**DL Assignment 2**

**Report**

**Group ID : 4**

**Methodology for Part A: Implementing and Benchmarking a Custom Encoder-Decoder Model**

**SmolVLM Baseline Evaluation**

* **Overview:** To establish a performance baseline, the **SmolVLM** model (HuggingFaceH4/smolvlm-1.6b-patch14-336) was utilized for zero-shot image captioning on the test dataset3. No training or fine-tuning was performed on SmolVLM.
* **Implementation:** Captions were generated using the model's default settings, with \_attn\_implementation="eager" used as necessary to ensure compatibility3.
* **Metrics:** Standard image captioning metrics (BLEU, ROUGE-L, METEOR) were calculated using the generated captions and the ground-truth captions from the test set to serve as a reference point for the custom model3.

**Custom Encoder-Decoder Model**

* **Overview:** A custom image captioning model was developed using a Transformer-based encoder-decoder architecture, specifically designed to operate within the memory constraints of a free-tier Google Colab GPU (e.g., T4 with ~15GB VRAM)3.
* **Architecture:**
  + **Image Encoder:** A pre-trained Vision Transformer (ViT), specifically vit-small-patch16-224, was used as the image encoder. This model processes input images and generates feature embeddings3.
  + **Text Decoder:** A pre-trained GPT-2 model (gpt2) served as the text decoder. It takes the image embeddings from the ViT encoder and autoregressively generates the caption text3.
  + **Integration:** The **VisionEncoderDecoderModel** class from the Hugging Face Transformers library was used to combine the ViT encoder and GPT-2 decoder into a cohesive sequence-to-sequence model3. The cross-attention layers were initialized randomly and trained.
* **Training:**
  + **Dataset:** The model was trained on the provided custom\_captions\_dataset3.
  + **Optimization:** The Seq2SeqTrainer from Hugging Face was employed for training. Key hyperparameters included:
    - Learning Rate: 5e-53
    - Number of Epochs: 53
    - Batch Size (Train/Eval): 8 per device3
    - Gradient Accumulation Steps: 2 (effective batch size of 16)3
    - Weight Decay: 0.013
  + **Constraints:** Mixed-precision training (fp16=True) was enabled to manage memory usage and speed up training on the T4 GPU3. The model configuration and batch size were chosen to fit within the ~15GB memory limit.

**Methodology for Part B: Studying Performance Change Under Image Occlusion**

**Image Occlusion Strategy**

* **Patching:** Each image in the test set was divided into a grid of 16x16 non-overlapping patches3.
* **Masking:** For each image, a specific percentage (10%, 50%, or 80%) of these patches was randomly selected. The pixel values within the selected patches were set to black (0), effectively masking parts of the image content3. This process was repeated for each of the three occlusion levels (10%, 50%, 80%), creating three distinct perturbed versions of the test dataset**.**

**Evaluation Under Occlusion**

* **Models Evaluated:** Both the zero-shot SmolVLM model and the trained custom Vision Transformer-GPT2 model were evaluated on these occluded datasets3.
* **Metrics Recorded:** BLEU, ROUGE-L, and METEOR scores were computed for the captions generated by each model on the occluded images at each perturbation level (10%, 50%, 80%)3.
* **Change Calculation:** To quantify the impact of occlusion, the change in each metric was calculated relative to the model's performance on the original (non-occluded) test images3:  
  ΔMetric=Metricoccluded−MetricoriginalΔMetric=Metricoccluded−Metricoriginal  
  This calculation was performed for both models across all three metrics and all three occlusion percentages.
* **Data Storage:** The original captions, generated captions (from both models for original and occluded images), and the corresponding perturbation levels were saved to facilitate the classification task in Part C13**.**

**Results for Part A: Model Performance Comparison**

| **Model** | **BLEU Score** | **ROUGE-L Score** | **METEOR Score** |
| --- | --- | --- | --- |
| **SmolVLM (0-shot)** | **0.0334** | **0.2293** | **0.2594** |
| **Custom Model** | **0.0421** | **0.2306** | **0.2953** |

**Results for Part B: Performance Change Under Occlusion**

The following tables summarize the change in performance metrics for both SmolVLM and the Custom Model when evaluated on images with 10%, 50%, and 80% patch occlusion compared to their performance on the original images3**.**

**SmolVLM Performance Change**

| **Perturbation (%)** | **BLEU Change** | **ROUGE-L Change** | **METEOR Change** |
| --- | --- | --- | --- |
| **10** | **0.003463** | **0.008549** | **-0.008027** |
| **50** | **0.003327** | **0.014796** | **-0.013452** |
| **80** | **-0.013117** | **-0.028565** | **-0.070910** |

**Custom Model Performance Change**

| **Perturbation (%)** | **BLEU Change** | **ROUGE-L Change** |
| --- | --- | --- |
| **10** | **0.000324** | **0.000406** |
| **50** | **0.000359** | **-0.003229** |
| **80** | **-0.001138** | **-0.010673** |

**Methodology for Part C: Caption Source Classification**

**Overview**

Part C involves building a classifier to determine whether image captions were generated by **Model A (SmolVLM)** or **Model B (Custom)**. This classification task is essential for detecting AI-generated content and identifying the specific model responsible for a given caption.

**Dataset Construction**

The dataset was constructed by combining captions from both SmolVLM and a custom model with various perturbation percentages:

1. **Input Format**: <original\_caption> <SEP> <generated\_caption> <SEP> <perturbation\_percentage>
2. **Labels**: Binary classification (0 for Model A/SmolVLM, 1 for Model B/Custom)
3. **Perturbation Levels**: Three distinct perturbation percentages (10%, 50%, 80%)

To prevent data leakage, the dataset was split (70:10:20) based on unique original captions, ensuring that variations of the same caption don't appear across different splits.

**Model Architecture**

The classifier uses a fine-tuned BERT model with additional layers for classification:

Sigmoid (implicit)

Linear (128 -> 1)

ReLU

Linear (768 - > 128)

BERT Base Model

The model architecture consists of:

1. **BERT Encoder**: Pre-trained BERT (bert-base-uncased) that converts text into contextual embeddings
2. **Classification Head**:
   * Linear layer reducing dimensionality from 768 to 128
   * ReLU activation for non-linearity
   * Dropout (0.3) for regularization
   * Final linear layer outputting a single value for binary classification

This design leverages BERT's powerful language understanding capabilities while adding task-specific layers to discriminate between caption sources.

**Training Methodology**

**Hyperparameter Tuning**

A comprehensive grid search was performed across:

* **Learning rates**: 1e-5, 3e-5
* **Batch sizes**: 16, 32
* **Training epochs**: 3, 5, 10

The best model was selected based on the highest validation F1-score, as this metric balances precision and recall, which is important for this classification task.

**Optimization Details**

* **Loss Function**: Binary Cross-Entropy with Logits (BCEWithLogitsLoss)
* **Optimizer**: AdamW with weight decay
* **Text Processing**: Tokenization with truncation/padding to 128 tokens

**Evaluation Procedure**

The model was evaluated using:

1. **Accuracy**: Overall correct predictions
2. **Precision**: Ratio of true positives to all positive predictions
3. **Recall**: Ratio of true positives to all actual positives
4. **F1-Score**: Harmonic mean of precision and recall

The final evaluation was conducted on the held-out test set using only the best-performing model from hyperparameter tuning, ensuring an unbiased assessment of the model's performance.

**Implementation Details**

* The model was implemented using PyTorch and Hugging Face Transformers
* Training was conducted on GPU when available, with automatic device detection
* Dataset batching and preprocessing were handled by custom PyTorch Dataset classes

This methodology enables accurate classification of captions by source model, providing insights into the distinctive patterns that differentiate captions generated by SmolVLM versus a custom model.

**Results for Part C:**

**Accuracy : 0.9866310160427807**

**Precision : 1.0**

**F1\_score : 0.986449864498645**