

1. Model

In this 2b assignment, I used ``t5-small-chinese-cluecorpussmall`` as the T5 model, and for GPT-2, I used ``gpt2-distil-chinese-cluecorpussmall``. Both models were pre-trained using CLUECorpusSmall, which is a Chinese dataset.

T5 is composed of an encoder and a decoder, excelling in sequence-to-sequence tasks like question answering, translation, and summarization. It is pre-trained using a combination of unsupervised and supervised learning.

GPT-2, on the other hand, has only a decoder and is good at text generation. It generates text based on the preceding context and is pre-trained using unsupervised learning.

2. Dataset

For the T5 model, I used ``getitem`` to process the input data, setting the input as text and the label as a summary. These are then encoded into PyTorch tensors, followed by padding the output token ID list to ensure uniform length. The token ID for input padding is set to ``tokenizer.pad_token_id``, and for label padding, it is set to -100, so the model does not calculate padding loss.

For the GPT-2 model, I used three functions to handle the input data: ``getitem``, ``getitem2``, and ``collator_fn``.

In ``getitem``, the input is set as (text token ID list) + (SEP) + (summary token ID list), and the label is set as ``-100 * len(text token ID list) + (summary token ID list)``.

This method teaches the model that SEP marks the start of the summary. Therefore, during evaluation, inputting the text token ID list + SEP prompts the model to generate a summary. Setting the label as ``-100 * len(text token list + SEP) + (summary token ID list)`` ensures that the model only calculates loss on the summary part.

In ``getitem2``, the input is set as (text token ID list) + (SEP), and the label is set similarly.

Finally, ``collator_fn`` converts every element of input and label into PyTorch tensors and pads them. The padding token ID in the label is set to -100, preventing the model from calculating loss on the padding.

3. Training

The hyperparameters used for model training are:

- Batch size = 32
- Learning rate = 0.002
- Epochs = 35

First, clear gradients from previous iterations, pass the dictionary contents of the batch as parameters to the model, compute the loss, calculate the gradients, and then update the model parameters.

Below is the ROUGE score of the T5/GPT-2 models.

[illegible]

GPT-2:

We observe that the ROUGE-P score of GPT-2 is lower than T5, while the ROUGE-R and ROUGE-F scores for both are similar. This indicates that T5 includes more content from the reference summary than GPT-2.