CSIE5431 Applied Deep Learning

Homework 2 Report

Name: 高榮浩 **ID:** R12922127

1. Model

- ### Model Architecture and Text Summarization Process

The text summarization model employed for this homework is mT5 (**m**ultilingual **T5**), a multilingual variant of T5 (**T**ext-**T**o-**T**ext **T**ransfer **T**ransformer). mT5 adheres to the same design principles as T5 and undergoes pre-training on a dataset known as mC4, which contains text in 101 languages sourced from web scraping through the Common Crawl. This multilingual pre-training equips mT5 with the capability to proficiently handle text summarization across a multitude of languages.

Similar to T5, mT5 utilizes a text-to-text format, wherein both input and output are treated as text. This format offers remarkable flexibility in addressing a wide array of NLP tasks, including summarization. The model learns to generate concise summaries based on textual inputs, making it a versatile tool for various summarization tasks.

- ### Preprocessing Methods for Text Summarization

I employed the `T5Tokenizer` for tokenization purposes. To provide a simplified explanation of the tokenizer algorithm, I'll illustrate it with the following example.

When tokenizing the input `"Hello 作業二"`, the resulting tokens are as follows: `['▁Hello', '▁', '作業', '二']`. For more comprehensive details about this tokenization process, please consult the table below.

`'input_ids'`	`[30273, 259, 30407, 3178, 1, 0, 0, 0, 0, 0]` List of token ids to be fed to a model
`'attention_mask'`	`[1, 1, 1, 1, 1, 0, 0, 0, 0, 0]` List of indices specifying which tokens should be attended to by the model

It's crucial to highlight that the trailing zeros in each row stem from configuring a maximum sequence length of 10, which necessitates padding the sequence to fulfill this requirement. Moreover, the values of `1` in the `'input_ids'` correspond to special tokens. Below is a table elucidating these special tokens:

Token	*ID*	*Usage*
<*pad*>	0	The token used for padding.
</*s*>	1	The end of sequence token.
<*unk*>	2	The unknown token.

2. Training

- **Hyperparameter Selection and Decision Process**

I use the hyperparameters listed in the table below.

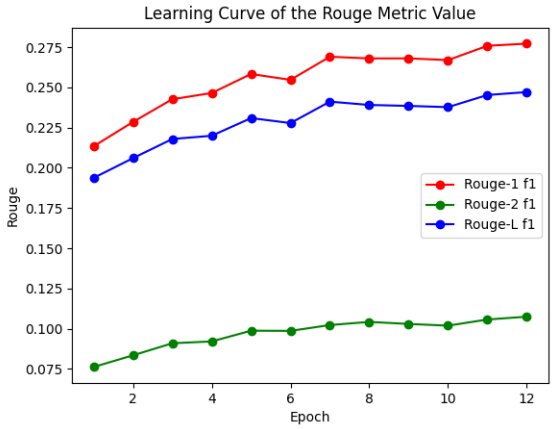
Model	google/mt5-small
Optimizer	AdamW
Epoch	12
Learning Rate	5e-4
Batch Size	16
`per_device_train_batch_size`	4
`gradient_accumulation_steps`	4
`max_source_length`	256
`max_target_length`	64

I determined the values for `max_source_length` and `max_target_length` by following the recommendations provided in the homework guidelines. As for the learning rate, I initially used the default value, which is 5e-5. However, I found that the performance did not meet my expectations. Therefore, I conducted manual adjustments to the learning rate, eventually achieving the desired results.

When it came to deciding the number of epochs, I initially trained the model for 24 epochs, which took approximately 4 hours and aligned with the suggested training time in the homework instructions. After closely monitoring the performance, I observed that the model's improvement plateaued at around 10 epochs. Consequently, I decided to settle on 12 epochs for the final training duration.

To adhere to the 8GB VRAM limitation specified in the homework, I used the default values for `per_device_train_batch_size` and `gradient_accumulation_steps`, which are 8 and 1, respectively. I closely monitored GPU usage and made simultaneous adjustments to the batch size (`per_device_train_batch_size` and `gradient_accumulation_steps`) to meet the hardware limitations. Ultimately, I settled on the values as outlined in the table above.

- **Learning Curve of the Rouge Metric Value**



3. Generation Strategies

- **Text Summarization Generation Strategies in Detail**
  - **Greedy**

    The greedy strategy involves selecting the word with the highest probability. Nevertheless, this approach may lead to suboptimal results because it lacks the capability to backtrack. In other words, unexplored paths might have a higher probability, and it is infeasible to explore every possible path.

  - **Beam Search**

    The beam search strategy involves maintaining a record of the k most likely sequences while continually seeking better alternatives. With a smaller beam size, it can lead to problems such as grammatical errors, awkwardness, inaccuracies, and so on. On the other hand, using a larger beam size can alleviate some of these issues, but it comes at the expense of increased computational demands.

  - **Top-k Sampling**

    The top-k sampling strategy involves selecting the next word with an element of randomness by randomly sampling a word from a probability distribution, rather than simply choosing the word with the highest probability (argmax). However, this random sampling is constrained to the top-k most likely words. When k equals 1, it mirrors the greedy strategy, and when k equals the total number of words, it becomes pure sampling. Increasing the value of k tends to produce more diverse and daring output, while decreasing k results in more conventional and conservative output.

■ Top-p Sampling

The top-p sampling strategy involves sampling from a subset of the vocabulary that contains the highest probability mass. This approach can adaptively expand or contract, effectively encompassing top-k sampling within its capabilities. It's worth noting that top-p sampling is also recognized as nucleus sampling.

■ Temperature

The temperature strategy involves applying a hyperparameter, denoted as τ, to the softmax function. When a higher temperature is used, the probability distribution becomes more uniform, resulting in a more diverse output. Conversely, when a lower temperature is applied, the probability distribution becomes more concentrated, leading to a less diverse outcome. It's important to note that softmax temperature is not a decoding algorithm in itself; rather, it's a method for controlling diversity during testing, which can be integrated with any decoding algorithm.

● Comparison of Text Summarization Strategies and Final Selection

■ Greedy

`num_beams`	**1**	**1**	**1**
`top_k`	0	0	0
`top_p`	1.0	1.0	1.0
`temperature`	**0.8**	**1.0**	**1.2**
`do_sample`	False	False	False
Rouge-1 F1	**0.20367**	0.16523	0.11974
Rouge-2 F1	**0.06892**	0.05212	0.03253
Rouge-L F1	**0.18201**	0.14796	0.10786

I set the `num_beams` value to 1 for the table above, implementing the greedy strategy. However, the performance of these configurations did not meet my expectations. Note that the only parameter related to the greedy strategy is `num_beams`, which can only be assigned to 1. Therefore, I attempted to adjust the temperature value to obtain more results to meet the requirements of the report.

■ Beam Search

num_beams	**5**	**10**	**15**
top_k	0	0	0
top_p	1.0	1.0	1.0
temperature	1.0	1.0	1.0
do_sample	False	False	False
Rouge-1 F1	0.25191	0.25623	**0.25877**
Rouge-2 F1	0.09615	0.10096	**0.10317**
Rouge-L F1	0.22620	0.23115	**0.23377**

To enhance the table above, I increased the `num_beams` value to implement the beam search strategy. It is evident that when the `num_beams` value is set to 15, the performance surpasses all other configurations. While increasing the `num_beams` value may lead to an increase in computational time, I believe this compromise is acceptable for the improved performance.

■ Top-k Sampling

num_beams	1	1
top_k	**10**	**5**
top_p	1.0	1.0
temperature	1.0	1.0
do_sample	True	True
Rouge-1 F1	0.21634	**0.22533**
Rouge-2 F1	0.06983	**0.07572**
Rouge-L F1	0.19187	**0.19963**

I adjusted the `top_k` value for the table above to implement the top-k sampling strategy. However, the performance of these configurations did not align with my expectations.

■ Top-p Sampling

num_beams	1	1
top_k	0	0
top_p	**0.75**	**0.5**
temperature	1.0	1.0
do_sample	True	True
Rouge-1 F1	0.19924	**0.22195**
Rouge-2 F1	0.06616	**0.07798**
Rouge-L F1	0.17706	**0.19854**

I adjusted the `top_p` value for the table above to implement the top-p sampling strategy. However, the performance of these configurations did not meet my expectations.

■ Temperature

num_beams	15	15	15
top_k	0	0	0
top_p	1.0	1.0	1.0
temperature	**0.8**	**1.0**	**1.2**
do_sample	False	False	False
Rouge-1 F1	0.24145	**0.25877**	0.25466
Rouge-2 F1	0.09479	**0.10317**	0.09542
Rouge-L F1	0.22071	**0.23377**	0.22659

Utilizing the best configuration from all previous experiments for the table above, I modified the `temperature` value to implement the temperature strategy. However, even when the `temperature` value is set to the default value of 1, the performance remains better than any other configurations.

■ Final Generation Strategy

Based on the results of the above experiment, I have chosen the beam search strategy with a `num_beams` value of 15. The following table provides detailed information about my configuration.

num_beams	top_k	top_p	temperature	do_sample
15	0	1.0	1.0	False

Rouge-1 F1	*Rouge-2 F1*	*Rouge-L F1*
0.25877	0.10317	0.23377