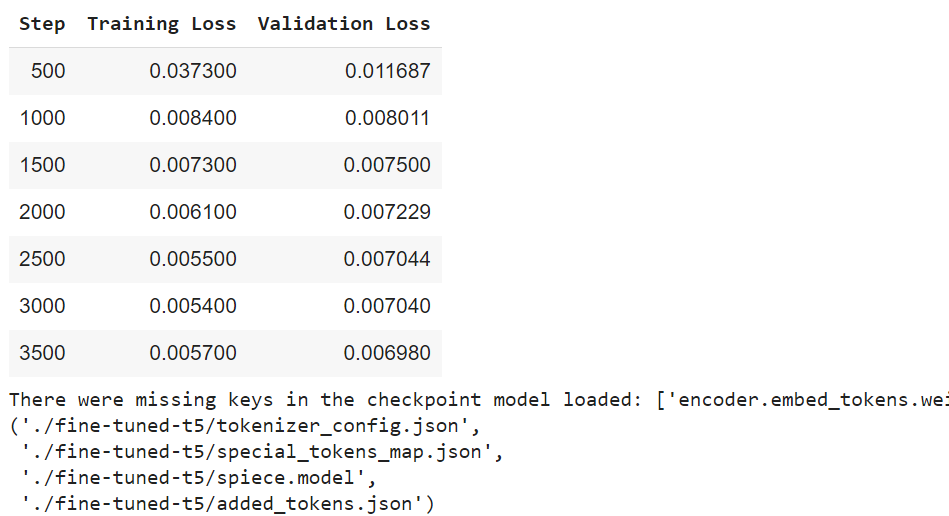
These results are promising, indicating that the BART model performs well on this task.

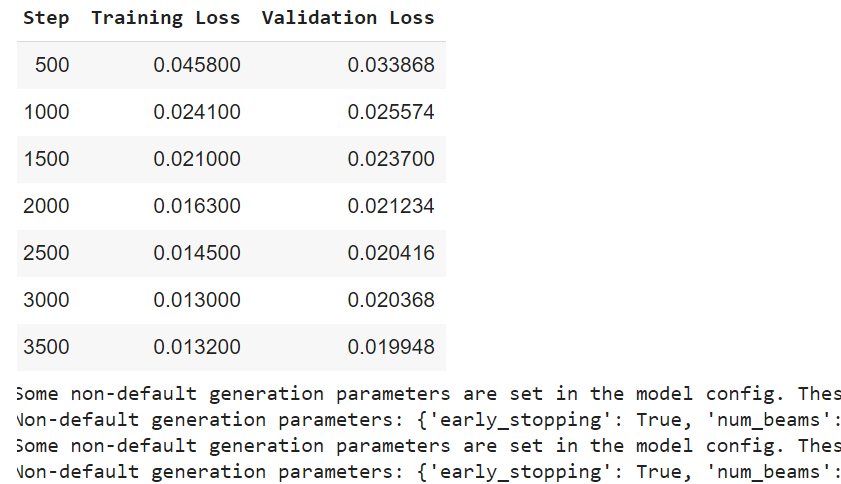
**Avoiding Overfitting**: For the next set of experiments with both the T5 and BART base models, additional measures were taken to prevent overfitting:

1. **Early Stopping**: Implemented with a patience of three steps, meaning that if the model's performance did not improve over three consecutive steps, training would stop.
2. **Weight Decay**: Set to 0.01, a form of L2 regularization, which penalizes large weights in the model.
3. **Learning Rate Scheduler**: A linear rate scheduler with 500 warm-up steps was used to gradually increase the learning rate at the beginning and then decay it slowly.
4. **Training Parameters**: The number of epochs was set to three, and the batch size was increased to eight to control the number of overall steps, ensuring better generalization and reducing the risk of overfitting.



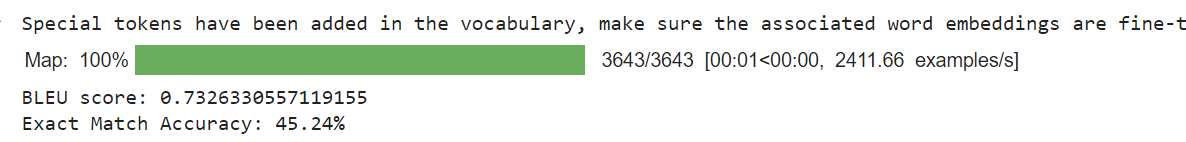


The training and validation losses for both the T5 and BART base models are shown below:

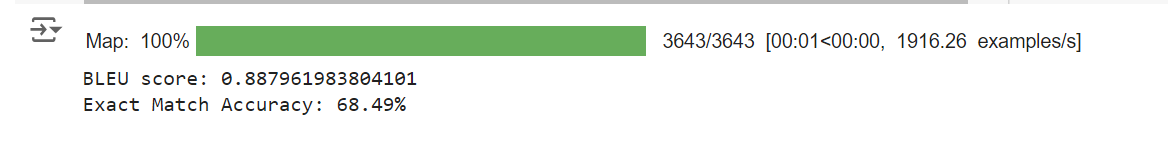


The measures implemented successfully prevented overfitting, as evidenced by the continuous decrease in both training and validation losses. However, the results suggest that the models might be slightly undertrained, as the validation loss does not plateau, though the decreasing difference between the losses towards the end is a positive sign.

**Comparative Results**: The performance of the models under these parameters is summarized below:



* **T5 Base Model**:
  + BLEU Score: 0.733
  + Exact Match Accuracy: 45.24%

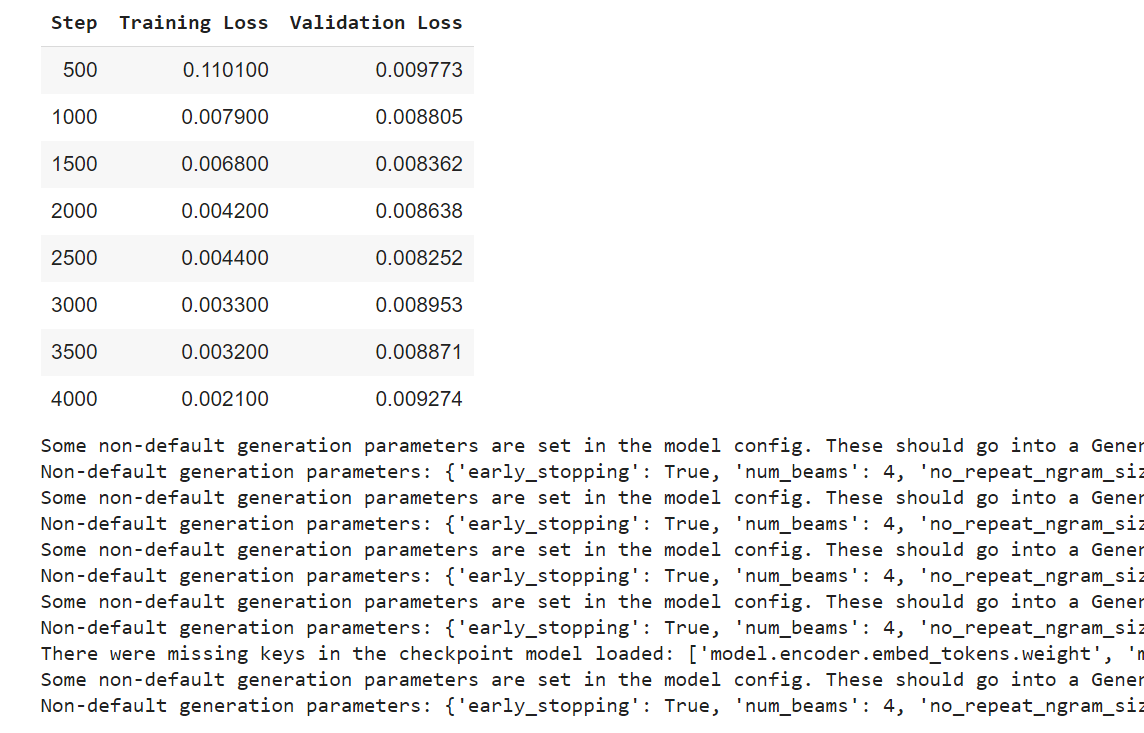


* **BART Base Model**:
  + BLEU Score: 0.888
  + Exact Match Accuracy: 68.50%

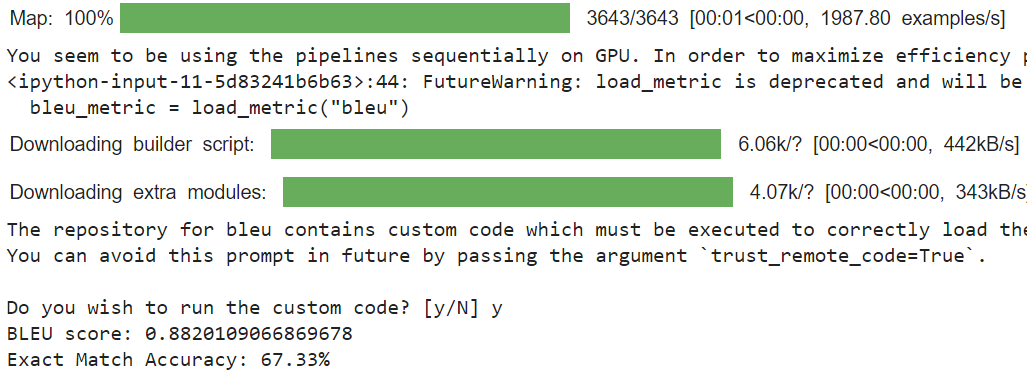
It is evident from these results that the BART base model outperforms the T5 model for this specific task. This outcome can be attributed to BART’s design, which is tailored for text correction, whereas T5 is a more generalized NLP model.

**Final Experiment**

To further explore the BART model's potential, it was trained for one additional epoch using the same parameters to observe any changes in performance. The following outcomes were recorded:



The validation loss showed a slight upward trend towards the end, indicating potential overfitting. However, the final performance metrics were as follows:



* **BLEU Score**: 0.882
* **Exact Match Accuracy**: 67.33%

These results, were slightly lower than the previous setup. This could be possible because this specific part was run in a different run-time but still demonstrate that the BART model performs well and can be effectively used for tasks involving the correction of contextually disfluent sentences and the parameters in the previous experiment setup were ideal.

**Conclusion and Remarks**

In conclusion, the BART base model outperforms the T5 base model for the given task. By utilizing parameters such as a batch size of eight, three epochs, and early stopping, the model is prevented from overfitting and can achieve satisfactory performance. Although additional steps such as using a larger model, reducing token size, or extending training could further enhance performance, limitations in computational resources and time constraints prevented these explorations.

This document serves not as a traditional research paper but as a detailed overview of the coding process and the rationale behind the choices made during the experiment.