Arshia Aflaki's Assignment Report

Arshia Aflaki

August 26, 2024

Contents

1 Coding

The codes can be found here: Github

2 Results

2.1 Model Evaluation

Table 1 provides a detailed evaluation of the fine-tuned T5 model. The computational cost and performance metrics such as BLEU, METEOR, and ROUGE scores are reported for both training and validation datasets.

Metric	**Description**	**Fine-tuned T5**
Computational Cost	Time taken for training	1 hour and 24 minutes
Average BLEU Score	Training	0.8285
	Validation	0.8268
Average METEOR Score	Training	0.9330
	Validation	0.9324
Average ROUGE Score	Training	0.9160
	Validation	0.9127

Table 1: Model Evaluation Metrics for Fine-tuned T5

2.2 Training Hyperparameters

Table 2 lists the hyperparameters used for training the fine-tuned T5 model. These parameters were carefully selected to balance training efficiency and model performance.

Hyperparameter	**Value**
Evaluation Strategy	`epoch`
Learning Rate	2e-5
Train Batch Size per Device	16
Eval Batch Size per Device	16
Weight Decay	0.01
Save Total Limit	3
Number of Epochs	4
Predict with Generate	`True`
FP16	`True`
Logging Steps	10

Table 2: Hyperparameters for Fine-tuning the T5 Model

2.3 Loss Comparison

Figure 1 compares the training and validation cross-entropy loss over epochs. This comparison helps in understanding the model's performance and generalization capability.

2.4 Analysis

The fine-tuned T5 model demonstrates excellent performance with high BLEU, METEOR, and ROUGE scores, reflecting its strong ability to generate accurate text predictions. The low variance between training and validation metrics suggests effective generalization and minimal overfitting.

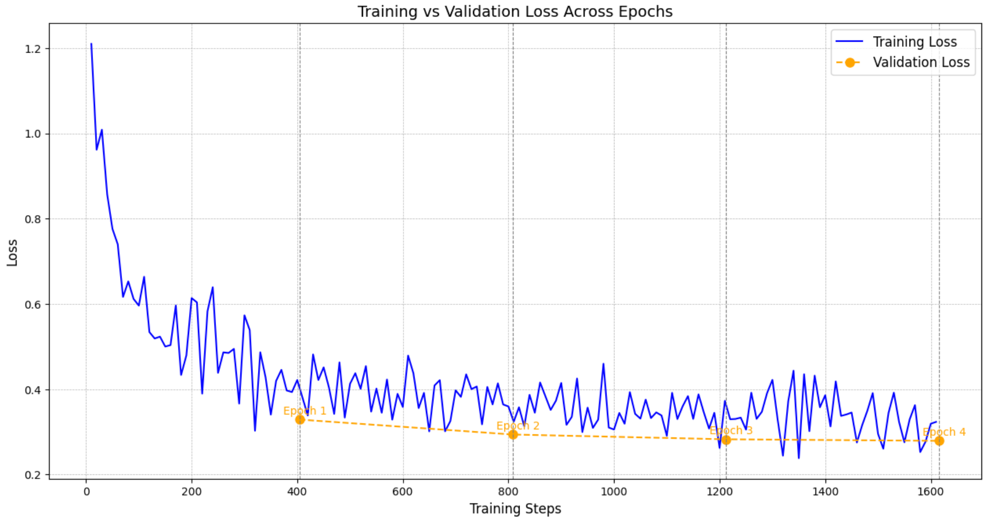

Figure 1: Training vs. Validation Cross-Entropy Loss

The loss comparison plot (Figure 1) further corroborates the model's stability, as the validation loss closely follows the training loss without significant divergence, indicating no major overfitting issues.

3 Reports

3.1 Research and Development Process

The task was to develop a question rewrite model for the Disfl QA benchmark dataset [1]. This dataset involves rewriting disfluent questions into fluent ones, which is a text summarization task rather than text generation.

Initially, two approaches were considered:

- **Autoencoder-based Neural Network**: Using LSTMs, which was computationally intensive.
- **Pretrained Summarization Model**: Chosen due to limited computational resources and a small dataset size.

Model Selection: Several models were evaluated:

- **BERT**: Large and resource-heavy.
- **BART**: Considered but not used due to its resource requirements.
- **T5**: The T5-small model was selected for its efficiency and suitability for summarization tasks. This model has 50M parameters and was fine-tuned due to its balance between performance and computational cost [2].

Process:

1. Data preprocessing and tokenization.
2. Fine-tuning of the T5-small model.
3. Achieved a validation ROUGE score of 0.9127 through hyperparameter tuning and optimization.

3.2 References

References

[1] Google Research. (n.d.). Disfl QA dataset. *Retrieved from* `https://github.com/google-research-datasets/Disfl-QA`

[2] Raffel, C., Shinn, C., Roberts, A., et al. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140), 1-67.