ADL HW2

標籤	Course HW
優先級	⭐⭐
截止日期	@October 24, 2024
狀態	進行中

- Model (1%)
  - Describe the model architecture and how it works on text summarization.
    - T5 is a transformer-based encoder-decoder architecture designed for various sequence-to-sequence tasks, including text summarization.
    - **Key Components:**
    - **Encoder-Decoder Structure:**
      - **Encoder:** Processes the input text (`maintext`) and encodes it into continuous representations.
      - **Decoder:** Generates the output text (`title`) by predicting one token at a time, using both the encoder's representations and its own previously generated tokens.
    - **Self-Attention Mechanisms:**
      - Both the encoder and decoder use self-attention layers to capture dependencies between tokens in the input and output sequences.
    - **Cross-Attention:**
      - The decoder incorporates cross-attention layers to focus on relevant parts of the input sequence when generating each output token.

    Function in Text Summarization:

- **Input Processing:** The encoder transforms the main text into a context-aware representation.
- **Sequence Generation:** The decoder generates the summary by attending to the encoder's output and previously generated tokens.
- **Learning Objective:** The model is trained to minimize the difference between the generated summary and the reference summary, effectively learning to produce concise and informative summaries.

- Preprocessing (1%)
  - Describe your preprocessing (e.g. tokenization, data cleaning and etc.)

    Data Preprocessing Steps

    1. Tokenization:

    - **Inputs (`maintext`):**
      - Tokenized using the `AutoTokenizer` associated with the model.
      - Converts text into token IDs that the model can process.
    - **Targets (`title`):**
      - Tokenized similarly, using `tokenizer.as_target_tokenizer()` to ensure correct handling.

    2. Truncation and Padding:

    - **Inputs:**
      - Truncated to a maximum length of **256 tokens** to handle long texts and fit GPU memory constraints.
      - Uses `truncation=True` to cut off sequences longer than the maximum length.
    - **Targets:**
      - Truncated to a maximum length of **64 tokens** to focus on concise summaries.

    3. Label Preparation:

    - The tokenized target sequences are assigned to the `"labels"` key in the model inputs.

- Ensures the model computes the loss between its predictions and the actual summaries during training.

4. **Data Cleaning:**
   - The code assumes that the dataset is already clean.
   - No explicit steps for data cleaning like removing special characters or handling missing values are included.

5. **Dataset Column Removal:**
   - Original columns are removed after preprocessing to keep only the necessary tokenized data for training.

- Hyperparameter (1%)
  - Describe your hyperparameter you use and how you decide it.

1. **Number of Training Epochs (`num_train_epochs = 10`):**
   - Trains the model over 10 full passes of the training dataset.
   - Chosen to provide enough iterations for learning without overfitting.

2. **Batch Size (`batch_size = 8`):**
   - Small batch size to accommodate GPU memory limitations.
   - Effective batch size becomes `batch_size * gradient_accumulation_steps = 64`.

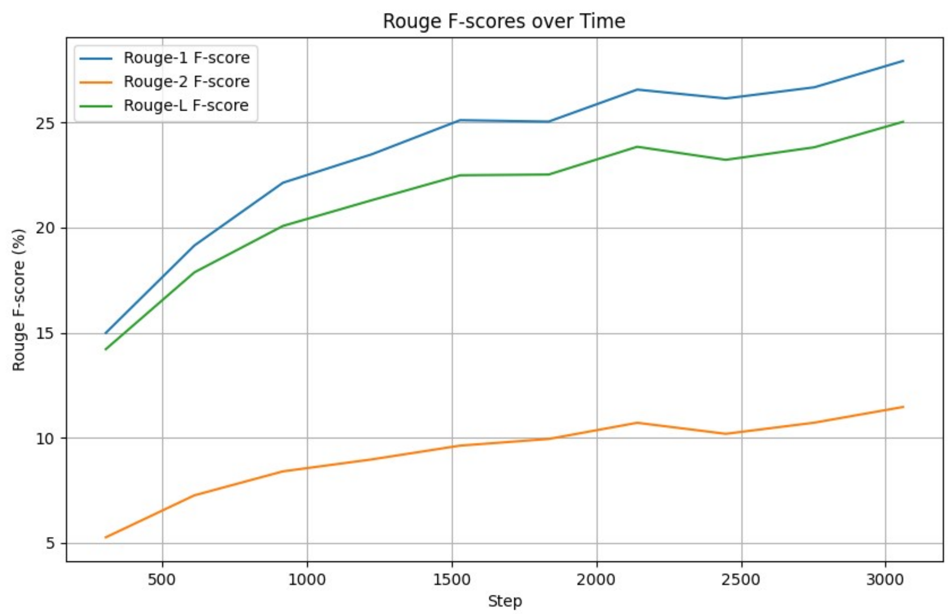
3. **Gradient Accumulation Steps (`gradient_accumulation_steps = 8`):**
   - Accumulates gradients over 8 steps before updating model weights.
   - Allows for a larger effective batch size without exceeding memory limits.

4. **Maximum Sequence Lengths:**
   - **Inputs (`max_length=256`):** Captures sufficient context from the main text.
   - **Targets (`max_length=64`):** Focuses on generating concise summaries.

5. **Optimizer Settings:**
   - **Optimizer:** `Adafactor`, suitable for large models like T5.
   - **Learning Rate (`lr=None`):**

- Relies on a relative learning rate with warmup (`relative_step=True`, `warmup_init=True`).
- Automatically adjusts learning rate based on training progress.

- Learning Curves (1%)
  - Plot the learning curves (ROUGE versus training steps)



- Stratgies (2%)
  - Describe the detail of the following generation strategies:

    1. Greedy Search:

    - **Mechanism:**
      - At each decoding step, selects the token with the highest probability.
    - **Characteristics:**
      - **Deterministic Output:** Always produces the same summary for a given input.
      - **Limitations:** May lead to suboptimal summaries due to lack of exploration.

    2. Beam Search:

    - **Mechanism:**

- Keeps track of the top `k` (beam width) most probable sequences at each step.
- Explores multiple hypotheses simultaneously.

- **Parameters:**
  - **`num_beams`:** Number of beams (e.g., 5).
- **Characteristics:**
  - **Balanced Exploration:** Increases the chance of finding a better overall summary.
  - **Computational Cost:** More beams require more computation.

3. Top-k Sampling:

- **Mechanism:**
  - At each step, samples the next token from the top `k` most probable tokens.
- **Parameters:**
  - **`top_k`:** Limits the number of tokens to sample from (e.g., 50).
- **Characteristics:**
  - **Controlled Randomness:** Introduces diversity while limiting unlikely tokens.
  - **Use Cases:** Creative tasks where variability is desired.

4. Top-p (Nucleus) Sampling:

- **Mechanism:**
  - Samples from the smallest possible set of tokens whose cumulative probability exceeds `p`.
- **Parameters:**
  - **`top_p`:** Cumulative probability threshold (e.g., 0.9).
- **Characteristics:**
  - **Adaptive Sampling:** The size of the token pool varies dynamically.
  - **Advantages:** Balances the need for diversity with maintaining coherence.

5. Temperature:

        - **Mechanism:**
            - Adjusts the probability distribution by scaling logits before applying softmax.
        - **Parameters:**
            - `temperature`: Controls the randomness (e.g., 0.7).
        - **Characteristics:**
            - **Lower Temperature (<1):** Makes the model more confident and outputs more common tokens.
            - **Higher Temperature (>1):** Flattens the distribution, increasing randomness.

- Hyperparameters (4%)

  Experiments with Different Generation Strategies

  1. Greedy Search Experiments:

    - **Setting 1:**
        - **Parameters:** `num_beams=1`, `do_sample=False`
        - **Results:** Lower ROUGE scores; summaries lacked depth.
    - **Setting 2:**
        - **Parameters:** Adjusted `max_length=64`
        - **Results:** Slight improvement but still inferior to beam search.

  2. Beam Search Experiments:

    - **Setting 1: Beam Width 3**
        - **Parameters:** `num_beams=3`, `do_sample=False`
        - **Results:** Improved ROUGE scores over greedy search.
    - **Setting 2: Beam Width 5**
        - **Parameters:** `num_beams=5`, `do_sample=False`
        - **Results:** Best ROUGE scores; summaries were coherent and informative.

  3. Top-k Sampling Experiments:

    - **Setting 1:** `k=50`

- **Parameters:** `do_sample=True`, `top_k=50`
- **Results:** Diverse but less coherent summaries; lower ROUGE scores.

- **Setting 2:** `k=100`
  - **Parameters:** `do_sample=True`, `top_k=100`
  - **Results:** Increased diversity; slight improvement in ROUGE scores but still below beam search.

4. Top-p Sampling Experiments:

- **Setting 1:** `p=0.9`
  - **Parameters:** `do_sample=True`, `top_p=0.9`
  - **Results:** Summaries sometimes strayed off-topic; lower ROUGE scores.
- **Setting 2:** `p=0.8`
  - **Parameters:** `do_sample=True`, `top_p=0.8`
  - **Results:** Slightly more coherent; marginally better ROUGE scores.

5. Temperature Experiments:

- **Setting 1:** `temperature=0.7`
  - **Results:** Summaries were more coherent; ROUGE scores improved.
- **Setting 2:** `temperature=1.0`
  - **Results:** Baseline; standard randomness.
- **Setting 3:** `temperature=1.5`
  - **Results:** Increased randomness; summaries were less coherent; ROUGE scores decreased.

Final Generation Strategy

Based on the experimental results, the final generation strategy combines **Beam Search** with **Temperature Adjustment**:

- **Parameters:**
  - **Beam Search:**

- `num_beams=5`
- `do_sample=False`

▪ **Temperature:**

- `temperature=0.7`

▪ **Additional Parameters:**

- `max_length=64` (to ensure concise summaries)
- `early_stopping=True` (to stop generation when an end-of-sequence token is generated)